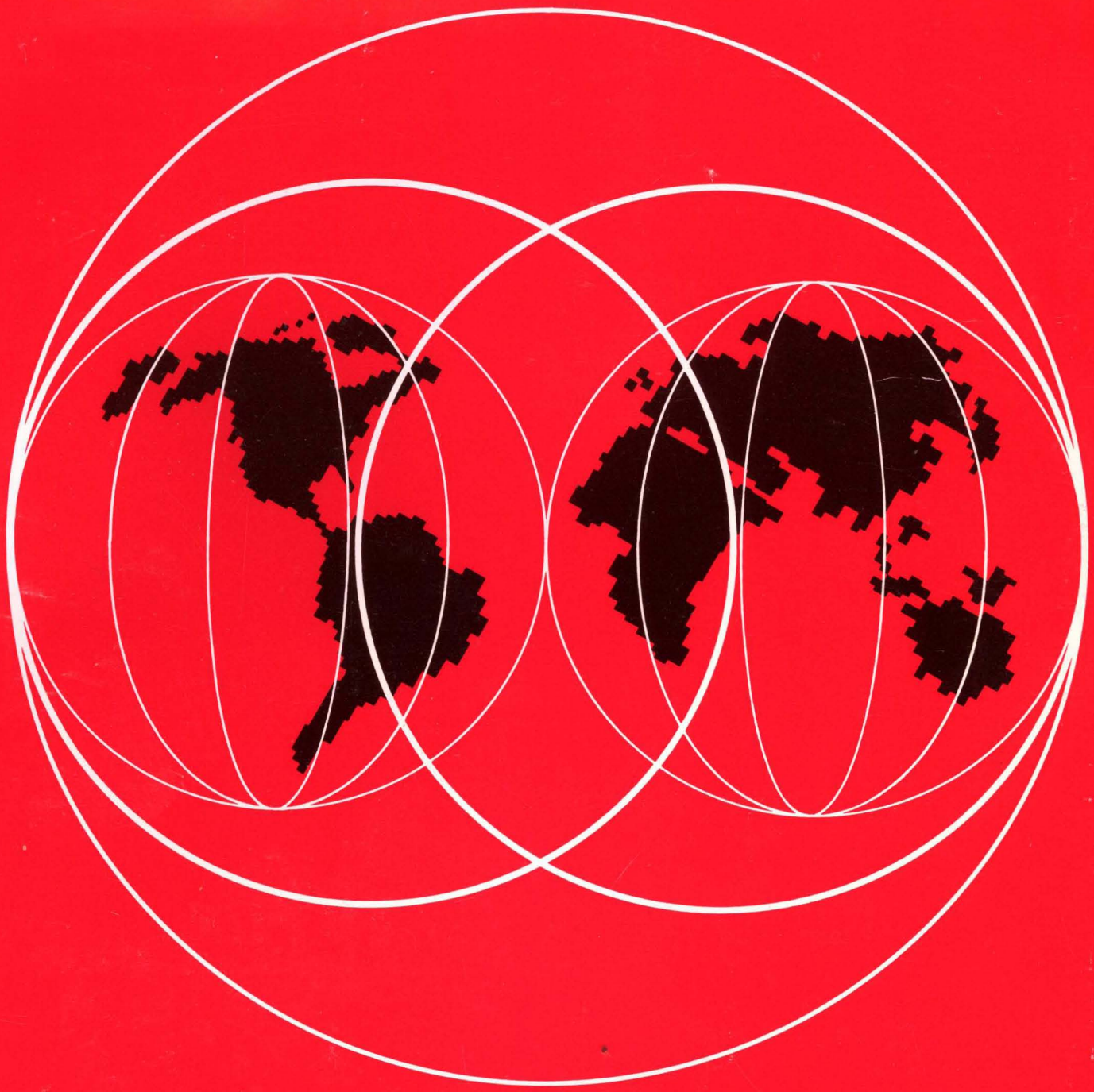


IBM

International Technical Support Centers

ENTERPRISE NETWORKING WITH SNA TYPE 2.1 NODES

GG24-3433-00

Enterprise Networking with SNA Type 2.1 Nodes

Document Number GG24-3433

November 1989

International Technical Support Center

Raleigh, North Carolina

FIRST EDITION, November 1989.

This is the first edition of GG24-3433.

Abstract

This publication presents a broad tutorial on the new "any-to-any" networking capability of SNA.

Named "SNA Type 2.1 node support," this is a key new function of SNA. "SNA Type 2.1 node support" also provides for the integration of small ("Low Entry Networking" and "APPN" networks) into the SNA wide area network and for the full support of the new function provided by "APPC" (or LU6.2) protocols.

Primarily intended for technical planners, this book is written at a "technical understanding" (or conceptual) level. It discusses in some detail the technical characteristics of the new function and some of the new application possibilities that have become available.

A focus of the book is the newly announced SNA Type 2.1 node support for VM/VTAM on the IBM ES/9370 system although the discussion also applies to the similar function already available for the larger /370 systems through the "SNA Type 2.1 node support" feature of NCP.

CSYS (195 pages)

Acknowledgements

The author of this document is:

- Harry J.R. Dutton, IBM Australia Limited

Project Advisor:

- Roland Peschke, IBM International Technical Support Center, Raleigh

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

Thanks are due to the following people.

- Gerard Joseph, IBM Australia Limited

  for permission to reproduce the Appendix "SNA Type 2.1 Node Overview" from *An Introduction to Advanced Program-to-Program Communication (APPC), (GG24-1584-1).*

- Hans H. Pfeiffer, IBM International Technical Support Center, Rochester

  for permission to include material from *IBM AS/400 Advanced Peer To Peer Networking (APPN) (GG24-3287-0).*

- Bob Gibbs, IBM Communications Systems Division, Raleigh

  for technical consultation and advice.

- Jeff Mason, IBM Communications Systems Division, Raleigh

  for assistance with the Bibliography and the Glossary.

- Gail Wojton, IBM International Technical Support Center, Raleigh

  for editorial assistance and review.

Preface

This publication is a systems engineering technical paper, NOT a product manual. Its purpose is to assist the reader in understanding the wider issues relating to the interconnection of IBM products. It should be regarded by the reader in the same way as a paper published in a professional journal or read at a technical conference.

The perspective and orientation reflect the professional opinion of the author and not necessarily the policy or intent of IBM.

Detailed information about IBM products is given here incidental to objectives of the document and while every effort has been made to ensure accuracy, such information should not be considered authoritative.

Authoritative information about IBM products is contained in the official product manuals. A bibliography of some of these publications may be found in "Bibliography" on page 177.

Purpose and Scope

Over the past few years, IBM has introduced major new networking function into Systems Network Architecture (SNA). This function provides the user with a significantly wider range of interconnection possibilities and hence a much richer variety of application alternatives.

There have been many individual enhancements which despite their "jargon" names have added important functions to SNA. These enhancements are a little like a jigsaw. Taken individually, each is significant in and of itself. But taken together, they form a total capability which as a whole is substantially greater than the sum of its individual parts.

The most recent enhancement is called "SNA Type 2.1 node support". This was announced for large systems in 1987 and delivered in 1988. In 1989, IBM is extending this support to smaller IBM ES/9370 systems. T2.1 node support means that an SNA network is now able to provide direct communication between any two "boxes" attached to the SNA network provided that each "box" uses the T2.1 node attachment protocol. It also means that SNA networks and networks of small systems (called APPN networks) can be interconnected in a natural and convenient way. Further, T2.1 node support enables another new feature, called LU 6.2,[1] to be used to its fullest potential.

This book is about any-to-any connectivity in SNA networks. It is intended that the reader gain an understanding of the function provided, its advantages and limitations and, most important, the kinds of new application possibilities that are now open to the system designer.

Audience

This document is intended for persons wishing to gain an understanding of the new SNA functions and their use.

Systems designers wishing to implement applications solutions incorporating distributed processing will find that the discussion in Chapter 3 will suggest a new range of options.

[1] Also called APPC or Advanced Program-to-Program Communication

Network planners will find that Chapters 2 and 5 will convey an understanding of the new function and will provide the necessary understanding for network planning.

Systems programmers will find a useful background to the product details to be found in other documents.

IBM systems engineers should find information to suggest new problem solutions and application possibilities.

Chapter 3, "Systems Examples" on page 21 deals with the additional function available to SNA users as a result of SNA Type 2.1 node support. As such it is the most important part of the document. Other chapters serve either to introduce and provide background to the subject for the generalist reader or to provide some detail for the technically inquisitive.

Organization

This publication is divided seven chapters and four appendices:

Chapter 1, "Introduction" on page 3

Introduces the concept of the SNA Type 2.1 node and places it in the context of Enterprise Networking.

Chapter 2, "Overview of New Functions" on page 7

This chapter spells out the functions of VTAM V3 R3 and NCP V5 R3 as they relate to the T2.1 node connection support.

Chapter 3, "Systems Examples" on page 21

This is perhaps the most important section in the document. Addressed to network planners and IBM systems engineers, it contains a discussion of system design possibilities in relation to other IBM networking products. It is intended to suggest how the T2.1 node support can be used to solve some common systems design problems.

Chapter 4, "Technical Background" on page 45

This chapter contains a "technical conceptual" discussion of the developments in SNA that led up to the "T2.1 node support" and of its objectives.

Chapter 5, "Principles of Operation" on page 65

A more detailed technical description (again at a conceptual level) is presented in this chapter. It is intended to introduce a reader who has some prior SNA knowledge to the concepts relevant to T2.1 node connection.

Chapter 6, "Token-Ring (TRN) Connection" on page 83

Token Ring "gateways" are very necessary to the enterprise network. However, TRN gateways offer different levels of T2.1 node capability. This chapter discusses the issues relating to use of TRN gateways.

Chapter 7, "Technical Details" on page 89

This chapter is not for the faint-hearted. It contains a sometimes detailed discussion of individual characteristics of SNA as they relate to the T2.1 node attachment. Readers of this chapter are expected to have considerable prior SNA knowledge (albeit at the conceptual level).

Chapter 8, "VTAM V3.3 Installation Planning" on page 119

This chapter discusses the T2.1 node implementation in ES/9370 VM/VTAM.

Appendices

- Appendix A, "SNA Type 2.1 Node Overview" on page 135.

- Appendix B, "An Introduction to APPN Networking" on page 143.
- Appendix C, "Independent (X.25) Packet Networks Compared to SNA" on page 163.
- Appendix D, "An Introduction to X.25 Concepts" on page 169.
- "Bibliography" on page 177.

Terminology

As SNA has evolved in function, new concepts have been added and old ones changed. This process has caused many changes in the language used to describe SNA. This document uses the most recent terminology.

Readers familiar with SNA publications should note the following:

SNA Type 2.1 node (T2.1 node).

In the past this was called the "SNA PU 2.1" or just "PU 2.1."

SNA Type 2.0 node (T2.0 node).

This was called the "SNA Physical Unit Type 2," the "PU 2" or the "PU 2.0."

In general, the term "SNA physical unit" has been replaced by the term "SNA node." This change reflects the change in concept and function that has taken place in the architecture.

Related Publications

Systems Engineering Technical Publications

APPC

- *An Introduction to Advanced Program-to-Program Communication (APPC)* (GG24-1584-01).

Previous NCP/VTAM T2.1 node support

- *VTAM V3 R2 and NCP V4 R3 - Planning Guide for New Functions* (GG24-3121-0).

AS/400 APPN

- *IBM AS/400 Advanced Peer-to-Peer Networking (APPN)* (GG24-3287-0).

Network Management

- *Management of AS/400 in SNA Subarea Network Using NetView Products* (GG24-3289).

Interconnection with X.25

- *Integrating X.25 Function into Systems Network Architecture Networks* (GG24-3052-1).

PS/2 OS/2 Extended Edition

- *OS/2 Extended Edition Cookbook* (GG24-3359 and GG24-3360).

SNA Architecture Publications

- *Systems Network Architecture Concepts and Products* (GC30-3072).
- *Systems Network Architecture Technical Overview* (GC30-3073).
- *Systems Network Architecture Reference Summary* (GA27-3136).
- *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic* (SC30-3112).
- *SNA Format and Protocol Reference Manual: Architecture Logic for Type 2.1 Nodes* (SC30-3422).
- *Systems Network Architecture Format and Protocol Reference Manual: Management Services* (SC30-3346).
- *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2* (SC30-3269).
- *Systems Network Architecture: Transaction Programmer's Reference Manual for LU Type 6.2* (GC30-3084).
- *Systems Network Architecture: Sessions Between Logical Units* (GC20-1868).

Product Specific Publications

A list of some of the related product specific publications is given in "Bibliography" on page 177.

Contents

Figures

Section 1. Systems

Chapter 1. Introduction

Background

Since its inception in 1974, Systems Network Architecture (SNA) has been changing constantly to meet the changing technological environment and to accommodate the needs of IBM customers. In the early 1980s IBM began a process designed to:

1. Review and redesign parts of SNA to take advantage of the lessons learned over the many years since SNA was first introduced.
2. Take advantage of hardware and software technology improvements that offer new systems opportunities. (For example the improvements in hardware cost/performance made true peer-to-peer networking in small computers an important requirement.)
3. Adjust to the changes in telecommunications facilities available from the carriers.

Of the many results of this process two are important to note here:

1. The development of LU 6.2.

   Also called APPC for Advanced Program-to-Program Communication, LU 6.2 is a major redesign of the end-to-end protocol parts of SNA. It involves major changes in many SNA concepts.

2. The extension of peer-to-peer networking in SNA from peer-to-peer connection of S/370 hosts only to peer-to-peer connection for any small, link-connected processor.

   In the past, small link-connected processors could only communicate with each other through the use of a host resident relay program. (Later Network Routing Facility (NRF) enabled a peer-to-peer routing through a relay program in the 37xx but this function is a data relay, NOT a direct SNA session connection.)

   This involved first the standardization of a "direct link" box-to-box SNA protocol. Second, it meant the development of an SNA extension called APPN (Advanced Peer-to-Peer Networking) which allows for the creation of networks of small systems without the need for mediation by a traditional SNA host computer. Third, it means the integration of the above two forms of SNA networking into the traditional wide area network environment.

This book is about the new SNA environment created by the integration of the above developments.

The Enterprise Network

In the 1970's the focus of networking for many organisations was the wide area network. This was because, in the technology of the time, the wide area network was the place where the most business benefits were to be found. Long line bandwidth was extremely expensive and it made sense to build a "packet" network to allow the optimisation and sharing of an expensive, scarce resource.

Today, as we enter the 1990's most of the technological challenges of the wide area network have been solved. In addition to this the cost of communication bandwidth has reduced dramatically as the telephone companies introduce digital techniques into their networks.

In the computer industry, small processors have continued to become more capable and cost effective every year.

This has meant that within many large organisations the focus of data communications has shifted from the wide area part of the network to local area networks.

The development of SNA, the introduction of L.E.N. and then APPN networking, and the transition to LU 6.2 should be seen in the above context.

Today, the considerable challenge is to construct a single, integrated network for an enterprise. In many cases this means the sharing of data and voice on the same links but in all cases it means having a single, integrated, data network. Such a network must:

- Be able to handle the traditional wide area network traffic.
- Fully integrate local area and wide area network technologies.
- Be able to take advantage of the most effective data transmission facilities available.
- Fully support any-to-any communication for small processors such as PCs etc.
- Have standardised interfaces throughout such that users are able to take full advantage of the facilities offered without extensive retraining for every new device.
- Be able to support industry standard interfaces such as X.25.
- Be manageable as a single integrated entity.
- Be able to integrate as needed with voice networks.

SNA Type 2.1 node networking should be seen as a major step toward fulfilling these requirements for SNA users.

SNA Type 2.1 Node Networking

The name "SNA Type 2.1 node" does not, of itself, identify a network or networking function. SNA Type 2.1 node identifies two things:

1. A type of *interface* to an SNA network *but not the function provided by that network.*[2]
2. The generic structure of one type of SNA node.

While the two aspects are related by far the most important in the context here is the "network interface" function. T2.1 node can be used as an interface for a box-to-box direct link, an interface to an SNA APPN network or as an interface to an SNA wide area (also called a subarea) network.

T2.1 node support in the SNA subarea network[3] means the following new capabilities have been added to traditional SNA:

1. The ability to communicate directly between "boxes" using the T2.1 node protocols across an SNA subarea network *without* the need for any form of "relay" application (whether host or 37xx based).
2. The ability to communicate between an SNA APPN network and an SNA Type 2.1 node *through* an SNA subarea network.
3. The ability to connect SNA APPN networks to one another through an SNA subarea network.
4. The ability to allow for full-function LU 6.2 support in peripheral boxes connected directly to the SNA subarea network or indirectly through a connected APPN network.

---

[2] In this sense it can be compared to X.25 because X.25 identifies an interface to a network and not the network function provided also. A comparison with X.25 is provided later in Appendix C, "Independent (X.25) Packet Networks Compared to SNA" on page 163.

[3] The traditional SNA network consisting of S/370 hosts, 37xx communication controllers using VTAM and NCP communication program products.

5. The ability to connect a large mainframe S/370 host to an SNA network as a channel-connected T2.1 node, thus making that host into an independent processor attached to, but separate from, the subarea network.
6. The ability, under some circumstances, to connect two SNA subarea networks together using a T2.1 node interface. This is an attractive option in some situations because while it offers a lower degree of function than traditional SNA Network Interconnection (SNI), it is extremely simple and offers excellent isolation characteristics.

These capabilities were added to the SNA subarea network for the larger S/370 based networks through new releases of VTAM (V3 R2) and NCP (V4 R3 and V5 R2). This capability is now extended to the IBM ES/9370 system using VM/VTAM Version 3 Release 3.

It is important to note that a T2.1 node connection may use a number of different means of physical connections. It does not need to be a direct SDLC link connection. The connection could be:

- An SDLC link connection provided over an analogue or digital service (including an ISDN service)
- A Token Ring Local Area Network (TRN_LAN) connection
- An X.25 network
- A System /370 channel connection

SNA is structured in such a way that in the future other link protocols could be added to the list relatively easily. The following are examples of possible future link level connections:[4]

- ISDN (Integrated Services Digital Network) both at basic rate and at primary rate.
- Ethernet™ [5] local area network.
- FDDI (Fibre Distributed Data Interchange) for very high-speed campus area connections.
- DQDB (Distributed Queue Dual Bus - IEEE 802.6) for metropolitan area network connections.
- Other technologies as they become useful.

These characteristics listed above are extremely important because, taken together, they can allow an SNA subarea network to form a complete networking structure for an enterprise. Such a network is fully supported by network management and may be operated, diagnosed and maintained as a single, integrated, whole.

Systems Application Architecture (SAA)

In the past many users found that building a truly distributed data processing system was an insurmountable challenge. While distribution of function throughout an enterprise network looked economic and attractive there were some severe problems.

The most common conception of a distributed system was for a hierarchy of "processing levels." In a bank, for example, one possible structure is as follows:

- Central site S/370 processors with a transaction processing subsystem such as CICS or IMS to manage the central data base.

---

[4] This is an indication of potential only. Future product announcements will be made according to IBM's business judgement at the time.

[5] Ethernet is a trademark of Xerox Corporation.

- Distributed regional computer centres with local "memo posted" data bases.
- Local processing in the branch for transaction edit, logging, offline operation and data compression, etc.
- A common SNA backbone network.

One realisation of the above structure might be to have S/370 processors in the major computer centres, IBM 8100 processors as regional data base processors and IBM 4700 systems in each branch.

This concept is attractive, economic and very difficult to install as a usable system.

SNA communications gives the ability to integrate all of these systems routinely into the same network and to communicate between application programs from end to end. This has been available and supported in SNA since 1974.

The main reasons a system of this kind was so difficult to install was that each processor type was optimised for its particular function. Optimisation led to the best price performance characteristics. This meant that each processor family was different:

- At the hardware level they had different instruction sets and structures.
- The operating systems or supervisors used by each system was totally different from the others as were the job control and operational characteristics.
- The programming languages used by each system, while in some cases common (all could use COBOL for example), had local optimisations which made them incompatible. Also, with this level of technology, optimisation of code at a very low level (assembler) was economically attractive.
- SNA offered end-to-end communication support between programs but this was for the sending and receiving of data blocks and for end-to-end protocol signalling. There was no good end-to-end protocol available in SNA which allowed for easy development of distributed systems. Users had to invent their own and some found that this was too complex a task.

The major problem in implementing this kind of system is people. Each communicating system developed its own cadre of specialists who had different and to some degree incompatible systems concepts and used different words to describe the same (or worse, slightly different) functions.

Programmers and analysts developing communicating applications found that they didn't understand the language or concepts being used by the people developing applications they had to communicate with.

Advanced Program-to-Program Communication (APPC or LU 6.2) was a major advance towards solving this problem. The LU 6.2 design took as its beginning the insistence that it be the same on all implementations down to the calls to the operating system and the error codes.

However, LU protocols were only a part of the problem and not the major part. Programming languages, operator interfaces, basic concepts and many other things were not standardised.

Systems Applications Architecture (SAA) was designed to address this problem. Often, SAA is explained as offering application portability. That is true but that is not the main importance here. SAA will allow for one group of people with common training to install, program, manage and use a network consisting of a wide variety of different processor types each optimised to its particular function and each vastly different from the others internally.

One component of SAA is called "Common Communications Support" and is based on LU 6.2 and T2.1 node communication. It is specifically designed to enable users to install distributed systems easily and thus to take advantage of the variety of different processor architectures (each optimised to its particular environment) offered by IBM.

Chapter 2. Overview of New Functions

The primary functions provided by "SNA Type 2.1 node support" are:

1. To allow attachment of an SNA Type 2.1 node to an SNA subarea network.
2. To provide the ability for independent LUs within attached T2.1 nodes to have direct sessions with one another through the network (without the use of a "host relay").
3. To allow the use of full-function LU 6.2 operation within a T2.1 node attached to the network. This means that an LU 6.2 within a T2.1 node may have multiple and parallel sessions with host-based applications (such as CICS) as well as the same functions when communicating with other T2.1 nodes across the network.

If the T2.1 node is to be attached to an IBM 37xx controller running the Network Control Program (NCP) then the appropriate support is:

- ACF/NCP Version 4.3 (for the 3725) or
- ACF/NCP Version 5.2 (for the 3720, or 3745) and
- ACF/VTAM Version 3.2 for MVS, VM and VSE host systems.

If the T2.1 node is to be connected to an ES/9370 system through a 9370 Telecommunications Subsystems Controller (TSC)[6] then the required support is:

- ACF/VTAM Version 3 Release 3 for VM/SP and VM/9370

Note: The T2.1 node functions of ACF/VTAM V3 R3 are NOT available to devices connected via the ICA feature of the IBM ES/9370 system under the MVS or VSE operating systems.

The features required for "casual" connection of two subarea networks to each other using T2.1 node protocols and for interconnection of devices in different SNA Networks (Non-Native Network Connection) are new with ACF/VTAM V3 R3. These functions are also provided for ACF/NCP based connections by ACF/NCP V5 R3 when used with ACF/VTAM V3 R3.

[6] The TSC is the ES/9370 implementation of an "Integrated Communications Adapter" (ICA). The term "ICA" has been used throughout this document as a generic term to describe this kind of device. The VTAM T2.1 node support however, only applies to the IBM ES/9370 and not to previous ICAs.

Physical Connection to the Network

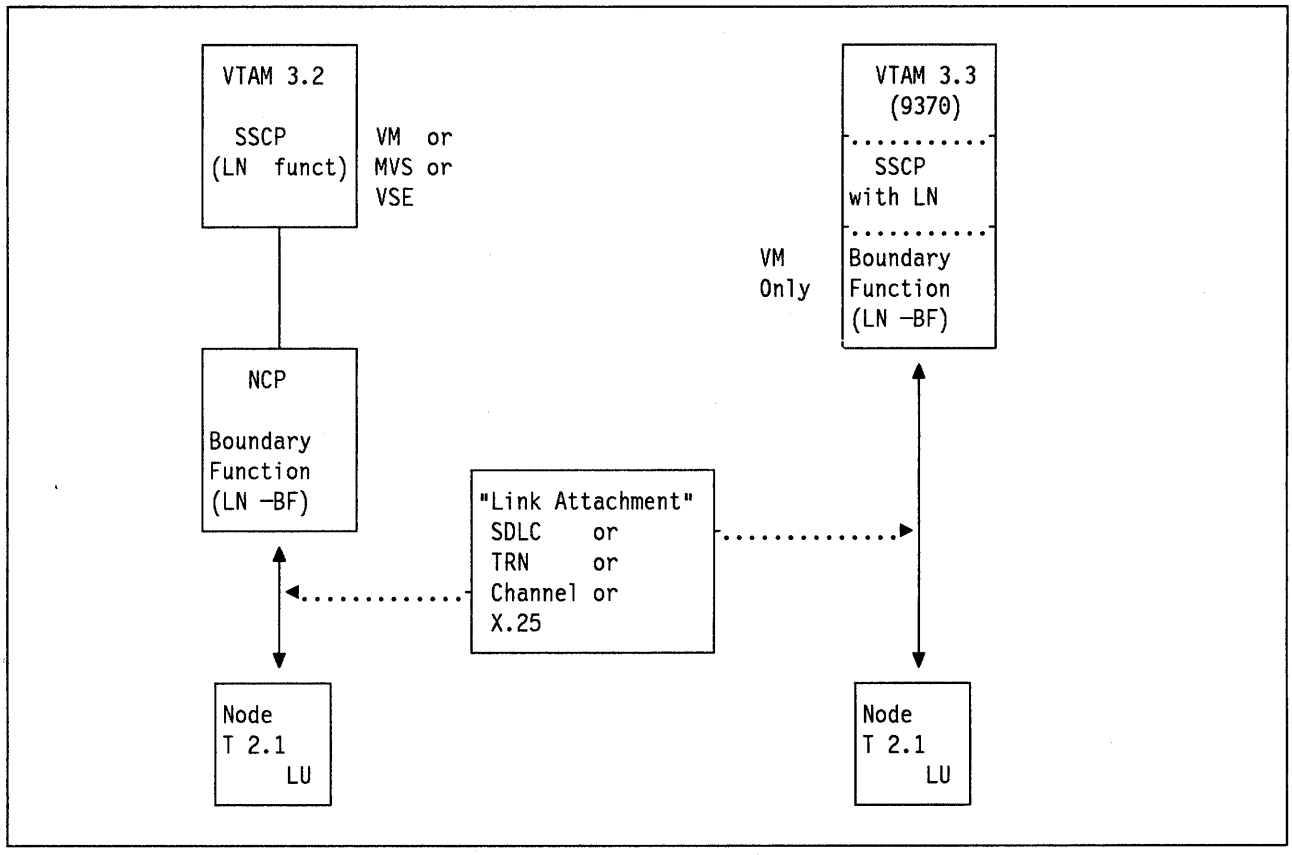


Figure 1. SNA Type 2.1 node Subarea Network Connection

Figure 1 illustrates T2.1 node connection. Support in VTAM V3.3 (for ES/9370) is equivalent to and compatible with the (older) T2.1 node support which was delivered in VTAM V3.2 and NCP V4.3 and V5.2.

Since the VTAM or VTAM/NCP system appears to the connected T2.1 node as though it were just another T2.1 node, it is now possible to connect VTAMs and NCPs to each other using the T2.1 node protocols. This was not possible in the first release of the previous VTAM/NCP support because at the link level, NCP insisted on being the primary. Because two primaries cannot connect to one another, NCP could not communicate with another NCP as a T2.1 node. However, coincident with the VTAM 3.3 (for 9370) announcement there has been an enhancement made to NCP V5R3 to allow NCP to be a secondary and hence allow this kind of connection.

Each T2.1 node connected to the network is treated in *exactly* the same way regardless of the physical means of connection. That is, except for the local link control function, there is no difference between T2.1 nodes connected through a channel, through X.25, a token-ring or via an SDLC switched or leased connection.

The figures on pages 9 through 12 show some of the possible physical connections.

Channel Connection

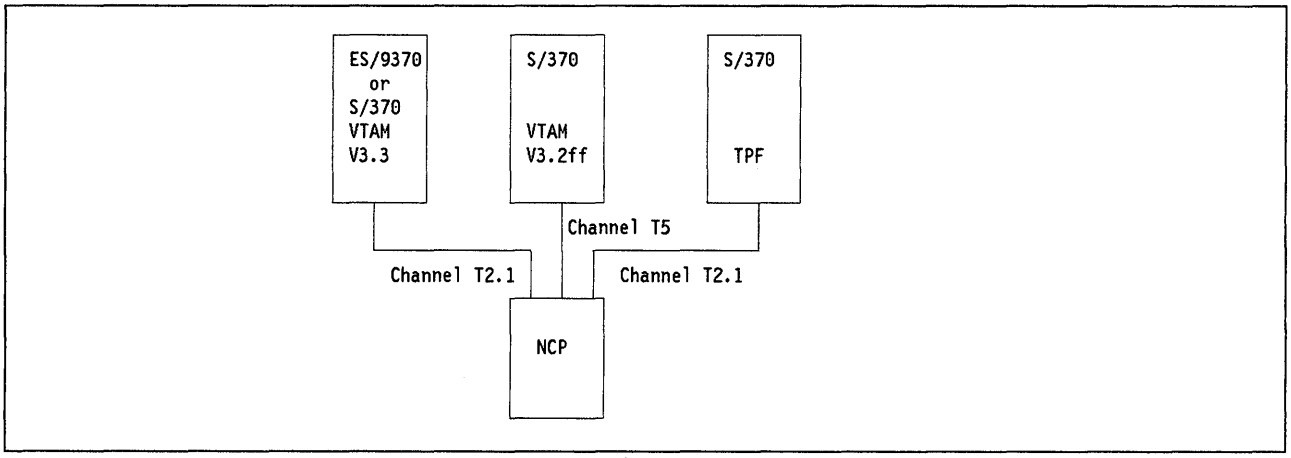


Figure 2. Channel Connection

Here VTAM 3.3 is shown channel connected as a T2.1 node to an NCP controlled by VTAM in another processor. Also shown is a S/370 host running TPF connected as a T2.1 node.[7]

When a channel is used for T2.1 node connection (unlike other types of connection) the channel becomes dedicated to the single point-to-point T2.1 node connection. In the figure, it would be possible for the VTAM 3.3 processor to have a Type 5/4 connection to the NCP in addition to the T2.1 node connection. If this were the case then an additional channel adapter would be required on the 37xx.

The case of T2.1 node host channel connection to the NCP is seen by NCP and VTAM as *exactly* the same as a point-to-point SDLC link. The only difference is that the link is a channel rather than a line.

The T2.1 node VTAM host may be *any* System /370 running VM with VTAM 3.3. It need not necessarily be an ES/9370. The T2.1 node channel connection ability may only be used to connect to a 37xx running NCP. Channel connection cannot connect two ES/9370 systems directly to one another (because there is no hardware channel-to-channel adapter available which allows this). Two ES/9370 systems can be connected to each other using the existing VTAM CTCA support.

[7] Transaction Processing Facility (TPF), formerly known as Airlines Control Program (ACP), is able to connect to NCP via a channel connection and appear as though it were a link connected T2.1 node to the network (VTAM).

Token-Ring Connection

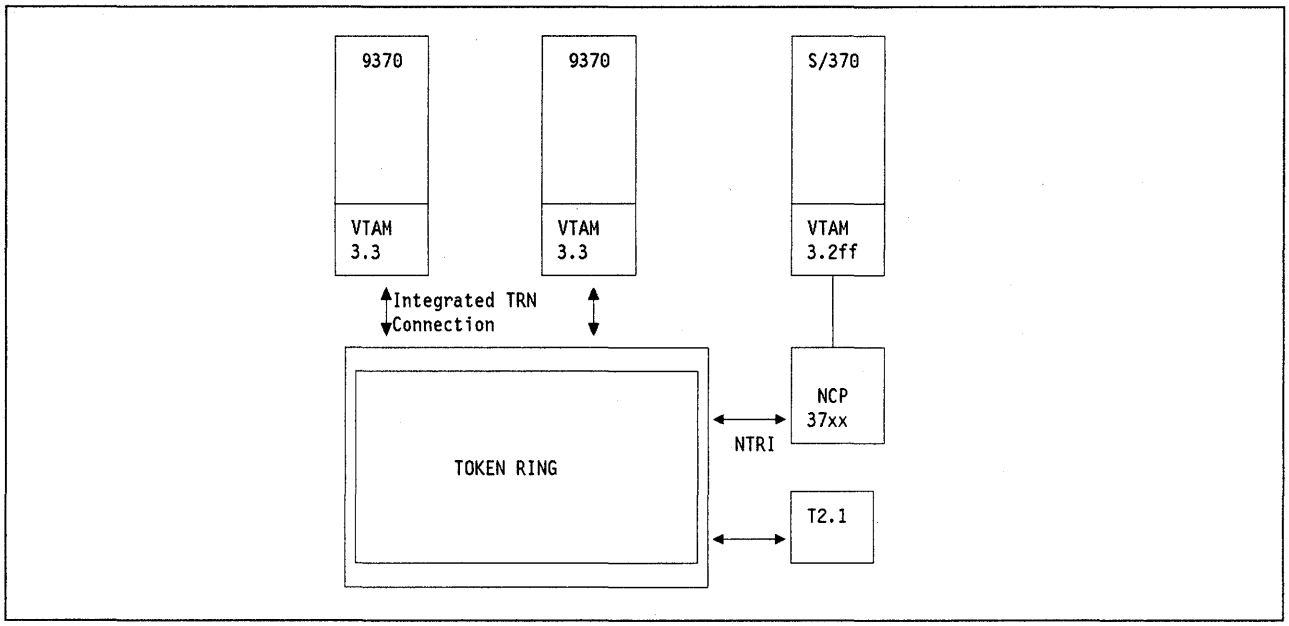


Figure 3. Token Ring Connection

All of the connections shown in this example may be T2.1 node connections. For example, each of the VTAMs shown views the other three devices on the ring as attached T2.1 nodes.

It is important to note that each VTAM system shown could have (in addition to the T2.1 node connections) connections as a Type 5/4 node. That is, it could have Intermediate Network Node (INN) connections with the other VTAMs as well as controlling other T2.0 nodes on the ring. No matter how many logical connections are used, only one TRN adapter is required for each VTAM to perform all these connections simultaneously.

In contrast to the SDLC link attachment case where only the primary (in control of the link) can communicate directly with other nodes on the link, TRN connected nodes can have simultaneous direct communications with every other node on the same TRN.

X.25 Connection

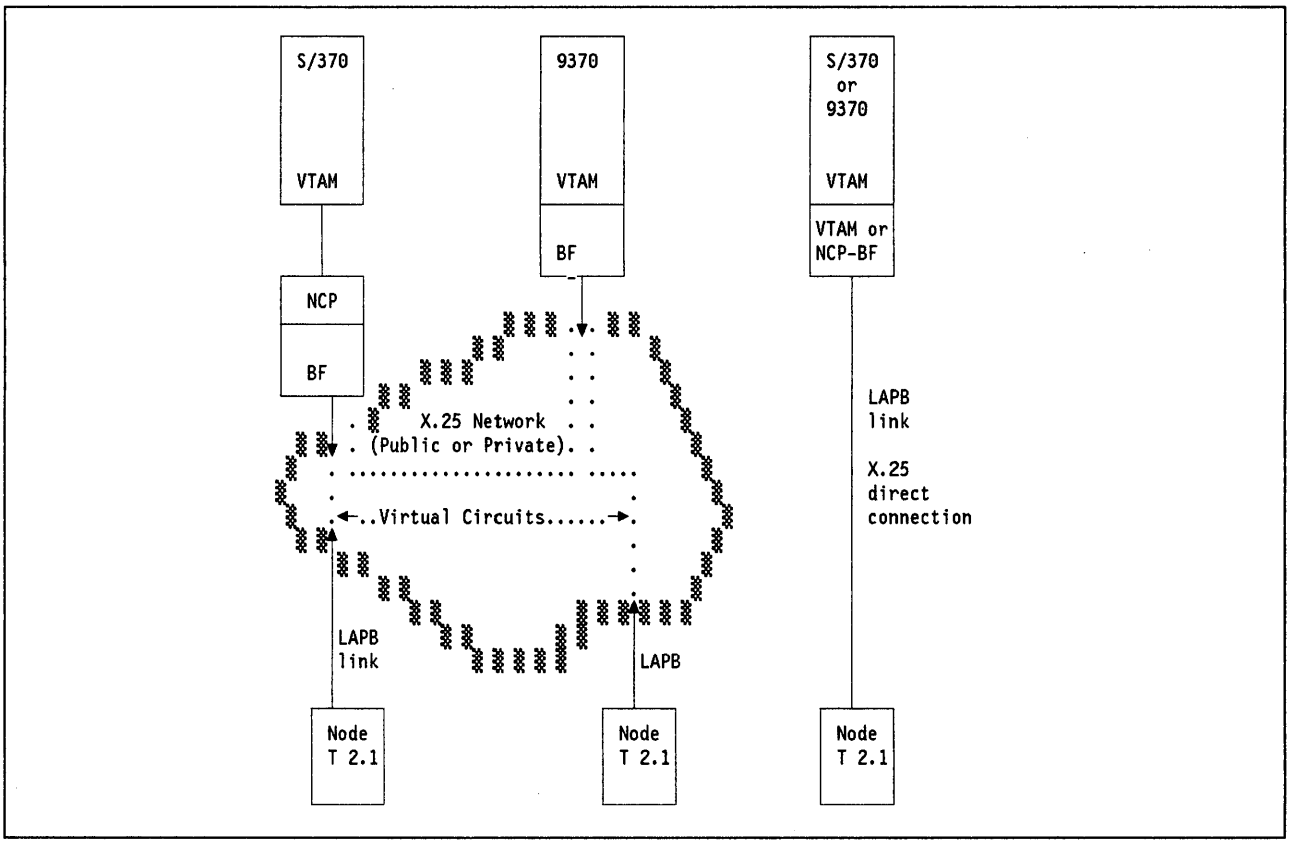


Figure 4. X.25 Connection

Connection to a T2.1 node through an X.25 virtual circuit is supported by both the NCP and the VTAM/9370 implementations. Connection may be through a real X.25 network or via a direct link connection as illustrated. Direct connection is sometimes a useful method of connection in product specific cases. Refer to "Direct X.25 Link Connection" on page 18 for a discussion of direct connection using X.25 protocols.

Both NCP and VTAM(ICA) connections to X.25 allow other types of connection to share the same physical link to the X.25 network. For example, it is possible (through the use of multiple virtual circuits) for VTAM or NCP to connect to other VTAMs or NCPs (cross-network or INN connection), and to traditional T2.0 nodes using only one physical connection to the X.25 network.

The logical link control (LLC) type used for the connection is "QLLC" (or LLC3) **only**. Neither the old "PSH" (LLC_2) nor the extended type "ELLC" is supported for this connection to VTAM (ICA) or to NCP.

Note: The X.25 connection between the T2.1 node and the "X.25" network may be a leased, point-to-point link only. (Channel, TRN and SDLC connection is not possible.) The X.25 link protocol ("LAPB") is a point-to-point protocol and may not have "multidrop" connections.

SDLC Link Connection

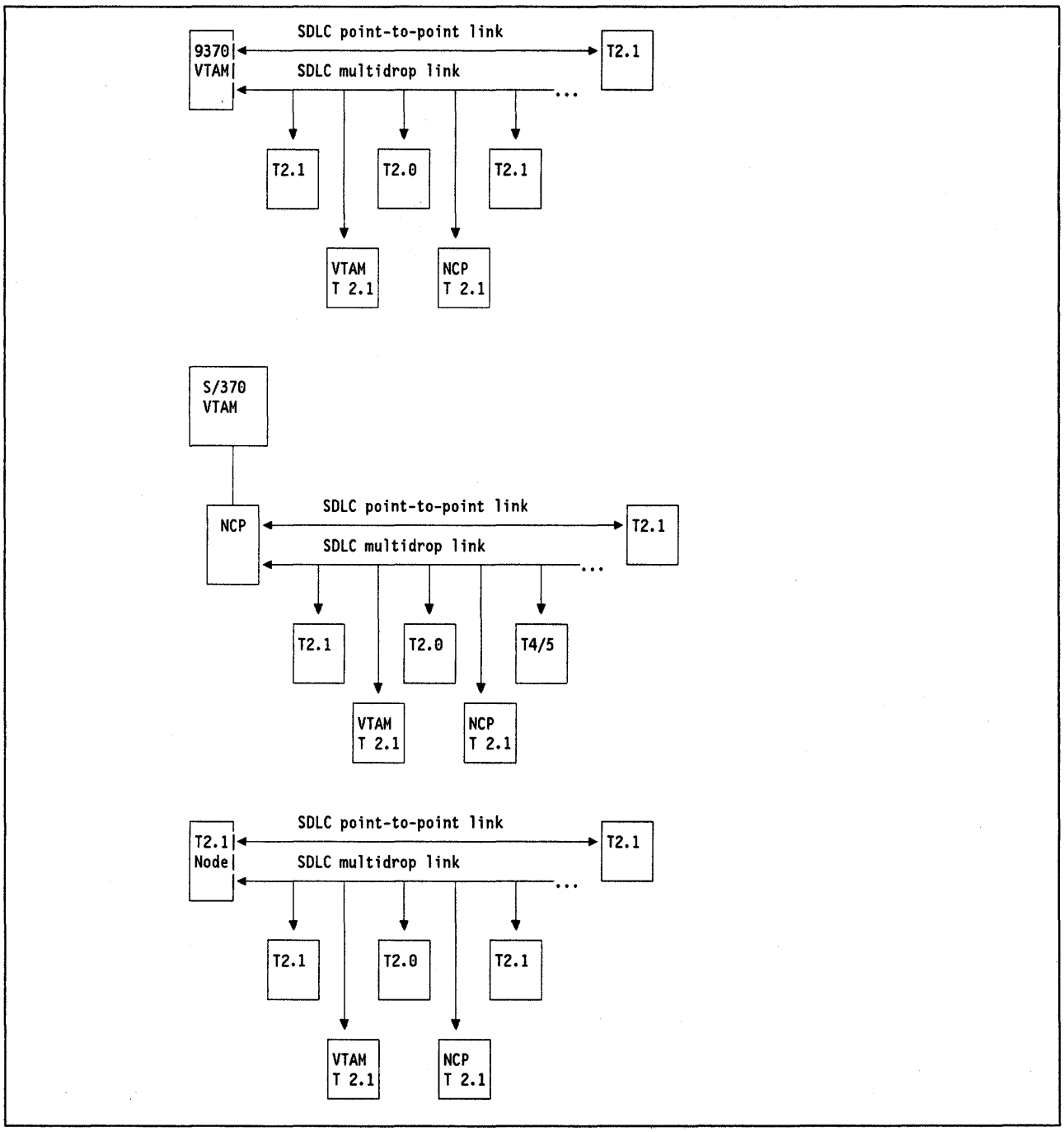


Figure 5. SDLC Link Connections

Both VTAM(ICA) and NCP have identical logical capability to support T2.1 node connections on SDLC links. Of course, there are differences in speed and link attachment capabilities between the attaching hardware devices. These limitations restrict what VTAM and NCP can do in any particular situation.

For example, the 3745 can use SDLC at two megabits per second; the 3720 and the 9370 ICA cannot.

In general, SDLC is supported over both switched and leased connections and over a range of hardware attachment interfaces (X.21, V.24, V.35, G.703, etc.).

The SDLC link may be shared by other types of connection as shown in the diagram.

In the case of SDLC "leased" connection (for either multidrop or point-to-point) the role of primary or secondary on the link must be specified when the system is defined. In the case of "switched" point-to-point connection the link station role is negotiated at connection time and either end may become link primary or secondary.

Network Interconnection

APPN to Subarea Network Connection

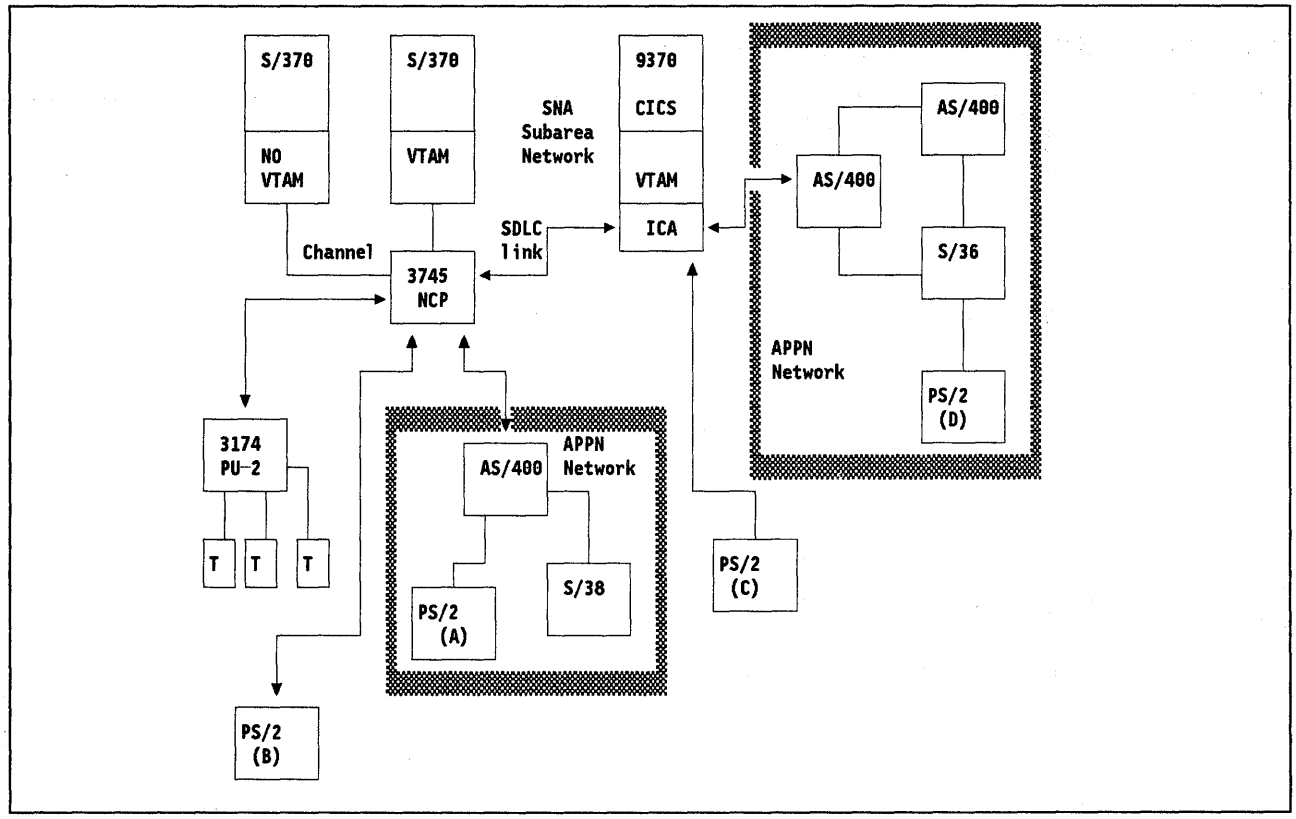


Figure 6. APPN to Subarea Network Connection

In "Physical Connection to the Network" on page 8, T2.1 nodes were treated as though each were a single small processor. In fact, as described in "Subarea Network SNA Type 2.1 Node Interface" on page 72 an entire APPN network may be interfaced to a subarea network using the T2.1 node interface. In this case each network will see the other as a "virtual L.E.N. node." Subarea networks may also be connected to each other in this way.

A subarea network with two connected APPN networks is shown here. The four PS/2's illustrated are assumed to be running OS/2 EE for the sake of the example. Some of the available connections are as follows:

- Any PS/2 illustrated can have multiple LU 6.2 sessions with any or all of the illustrated PS/2's.

  There is no difference in LU 6.2 function resulting between direct connection and connection through the APPN network.

  The APPN connected PS/2's have some restrictions on their 3270 session capabilities. They may connect through to the subarea network if they are directly connected to the AS/400 which is itself directly connected (via T2.1 node protocols) to the subarea network. (This is a feature of the AS/400.) However, APPN networks will not carry dependent LU traffic. This means that PS/2's connected through an APPN network may not have 3270 sessions with the subarea network. In the figure, PS/2 (A) may have 3270 sessions through to the S/370 hosts but PS/2 (D) may not.

- Any of the PS/2's shown could have multiple LU 6.2 sessions to a S/370 host such as the CICS system shown.

- Any of the link connections illustrated could be SDLC or TRN or X.25.
- Any AS/400 LU 6.2 application could have full-function connections with applications in any of the other AS/400's or the S/370 hosts.
- The functions available in the subarea network are unaffected by the addition of T2.1 node capability.

Subarea-to-Subarea Network Connection

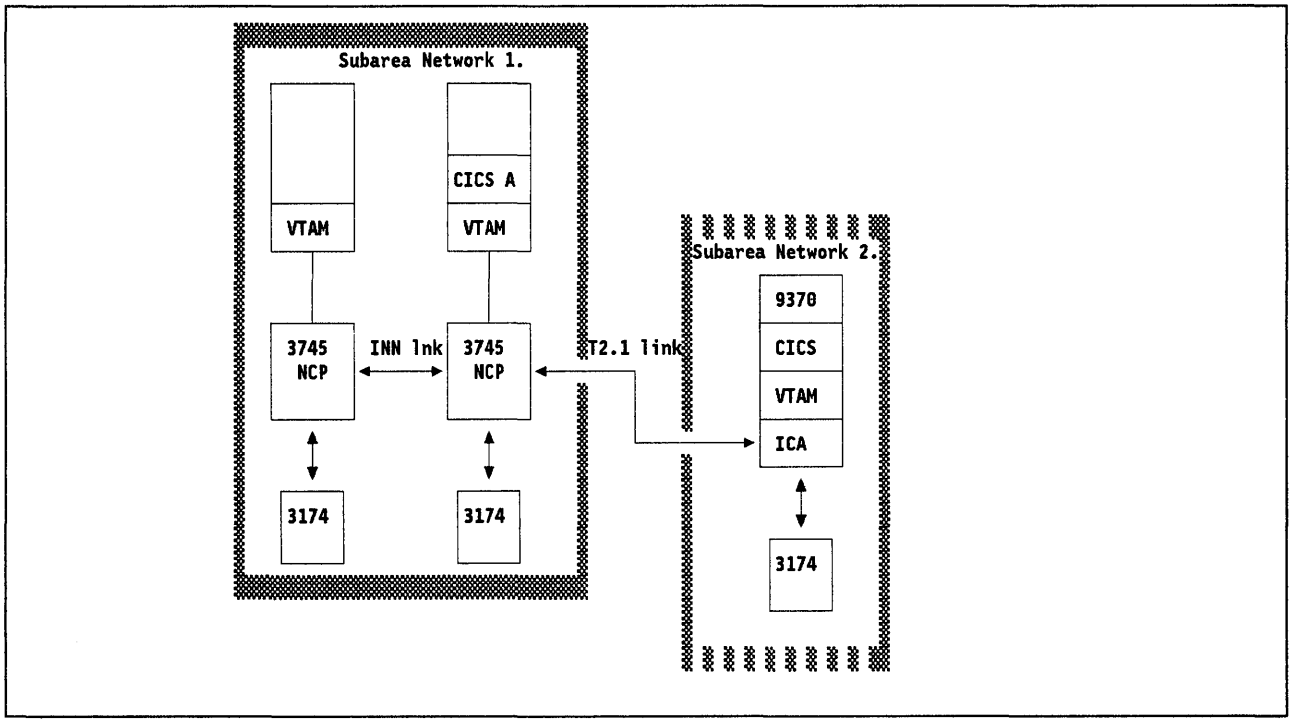


Figure 7. Subarea-to-Subarea Network Connection

In the subarea network interconnection case note:

- When two VTAM systems are connected to each other through the T2.1 node protocols, the two separate VTAM systems are isolated from one another and CANNOT form a part of the same SNA network (unless there is another non-T2.1 node connection present).
- No "3270" logon may take place across the network boundaries. This means that 3270 traffic is impractical unless special code is produced to initiate the logon from the host side. This is discussed in "Dependent LU to Type 2.1 Node Networking" on page 75.
- CICS-to-CICS LU 6.2 traffic is easily possible.
- There is no VTAM-to-VTAM traffic or NetView-to-NetView traffic possible across the network boundaries. This means that network management information may not cross the boundaries unless special code is produced to allow this function.

This facility is extremely easy to install and offers users a new kind of "casual" network interconnection as discussed in "Casual Network Interconnection" on page 27.

Any-to-Any Session Capability

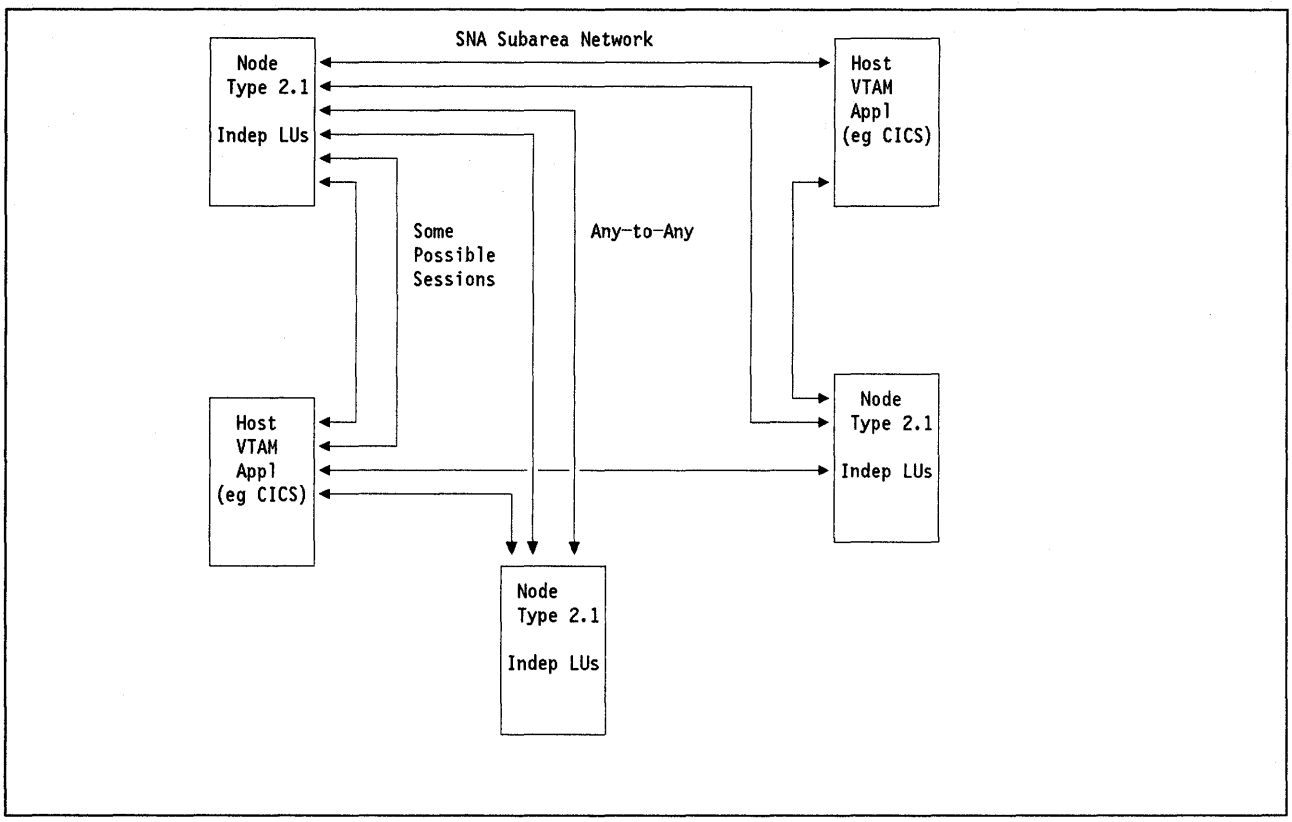


Figure 8. SNA Type 2.1 node Connection Capabilities

SNA Type 2.1 node allows for direct, any-to-any, sessions to be established between LUs within T2.1 nodes and other LUs within T2.1 nodes across the network. The figure shows some possible "direct" sessions. Independent LUs within T2.1 nodes and host LUs may have direct sessions throughout the network without the need for any intervening "relay" function as is required for dependent LUs to have sessions with one another. Dependent LUs are not shown in the diagram.

In general the rules are as follows:

- One LU in each session must perform the role of primary and one secondary. In general LU 6.2's are capable of either role but may be defined in the product implementation (or setup) to have maximum numbers of sessions in which it is a primary or a secondary. Dependent LUs are always secondary and can only have a single session. Host LUs may be primary only, secondary only, or capable of both depending on how they have been set up.
- Each communicating LU in a session must use the identical set of profiles that the other one uses. That is to say each LU must be of the same type and be capable of supporting the same set of characteristics.

  The subarea network is capable of supporting any LU type in any role. The LU type used is a function of the LUs themselves and not the network.

Product Specific Issues

Most of the discussion in this document centres on SNA architecture and network function. It is important to realise that individual products are built to perform specific functions. Few, if any, individual products have (or should have) all of the possible functions.

In the construction of a system to perform a specific role in a user organisation the specific capabilities of the products selected are paramount. Many times, specific systems solutions will be available which appear to contradict the general guidelines. One such case is discussed in "Direct X.25 Link Connection" on page 18.

Some product specifics in relation to connection to the network are discussed here to illustrate the point:

System /36

The System /36 implements a prior (prototype) version of APPN. When the System /36 is connected to a subarea network, it is capable of having both independent and dependent LU sessions with an SNA host in the subarea network. However, *in this case, dependent LUs and independent LUs cannot share the same physical connection to the subarea network.* This means that if both types of connection are required, there must be two links from the System /36 to the VTAM or NCP subarea node.

System /38

System /38 cannot be an APPN node. Nevertheless, it may contain both independent and dependent LUs and these may have sessions to and through the subarea network as described throughout this document. The System /38 does allow both independent and dependent LUs to exist on the same physical connection to the network. It just does not have the networking function of APPN.

When T2.1 node support was first introduced (to NCP and VTAM V3 R2), the System /38 could not connect using independent LUs. This was because the System /38 was the first implementation of L.E.N. architecture, introduced before APPN was developed and before the connection to the subarea network was planned. It allocated its LFSIDs according to a different algorithm from that used by the subarea network connection. This was fixed in March of 1989.

OS/2 EE TRN Gateway

OS/2 EE is capable of attaching directly to the subarea network through SDLC, X.25 or TRN connection and of having both independent and dependent LUs on the same connection. However, as discussed in "OS/2 EE SNA LAN Gateway Support" on page 85, OS/2 EE V1 R2 cannot be a gateway for independent LU sessions originating in LAN-connected PS/2's.

Also, OS/2 EE cannot be a LAN gateway for a TRN-connected LAN Manager PS/2. The LAN manager communicates with NetView through the SSCP-to-PU session and this session is *not* passed through by the gateway. (Nevertheless the LAN Manager can share the same PS/2 as the LAN gateway and then both functions are available normally.)

As stated above, products have functions appropriate to their objectives and most do not (and should not) have all functions. The system planner must take care to become familiar with the functions of the devices that are to be used.

Direct X.25 Link Connection

One of the possibilities for T2.1 node connection to the ES/9370 (or to the 37xx) is to use the X.25 protocol without having an intervening "X.25 Network." Under most conditions this would not be a preferred solution but an example is given here of one (product-specific) situation where it could be very effective.

PS/2 LAN Gateway Connection to the Subarea Network

OS/2 EE-CM may be used in a PS/2 microchannel machine (currently Model 50 or above) as a LAN gateway (for either Ethernet or Token-Ring LAN's) with X.25 protocols used for the upstream link. X.25 can be used as a link protocol but without an intervening packet switched network.

The reasons for this are as follows:

- Standard SDLC upstream connection for OS/2 EE-CM is limited by the hardware characteristics of the "multiprotocol adapter" through which the connection is made. This adapter, while low in cost, places a significant load on the PS/2 processor through the handling of interrupts at the character level. Furthermore it has a maximum link speed of 19.2kbps half duplex.
- The PS/2 X.25 Co-Processor in contrast is a true outboard processor which off-loads all of the PS/2 cycles involved in link control. (It is an 80186 processor with 512k bytes of storage.) Data transfer efficiency (in terms of PS/2 cycles used per block transferred) is significantly improved by using the Co-Processor as compared with the "Multiprotocol" Adapter used for SDLC.
- The PS/2 X.25 Co-Processor is capable of sustaining a link speed of 128kbps *full duplex*.
- The LAPB link control used by X.25 does NOT poll. This means that overheads associated for unused polling cycles do not occur.[8] An SDLC poll received by a PC takes a significant number of PC cycles to process even when there is no data available to send to the host. In order to achieve good response time many users poll at the rate of 10 times per second. This processing load effect has not yet been measured for the PC with OS/2 EE-CM, however in previous IBM cluster controller products, this "overhead polling" has been measured to take up to 30% of box cycles *when no data is being transferred.*

  Also, with X.25 as the link control, because there is no polling, there is no wait between when the data arrives in the PC from the LAN and an opportunity to send that data to the host. The PS/2 can send immediately without waiting for a poll. This improves response time.

  One disadvantage in this approach is that X.25 is a point-to-point protocol and thus multiple token-ring OS/2 EE-CM gateways (or other PS/2 containing the LAN Manager for example) cannot be multidropped from the same link. For capacity reasons (in the case of multiple LAN gateways) it is unlikely that this multidrop configuration would be used very often anyway.
- Another disadvantage is that (if the connection is to NCP) at the host the IBM Program Product "X.25 NPSI" must be used in the 37xx in addition to NCP. This, of course, adds to the path length and hence the load on the 37xx.[9] However, this effect can be minimised by tuning (principally by using a packet size larger than the largest SNA segment to be sent).

  At the host end the 37xx/NCP is also affected by unused polling and the use of LAPB saves the overhead of sending polls - this helps to offset much of the added path length of X.25 packet level processing.

---

[8] While polling has an overhead in performance, it also confers many advantages in other areas. For example, the process of polling is necessary if the link is to be shared using a multidrop procedure. Also, if the link is cut, a polled protocol detects the condition immediately where under some circumstances, a LAPB link may not detect the condition for some time.

[9] VTAM 3.3 provides the X.25 connection function for links connected through the ICA on the ES/9370 without the use of NPSI.

A firm recommendation on the use of this configuration must wait until after appropriate performance studies have been completed. However, based on a knowledge of the basic principles involved, this configuration of LAN gateway (OS/2 EE-CM with upstream X.25) looks attractive.

Supported Devices

The following are some of the current IBM devices which use the T2.1 node interface.

DEVICE OR SUBSYSTEM NAME	DEVICE-MODEL	PU TYPE
3820 Page Printer	3820	2.0/2.1
5520 Administrative System	5520	2.0/2.1
5550 Family	5550	2.0/2.1
8815 Scanmaster I	8815-1,3,4	2.0/2.1
APPC/PC	IBM Personal Computer	2.0/2.1
OS/2 EE Comms Mgr	IBM PS/2	2.0/2.1
AIX	IBM PC/RT	2.0/2.1
Series/1 RPS Version 7.1	Series/1	2.0/2.1
System/88	System/88	2.0/2.1
System/36 Release 4	System/36 Rel.4	2.0/2.1
System/36 Release 5	System/36 Rel.5	2.0/2.1
System/38 Release 8	System/38 Rel.8	2.0/2.1
AS/400	OS/400	2.0/2.1
Transaction Processing Facility - TPF 2.4	System/370	2.1

Figure 9. Supported Devices

Chapter 3. Systems Examples

SNA Type 2.1 node support in VTAM and NCP provides the user with a number of significant new functions. These functions open up a range of application possibilities that were either not possible before or were very difficult to accomplish. This chapter presents some of the systems solutions that are made possible with T2.1 node support.

The functions generally may be thought of as:

- Much improved distributed processing support
- Any-to-any connections through the network
- Casual network interconnection
- Interconnection with APPN networks.

Described in this chapter are applications in the following categories:

- Distributed processing
- Network interconnection (sharing)
- Casual subarea network interconnection
- Connection of non-SNA processors.

Distributed Processing

The primary intent of the new T2.1 node support in VTAM and NCP systems is to provide better support for distributed processing systems, that is, to assist users in developing "distributed processing" systems solutions using machines such as PS/2s and AS/400s attached to the subarea network.

IBM strategic direction for distributed processing is to use LU 6.2 protocols and T2.1 node connections via the SAA[10] specified user interface. This support, introduced in "Systems Application Architecture (SAA)" on page 5, is a quantum leap in usability for distributed systems and will bring many applications that previously were very costly to implement within the range of practical possibility.

Distributed Processing Example

The above example is typical of what many users wish to achieve. A corporate level data base has the authoritative copy of the data but distributed copies of local data are kept locally. Transactions are edited and formatted in local workstations. A small local server (here a PS/2) processes the user transactions against the local data base and also keeps a journal. In parallel, the user application handling the local data base enters transactions on to the main host system.

There are many, various systems flavors of this basic design. In this example, the work stations and local data base are all PS/2s on a token ring. The upstream host would probably be an IBM 3090.

The major point is that **the functions of LU 6.2 make installing a system such as this very much easier** than it would be without LU 6.2. For some users whose access to skills is limited, LU 6.2 can make applications like this possible. SAA will make it easier still.

But the facilities of LU 6.2 are built on the T2.1 node support and can only be separated with severe loss of function. Thus this was the major reason for T2.1 node support development in VTAM and NCP.

[10] Systems Application Architecture

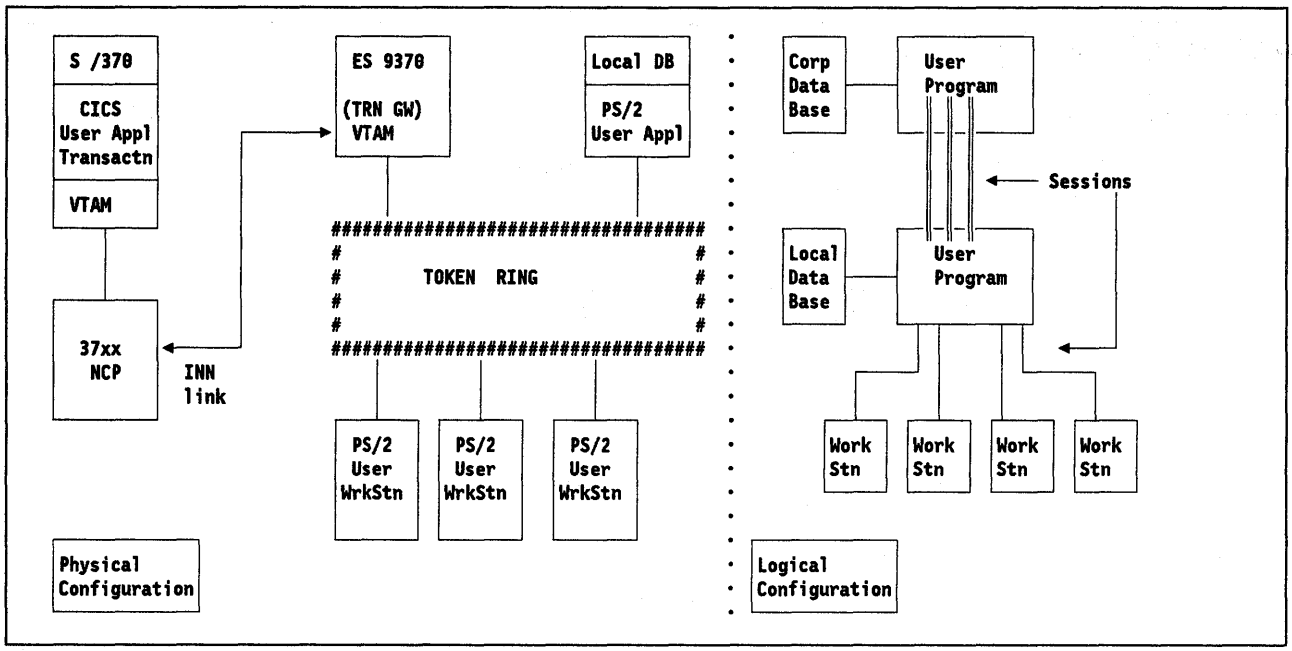


Figure 10. Distributed Processing Example. A small ES 9370 is shown here as a token-ring gateway to illustrate one possibility. Alternatively, the PS/2 acting as DB server could connect directly upstream using an SDLC link. Equally well, the ES 9370 could hold the local user DB and the DB server PS/2 would not be needed.

Network Sharing

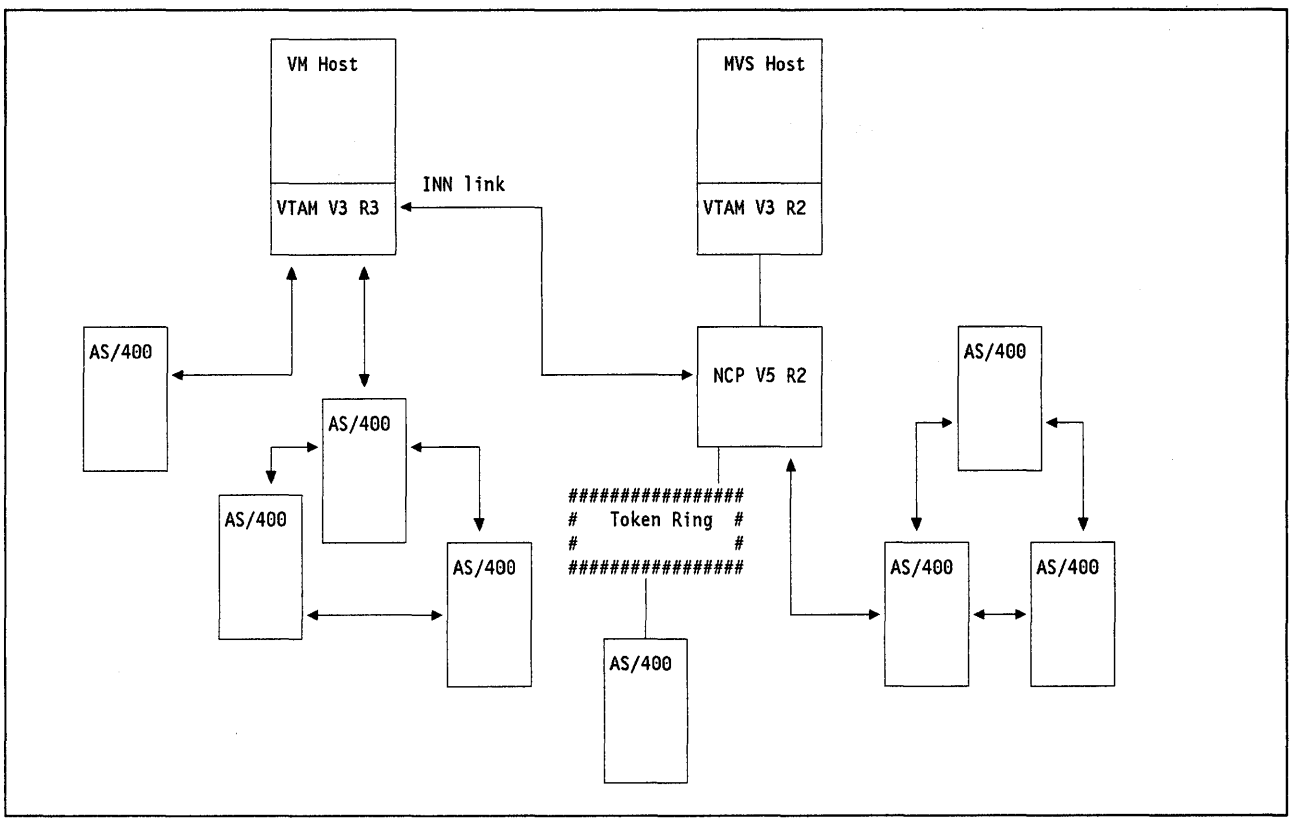


Figure 11. AS/400 to AS/400 Connection. Some possible connections are shown. In general, every AS/400 in the picture may communicate with every other AS/400 directly through the SNA subarea backbone. All other network functions are unaffected.

AS/400 to AS/400 Communication through the Subarea Network

This is the second major functional role envisaged for T2.1 node support in VTAM and NCP. Figure 11 illustrates this function.

The SNA subarea network can now be used as a shared transport medium between traditional SNA subarea network traffic and AS/400 to AS/400 APPN traffic.

This is a major function since many user organisations have a mixture of System /370 equipment and AS/400's.

Later in this chapter at "The "Universal Terminal" Concept" on page 34 it will be seen that this principle may be extended to cover access to many different kinds of host system.

It seems unlikely however, that the facility of "network sharing" will ever be used alone or in isolation. There will usually be a requirement for communication between AS/400 applications and System /370 mainframes as well as a requirement for terminal access to AS/400's through the network.

Terminal Access to AS/400 through the Subarea Network

This is an important function in any shared network.

An example of the function is provided by the "5250 Workstation Feature Program" (WSF) running on a PS/2 under control of OS/2 Extended Edition Version 1 Release 2. This gives the PS/2 the ability to be an interactive terminal to an AS/400 (or System/36) attached through the subarea network.

The PS/2 may be connected to the subarea network using SDLC, X.25 or Token Ring connections. It should be noted however, that this wide area network connection will *not* function if the PC in question is connected to the SNA network through an OS/2 EE 1.2 SNA LAN gateway or through an IBM 3174 remote LAN gateway. This is because any-to-any connectivity in the SNA subarea network requires APPC (LU6.2) to be an independent LU and these gateways support only dependent LUs. For more information on TRN LAN gateway use in this environment see Chapter 6, "Token-Ring (TRN) Connection" on page 83.

It is important to note that it is possible to have 5250 emulation sessions (with up to 5 different AS/400s) simultaneously with 3270 sessions (with up to five different S/370 based applications). The PS/2 user is able to "hot key" between access to AS/400 and access to S/370 systems.

"Non-SNA" Network Sharing

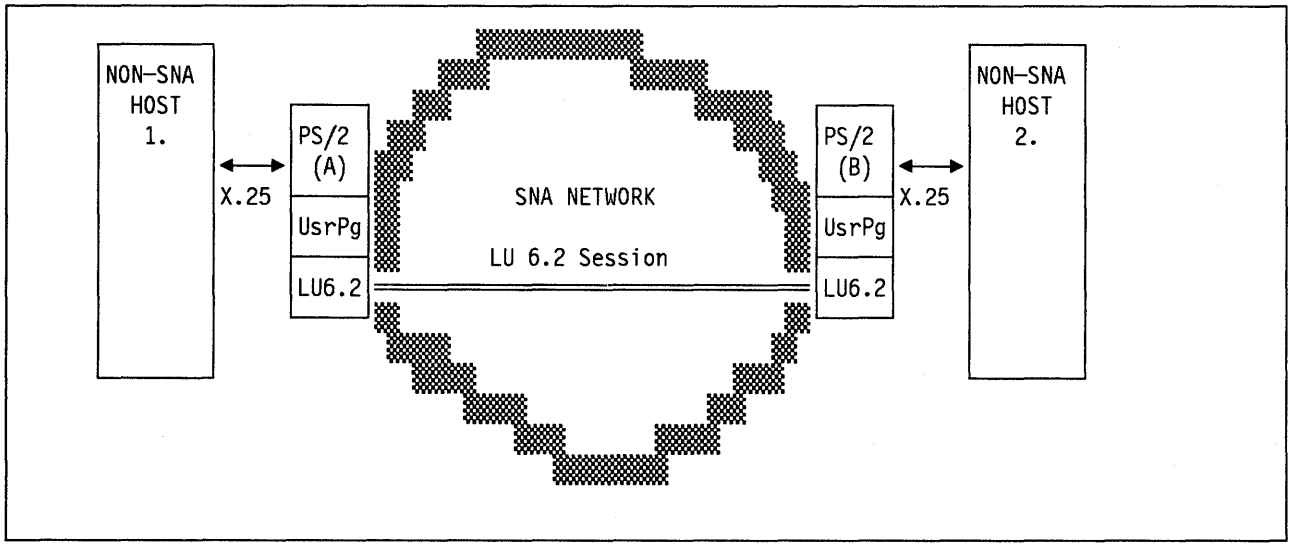


Figure 12. Non-SNA network sharing using PCs for attachment

The requirement to share an SNA network with other non-SNA traffic is an increasingly common one. Many users see large benefits in consolidation of multiple networks into one. Due to changes in technology the "line cost" reason for doing this is becoming less significant but the benefits of coordinated and consolidated network management are important both for cost and user service reasons.

Figure 12 shows an example of two non-SNA hosts communicating across an SNA network using PS/2's as interfaces to the network. As shown here the hosts are using X.25 as their connection protocol. This makes the SNA network appear to be an "X.25 Network" to the connecting hosts. The following points are important:

- There is an IBM product called "X.25 SNA Interconnect" (XI)[11] which allows an SNA network to carry X.25 traffic but this product runs within a 37xx communication controller and therefore is hard to justify in small, remote locations. However, XI is a formal, supported product and the approach described here is merely a technique and (to the knowledge of the author) no code using this technique is yet available on the market.
- This technique would be especially useful where the connecting PS/2's had other applications to perform.
- There is no reason that this technique should be limited to X.25 attachment. Cross-network transport of arbitrary protocols can be possible with this kind of technique. This would allow (for example) the transport of SDLC-non-SNA protocols and the like.

  When using *any* form of protocol transport across any kind of packet network it is essential that the timing characteristics of the protocol *and of the attaching processors* be carefully studied. Many protocols change as their timing characteristics change and many devices which appear well-behaved on a local link do not behave so well when the timing changes.
- There is an IBM "Special Bid" product called PC/Mux (available by special bid through IBM Systems Integration Division) which provides protocol transport (and many other) functions in a way similar to that described above.

  PC/Mux is an existing product and does not use T2.1 node protocols for attachment to the network but rather attaches as a T2.0 node. It is able to get cross-network routing (similar to the function provided by independent LUs) by using the IBM product "NRF" within the 37xx to relay the data between outboard PCs.
- LU 6.2 protocols are inherently half-duplex in nature reflecting their role as a higher-layer (ISO layer 6) end-to-end protocol. For protocol transport of any kind (including X.25), a full-duplex connection is required because there is no way of knowing the data flow characteristics of the data being transferred between systems. (Protocol transport is an ISO layer 3 function.) Therefore if LU 6.2 protocols are to be used for this connection then two sessions are required for each cross network connection. (This is no problem as parallel sessions is a characteristic of LU 6.2.)

  Some people believe that a much simpler LU protocol such as LU type 1 should be used for this kind of transport. (There is a subset of LU 1 that gives FDX data transfer.) However, there is no available support for LU 0 or LU 1 as an independent LU on the PC/PS.

The T2.1 node functions of NCP and VTAM offer some exciting systems possibilities for this kind of interconnection.

---

[11] XI and its use within an SNA network is described in *Integrating X.25 Function into Systems Network Architecture Networks*, GG24-3052-1.

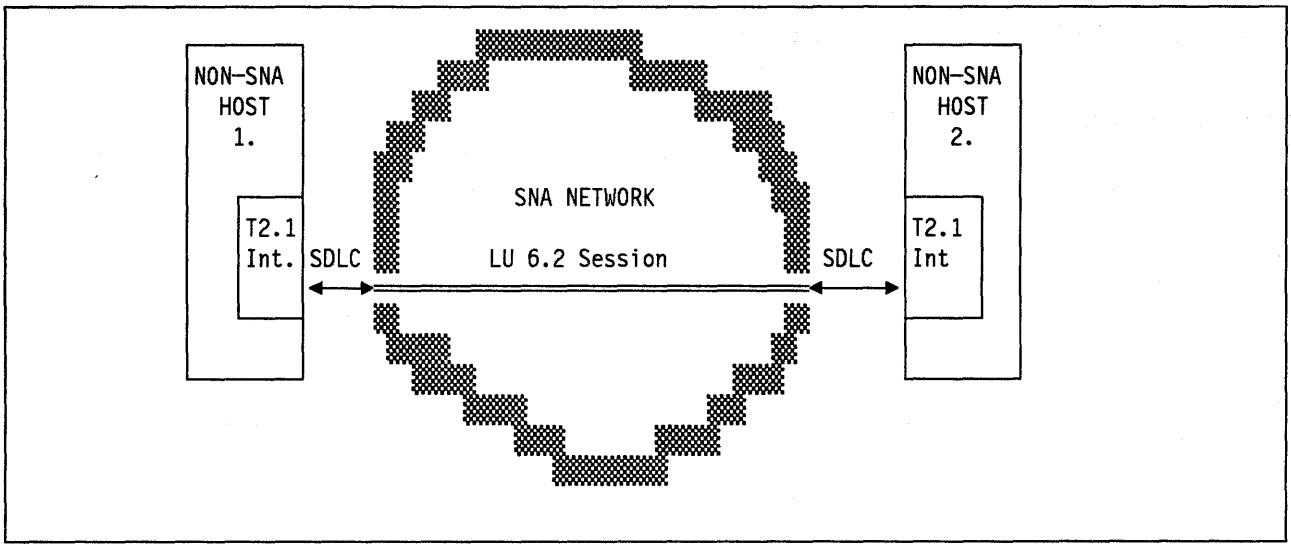


Figure 13. Non-SNA network sharing

Figure 13 is almost a contradiction in terms. It shows the connection of two non-SNA hosts together using the SNA subarea network for transport by implementing T2.1 node protocols within those hosts for attachment. It would be possible to argue that the hosts are no longer non-SNA by virtue of their having an SNA attachment capability. However, the meaning is clear.

The T2.1 node interface to the subarea network is clean, clear, relatively simple and fully published. The IBM announcement letter 286-410 of 18th September 1986 (US Announcement) declares the "Low-Entry Networking or Type 2.1 node" to be one of the "Open Communication Architectures" of IBM. That announcement states:

> "The major goals of opening SNA are to enable telecommunication users and vendors to ... Interconnect communication networks..."

In a migration situation where a user is changing from an older type of equipment to a newer, the best way of integrating communications networks will be through the use of PS/2's as protocol converters or protocol envelopers.

This is because:

1. After the conversion the PS/2's become available for other purposes.
2. It is often difficult when working with older style equipment to change or add interfaces to that equipment.

In the case where a user has different types of equipment which is planned to exist side-by-side for an indefinite time, then the implementation of T2.1 node protocols within the non-SNA equipment may be considered as an implementation alternative.

Casual Network Interconnection

Electronic Data Interchange (EDI)

One application area that is rapidly growing in importance around the world is the electronic transfer of information between organisations. This is often done by using third party "Value Added Networks" (VANs) such as the IBM Information Exchange (IE). However, it may also be performed "directly" (over a dialed connection) without the intervention of an intermediate VAN. For example:

- A manufacturing organisation may wish to send orders, invoices, shipping notices and the like to its customers and suppliers in electronic form. This is one of the fastest growing application areas because it saves the rekeying of data from orders or invoices into the destination system.
- A shipping company or an airline may wish to have shipping documentation presented to the customs department *before* a ship arrives in port, enabling a faster customs clearance.
- A group of banks may wish to exchange value data with each other electronically. In worldwide terms the banking industry is a leader in this area because they started earlier than other industries by exchanging "clearance" data on magnetic tape. This exchange of tapes caused them to set industry agreed data formats, etc. and when the opportunity came to send the data directly they were well-positioned to take advantage of the new system.
- In a similar way to that mentioned above, a bank may offer a service to its customers for the electronic processing of payrolls.[12]

This, of course, is one kind of electronic mail and can be processed through any electronic mail system. However, the application characteristics and requirements are often sufficiently different to warrant a different type of systems solution.

Application Characteristics

The important characteristics of this application are:

1. Regular but infrequent transfer of small data files.
2. The data files are in a fixed, predetermined format.

   Many industries now have standards for document layout and coding intended to aid in electronic document exchange.
3. Any individual organisation has a large (but finite) population of communicating partners.
4. Processing is done in small batches. There is no requirement for real-time response to individual items. That is response time is measured in minutes or hours rather than in seconds.
5. Security is a concern as documents represent items of value.
6. A means of delivery confirmation is required to prove confirm receipt of important documents.

Principle of Operation

One approach to the design of this application is to use a public switched network (telephone or data).

The overall method of operation is simple:[13]

- Each organisation has an application subsystem capable of sending and receiving small files. This application is typically active at all times.

---

[12] This latter application is a standard one which is usually processed by the exchange of magnetic tapes or "floppy disks."

[13] It is interesting to note that this principle is used to very great effect by the informal network of hobbyist "bulletin board" users of personal computers. It is a very simple and very effective principle.

- When one organisation has a file to send to the other a call over the switched network is made and a link established.
- After link establishment the two application subsystems establish communications and exchange identification. At this stage, other forms of security such as encryption may be put into effect from application system to application system.
- The file(s) is sent from one application to the other and the receiving application stores the data securely on disk and sends an acknowledgement. For security and verification it is common to place a unique verification number in this response.
- When traffic is complete, the communication is terminated and the call is "cleared."

Existing Approaches

If both organisations are SNA users, it makes sense to communicate network to network.

This is done in a limited way already. The only standard IBM product available which allowed this before the T2.1 node support in VTAM and NCP was the X.25 attachment product NPSI. This allows casual connection of the kind needed by supporting "X.25 non-SNA" connection between users over a public packet-switched data network. Each user sees the other as an SNA LU Type 1 but *no* SNA protocols pass between communicating users. Data is sent without any SNA headers. This gives complete isolation of one network from the other but requires the user to invent a set of end-to-end protocols to communicate between the application systems.

An IBM Special Product called "Establishment Communication Management System" (ECMS) is available in Japan. This provides special code in the NCP to allow a call to be placed using SDLC from one SNA network to another. Each network "sees" the other network as an SNA Node Type 2 with an LU type 0 session. The file transfer application works as described above. This software is not offered outside Japan.

Casual Connect Compared to SNI

The EDI application requires the infrequent connection of hundreds or thousands of other networks. The most important systems requirement therefore is simplicity and ease of use.

SNA Network Interconnection (SNI) is the existing way of connecting SNA networks with one another. SNI offers full SNA function over a secure connection between separate SNA networks. However, because of its high function, SNI also requires careful planning and installation. The systems programmers of both communicating organisations need to carefully plan the connection together. While this offers high function, it is often considered too complex to be used for the casual connection situation. Also, in the case where the user wants to have a switched connection between networks, the SNI switched connection must be initiated by a network operator command.

SNA Type 2.1 node connection between networks requires no such detailed planning. Planning and installing this connection is discussed in "Planning for a "Casual" Network Connection" on page 131.

Using the T2.1 node support a call may be dialed automatically when the application has data to send. It is often typical to initiate this type of application at fixed intervals based on a timer.

Security

In any dial connection situation, and especially where network interconnection is required, security is a concern. A discussion of ways of limiting cross-network access to intended and authorised connections is discussed in "Security" on page 115.

EDI Standards

T2.1 node support can provide a convenient, direct way of connecting two organisations together for exchange of information. Exchange of blocks of binary data will be of little use however, unless the communicating organisations understand the exact format and meaning of the data exchanged. This covers questions like the format of headers and trailers, the binary coding of alphabetic data, the length and format of numeric fields and the exact format of specific record types.

A clearly defined set of data standards is essential to successful operation.

Some organisations have established proprietary systems for direct data interchange and are achieving significant benefits. Few, however, have enough resources to install separate mainframe applications for every other company with which it wishes to do business.

The use of data standards facilitates electronic data interchange. Standard data formats allow data to be recognized by a variety of data processing applications while allowing participating organizations to maintain control over final document format. There are already many industry, national and international data standards emerging for EDI use. Some examples of these are:

- ANSI X12 based standards, the cross-industry data record format developed by the American National Standards Institute (ANSI),
- UCS, the Universal Communications Standard developed by the grocery industry, and
- ACORD, the data standard for the property and casualty insurance industry.

Any organisation planning to interchange data with another organisation (especially for trading purposes where it is value data that is being exchanged) would be well advised to seek out and adopt, *unchanged* the appropriate national and industry standard.

Casual Interactive Connection

Terminal Access to AS/400

This is the case of an AS/400 connected to one subarea network communicating with a PS/2 using 5250 emulation connected to another subarea network. This is shown in Figure 14.

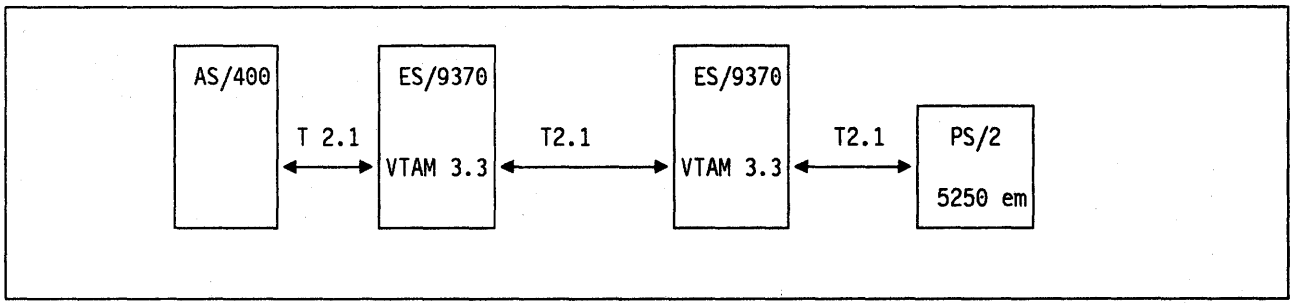


Figure 14. PS/2 to AS/400 through a Subarea Casual Connection

Since the communication uses LU 6.2 protocols and the PS/2 can initiate the session without SSCP assistance, this configuration will function quite well.

Terminal Access to Application Supporting 3270 Data Streams

This is the case of a 3270 application in one subarea network and a 3270 device in the other subarea network. Both subarea networks are connected together using "casual connection," T2.1 node protocols.

Because there can be no VTAM-to-VTAM SSCP session across the casual connection there is no way for the 3270 device to request a LOGON from the application.

See the discussion in "Dependent LU to Type 2.1 Node Networking" on page 75 and "Example of 3270 Logon to a Type 2.1 Node (TPF)" on page 76.

CICS-to-CICS for Infrequent Enquiry

When a CICS transaction requires access to a process situated within another SNA network "casual" connection is a potential solution to setting up the connection. If the user wants to have a switched connection with the other network and data traffic is very infrequent, then this method of connection may be appropriate.

- The major advantage is that the system can be set up in such a way as to automatically initiate a call when the transaction is required.
- Also, as was discussed earlier, casual connection is very much easier to set up.
- The disadvantage is that a connection must be established and sessions set up for (perhaps) a single transaction.

CICS-to-CICS for Remote Function

Casual connection is also appropriate in the situation where there is a permanent (or semi permanent) connection but relatively few sessions are required. The required sessions are application to application sessions.

A good example of this kind of connection is in the "credit bureau" application. An example is as follows:

- An automotive dealer has an "agency" for a credit provider such as a finance company.
- A customer wishes to purchase a car and wants to obtain finance from the finance company used by the car dealer.
- Typically, the finance company will place a terminal in the car dealership and the dealer's staff will operate it.
- During the online preparation of the credit application it is necessary to check with a credit bureau for the applicant's credit history. (This process differs in different countries because of the differences in legal systems.)
- It is common for the requirement to be expressed for the user (in this case the dealer) terminal to have access to the credit bureau. (This brings with it all kinds of protocol and attachment problems.)

  The application is often conceived as a requirement for the user to disconnect from the credit approval application, log on to a credit bureau for the credit check, and then resume with the credit approval carrying data from one process to the other by writing it down with a pencil and paper!

  A good answer to this situation is for the user terminal to have access only to the finance companie's credit approval application. The finance company would then, in turn, communicate with the credit bureau from application to application. This gives a much better interface to the end user and is more secure since the end user does not see the response from the credit bureau.[14]

---

[14] Many users are finding that expert systems offer a good way of processing this application.

- If both organisations use CICS (or any compatible host subsystem) then the connection between the finance company and the credit bureau could be a T2.1 node "casual" connection.

  Since there are only two communicating LUs (in this example two copies of CICS) with multiple sessions, the connection is very easy to install and to use.

In general "casual connection" can be a good solution where there are relatively few (probably host-based) LUs required to communicate across the connection.

NON-SNA Processor Access

X.25 "Split PAD"

One common requirement among IBM SNA users has been for access to non-SNA hosts of one kind or another from terminal devices in the SNA network. This requirement is often for interactive terminal log-on access from across the SNA network.

There are many approaches to this kind of need such as the use of the "GTM_OSI Pad Emulation Services" product which does a host-based protocol conversion from 3270 data stream to "ASCII terminal connected to a PAD" appearance.

T2.1 node support offers another method which offers significant benefits when compared to methods used in the past.[15]

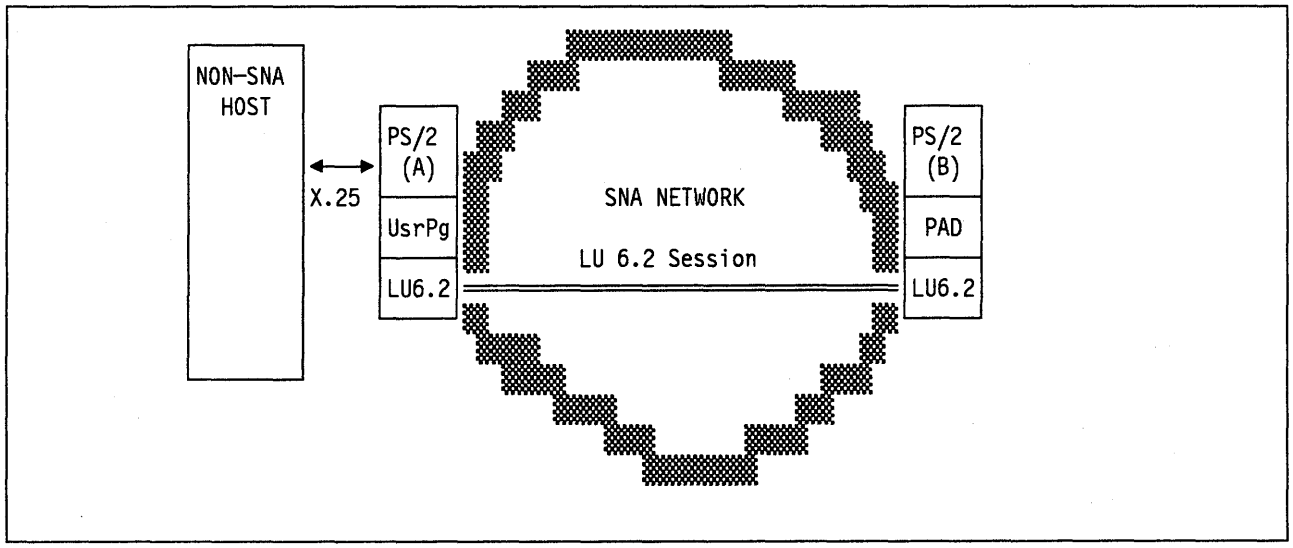


Figure 15. "PAD" Access to a non-SNA Host

In Figure 15, PS/2 (B) is operated by the end user. PS/2 (A) is a gateway connection from the SNA network to the non-SNA host using X.25 protocols.

[15] The technique described here is technically sound. It could be implemented using IBM supplied standard hardware and software. However, the function described is not performed by any IBM product currently available. The technique is discussed here to illustrate system capability and must not be taken to indicate any intent on the part of IBM to produce a product of this kind.

A program in B emulates a PAD-connected ASCII terminal[16] but instead of sending the data over an X.25 connection it sends the data to A using LU 6.2 protocols. The "UsrPg" (user program) in A takes the data from the LU 6.2 session and places it on the X.25 connection to the non-SNA host.

The non-SNA host "sees" an ASCII terminal connected to a PAD but the user is actually a PS/2 connected through the SNA network.

The advantages of this technique are obvious:

- The user's terminal (PS/2 or PC) may be connected to the SNA network (as a T2.1 node) in any one of the supported ways. (On an SDLC switched or leased connection, on a token-ring or through an X.25 connection.)

  This allows great flexibility in the user's network.

- There is a single, fully manageable and integrated SNA backbone network over which the ASCII/PAD traffic flows. This reduces the problem of building duplicate networks.

- The user's PC or PS/2 could contain other emulations and perform other functions simultaneously. This is further described in "The "Universal Terminal" Concept" on page 34.

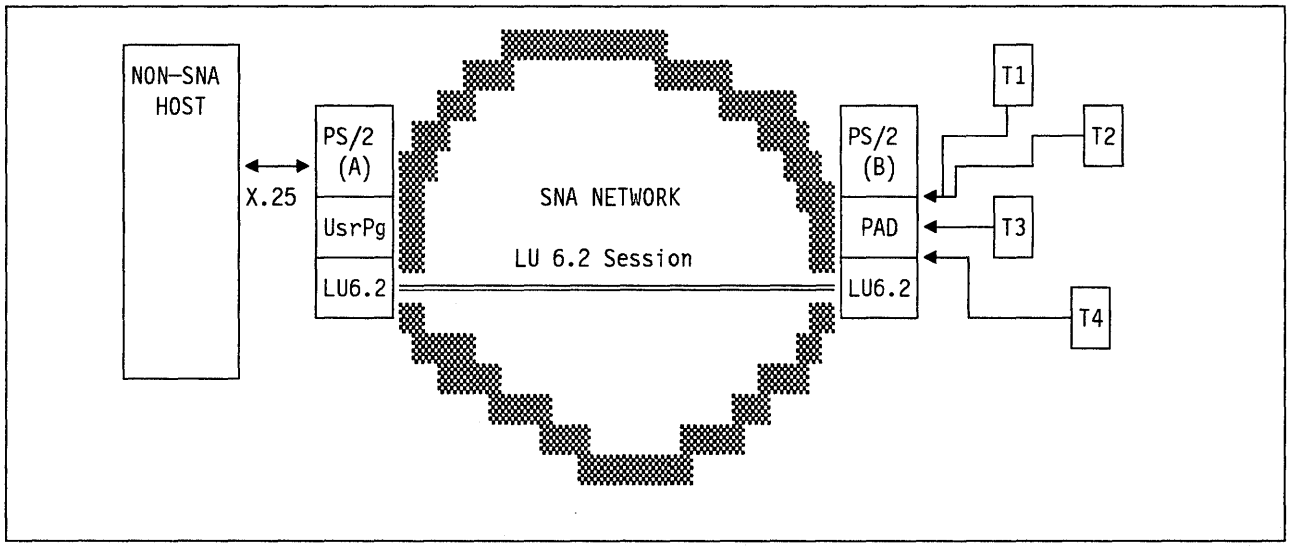


Figure 16. "PAD" Access to a Non-SNA host for ASCII Devices

Figure 16 shows the same *function* as was shown in Figure 15 on page 31. However, instead of the PAD function being performed only for the integrated screen/keyboard of the PS/2 involved, the PS/2 is used to connect external ASCII terminals to the PAD function. In this configuration a *standard* SNA network can perform the same function *exactly* as an X.25 network using PAD access.

A method of connecting the ASCII devices to the PS/2 is available in the form of the "Realtime Interface CoProcessor" (RIC) family of cards which allow up to eight lines to be attached per card.

Again, to the knowledge of the author, there is no product that performs this function in this way available on the market today. (Although there is at least one which comes very close.) The function is described here to illustrate potential only.

A similar function (already commonly available) is protocol conversion from ASCII to IBM 3270 protocols. A PS/2 application that allowed *both* PAD and 3270 access from an ASCII TWX device would be a very

[16] A short description of the PAD function is given in "The PAD Function" on page 174.

powerful way gaining access to multiple IBM and non-IBM host equipment from a single standard SNA network.

Other Terminal Emulations

The technique described above for PAD emulation could equally well be used to emulate *any* non-SNA device protocol provided that it had the following characteristics:

1. The protocol must be capable of operating using "block mode" transmission.

   This means that data must be sent in blocks and not character by character.[17]

2. The physical characteristics of the PC or PS/2 must be compatible with the device characteristics expected by the communicating host.

   For example, screen size and shape, and keyboard characteristics do not need to be the same as the emulated device but there must be an acceptable mapping from the virtual device to the real physical device.

3. The communicating host must be able to communicate with the emulated device type using X.25 protocols.

A user application program would need to be written which ran in the PS/2 labeled "B" in the diagram and emulated the device characteristics expected by the non-SNA host. Data would be sent to another PS/2 (A) using LU 6.2 protocols over standard SNA connections. The PS/2 (A) would have another user program which took the data and presented it to the non-SNA host using X.25 protocols.

Most computer suppliers support the X.25 connection of their proprietary screen/keyboard terminals. This approach could be used to provide input from PS/2's and PC to these non-SNA systems through a standard SNA network.

Of course, the upstream host connection does not need to be X.25. It could be whatever link protocol is supported by the "foreign" host. In order to support arbitrary link connections however, it will usually be necessary to write user code within the RIC[18] PS/2 attachment card. This can be a significant effort.[19]

OSI Virtual Terminal

In a way similar to that described above for both PAD and non-SNA proprietary systems, the same principle could be applied to implementing OSI virtual terminal protocols. This would give connection to any OSI host from PC and PS/2 devices through a standard SNA network.

---

[17] Some protocols (for example "ASCII TWX") as they are implemented by some applications require interactive response at the character level. It is often not obvious that this is a characteristic of operation unless the protocol is studied carefully. The principle described here would be very inefficient if each character had to be sent as a separate block.

[18] Realtime Interface Coprocessor

[19] However, the IBM "special bid" product PC/Mux offers a supervisory system which runs in the RIC card and facilitates user coding. Further information may be obtained from IBM Systems Integration Division.

The "Universal Terminal" Concept

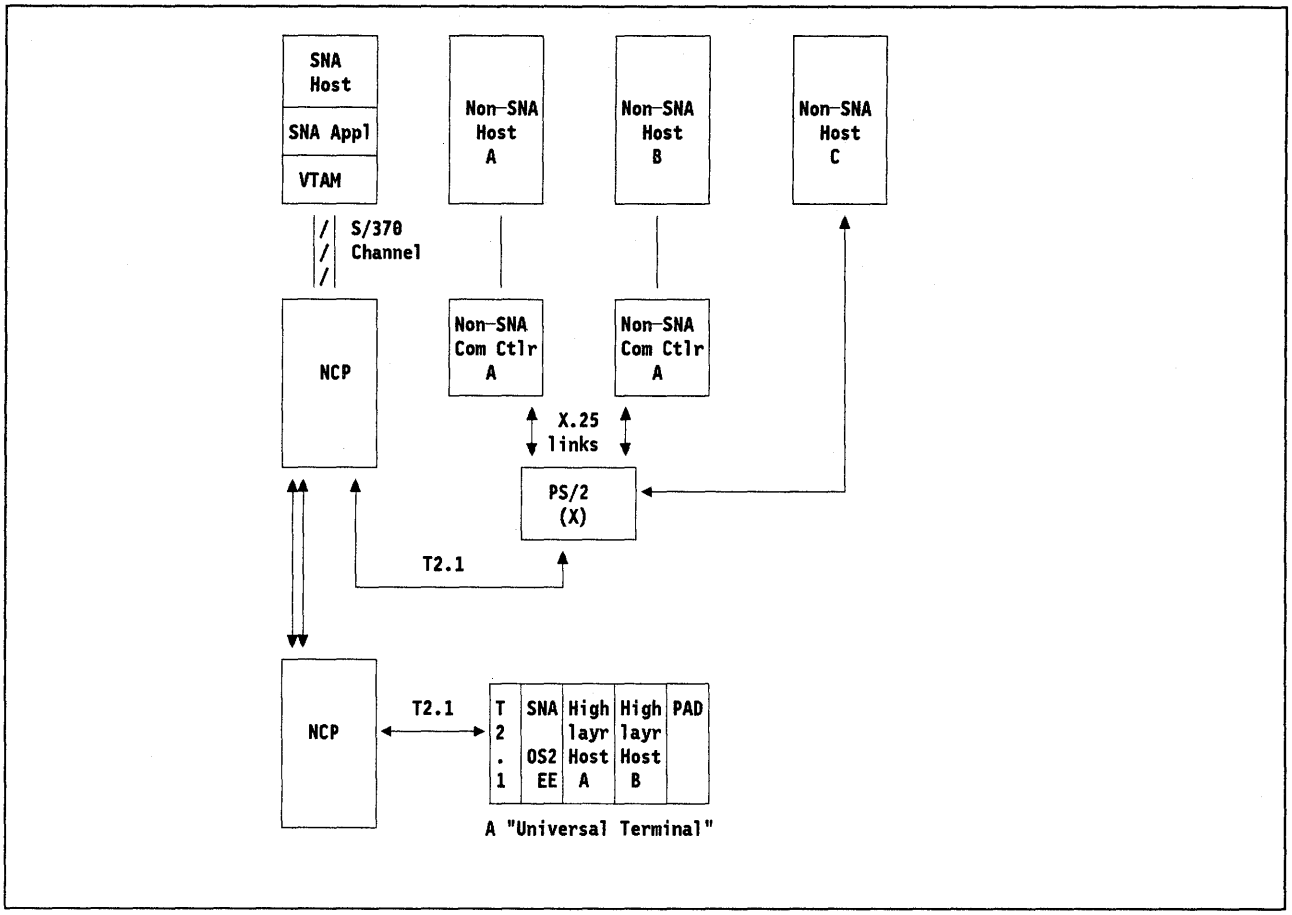


Figure 17. A "Universal" Terminal - Network Connections

Figure 17 illustrates a very common user requirement. The user has many incompatible hosts from different manufacturers. The leftmost one in the diagram is an IBM SNA host. The rightmost one is a "supermini" host that supports direct link connection via X.25 through to PAD-connected ASCII terminals. The other two are traditional "DP" hosts connected to the X.25 network through "communication front ends". Most manufacturers of such hosts now provide the ability to access their proprietary synchronous screen/keyboard devices through an X.25 network.

Many kinds of users, particularly in government require the ability to access many different kinds of processors from a single type of terminal. Since most manufacturers support connection of the ubiquitous ASCII TWX via a PAD then this could be the "universal terminal" in question but these terminals are very basic and this type of connection is regarded by some as reversion to the lowest common denominator. What is required is a terminal (small processor) that can obey (emulate) the protocols required by each manufacturer both for the higher layers of connectivity and also for the meaning of the data stream etc.

Such a device could be a Personal Computer. Figure 17 shows the structure of such a system. The PS/2 would use a multitasking operating system such as OS/2 EE and code for each protocol emulation required. Communication with the IBM SNA host would use both 3270 data streams in the usual way. Emulation of other non-SNA devices would take place as application code within the PS/2. A user would then be able to communicate with each system simultaneously perhaps changing between them via a function key.

While this configuration is an interesting technical possibility and while some users have implemented partial solutions, **to the knowledge of the author, no commercial product addressing these functions is available**

today. Discussion of this architecture here is done for the purpose of suggesting potential ways of solving a problem to users. It is not meant to imply that a commercial product implementation is planned.

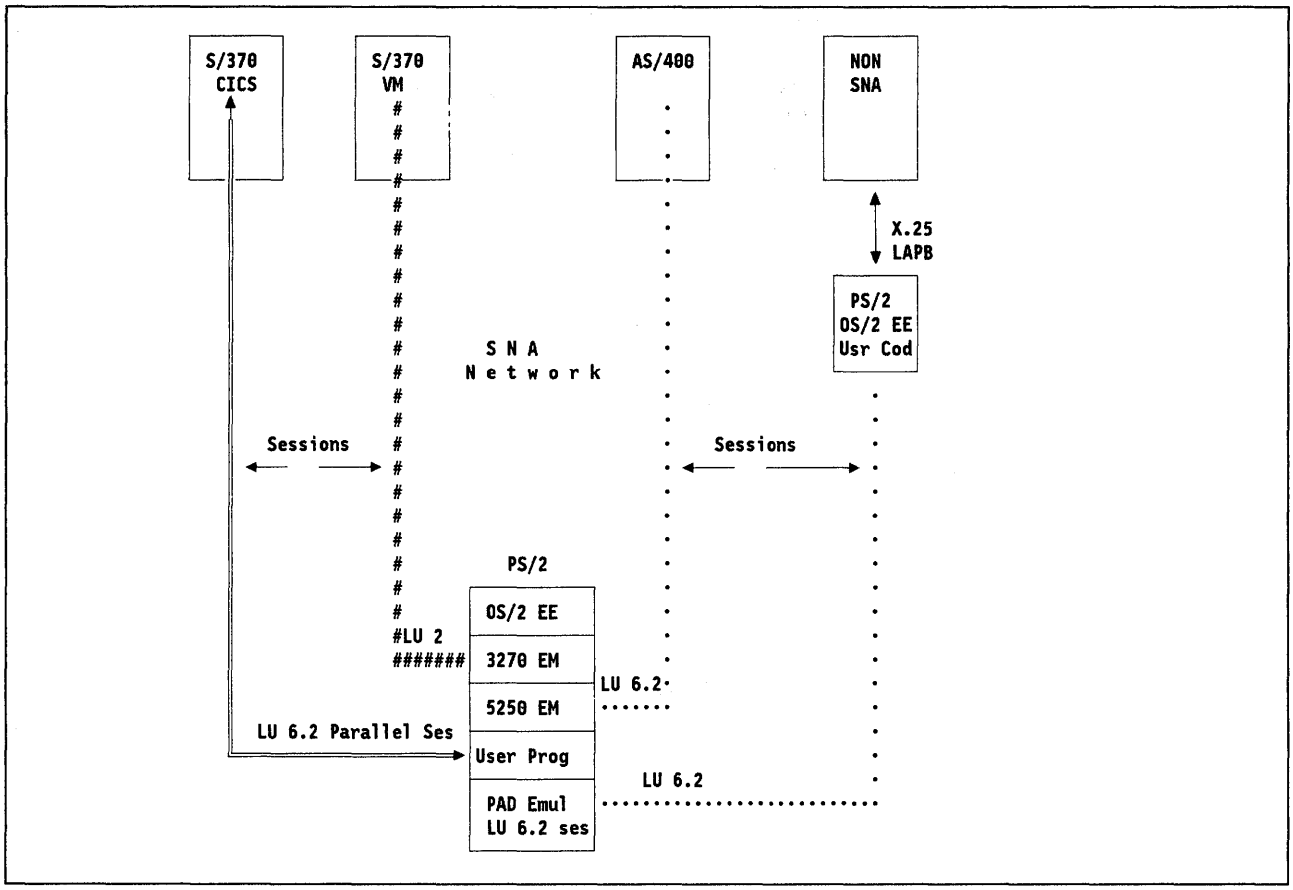


Figure 18. A "Universal" Terminal - Session Connections

Figure 18 shows a *potential* solution to the universal terminal requirement.

The physical network is not shown in detail since it is irrelevant to the function. All that is required is to set up the relevant sessions through the SNA subarea network. Connection to the network could be by any physical mechanism appropriate to each device. The PS/2 could be connected by SDLC, X.25 or token-ring network.

The functions provided are as follows:

- The PS/2 may have up to four simultaneous 3270 sessions. These are 3270 data stream and operate in a business-as-usual way.

  This connection is standard and supported by OS/2 EE today.

- The 5250 Data Stream on the connection to the AS/400 is carried on an LU 6.2 connection.

  This support is provided as a standard part of OS/2 EE Version 1.2.

- Communication with CICS could be one of the 3270 data stream sessions but is here illustrated as a set of parallel LU 6.2 sessions.

  This capability is supported by the current release of OS/2 EE.

- The X.25 PAD could be constructed as described above in "X.25 "Split PAD"" on page 31.

To the knowledge of the author there is no commercial product available today which performs the illustrated function. However, all of the needed system functions are available and the application looks attractive as a solution to many user requirements.

Other terminal emulations could also be present as discussed in "Other Terminal Emulations" on page 33.

Non-SNA host access example

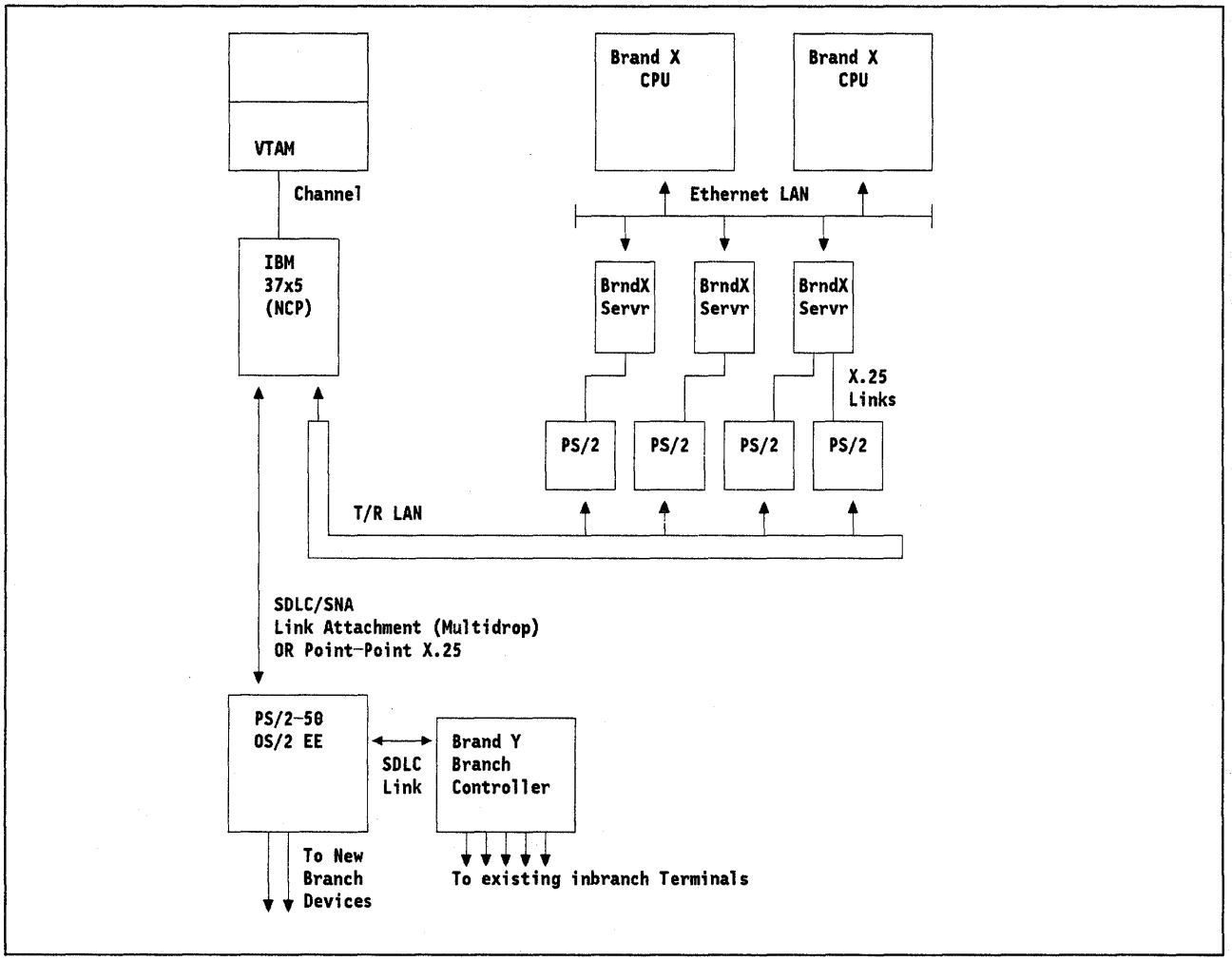


Figure 19. SNA Solution. The total network is SNA and the Brand X to Brand Y traffic is carried on LU 6.2 sessions.

The following is an example of a real user situation.[20]

The user is a large bank with two separate systems. One system is a retail banking system which was installed some years ago. The processors are non-IBM as is the network. The user has installed an IBM processor complex for additional branch applications and now has two networks going to each of over 500 branches. The user is unable to change the retail banking system for some time but wants to have a single network for all of the usual management, operational and cost reasons.

The "Brand X" network uses SDLC link control but *not* SNA. The IBM-based network uses SNA.

[20] In the discussion the terms "Brand X" and "Brand Y" are used to identify the non-IBM equipment.

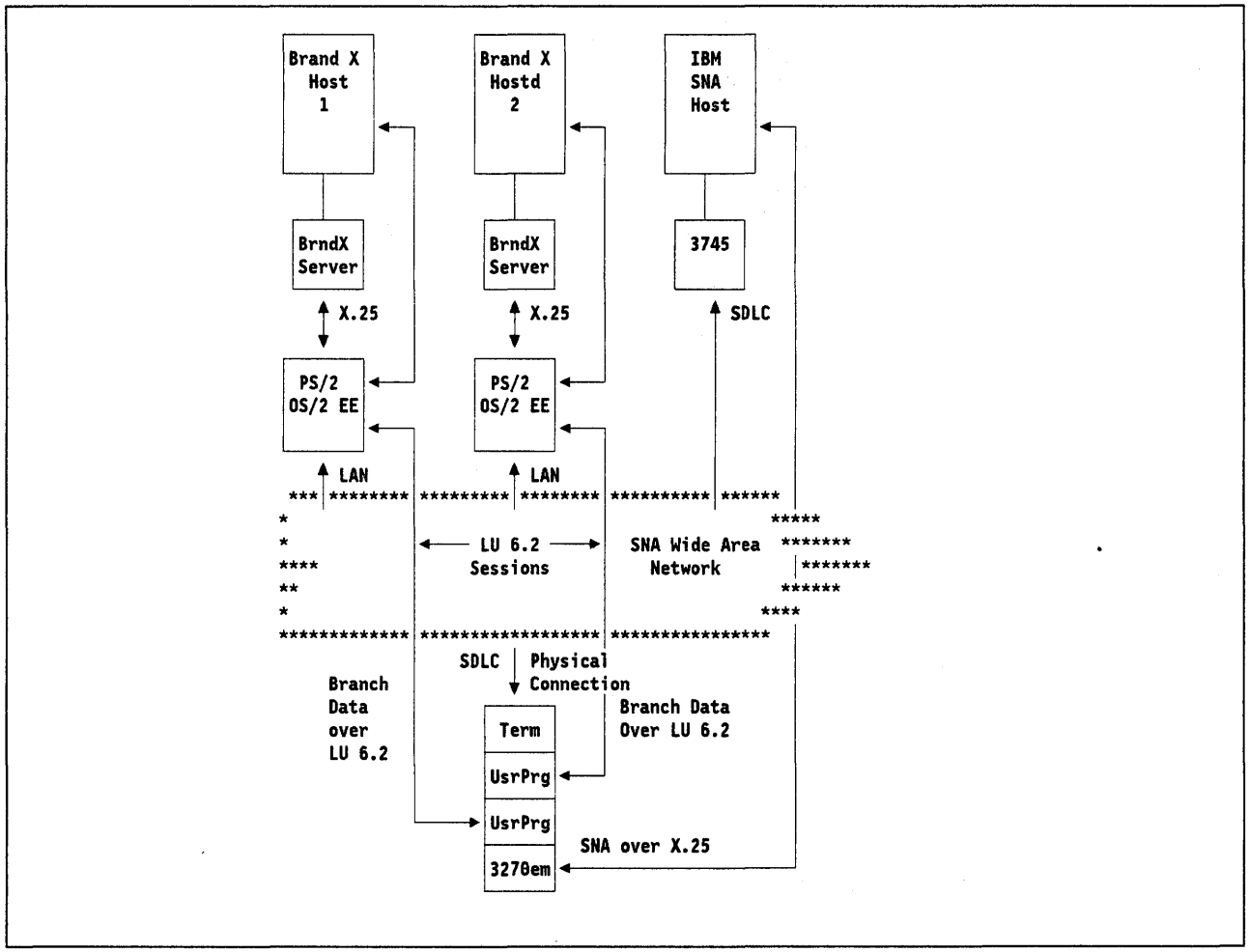


Figure 20. Logical Connections mapped to Physical Connections

The requirement expressed by the customer is for a wide area network using slow-speed (up to 19.2kbps) analogue telephone lines. This network should have the capacity for the transparent transport of data from any attachment point to any other regardless of link or network protocols used by the attaching device. The customer has expressed several central requirements:

1. There should be one coherent network to handle all the bank's traffic
2. This network must be transparent to all existing protocols
3. Protocol transparency should include future protocols that "may be of interest."

Many of the protocols used, such as IBM Financial B-loop, are highly timing-and-synchronisation dependant. Asynch ASCII protocols are also often used in a time-critical or logic-critical manner. Therefore the *only* kind of solution that can satisfy this specification exactly as it was written is a network of time division or statistical division multiplexors (TDMs or SDMs). Such devices derive slow-speed data channels from fast ones and allow for the overlaying of multiple logical networks over a single physical carrier. The advantage of this approach is that the equipment is simple, low-cost and reliable. Modern versions come provided with quite sophisticated network management systems.

In this specific customer environment it was considered that multiplexors would be a very short-term solution to the problem. In such a system:

1. There is no way of switching transactions between networks. (In the future it will be necessary to select which host individual transactions are to be sent to. This will need to be done either in the branch or in the network. A TDM solution will not answer this anticipated requirement.)

2. Network management of the individual physical network and the overlaying logical networks is not and cannot be integrated. This means that there are several disparate networks to operate and to manage. (This would also be true of an X.25 solution for example).
3. In the near future new Telecom offerings (such as ISDN) will change the whole character of data networking. TDM/SDM equipment installed now has no growth path to these future offerings.

The proposed solution rests on the functions of T2.1 node support. The logical connection is shown in Figure 19 on page 36.

The concept is that there is a single SNA network which carries all of the data traffic. This network uses standard and unmodified IBM hardware and software and can be managed as a single entity using IBM NetView products. The non-SNA data is carried over the SNA connections using LU6.2 connections.

Branch Configuration.

In the branch a PS/2 is used for connection to the SNA wide area network. The existing Brand Y branch controller is connected to this PS/2 using the existing SDLC non-SNA protocols. Connection to the SNA network is through two SDLC or X.25 link connections to the IBM 37xx controllers

The Branch PS/2's use OS/2 EE as their basic operating software system. OS/2 EE gives excellent functions in the areas required and while it may be possible to cobble some solution together using DOS at (perhaps) a slightly reduced cost, OS/2 EE is much easier and more cost effective to program, install, operate and manage.

The link between the PS/2 in the branch and its locally attached Brand Y controller is point-to-point using a "modem eliminator" at 2400 bps. In the PS/2 user code will need to be written for the interface. This user code controls one PS/2 "Multiprotocol Adapter" and support the Brand Y SDLC non-SNA protocol. (This is needed since OS/2 EE does not support SDLC attachments except through SNA.) In any case in this type of attachment the detail of how the SDLC protocol is used (it is very flexible) is quite critical and must be built into the attaching code.

The user program communicates with another user program in the "host interface" PS/2 via standard LU 6.2 protocols.

Upstream SNA link to the Computer Centers.

One requirement in the specification is for two hardware link connections per branch upstream to the computer centres. In this respect the approach is very flexible. Each PS/2 has two upstream SDLC links, each of which may be part of a multidrop arrangement with other branches in order to save costs, if required.

The SDLC attachments on a PS/2 have a recommended maximum data rate of 40kbps combined (meaning two 19.2kbps links). This is because the SDLC protocol has been implemented in a highly cost-effective manner by placing most of the link control in the PS/2 and thus minimising the cost of attachment hardware. The use of SDLC multidrop will allow for economical route design. Alternatively, X.25 point-to-point connection to the host offers a much higher performance option for large suburban branches.

Attachment of New Branch Devices.

One of the motivations for this tender is to allow the customer to attach additional devices in each branch that are *not* connected to the Brand X controllers. When new devices are required they can be either:

1. PS/2's (or PCs) connected to the branch PS/2 controller via a token-ring connection. In this case OS/2 EE has a token-ring gateway function built in.
2. Additional controllers may be situated in the branch and multidropped from the same lines as the branch PS/2 (through a "splitter" - behind the modem).

Attachment of IBM Financial B-Loop Protocol.

In this proposal the IBM cash dispensers remain as a separate network using their B-loop protocol as they currently exist. However, the PS/2's in the branches could be given the capability of attaching to IBM B-loop financial protocols. Provided the application code that now resides in IBM 4700 equipment control-

ling the ATMs can be implemented within the PS/2's, these controllers could be attached to the branch PS/2 also.

Another way would be to change the cash dispenser attachment to use SDLC and just multidrop them from the branch links.

Brand X Host Connection.

It is assumed that the user is able to reconfigure the servers to allow the use of X.25 protocols for connection as shown in Figure 20 on page 37. X.25 is the interface of choice between the IBM SNA network and the Brand X host.

Note: A user program in the IBM PS/2 host servers is required to interface the APPC protocols within the SNA network to the X.25 protocols used for interface to the Brand X equipment.

The major reason for choosing X.25 for this host interface was that X.25 can handle many virtual circuits (one for each connection to a branch). If the obvious solution of SDLC "enveloping" was used one SDLC interface would be required for each existing host SDLC connection, thereby increasing the number of PS/2s required at the host substantially.

This example was introduced to show some of the potential for systems solutions that exists as a result of T2.1 node support in VTAM and NCP. It is anticipated that many users will take advantage of the new facilities in ways similar to the above.

An SNA Type 2.1 Node VTAM network

Connecting two SNA networks together by "casual" connection is extremely attractive to many people.

- It requires very little planning
- There is no need to plan routes and distribute definitions etc.
- Connections can be made by autodial as needed.
- Nodes can develop and change software at their own pace without the constraint of matching software levels in other linked nodes.

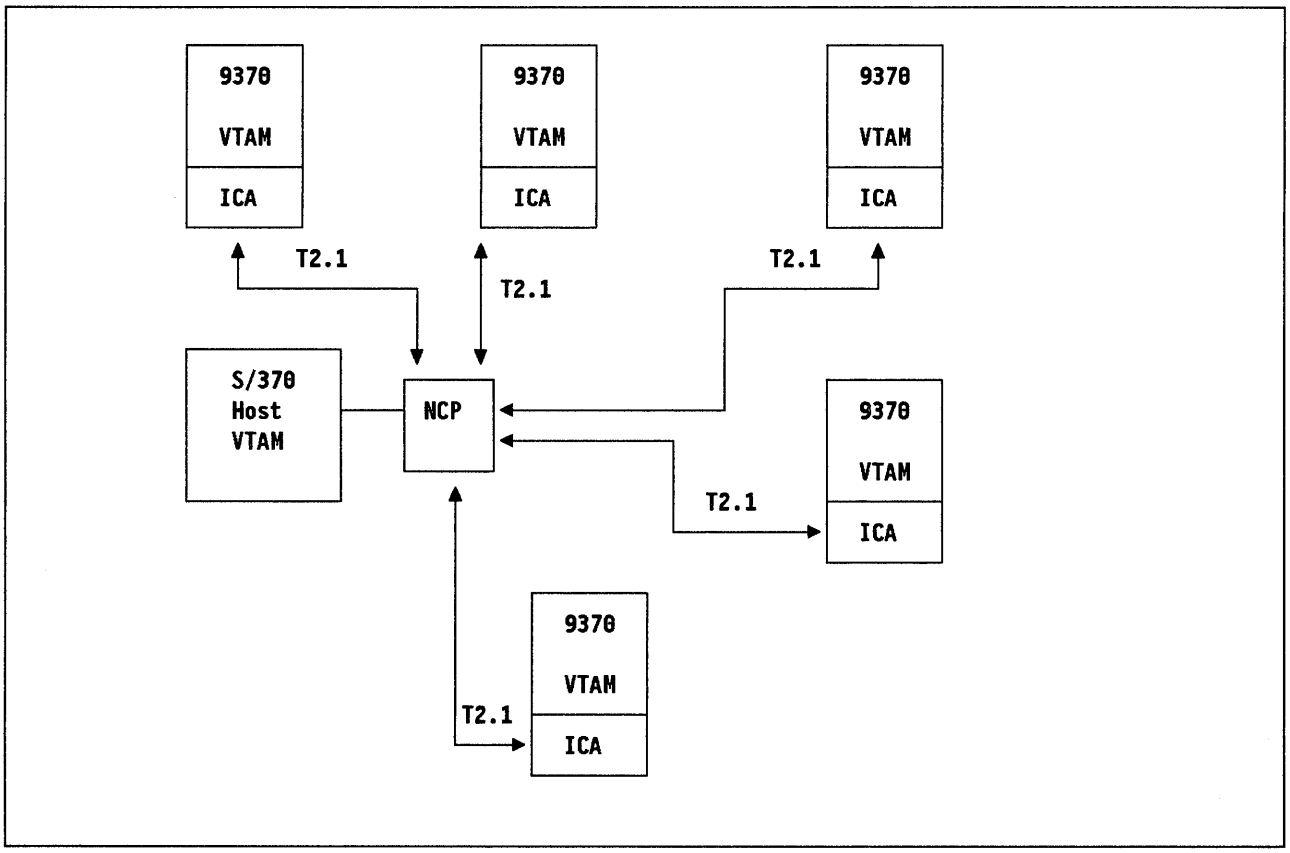


Figure 21. Hierarchical Network of Subarea Networks. All link connections shown are T2.1 node casual connections.

In Figure 21 a network consisting of a central VTAM/NCP mainframe complex is connected to a number of distributed 9370 systems using T2.1 node connection only. For example, the S/370 shown could be in a company that provides information or services to small companies which use ES/9370's. Also, a very decentralised business operation where a number of autonomous departments use ES/9370's (say for electronic mail) may wish to use this type of connection because of its simplicity.

In this configuration:

- Each processor in the complex forms a separate local SNA network which would need to be installed and maintained separately from each other separate, local, SNA network.
- There are no VTAM-to-VTAM sessions between the ES 9370's and the S/370 host (they are not needed since this is a casual connection, not an INN link).
- There is no NetView-to-NetView communication or centralised network management.

- Central site control of automated operations is not possible. In the examples cited, this is not required since each 9370 is controlled by its own autonomous user organisation.
- 3270 terminal traffic cannot "log on" to any host other than its locally attached one. That is, the user at the 3270 console cannot send a LOGON request. It is possible to use 3270 applications but only if the application has other means of discovering that the terminal user wants to log on. This subject is discussed in "Dependent LU to Type 2.1 Node Networking" on page 75 and "Example of 3270 Logon to a Type 2.1 Node (TPF)" on page 76.
- Host-to-host communication is possible if the host application requesting the session is the primary (actually initiates the session).
- The initiation of communication by the terminal to a non-local host is possible only for PCs using LU 6.2 sessions.
- There can be only one path or route between any two applications or end users. (No alternate routing is possible.)
- No priority selection for sessions is possible.

From the above it is apparent that T2.1 node connection is not and was not designed to be a replacement for regular SNA MSNF (Multisystems Networking Feature) or SNI (SNA Network Interconnection). The appropriate, strategic, architectural way for building SNA subarea nodes into networks is MSNF. The best way to interconnect SNA networks with each other is SNI. "Casual" network interconnection has a different purpose - as described above.

Nevertheless, some users will find that there is considerable advantage in using casual interconnection **if the level of function provided by casual connection is sufficient for the application.**

In general if host-to-host communication at the application level is required within a loose organisation of semi-autonomous user communities, then "casual" connection is probably the most appropriate method available.

For example, the case of a single conglomerate corporation where offices were distributed and independent and the requirement is for application-to-application file transfer traffic (such as a PROFS/RSCS application), with limited interactive terminal-to-distant host traffic such a network approach should be seriously considered.

It would also be possible to have a large network of interconnected SNA subarea nodes where the connections were arbitrary and numerous. Casual connection was not designed for this environment. MSNF does this job very well and should be used instead.

Section 2. Technical

Chapter 4. Technical Background

The traditional SNA Network (often referred to as an SNA subarea network), is constructed using IBM System /370 host computers and IBM 37xx communication controllers.[21] This kind of SNA networking system has been significantly changed and enhanced over many years.

In the early 1980s it became clear that a new form of peer-to-peer communication was required for distributed processing. Traditional SNA was organised for highly efficient operation in a hierarchically organised processing network design. This meant that much better end-to-end protocols were required within the network for end user peer-to-peer communication. In addition it meant that a new form of SNA network was required for the interconnection of "small" processors *without* the need for mediation by a large-scale System /370 host processor.

These requirements led to the development of:

- "Advanced Program-to-Program Communication" (called APPC or LU 6.2). APPC is central to the provision of improved end-to-end communication within SNA networks. However, full implementation of APPC requires a number of technical changes to the traditional SNA subarea network.
- L.E.N. (Low Entry Networking) is an architecturally specified means of direct, point-to-point connection of two (and only two) small SNA machines.

  Before L.E.N. architecture there were several SNA-like[22] peer-to-peer connection techniques available in IBM products, but they only worked in the specific situations for which they were built.
- APPN (Advanced Peer-to-Peer Networking) provides a new kind of network interconnection for small and intermediate range IBM systems. This allows the construction of arbitrary networks of "small" IBM machines without the intervention of a System /370 host processor. APPN has been implemented on the IBM System/36 and the IBM AS/400 System.
- SNA Type 2.1 node (T2.1 node) is the technical method of interconnection used by L.E.N. and APPN and is required for the full support of APPC functions.

In 1988 and 1989 the SNA subarea network has been enhanced to provide support for the direct connection of L.E.N. nodes, the interconnection APPN networks and for the full support of APPC protocols within the network. This is provided for VTAM/NCP based networks by VTAM Version 3 Release 2 and NCP Version 4 Release 3 and Version 5 Release 2. For the IBM 9370 processor (which uses integrated line attachments) a new release of VTAM (Release 3.3) provides the new functions.

This chapter discusses some of the technical "background" which gave rise to the development of T2.1 node support in VTAM and NCP.

---

[21] The notation "37xx" is used throughout this document to refer to the family of IBM communication controllers including the IBM 3720, 3725 and 3745.

[22] Incompatible extensions and variations of SNA

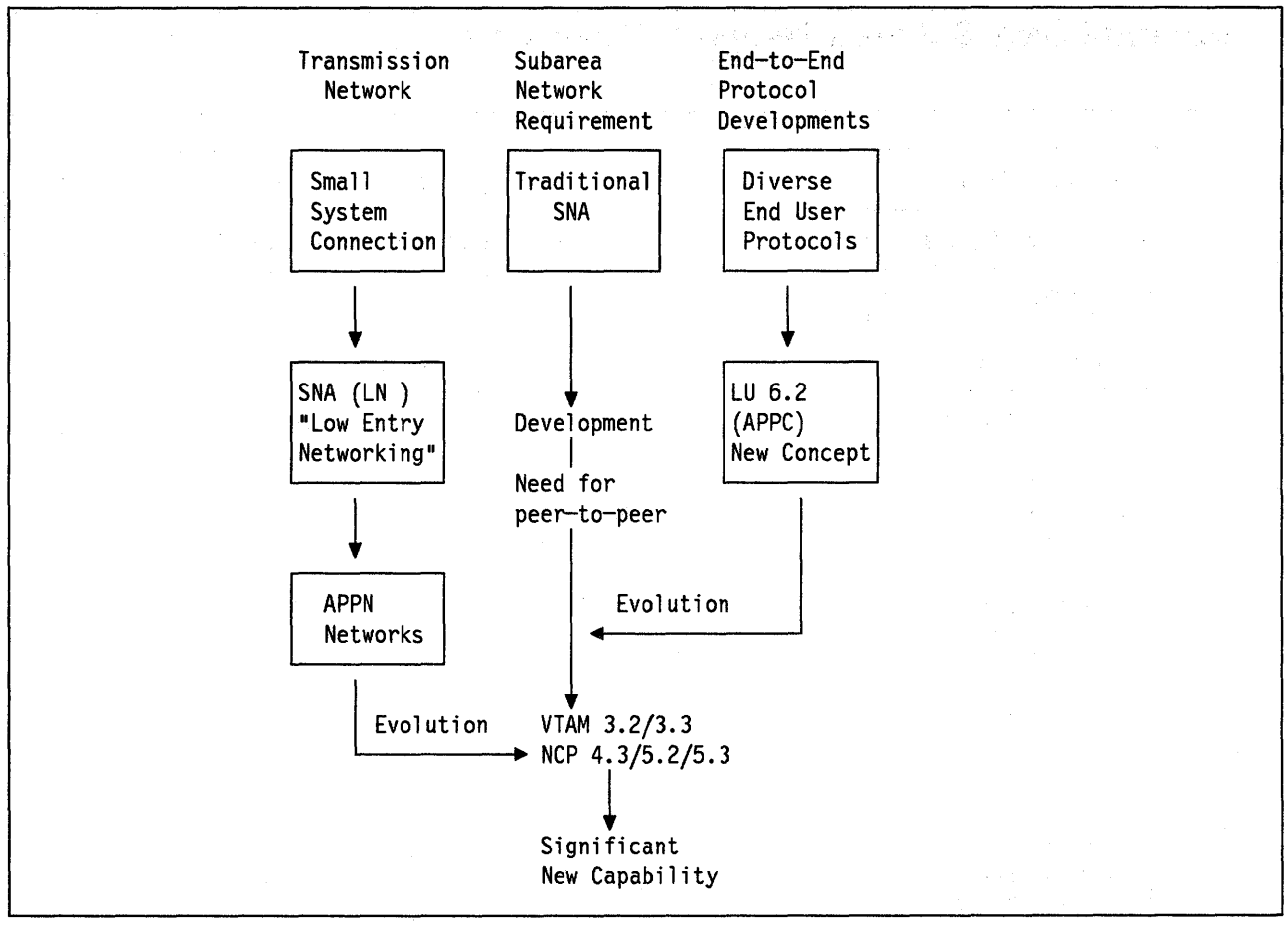


Figure 22. Development Roadmap

Figure 22 shows the general logic the discussion in this chapter will follow.

In the transmission network before T2.1 node there were several different methods of SNA peer-to-peer interconnection.

This was replaced by SNA Low Entry Networking (L.E.N.) which gave a single architecturally defined interface.

Later this was developed into APPN.[23]

In the SNA subarea network a need developed for small processors which were not S/370 hosts to be able to send data to each other directly through the network.

A uniform program-to-program protocol became necessary to allow for standardised communication between end-user programs throughout the network. This development gave rise to significant changes in the concepts of SNA. It also meant that the SNA wide area network had to change to allow for the new protocol.

The VTAM/NCP support for SNA Type 2.1 node provides for interconnection of APPN and L.E.N. devices through an SNA subarea network and for full functionality for APPC communication through the network. This means that SNA systems can be much more powerful and can perform many applications in new, more efficient, ways.

---

[23] Actually APPN exists in two slightly different but compatible forms: one on the IBM System /36 and the other on the IBM AS/400.

Traditional SNA Subarea Network (Pre-VTAM 3.2)

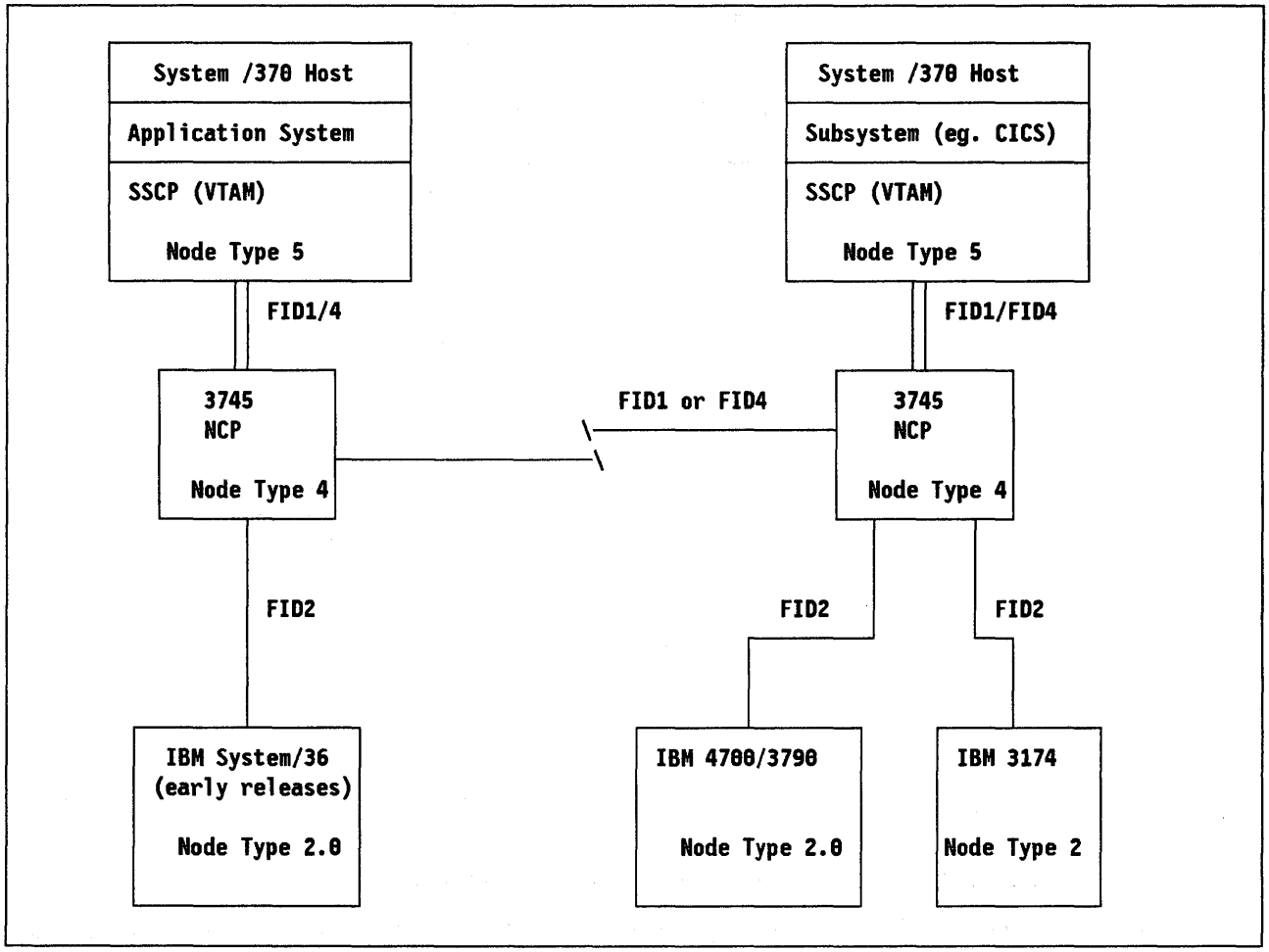


Figure 23. SNA Subarea Network

Subarea Network Structure

An SNA wide area network is constructed by interconnecting a series of SNA nodes. SNA nodes may be of different types depending on the functions they are required to perform in the network.

Node Types 1 and 2 provide services for LUs attached locally

Node Type 4 provides wide area network routing of data blocks and provides a network boundary function for attachment of node Types 1/2 to the wide area network.

Node Type 5 provides all of the function of a Type 4 node but also provides network directory, control and routing functions to the network. This control function is called a "System Services Control Point" (SSCP) and provides the necessary directory and control needed to set up connections between end users of the network.

Logical Units (LUs) are the communicating end points of the network. They can be regarded as end users or as automatons. See "Logical Units" on page 58 for a brief discussion of LUs.

Node types are distinguished by the range of functions that are provided to the LUs within their domain.[24]

SNA Type 2 nodes (such as the 3274, S/36, in the figure) provide services only to the LUs within their node and do not provide any intermediate node session routing.

SNA Type 4 nodes (such as the NCPs) in the figure provide intermediate node session routing, but do not provide any session services facilities.

SNA Type 5 nodes (such as the hosts) illustrated in Figure 23 provide both intermediate node routing and session service facilities for LUs.

SNA has been enhanced throughout the years and now current networks (post SNA Version 4.2) provide facilities to enhance network resource availability.

Explicit routes, virtual routes and transmission groups allow the definition of multiple paths between LUs. Failure of links or nodes will cause alternate paths to be chosen for routing sessions through the transport network.

The more sophisticated transport functions are only available between Type 5 and Type 4 nodes.

Data that flows within the network has appended to it a header containing information needed to route the data correctly. Within the "wide area" part of SNA (between Type 4 and Type 5 nodes) a header containing extensive information for network-wide routing is needed. The format of an SNA data block is identified by a "Format Identifier" (FID). Data blocks (Path Information Units -PIUs) flowing between Type 4 and Type 5 nodes use a FID Type 4 (or Type 1 in early SNA). The FID 2 Format Identifier is used between Node Type 4/5 and SNA Type 2.0 node/SNA Type 2.1 nodes.

Software

In the SNA wide area network the functions are performed by a number of major software products:

Virtual Telecommunications Access Method (VTAM) is the control program that runs in the S/370 host. It performs the following functions:

- Provides access to the SNA network for application programs that reside in the S/370 host. Often the interface to VTAM is provided by an IBM-supplied subsystem such as CICS, IMS or TSO.
- Performs the network SSCP (Directory and Control) functions described above.
- Performs the routing and boundary functions of a Type 4 node for IBM mainframe hosts which use an Integrated Communications Adapter (ICA) for link attachment rather than an external IBM 37xx.

Network Control Program (NCP) is the program that runs in an IBM 37xx[25] NCP running in a 37xx communications controller performs the following functions:

- "Off loads" the function of terminating and controlling communication links from the host.
- Provides an interface to the host for the communications network.
- Routes data blocks (variable length packets) to/from the host.

---

[24] In early SNA documentation SNA nodes are called PUs (Physical Units). An SNA Physical Unit has an SNA name and an SNA network address. Type 2.1 nodes do not necessarily have either a "PU Name" or a network address in the subarea network. The terminology therefore has been changed from "PU type" to "Node Type" in order to include the new mode of operation.

[25] "37xx" is a generic way of identifying the IBM family of communication controller nodes. Current controller products that use NCP are the IBM 3725, the IBM 3720 and the IBM 3745.

- Routes data from link to link and thus performs the functions of remote concentrator node and route switch.
- Contains the boundary of the SNA transport network and as such it is the junction between local and wide area network parts of SNA.

NetView is the name for a control subsystem and integrated set of programs that provide network-wide management for SNA networks.

Subarea Network Functions

Logical units are implemented in products like CICS and S/36 that provide connection to the network transmission facilities used by the end user (like display terminals or application programs).

Sessions are the logical connections between LUs on which SNA transports the actual data that applications send and receive. They are logical entities that do not need to be aware of either the physical connections or the topological relationships between the nodes.

Subarea Network Characteristics

There are a number of significant characteristics to note about traditional subarea SNA in order to contrast with APPN. These contrasts need to be accommodated when the subject of interconnection of APPN and SNA subarea networks is discussed.

1. Initially, SNA networks were "static" in that they relied totally on a system definition and generation process for installation. This was a necessity in the technology of the time and fulfilled customer requirements in the very small (by today's standards) networks of the time.

   As SNA architecture has evolved, there has been rapid progress to add dynamism to many areas. However, the structure and framework of subarea SNA was designed for static definition and this means that many things need to change when interfacing to a dynamic structure such as APPN.[26]

2. Subarea network SNA is still strongly hierarchical in its structure. This was a parameter of the basic design at a time when the only system that could reasonably process the compute and data storage intensive tasks of network control and directory management was a host processor.

3. Subarea SNA has strong primary/secondary relationships in many of its protocols. This was introduced because such relationships greatly simplify error recovery processes and reduce the compute load needed to implement any given protocol. This lowered the cost of secondary LU (SLU) implementation at a time when processor cycles were expensive and reliable low-cost disk storage was a dream for the future.

4. Initially, the subarea network components (such as the T2.0 node) were designed to be managed locally and not require centralised network management. This changed very rapidly as it was realised that in very large networks many customers needed centralised network management. Today, for most devices, the subarea network can be managed in either a centralised or distributed way.

---

[26] In later chapters of this document subjects like adaptive pacing and extended bind are discussed. These are things that needed to be added to subarea SNA in order to interface properly to APPN.

Peer-to-Peer before L.E.N.

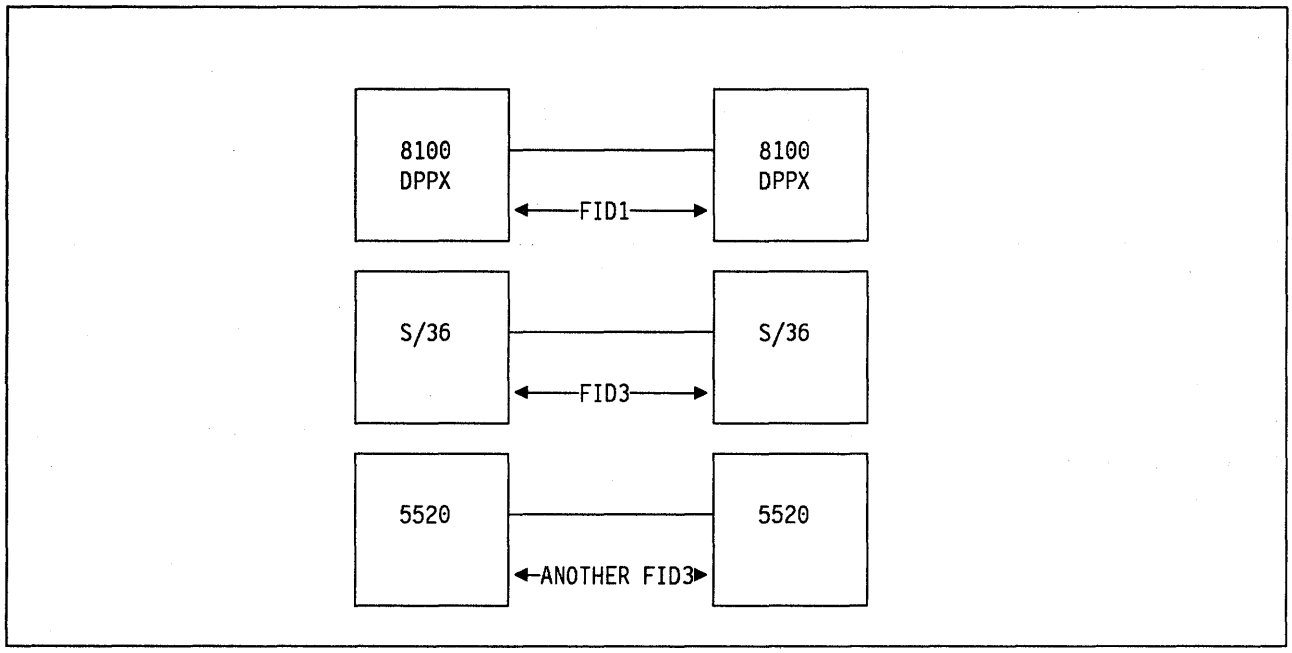


Figure 24. SNA Small System Connection before L.E.N.

Before SNA developed a formal architecture for point-to-point connection there were several, incompatible, SNA subsets in use for this purpose by various products. This is shown in Figure 24. Direct communication between these SNA nodes depended on the particular implementation.

In this example, the IBM 8100 uses a FID1 format for direct 8100-8100 communication, while the S/36 uses a FID3. Even between systems which use the same format identifier direct communication is not possible. For example, the S/36 and the 5520 both use FID3 for intersystem communication; however, they cannot communicate with each other because the session protocols were incompatible with one another.

SNA Low Entry Networking (L.E.N.)

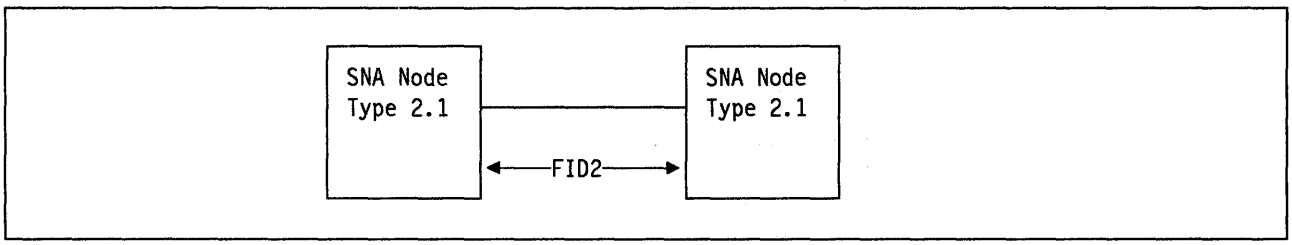


Figure 25. Low Entry Networking

In 1983, IBM defined the SNA Type 2.1 node[27] and the LU Type 6.2. This architecture defines a common set of protocols for peer-to-peer connection.

SNA Type 2.1 node permits arbitrary connectivity, using a modified FID 2, between a set of peer nodes. Some nodes that have implemented T2.1 node are the 5520, Displaywriter, ScanMaster, S/36, S/38, PC, PS/2 and the AS/400.

Since LUs inside a T2.1 node do not necessarily have communication with a network system services control point another method needed to be found to provide control point functions to the LU. A Single Node Control Point (SNCP) was defined which allows sessions to be established between peer nodes without the aid of an SSCP.

Although a node using the T2.1 node architecture can theoretically have any type of LU protocol within it, it was primarily intended that LU 6.2 protocols be used. T2.1 node complements LU Type 6.2 (and provides necessary additional function) in a peer environment. The LU 6.2 session partners have equal capability for initiating and terminating conversations, transmitting and receiving data, and performing error recovery.

L.E.N. Transport

L.E.N. nodes do not require the intervention of a host-based SSCP in order to communicate. In addition they can have multiple and parallel sessions between them and thus are able to support full-function LU 6.2 protocols.

There are still some restrictions that L.E.N. nodes have in a SNA Type 2.1 node/LU 6.2 environment:

1. A peer link node is either primary or secondary. Although there may be multiple links or a multi-dropped link attached to a T2.1 node, the link connections must be defined either as primary or secondary. This restriction can cause additional links to be required in order to achieve any-to-any connectivity.
2. L.E.N. nodes are incapable of intermediate network node routing, unlike the NCP and hosts.
3. Network definitions are pre-defined (static) like many of the host definitions.
4. Only a single link between any two L.E.N. machines is possible.

The combination of subarea and peer restrictions led to a set of requirements that set the stage for APPN development.

[27] At the time named "Physical Unit 2.1 (PU 2.1)"

APPN

Advanced Peer-to-Peer Networking (APPN) was developed because it was felt the traditional SNA networking, while highly functional, was too complex for the rapidly evolving small systems. Small systems had different requirements and different characteristics. Also, most important, in the ten years between the commencement of SNA development and that of APPN, IBM had learned a lot about data communications and decided to use that knowledge in developing APPN.

User Requirements

The important requirements that led to the development of APPN were:

Ease of use:

The most compelling need for any function involving small systems is ease of use. The history of office communications and of personal computers shows that if there is a large initial learning experience necessary before the system can be understood well enough to do useful work, then the system will tend to be used by an elite group and not by a wide class of users. This meant that the system must:

- Remove the need for coordinated system definition
- Require no full-time operator
- Require no specialized networking nodes.

Peer decentralized network control:

A peer-to-peer dynamic style of networking control was perceived to be important to many small system users who felt that the local user would like to maintain control over when their processor joined and left the network. This peer emphasis is a reflection of the way small systems are used in organizations. Large information processing hosts are usually the responsibility of information centres or headquarters operations, while small machines are usually found at the department or subdepartment level.

Arbitrary topologies:

As with large systems, the ability to configure any topology easily, rather than being restricted to a star, a bus or a hierarchy is an important flexibility advantage for the network growth and for cost or performance optimizations.

Connection flexibility:

Also as with large systems, a variety of connections should be supported, including LANs, X.25, ISDN, X.21, switched lines and leased lines.

Interworking with SNA subarea:

There were over 20,000 SNA networks of various sizes installed. It was expected that pure APPN networks would need to do interwork with SNA subarea networks more frequently than with non-SNA networks.

Design simplicity:

It was felt that conceptual simplicity and openness would be critical for users of small systems. By this, we mean that the design should be cleanly layered, easy to understand and modify, and well and clearly documented.

Continuous operation:

The need for continuous, nonstop operations has been expressed as a requirement for small systems in recent years, reflecting the emphasis visible earlier for large systems. It was envisioned that some

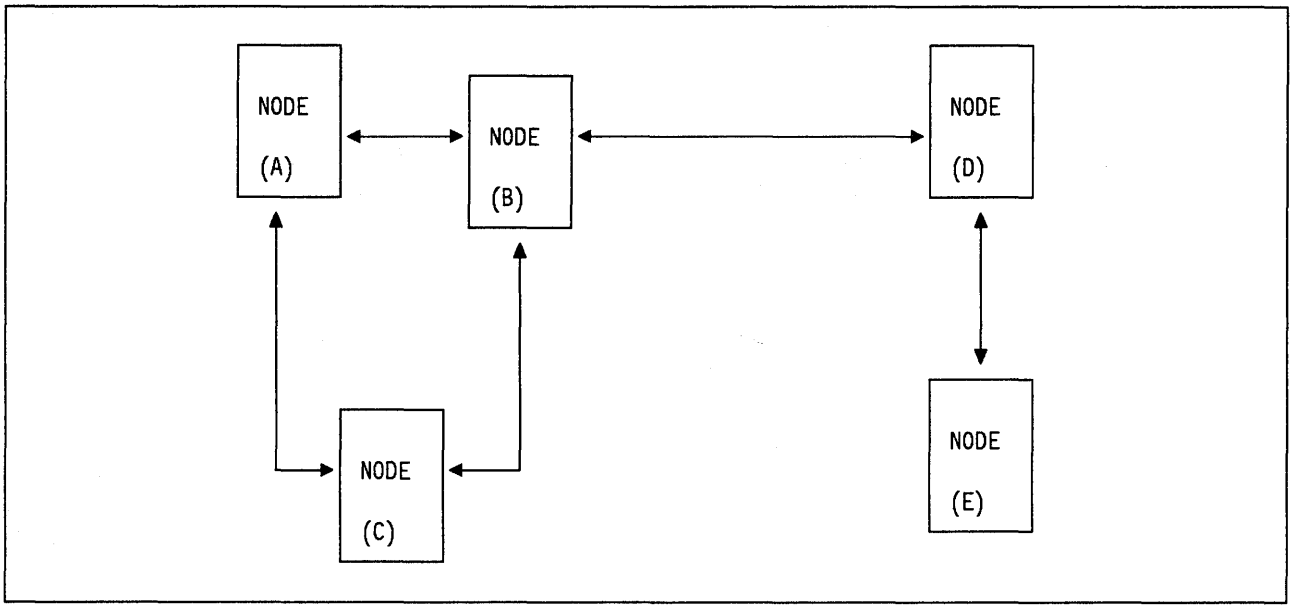


Figure 26. APPN Network

enterprises would commit their entire business to networks of small systems, just as they have in the past with networks having one or more large mainframes.

Cost competitive:

It was considered essential that APPN should be able to be implemented as an application within a using processor and not require the expense of a separate communications controller such as an IBM 3745. (The very high throughput obtainable from a 3745 is not a requirement here since the small processor cannot process that much data anyway.) This meant that the entry cost of using APPN should be minimal.

These requirements have largely been met:

1. APPN Networks are "peer" network that require no centralised host intervention.
2. Topology is arbitrary.
3. There can be any kind of link connection but this is limited by individual product implementations.

   Some of the connections supported by APPN products include: SDLC (over X.21, V.24, V.35 and ISDN connections), X.25, LAN, and S/370 channel.
4. It is dynamic and requires no fixed route definitions and the minimum of other fixed definitions.
5. Control is decentralised throughout the network. It resides in each full-function network node (NN).
6. Network Management may be centralised or decentralised depending on the user's requirements.

APPN Network Structure

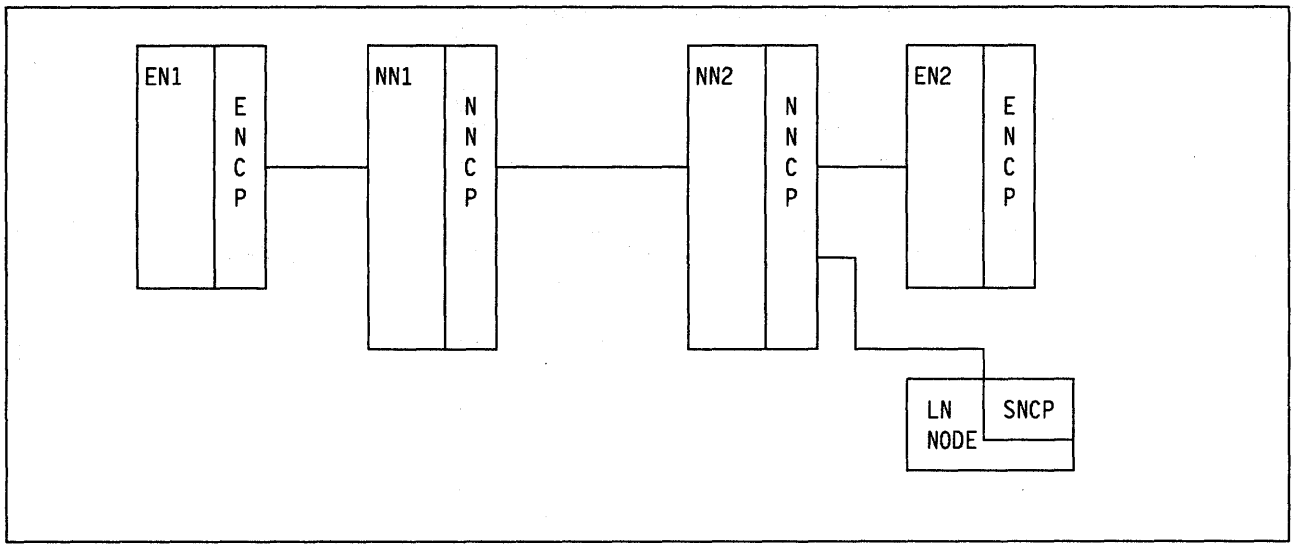


Figure 27. Function Definitions

An SNA/APPN network is made up of three types of nodes.

The L.E.N. Node, as discussed above, contains an SNCP but that control point does not have communication with any other control point. That is, the node has no knowledge of any network it may be attached to. All it "sees" across the T2.1 node interface is another L.E.N. node like itself.

A L.E.N. node is an End Node that has no APPN support at all. These non-SNA/APPN nodes can be attached to a NN and get part of the APPN benefits, (that is, sessions with non adjacent nodes).

The Peripheral Node was the original name for the L.E.N. node discussed above. Today the PN is a generic term used to mean both L.E.N. and EN style nodes. The term "Composite Peripheral Node" is used to describe the "virtual" node appearance of the subarea network at the T2.1 node interface.

The term Peripheral Node Control Point (PNCP) is also used as a generic term to cover both ENCPs and SNCPs.

The End Node (EN) is the same as the Peripheral Node of the SNA Type 2.1 node architecture except that a CP-CP session exists between this End Node and the Network Node owning it. It contains an Ending Node Control Point (ENCP) which will perform session services for its own LUs. An Ending Node can establish a session with another node without the aid of an SSCP.

The Network Node (NN) can, in addition to performing session services for its own LUs, also perform network services for attached LUs. These network services are routing and directory services. A network node can also provide routing services for all other LUs in the network. Network Nodes contain a Network Node Control Point (NNCP).

ENCPs and NNCPs are like mini-SSCPs in that they perform control point functions in support of their respective nodes. An NNCP can also perform some functions in support of directly connected L.E.N. and EN nodes.

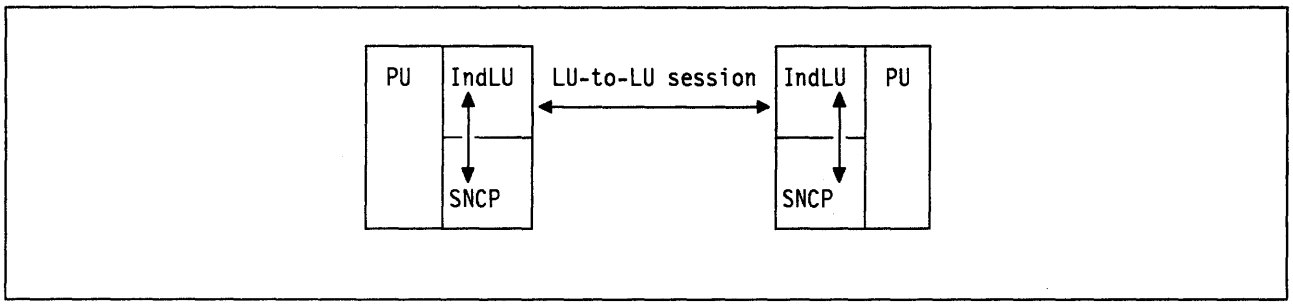


Figure 28. A L.E.N. Node Has Only LU-LU Sessions. The label IndLU identifies an Independent LU

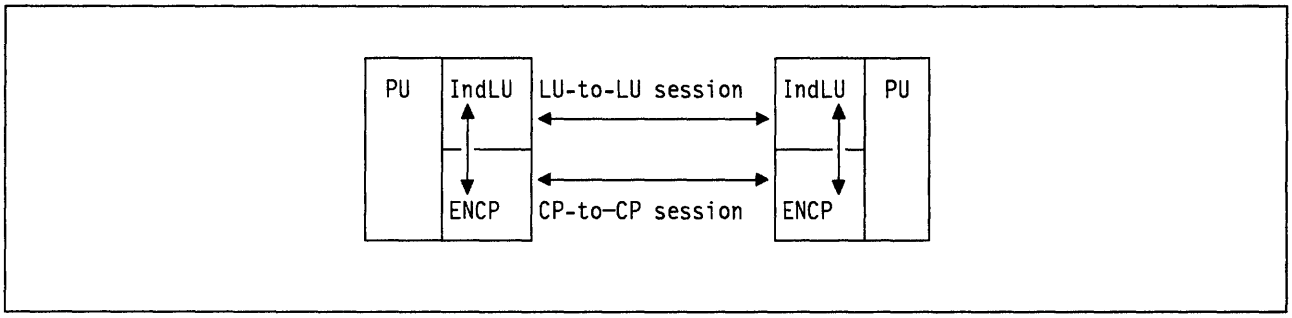


Figure 29. An APPN EN Can Have a CP-to-CP Session as Well

APPN Node Structure and Functions

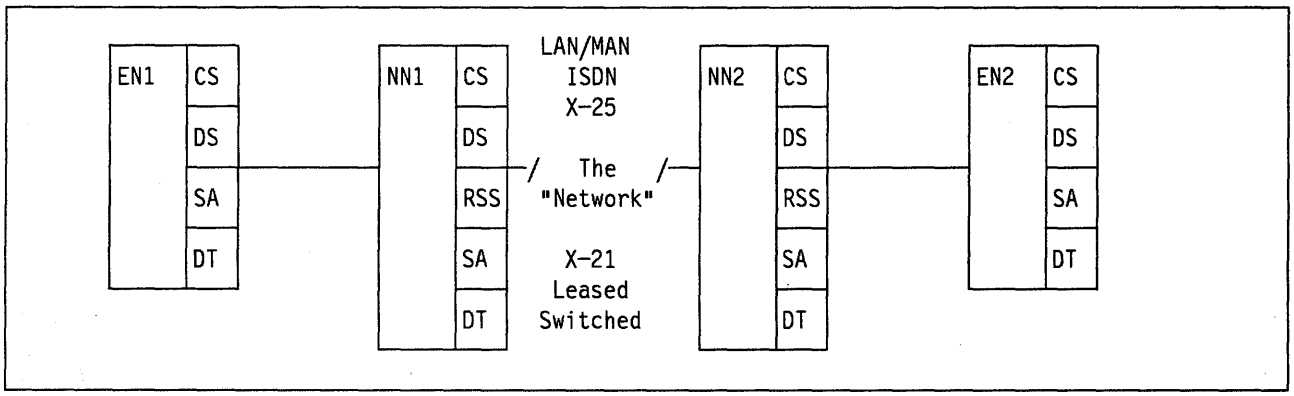


Figure 30. APPN Control Point Functions

Every node, whether it is an End Node or a Network Node, has within it a control point. This control point is responsible for managing and coordinating all the networking control functions performed by the node.

Network Nodes

Network nodes contain the following control point functions:

Connectivity Services

Activates a new link or node as part of the network.

Directory Services

Identifies which node contains a specified remote logical unit.

Route Selection Services

Identifies the preferred route to the remote node.

Session Activation

Establishes the session with the remote LU using the preferred route.

Data Transport

Manages traffic flows on the session between the local and remote LUs.

End Nodes

End Nodes contain the following control point functions:

Connectivity Services

Activates a new link.

Directory Services

Identifies a resource for its own LUs.

Session Activation

Establishes sessions on behalf of its own LUs.

Data Transport

Manages traffic flows on the session between the local and remote LUs.

Network Node Control Point Functions

Network nodes provide the full suite of APPN functions while the EN and L.E.N. nodes provide a subset of APPN functions. In particular, L.E.N. and EN nodes do not perform intermediate session routing. APPN enhances the transport mechanism of T2.1 by providing the following services:

- **Control Point Manager Services (CP)**

  A Control Point is present in each node in an APPN network, and effectively is the general manager of that node, controlling all APPN functions and communicating with other Control Points (CPs) in other nodes. The **Control Point (CP) name** uniquely identifies a node to other CPs.

  Network nodes in an APPN network must exchange information about the network topology (that is, information regarding the nodes and links in the network), and it is by establishing a **Control-Point-to-Control-Point (CP-to-CP) session** (between CPs in each node) that such information may be exchanged.

  Once APPN has established that a session may be routed through the network, the CP in each local node can **automatically create and start** the necessary device descriptions for remote locations.

  In summary, the control point performs the following functions:

  1. Overall control of APPN functions.
  2. Management of the CP-to-CP session between adjacent control points.
  3. Automatic creation and starting of device descriptions for remote locations.

- **Topology Routing Services (TRS)**

  There may be multiple possible routes between two nodes in a network. The function of Topology Routing Services (TRS) is to determine the best route to take. There are two kinds of information which TRS must compare in order to determine the best route over which a particular session should be established:

  - Topology database.
  - Class of Service Table.

  The topology database contains information about nodes and links in the network and their respective characteristics. Each node contains a topology database which is updated each time a new node or link is activated in the network or when node or link characteristics change. Updates are sent between nodes in the network via the CP-CP session that is established between them. Local node information is extracted

from three places: the network attributes, controller descriptions, and line descriptions. When a controller description is varied on the remote control point name and TG number represents a single transmission group and the characteristics from the line description that the controller description is attached to are used to represent the characteristics for the activated transmission group.

There are some nodes in an APPN network for which support for the topology database is different (end nodes and L.E.N. nodes); this is discussed in "APPN in a Network Containing EN and L.E.N. Nodes" on page 157.

The user defines a relative preference for certain characteristics of a link or node (topology characteristics) in a **Class-of-Service (COS) table**.

Thus, at session establishment time, TRS compares both of these two sources of information (COS table and topology database) and is thus able to place a relative preference on each possible route. The route chosen is the most preferred route.

The transmission of data may be prioritized by specifying a transmission priority to be used for a particular session. A transmission priority is associated with each COS table. When a transmission priority is specified at session establishment, APPN will automatically control or change how data is prioritized in the network according the transmission priority selected. Thus, data flow through the network may be better managed; for example, batch transmissions may be given a lower transmission priority than interactive users.

In summary, TRS provides the following APPN functions:

1. Topology database.
2. Route selection.
3. Class of service.

- **Directory Services (DS)**

  A node in an APPN network may have multiple **location names** or "nicknames" defined, by which it may be known, apart from its CP name. DS provides a database of nicknames (location names) for remote nodes in the network; this is called the **Directory database.**

  Thus, if a session is requested to a remote location, DS will either immediately recognize that name (by searching its local database) or will perform a search of the network (by contacting the CPs in other nodes) in order to find the CP which owns the requested location. Once a node's CP name is known then it can be uniquely identified and contacted. Thus the main task for DS is to find the CP name of any requested nickname.

  In summary, DS provides the following APPN functions:

  1. Maintenance of a local directory database.
  2. Inquires about information in the local directory database.
  3. Participation in distributed database searches through the network.

- **Intermediate Routing Services (DS)**

  The transport layer in each NN provides the APPN support to enable non-adjacent nodes to appear adjacent; that is, perform **Intermediate Routing** for sessions in which it is not the origin or destination.

  At the same time that a NN is providing intermediate routing it may also be a session end point for other sessions.

APPN Operation

Figure 26 on page 53 shows five APPN Network Nodes (NN).

When the links are being established, A to B to C (in Figure 26 on page 53) a **Control-Point-to-Control-Point Session** is established between each adjacent node in order to allow transfer of APPN information. The following information is transferred when links are established:

1. Information regarding the APPN node type of each node (in this case they are all network nodes).
2. Updates of the topology databases (if the nodes are network nodes).
3. If a node is an end node, information regarding the local location names defined within that end node is sent to the adjacent network node that the end node was configured to have a CP-to-CP session with.
4. Network searches. For example, if system A requests a session with system C, then system A sends a search request to the network (that is to DS in node A). If the remote location name requested by system A is not known, then **Directory Services (DS)** broadcasts the search request to all adjacent network nodes in the network who pass on the request, in order to find the CP who owns the requested **location name** (i.e. the nickname which system A used to request a session with system C).
5. Network node C will return a positive response to system A, which will then use **Topology Routing Services (TRS)** to calculate the best route from system A to system C and establish the session via that route.

A peer device will be automatically created, on system A to represent the remote and local location pair, and, on system C to represent the same remote and local location pair. These devices will be automatically activated beneath the appropriate controller description for the chosen route. The session between A and C has then been established.

Advanced Program-to-Program Communication

Logical Units

A **logical unit** (LU) is the end user's interface to an SNA network. LUs are attached to the path control network, which is responsible for addressing and routing data packets.[28] Two LUs communicate over a **session**, a logical connection established when one LU sends another an SNA request known as the **BIND**, which specifies the protocols that both partners agree to abide by for the duration of the session. The BIND sender and receiver are known respectively as the **primary LU** (PLU) and **secondary LU** (SLU).

LUs exist within **nodes** (not pictured in Figure 31 on page 59), which in traditional SNA networks are categorized as either:

1. Subarea nodes

   These correspond to hosts (**Type 5 (T5)** nodes) and 37xx controllers (**T4**), which can route packets from one node to another.

2. Peripheral nodes

   These correspond to small processors, cluster controllers, and terminal devices (**T2**, sometimes **T1**), which depend on attachment to a subarea node providing **boundary function** for all communication with other nodes.

The path control network is implemented by the physical network of interconnected nodes and links.

---

[28] Data is routed through an SNA network in the form of variable length blocks here called "packets". The correct term is "RU" meaning "Request/Response Unit". An RU may contain either data or control information or both.

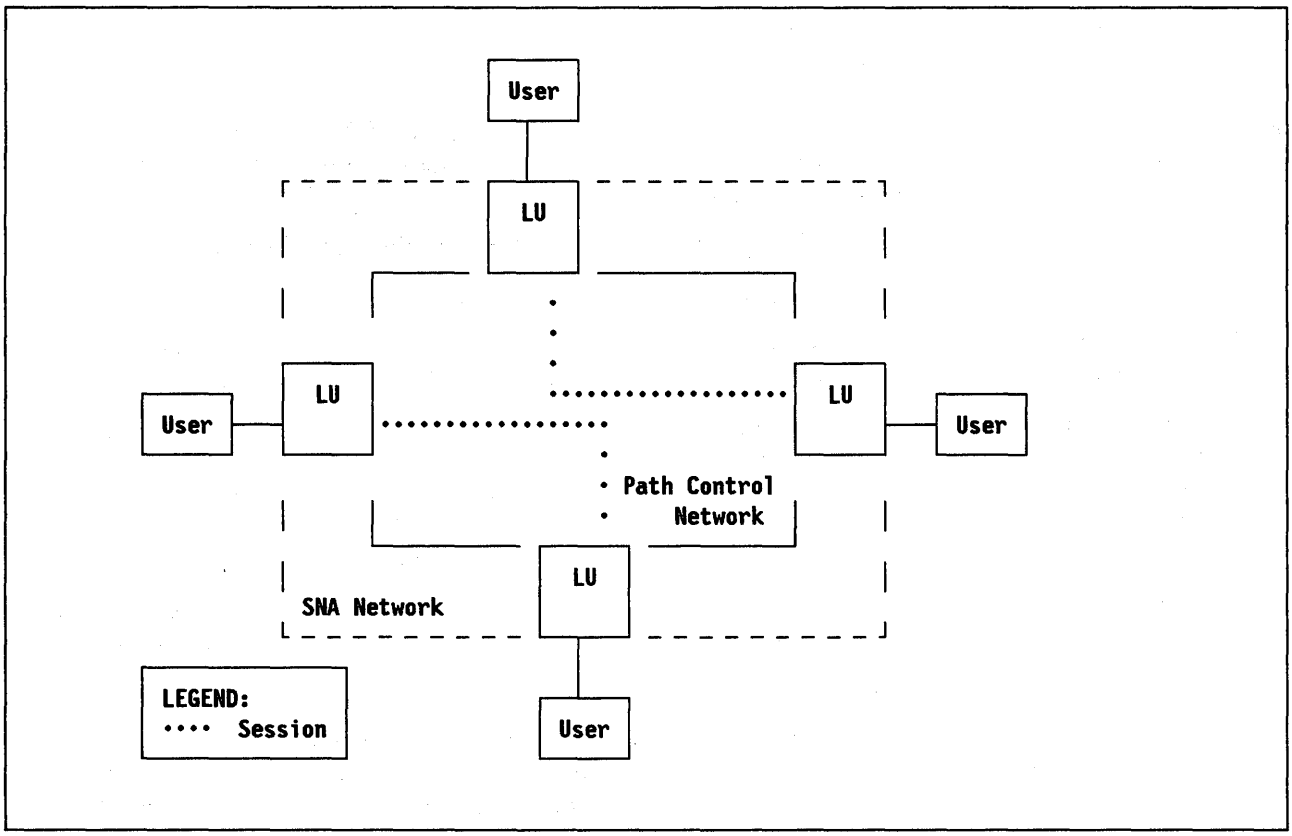


Figure 31. Logical View of an SNA Network

The layering of SNA enables changes to be made at different layers independently. In practice, connectivity enhancements are generally classified as belonging to either the path control layer or the session layer. The multi host support mentioned above is an example of a connectivity enhancement at the path control layer. The discussion of APPC primarily concerns the session layer and the presentation services layer, though there is an associated path control element which are discussed in a later section of this chapter.

LU Types

SNA products are classified according to the SNA functions that they support. The logical unit(s) that reside in each product assumes that classification. The classification type of each LU designates a particular subset of SNA functions that the product can perform when its LU is in session with a LU in another product.

To accommodate various often-used sets of protocols, a number of **LU session types** (or **LU types**) were developed. These LU types specify the session parameters within certain *profiles*, and fall into several categories:

- *Host-to-terminal* LU types can be regarded as defining "generic devices", so that the peculiarities of individual products are invisible at the session layer, thus reducing the cost of developing host applications for different terminal implementations. They include LU 1 (generic printer device with local resources such as keyboards and diskettes), LU 2 (3270 displays), LU 3 (3270 printers), LU 4 (similar to LU 1, but also supporting terminal-to-terminal communication), and LU 7 (5250 displays).

  Terminal LU types generally incorporate restrictive assumptions that reflect the asymmetry of host-to-terminal communication. For example, LU 2 assumes the host is always responsible for error recovery.

- A generic LU type allows products to implement session protocols that are *not defined by SNA*. Several IBM products, including NCCF (Network Communication Control Facility) and FTP (File Transfer

Program), have used this LU type (described later as LU type 0) to implement their own program-to-program protocols.

- A *program-to-program* LU type is available which provides SNA-defined protocols for peer communication between programs. This LU type is described later under LU 6.0, LU 6.1 and LU 6.2.

LU types are characterized by their associated Transmission Services (TS) profile, Functional Management (FM) profile, and Presentation Services (PS) profile.

LU types 0 through 7 are defined in SNA.

LU type 0: These are implementation-dependent and do not fall within the groupings of profiles defined by SNA. An example is the IMS SLU type "P" used for IBM 4700 and IBM 8100 access to IMS systems.

LU type 1: Type 1 logical units are for application programs that communicate with terminals. They support single or multiple devices in an interactive, batch, or distributed processing environment. Devices may include consoles, printers, diskettes, disks and card units. This session uses the SNA character string (SCS).

LU type 2: Type 2 logical units are for application programs that communicate with single display devices in an interactive environment. The session uses the SNA 3270 data stream and operates in half-duplex flip-flop mode. The LU in the IBM 3174 that is associated with the IBM 3278 is an example of a type 2 logical unit.

LU type 3: Type 3 logical units are for communication between application programs and a single printer using the SNA 3270 print data stream. The LU in the IBM 3274 that is associated with a 3287 printer can be bound as a type 3 logical unit.

LU type 6.0: Type 6 logical units are for data communication between CICS subsystems on a program-to-program basis. CICS is an example of a Type 6.0 logical unit (perhaps the only example). A CICS logical unit can establish a session with another CICS logical unit.

LU type 6.1: Type 6.1 logical units are for data communication between IMS application subsystems. CICS and IMS are examples of type 6.1 logical units. A IMS logical unit can establish a session with another IMS logical unit or with a CICS logical unit.

LU type 6.2: Type 6.2 logical unit uses device independent protocol for process-to-process (or program-to-program) communication in a distributed environment. Its operation is symmetric (peer-to-peer), with both partners having equal control over the resources allocated by the SNA session over which they communicate. LU 6.2 was developed as a generic program-to-program protocol based on the knowledge gained from implementation of LU 6.0 and 6.1.

LU type 7: A type of logical unit for an application program that communicates with a single display terminal in an interactive environment, for example, a session involving an application program in S/36 and an IBM 5250 display.

Note: A product can support several LU types, as exemplified by CICS/VS, which supports all except LU 7.

In order for two LUs to BIND a session, both must support a common LU type, since the BIND request establishes the LU type for the session. Accordingly, any convergence of LU types increases session-level connectivity. In fact, LU 6.2 is now the base on which SNA support for distributed processing is evolving.

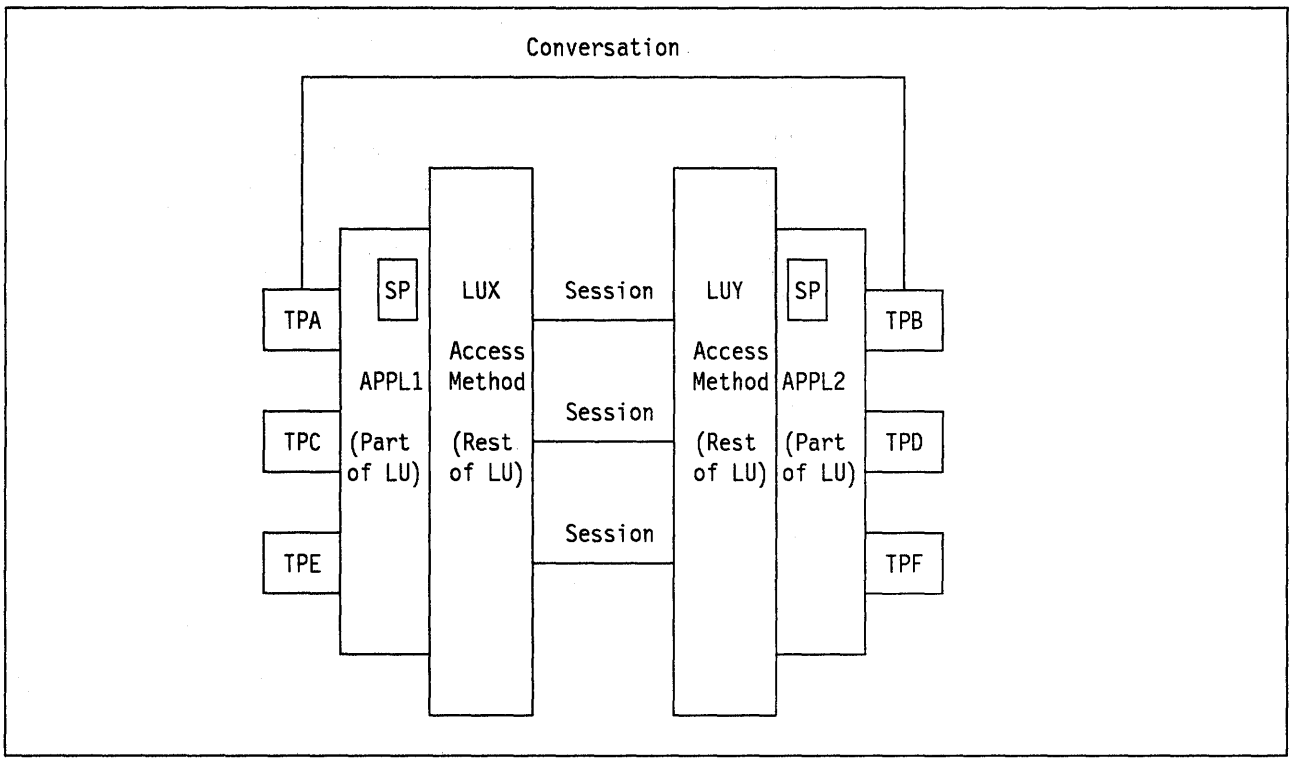


Figure 32. APPC Logical Structure

LU 6.2

LU 6.2 or APPC is much more than just another LU protocol definition.[29] LU 6.2 was developed in concert with the T2.1 node architecture and with APPN in order to implement a new concept in SNA communication.

Before the LU 6.2, an LU was regarded as the "end user" of an SNA network. An operator using a 3270 terminal has (typically) a single LU which may request a session with an application situated somewhere in the SNA network. If the operator wanted to communicate with another application then it was necessary to end the current session and log on to the other application.

Even in the first release of SNA, a program (user-written or IBM-supplied) could also "be" the end point of the network - the LU. This meant that SNA architecture did not extend to end devices connected downstream of an IBM cluster controller device (such as an IBM 3600 or IBM 4700 system) but stopped within the controller at the user program. It was never true that a one-to-one correspondence was required between a terminal and an LU. For example in the IBM 3650 Store System there was one LU per ***application*** with perhaps many hundreds of downstream terminals communicating with a single LU.

Designed primarily for program-to-program communication, LU 6.2 is a radical departure from previous SNA concepts. This is illustrated in Figure 32.

- The LU is no longer an end user or a user program. It is a *subsystem.* (For conceptual purposes, CICS is a good model of an LU 6.2 subsystem.)
- The end user is a transaction program that runs on the subsystem.
- Sessions are now established in advance of when they are needed and connect subsystems together.

---

[29] A good discussion of LU 6.2 may be found in *An Introduction to Advanced Program-to-Program Communication (APPC).*

- An end user communication with another end user is called a *conversation.*

  When a user transaction wishes to communicate with another transaction somewhere else in the network, the requesting transaction is given *exclusive* use of a vacant session which already exists with the subsystem on which the destination transaction is running.

  A transaction request may cause the destination subsystem to start the requested transaction if it is not already running. The communicating transactions are given exclusive use of a session for the duration of their conversation.

- Thus sessions are now pre-established "pipes" which are serially reused by transactions as needed. A transaction may have many conversations (either simultaneously or serially) with other transactions throughout the network.

- New sessions are only set up or taken down rarely in a running network. A transaction gains access to applications using existing sessions.

  This means that the system-wide rate of session establishment and disestablishment is minimised. Since there is no interaction with the SSCP for starting or ending a conversation, system overhead is minimised.

- The concept is that of a "distributed operating system" where transactions make standardised requests for communication with other transactions across the network without regard to where they are or in what type of equipment they are implemented.

- Communication is synchronous. Both communicating transactions must be present simultaneously for communication to take place. This is again "like CICS."

- There are many other features about APPC that are important such as its rigidly standardised user interface which enables users to standardize communication across a wide variety of types of hardware but the above characteristics are the relevant ones for the kind of network support required.

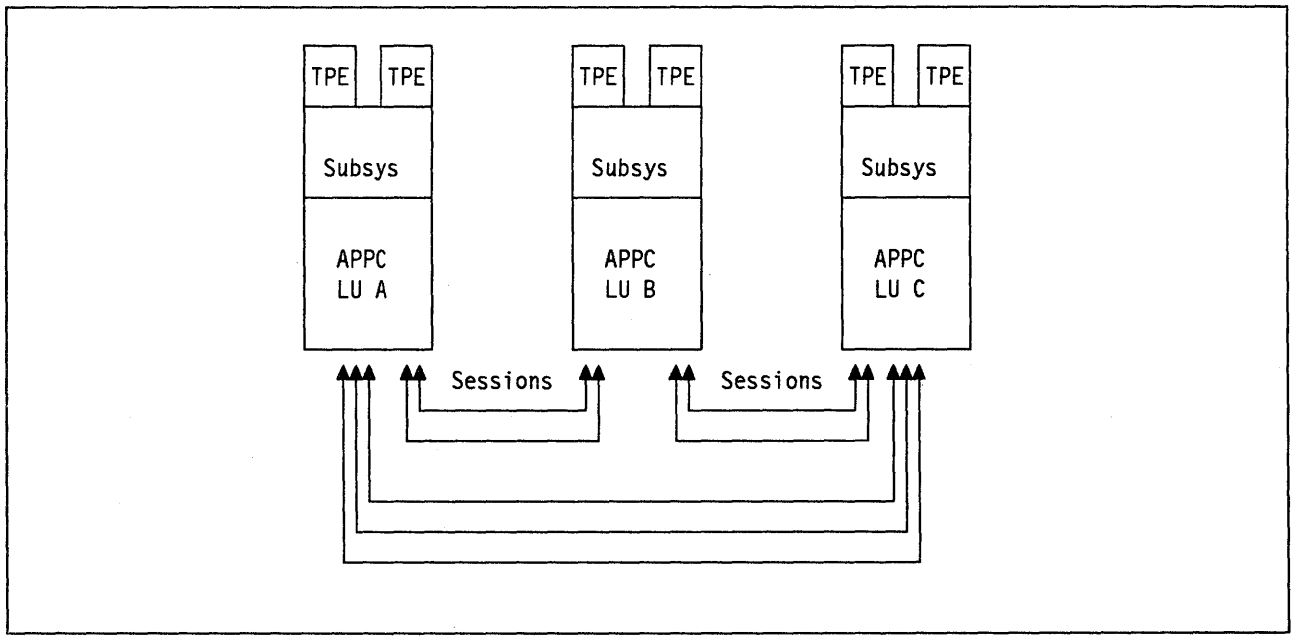


Figure 33. APPC Use of Sessions

Parallel and Multiple Sessions

Using LU 6.2, a subsystem LU may have many transactions active. These transactions must be able to communicate with other transactions elsewhere in the network. Thus, in order to support LU 6.2 communication properly the LU must be capable of having both "multiple" and "parallel" sessions (as shown in Figure 33.) This is the relevant requirement for T2.1 node support.

When an LU is capable of having a session with more than one other LU at a time it is said to have multiple session capability. This was true of the very first SNA host applications. In SNA a session is identified by the pair of network addresses of the communicating LUs. An LU with multiple (but single) sessions has only a single SNA network address but has a system structure which allows many sessions.

As soon as SNA became capable of having more than one host (in about 1977) there was the need to have more than one session between two communicating LUs. But the session is identified by the pair of network addresses used. Since an LU was identified by a name and a network address there was the problem that two "parallel sessions" between the same two LUs could not be separately identified. This problem was solved by removing the restriction that said an LU may have only a single network address. On the primary (or PLU) side, a new network address is allocated for each new parallel session started. Thus parallel sessions requires multiple network addresses to be allocated for the same LU name.

In order to support a full-function LU 6.2 both parallel and multiple session capability is needed.

There is a quasi-LU 6.2 function which works with traditional dependent LUs and has been in use for some time but it can only operate as a single session, secondary LU and thus much of the function of full LU 6.2 is lost.

New Network Functions - Integration

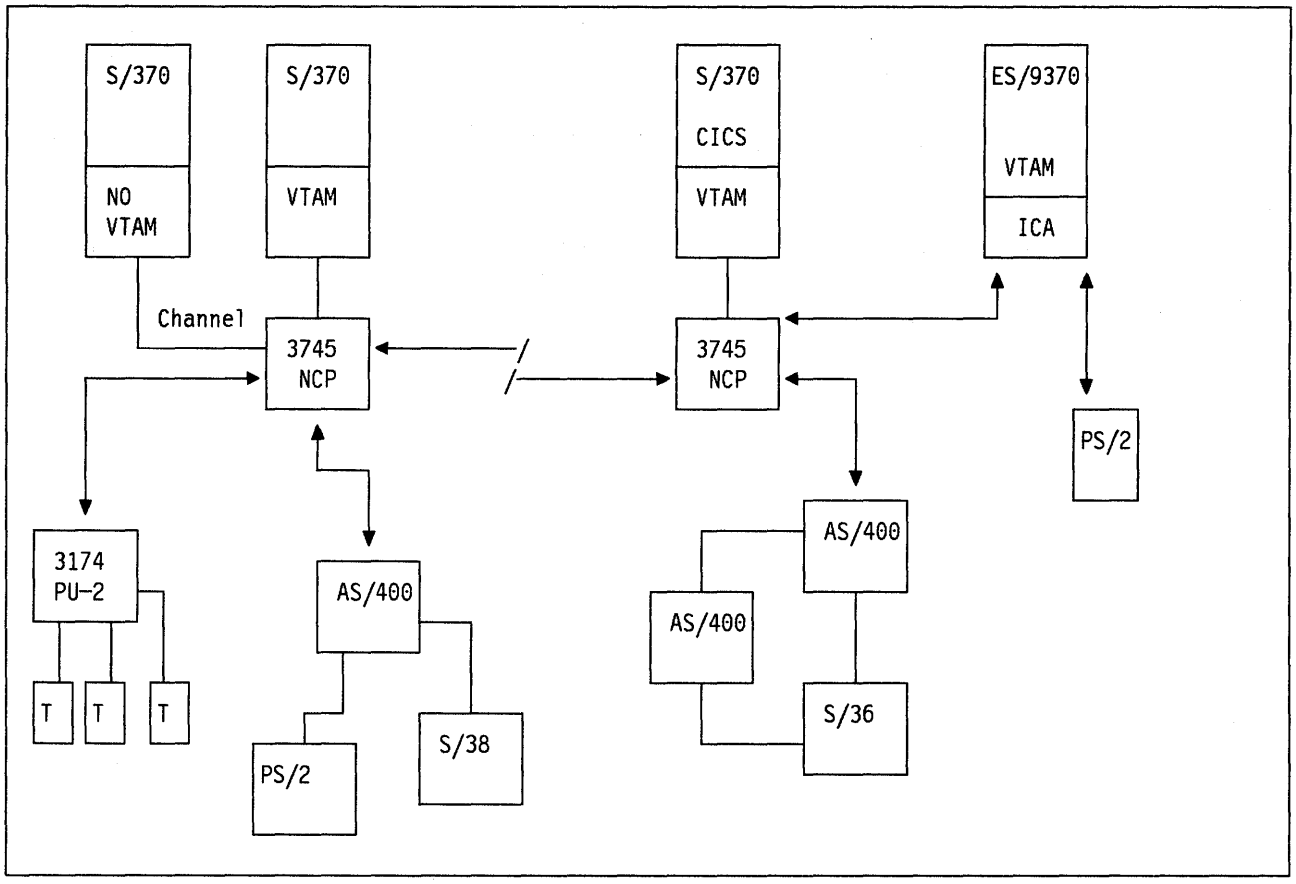


Figure 34. Integration

The objectives of the new T2.1 node support in NCP and VTAM are summarised in Figure 34. They are:

- Connect individual L.E.N.-style nodes to the SNA subarea network and integrate their operation fully with that network.
- Provide full-function LU 6.2 support to LUs within the connected L.E.N. nodes.
- Provide any-to-any connection between connected L.E.N. nodes and traditional SNA hosts.
- Allow APPN networks to connect to the subarea network and gain access to traditional SNA hosts.
- Interconnect APPN networks through the subarea network while retaining the ability to connect to a traditional SNA host.
- Allow large APPN/L.E.N. processors to connect freely L.E.N. nodes and other APPN nodes through the subarea network.

It is necessary to point out that the aim here is to *integrate* individual L.E.N. nodes with the subarea network but only to *interconnect* with APPN networks. Interconnected APPN and subarea networks provide a high level of function but are *interconnected* **not** *integrated.*

Chapter 5. Principles of Operation

In order to understand the new functions provided by the SNA Type 2.1 node it is first necessary to review the characteristics of its predecessor the SNA Type 2.0 node.

Operation of an SNA Type 2.0 Node

In a traditional SNA subarea network, the SNA Type 2.0 node is the SNA node type for "cluster controller" or "minicomputer" connection to the network. This was designed at a time when the costs of computing cycles and storage were very high. The design of the T2.0 node was aimed at minimising the amount of processing "overhead" necessary to the network attachment. At the time, when "distributed processing" technology was in its infancy, the immediate need was for hierarchically organised networks to optimise the connection to large-scale hosts.

These parameters produced an architecture for connection with the following characteristics:

- The node has a single session with and is controlled by a host resident control point (within VTAM).
- LUs residing within the T2.0 node are controlled from the host and may have a single session only with a single host-resident application.
- Both LUs and PUs are isolated from the characteristics of the transport network by a function called "boundary network node"(BNN). Implemented in NCP or in ES/9370 VTAM, BNN keeps control of such things as the real network addresses of the LUs and administers local flow control and message segmentation for buffer size matching.

Consider the characteristics of an LU implemented within a T2.0 node:

- It is always secondary.

  That is, it is capable of having a session only with a primary LU (PLU). Until recently, primary LUs only resided in hosts. Thus until recently, LUs within a T2.0 node could only communicate with host resident primary LUs. A new session is always initiated from the primary LU (PLU) in the host. (Albeit that the LU can send a request to its control point to pass a request for a session to a primary LU.)

- The LU has two sessions only.

  One session is with its control point and the other with a host application somewhere in the SNA network. Only one of the two sessions can carry application data.

- The LU is activated and deactivated (controlled) from its host resident control point, the System Services Control Point (SSCP). Although SSCPs have been implemented within other products (such as TPF and TCAM) the SSCP function is normally provided by VTAM.

A "Transmission Header" (message header) is prefixed to all data and control information sent between the T2.0 node and the network. This header is illustrated in Figure 35 on page 66. In this header an LU is identified by an eight-bit index number. This gives a maximum range of LU numbers of 0 to 255. (In practice there is a maximum number of 254 LUs allowed.) The use of an index number to identify the LU is very efficient when a message has to be processed (routed to the LU) but the index number is responsible for many of the limitations on the T2.0 node.

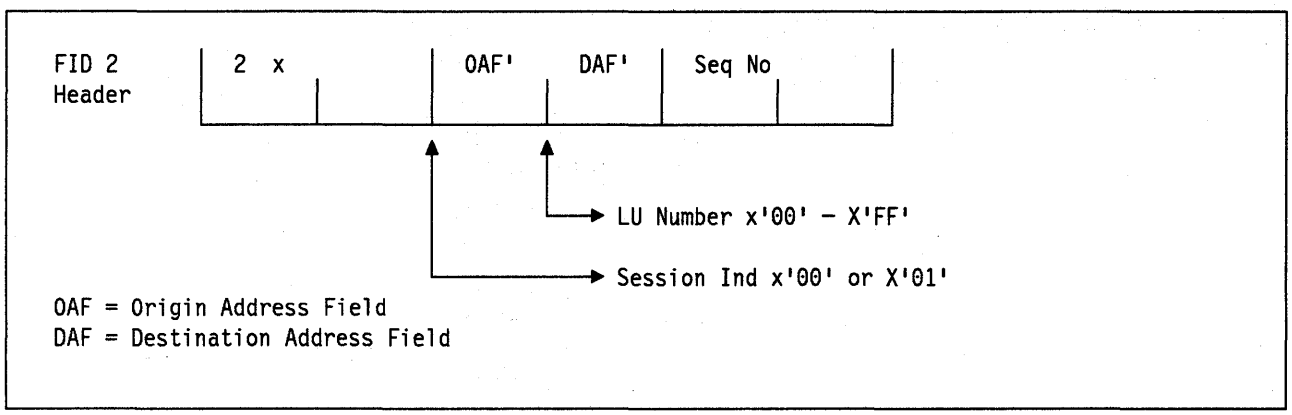


Figure 35. The FID 2 Transmission Header. As used by SNA Type 2.0 node.

The OAF' field (8 bits) can take only two values - 0 or 1. This field is used to identify which session the LU is referring to. Zero identifies the session with the SSCP (control session) and a one identifies the data session.

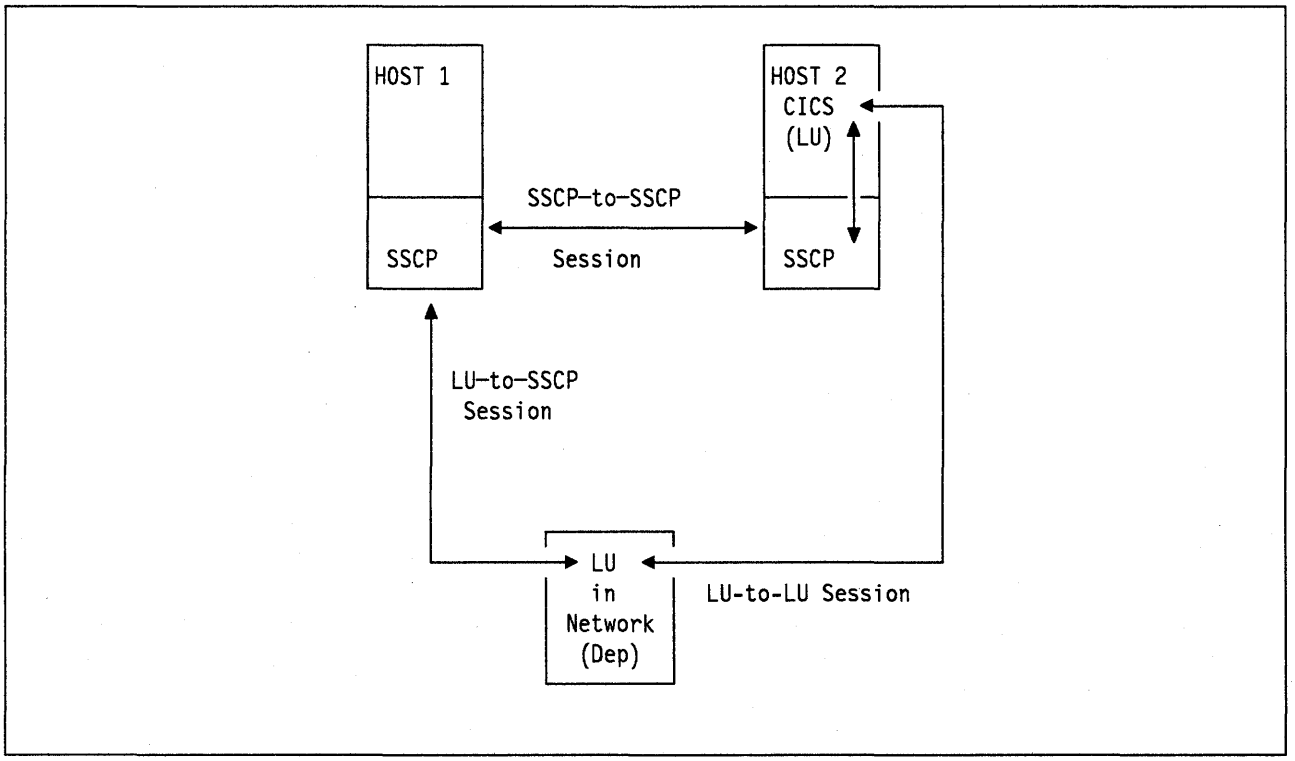


Figure 36. Dependent LU Sessions

These "traditional" LUs are called *dependent LUs* because they depend for their operation on having a session with a System Services Control Point (SSCP) in a distant System /370 host. All LUs that exist within a T2.0 node are dependent LUs. As will be seen later, the T2.1 node may have dependent LUs within it, but it can also have a new type of LU called an independent LU.

Session initiation with dependent LUs

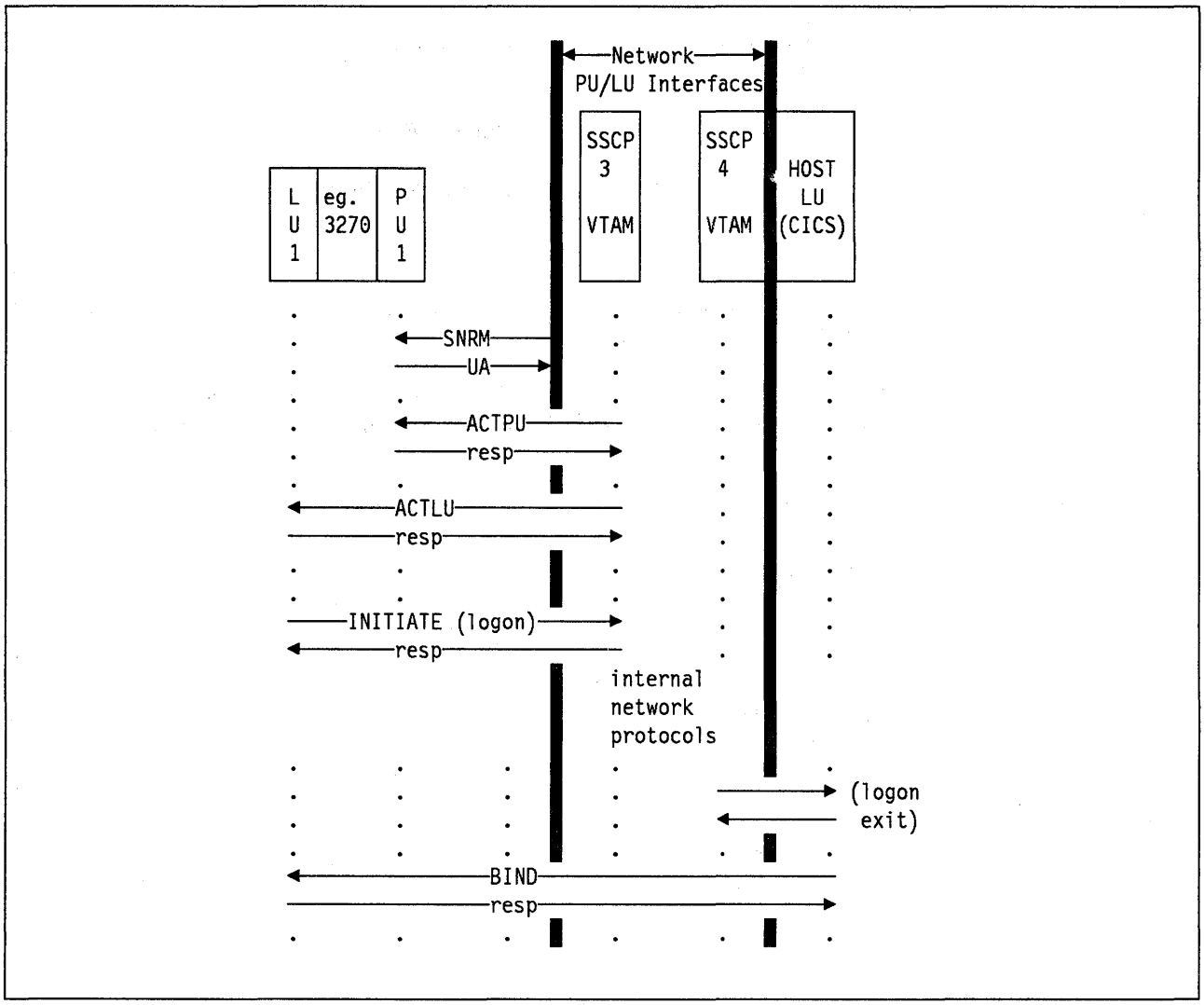


Figure 37. Session Initiation with Dependent LUs

Consider the protocol necessary to start a session between a dependent LU (SLU) and a host resident (also dependent) primary LU. This is illustrated in Figure 37. This might be an IBM 3270 style device establishing a connection to CICS in a host.

- When the secondary LU (in this case a 3270 device) needs to initiate a session it sends an INITIATE request (or a LOGON request) to its control point (VTAM) on its LU-to-SSCP session. Notice that before any initiate can be sent the control point must activate first the PU then the LU. These activations can be thought of as establishing the LU-to-SSCP session.
- The SSCP (VTAM) that owns (controls) the requesting LU then must locate the LU to which the request is addressed. In this case it is CICS operating under the control of a different copy of VTAM from the one which owns the 3270.

  At the time the request is sent, the address of the destination LU (CICS) is not known either by the initiating LU or by its controlling VTAM. Internal protocols between the two control points are used to inform the destination VTAM (SSCP 4) of the logon request.
- The destination VTAM (VTAM 4) then tells CICS that a request for a logon has been received from the 3270 called "LU1." This is done by driving the VTAM "LOGON" exit. This is really the initiate request being passed to its destination LU on an SSCP to LU session.

- CICS then issues a VTAM macro called "OPNDST ACCEPT" which results in a BIND command being sent on the LU-to-LU session from CICS to the requesting LU.
- Note that before this can take place the network will have to have set up an appropriate route etc. for the new session to use.
- When the BIND is received and a positive response sent from the 3270 in the network, communication is freely possible between the two LUs (CICS and the 3270).
- Note that the address used in the FID 2 header between the network and the T2.1 node contains only the LU index number - it does not contain the real network address of either session partner. These addresses are held within the network. Each LU knows the other *only* by its LU name *never* by a real address.

Operation of an SNA Type 2.1 Node

In the discussion of the T2.0 node above, it was understood that the phrase T2.0 node referred to both an architectural specification of the method of operation of particular kind of peripheral node and to the way in which that node interfaced to the network. SNA Type 2.1 node is *different* from SNA Type 2.0 node in that the phrase *primarily* specifies a method of interfacing to another "peer" box. There are several different kinds of nodes that interface either to the network or to each other using the T2.1 node specification. These are called SNA Type 2.1 nodes but in fact they are different from one another internally. In this document when the SNA Type 2.1 node is discussed it generally means the most simple L.E.N. node implementation. As discussed in Chapter 4, "Technical Background" on page 45, the node types that use the T2.1 node interface are as follows:

1. The Low Entry Networking (L.E.N.) node.

   This was the first SNA Type 2.1 node implementation.

2. The Advanced Peer-to-Peer Networking (APPN), End Node (EN).
3. The APPN Network Node (NN)
4. A Composite Node (CPN) consisting of either a combination of VTAM and NCP or of an ES/9370 VTAM alone.

   This Composite Node will be discussed in some detail later.

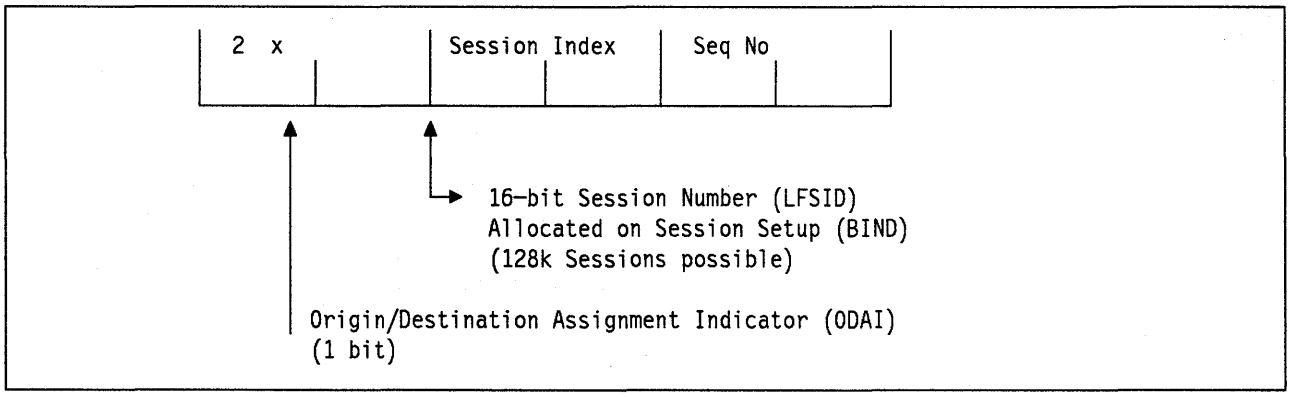


Figure 38. The FID 2 Header as used by the Type 2.1 node

Figure 38.[30] shows the header format used for the T2.1 node interface. The T2.1 node header (discussed in the previous section) contains a "session index" and an "LU index." The T2.1 node treats the two eight-bit fields of the FID 2 header as a single 16-bit number. This number, in conjunction with another bit called the "Origin/Destination Assignment Indicator" uniquely identifies a session.

The interface has the following characteristics:

- There is no fixed relationship between an LU within the node and the session number within the header. Numbers are allocated dynamically when the session is started and freed when the session is terminated.

  Note: In the T2.0 node connection an LU is identified by its local address. In the network (in NCP or in VTAM) LUs are defined in terms of their address. In the T2.0 node LUs may have names (in some implementations) but the network name of the LU as known to the network (NCP/VTAM) does not have to be the same as the name the LU is known by within the T2.0 node The sole identification of the LU to the network was its address.[31]

  In the T2.1 node architecture there is no fixed address to relate an LU definition in the network to an LU defined in the node. Therefore the "LU Name" (usually qualified by its network name) is sent in the BIND for both the origin LU name (BIND sender) and the destination LU name (BIND receiver).

  This is discussed in more detail in "Initiating Sessions" on page 94.

- A session is established using a BIND command - there is no requirement for a "request" to be sent to the partner LU. (In fact a request CANNOT be sent to the partner LU because there is no LU_SSCP session to send it on.)
- There is no relationship between the LU at either side of the interface and any information within the T2.1 node header. The index number identifies a single session but not the LU.
- Since the index is allocated dynamically when the session is started and there is no relationship between the number and a specific LU, there can be as many sessions as desired for each LU. That is, LU may have both "multiple" and "parallel" sessions.

The SNA Type 2.1 node interface was designed as a peer-to-peer interface. This is shown in Figure 28 on page 55. Except for the link control,[32] the interface is intended for interconnection of "equal" boxes.

LUs within a T2.1 node do not need to be controlled from an SSCP. They receive their services from a control point within the T2.1 node. This control point is variously called:

- An End Node Control Point (ENCP)
- A Peripheral Node Control Point (PNCP)
- A Network Node Control Point (NNCP).
- In the case of the subarea network T2.1 node connection, the control point function within the subarea network is performed by the traditional SSCP.

LUs that do not have a session with an external SSCP are called *independent LUs.*

---

[30] A T2.1 node may contain either "dependent" (business as usual) LUs or a new type of "independent" LU. For the sake of clarity only operation with independent LUs is described here.

[31] LU 6.2 BINDs carry LU names (within the "user data" field) but these are not checked by NCP or VTAM.

[32] In SDLC there is a primary/secondary relationship between communicating boxes, partially because this allows for very much simplified error recovery and also because it allows for multidrop connection.

Session Initiation with Independent LUs

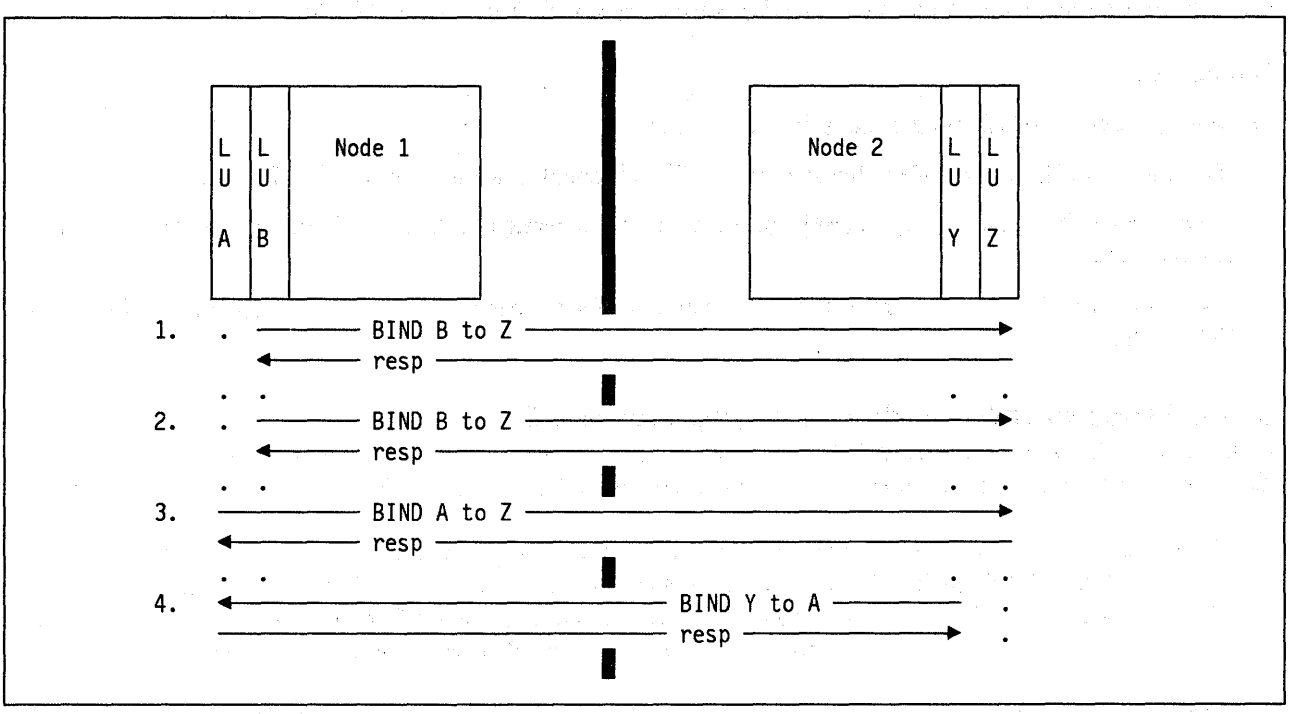


Figure 39. Session Setup for Independent LUs

Once a connection between two T2.1 nodes has been established, an LU within either node may request a session with any LU in the partner node. This is done by sending a BIND command from one LU to the other. In the BIND command the partner LUs are identified by their names *not* by any address information in the header.

The session identifier in the FID 2.1 header is allocated dynamically. This may be illustrated (**in concept**) by reference to Figure 39.

- LU B sends a BIND to LU Z.

  Since this is the first session to be started since the connection was established it is allocated a session identifier in the header of "1".

- LU Z responds with an acknowledgement to complete the session start.

  Since it would be possible for LU B to have sent more than one BIND to LU Z by now, the node and LU B both use the newly allocated session identifier to match the response to the previous BIND request.

- LU B starts another session with LU Z using another BIND.

  This session is given an identifier of "2".

- LU A then starts a session with LU Z and gets a session identifier of "3".

- Then LU Y starts a session with LU A in the same way.

  This session is also allocated an identifier of "1".

This leaves both the first session started and the fourth using the same session identifier!

The problem is that either side may send a BIND at any time. This means that there is always a potential for the same session identifier being allocated. In order to remove the conflict, another bit (called the Origin/Destination Assignment Indicator - ODAI) is used. When the link was set up, one side was allocated

a primary role and the other end a secondary role. Using this primary/secondary node designation (which itself may be dynamically determined at connection time) the ODAI indicates which side of the connection the BIND was sent from. This is described in more detail in "Transmission Header Usage" on page 139.

In summary:

- A single 16-bit field is used to identify the session
- The value of this session identifier is dynamically allocated at session setup (BIND) time
- Allocation takes place independently in each direction using the ODAI bit to identify the direction of session setup
- LUs that use this system are called "Independent LUs" because they cannot have a session with a (VTAM) SSCP.

Coexistence of Independent and Dependent LUs

In the case of the attachment of a T2.1 node to an SNA subarea network, the interface has been designed to allow the coexistence of both dependent and independent within a T2.1 node.

The FID 2 header may work in BOTH the new way with a session index and an ODAI bit and the old way with the OAF′ and DAF′ fields. This is done by starting the session index allocation for independent LUs at X′0200′. Thus if the ODAI bit is set to "0" a session index between X′0000′ and X′01FF′ is treated as OAF′ and DAF′ for dependent LUs. Independent LU address allocation starts with address X′0200′.

Dependent LUs within a T2.1 node have, of course, an SSCP-LU and an LU-LU session.

Note: If the T2.1 node contains dependent LUs it must have a PU-to-SSCP session. If it does not contain dependent LUs then it may or may not have a PU-to-SSCP session depending on information passed from the T2.1 node to the network in the XID3 exchange. In products like the AS/400 the user may decide to have an SSCP-PU session if S/370 host based network management is required.

Subarea Network SNA Type 2.1 Node Interface

Session Initiation on the Subarea Network

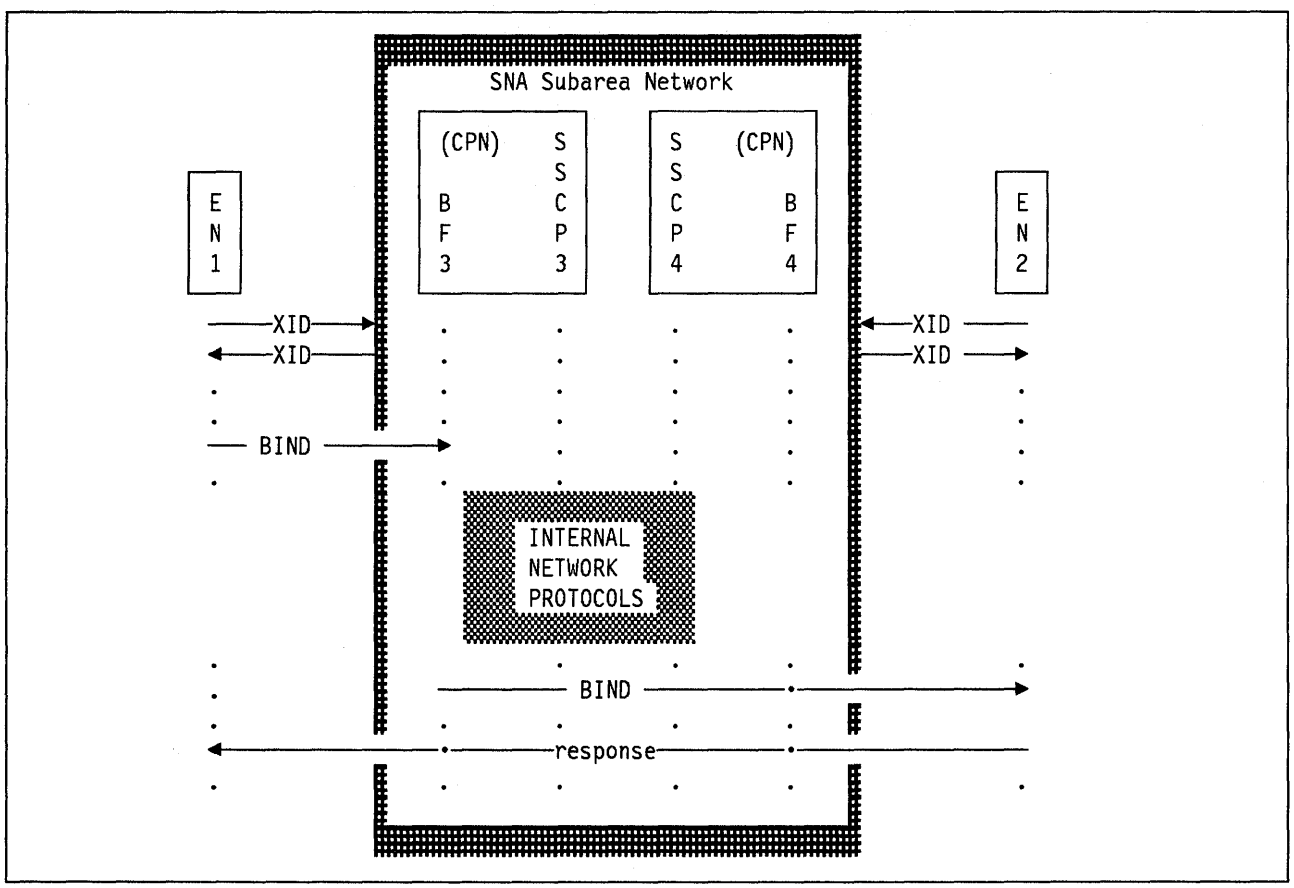


Figure 40. Session Activation with Independent LUs. EN 1 and EN 2 are L.E.N. nodes connected to the subarea network. A session is set up between an LU in EN 1 and another LU in EN 2.

While the interface to the subarea network (VTAM/9370 or NCP) strictly adheres to the T2.1 node definition, it is necessary to use the interface in a way slightly different from that used for peer connection.

Figure 40 shows a schematic representation of session setup between LUs in "EN 1" and "EN 2."

1. When the network is initialized, the first thing it will do is try to contact EN 1 and EN 2 by sending an XID.[33]

   The format 3 XID contains information about the characteristics of each box. This information about maximum block size (MAXDATA) and maximum number of SDLC blocks that will be sent without requiring a response (MAXOUT) is used to update the network definition of that node.

2. After link initialization, the LU within EN 1 sends a BIND to the network.

   This BIND contains the name of the destination LU but it cannot contain any routing information.

[33] A conceptual description of the XID protocol architecture used may be found in "DLC Activation" on page 136. The detailed link activation sequence is illustrated in Figure 57 on page 137.

Nevertheless the BIND must be either an LU 6.2 BIND or an Extended BIND. This is because only these BINDs contain the Adaptive Session Pacing information necessary for flow control over the T2.1 node interface.

This is discussed further in "Adaptive Session Pacing" on page 108.

3. When the BIND is received by the network, it is processed by "Boundary Function"(BF) within NCP (if the link is attached to a 37xx communications controller) or VTAM BF if the link is directly attached to the ICA of an IBM ES/9370.

   Boundary function determines some information about the requested session including the name of the destination LU. Since at this time BF does not know the network address of the destination LU or have any other routing information for it, BF sends a request to the SSCP for session initialization.

   Note: When the BIND arrives in BF there is no network address for the destination LU known by the BF. (Probably, the BF will not have a network address for the origin LU either since these are dynamically assigned.) The BIND contains the network names (perhaps qualified) of the origin and destination LUs. BF does not have a network directory or a control point and so communication with the SSCP is needed to get enough information to route the BIND to its destination and thus start the session.

   This point is more important when the BF in question is in the NCP since communication between the NCP and VTAM becomes necessary to start the session.

4. The session establishment functions within the subarea network are the same as for traditional session establishment. If the SSCP does not find the destination LU within its own domain, it will use its cross-domain definitions or go through the adjacent SSCP table to search for the LU.

5. The mode name (LOGON MODE name) is resolved into the class of service to be used within the subarea network through searching the MODETAB and DLOGMODE entries associated with the secondary (destination) LU.

6. When the location of the destination LU is found a Virtual Route (VR) may need to be established if one does not already exist. This VR and its underlying Explicit Route (ER) must be predefined.

7. If the other LU (and its boundary function) supports the extended BIND, the BIND is extended by appending the Fully-Qualified PCID, Transmission Priority/Class of Service and Mode control vectors. This extended BIND is then sent on the established route toward the SLU. The BF creates network qualified names in the BIND depending on the capability of the intervening and receiving nodes.

8. If the SLU cannot handle the extended BIND (notice that the SLU could be a dependent LU), the BF on the SLU side will transform the extended BIND into a conventional BIND.

9. A response to the BIND is sent from the SLU to the PLU (this may be an extended BIND response). The session is now started.

10. The owning SSCPs of both EN 1 and EN 2 are notified by their BFs of the successful session establishment.

Enhanced Session Capabilities

As a result of the architecture described above a T2.1 node when connected to the subarea network has significantly enhanced abilities compared to the T2.0 node which existed before.

- LUs within a T2.1 node can take full advantage of the APPC (or LU 6.2) architecture.

  Each LU within a T2.1 node may have many sessions with other LUs anywhere in the network (multiple sessions) and each LU may have many sessions with another similarly capable LU (parallel sessions).

  This capability is the same as was previously reserved only for application subsystems within S/370 hosts (for example CICS).

- LUs within a T2.1 node can be primary LUs (that is, they can send BIND directly without mediation from a central SSCP).
- LUs within a T2.1 node can have LU-to-LU sessions where each LU resides in a peripheral node of the network directly without the need for any relay mechanism in a host processor.

  For example two PS/2's can communicate with each other across an SNA subarea network without their data necessarily passing through a host processor.

  The total number of LU sessions allowed within a T2.1 node is essentially unlimited (up to 128 thousand). The total number of sessions will be limited by node storage and architecture. (For example in some nodes the maximum number of tasks supported by the task selection mechanism may limit the number of allowed sessions.)
- The maximum number of dependent LUs supported within a T2.1 node is still 254. (This is the same as before.)

Independent versus Dependent LUs

Summarizing the discussion above:

- Peripheral LUs (LUs within T2.1 nodes and T2.0 nodes), are now classified as either independent or dependent.
- **Independent LUs get new functions but not old function.**
  - Independent LUs may have multiple and parallel sessions and may be either primary or secondary to any session. (They can be primary for some sessions and secondary for others.)
  - Independent LUs do not have an SSCP-LU session and can exist in T2.1 nodes only.
- **Dependent LUs keep the old functions but do not get new functions.**
  - Dependent LUs may only be secondary and can have only a single session per LU.
  - An SSCP-LU session is present and is used to pass requests for service to the SSCP (and to receive responses).

Independent LUs cannot:

- Have an SSCP-LU session.
- Receive a "control initiate" (CINIT) from the SSCP. (CINIT is the message used by the SSCP to tell a PLU that a secondary has requested LOGON via an Initiate Self or character coded LOGON.)
- Initiate as an SLU (Send INIT SELF).
- Use unformatted system services (can't send an unformatted LOGON to the SSCP).
- Notify the SSCP of LU session capability changes
- Have a session "passed" to it using CLSDST(PASS).
- Use Extended Recovery Facility (XRF).
- Have a session cryptography key generated for it by the SSCP (Independent PLU).

Dependent LU to Type 2.1 Node Networking

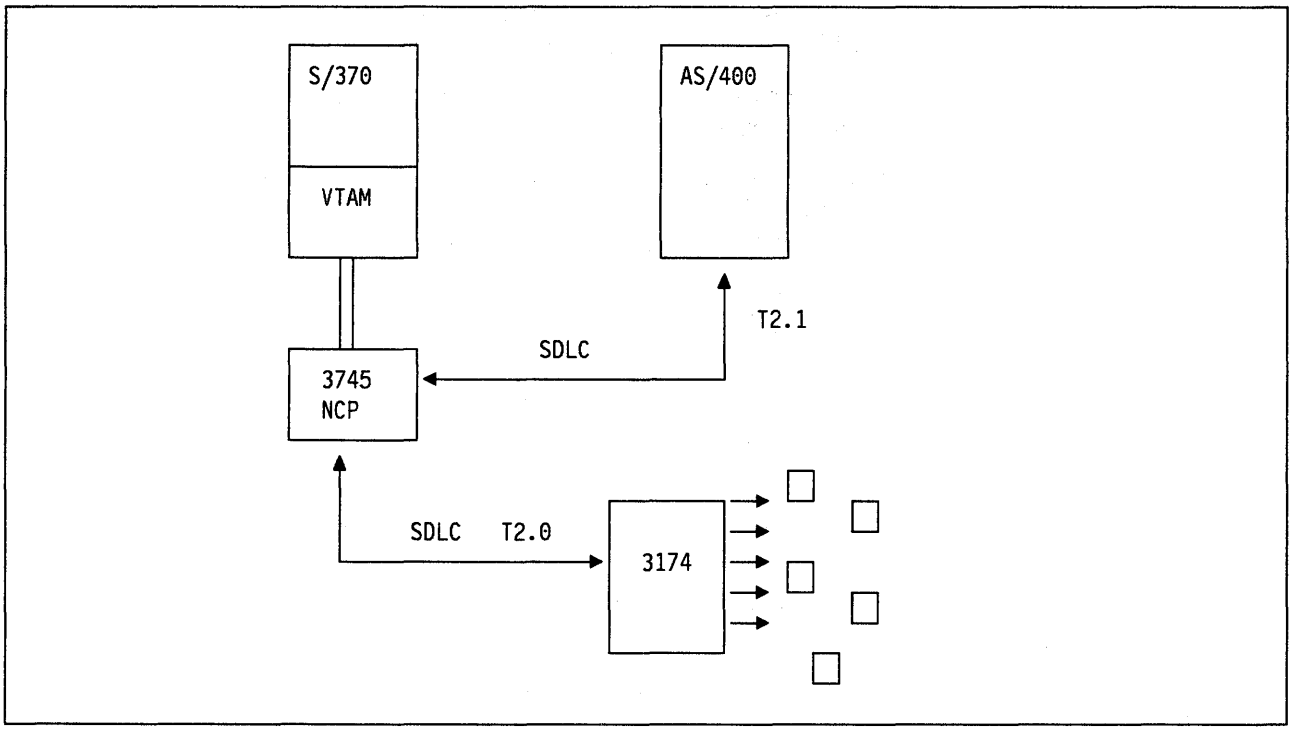


Figure 41. Dependent LU to Type 2.1 Node Networking

In Figure 41 an SNA subarea network is shown with an IBM 3174 controller and some 327x style screen/keyboards attached. Also attached to the network is an IBM AS/400 system. In this environment the AS/400 is a T2.1 node and may contain both independent and dependent LUs.

The 3174 attached devices can, of course, operate as usual requesting sessions with the IBM System /370 host.

Since the AS/400 system may contain primary LUs it would seem reasonable that devices attached to the 3174 could log on to applications in the AS/400. This (eminently reasonable) expectation is *not* fulfilled. Devices connected to the 3174 cannot LOGON to applications within the AS/400.

Leaving aside the question of AS/400 device support for the 3270,[34] The problem is that when a user tries to log on from the terminal, a logon request will go to the controlling SSCP (VTAM). The SSCP must tell the AS/400 LU that this 3270 device is requesting a session. In the traditional SNA architecture, this fact would be sent to the AS/400 in a CINIT (control initiate) command on the SSCP-to-LU session. However, since the only primary LUs that an AS/400 may have using this connection are independent LUs, there is no session to send the CINIT on.

This means that there is no way of asking the AS/400 to initiate a session with the requesting 3270 device.

In principle, as far as the VTAM/NCP support in the subarea network is concerned, the AS/400 could send a BIND to the 3270 involved and start a session. There is no restriction within the subarea network on the LU type of a session connection (the 3270 session type - LU 2 - will work in an independent LU).

[34] The AS/400 does support downstream connected 3270's but not for this type of network attachment - for the reasons described here.

However, there is just no way to tell the AS/400 to start the session. Because there is no session with the SSCP, an independent LU can neither send, nor receive, a request to start a session.

One architectural solution to the above problem would be to have the 3174 become a T2.1 node and allow the LUs within it to be primary and to send BIND. However, this would perhaps not be a very good solution because it would prevent a network monitoring application like NetView/Access or SAMON controlling network logon security.

The above discussion applies to *all* dependent LU-to-T2.1 node connections across a subarea network.

Example of 3270 Logon to a Type 2.1 Node (TPF)

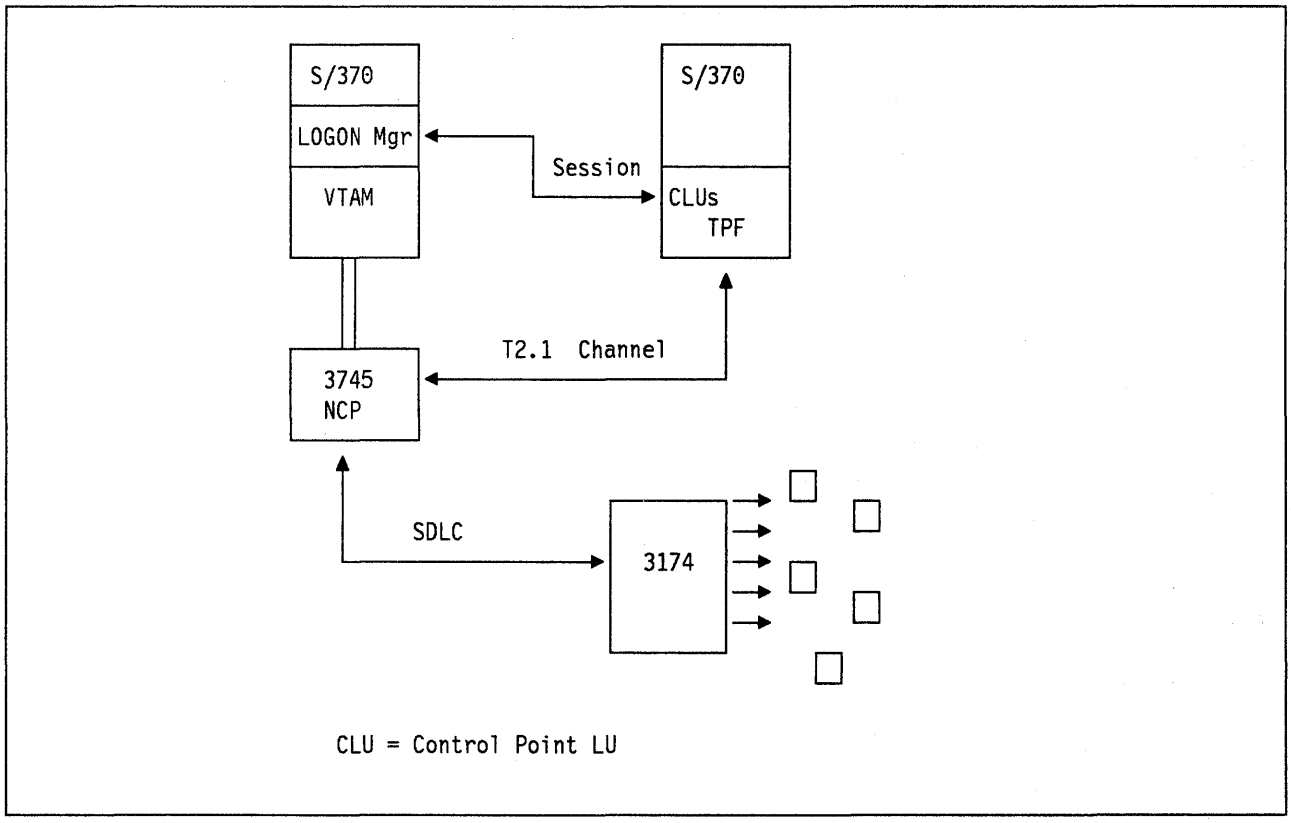


Figure 42. 3270 Logon to Transaction Processing Facility (TPF)

It *is* possible for an independent LU to have a session with a dependent LU provided that the independent LU starts the session. The problem discussed above was that there is no means for a dependent LU to tell an independent LU to do it.

One technique used to allow a dependent LU to request a session with an independent LU is the technique illustrated above. This is used by 3270-style LUs in a subarea network to gain access to the Transaction Processing Facility (TPF). TPF, although it resides in a System /370 host processor can be a T2.1 node attached to the subarea network through a channel.[35]

[35] In general T2.1 node connectivity is supported over many different kinds of physical connection. While connection through an SDLC link or through a token-ring LAN is far the most common method of attachment, a T2.1 node may connect through a channel or through an X.25 network connection.

Figure 42 illustrates this connection. The key to operation is a Logon Manager application within the subarea network which, existing within a VTAM-controlled S/370 host, can accept LOGON requests from a 3270 device. Although the Logon Manager is a separate application as far as VTAM is concerned, it is a supplied part of VTAM[36] and is documented in the VTAM product manuals. The method of operation is as follows:

- There is a pre-established session between the Logon Manager and each TPF Control Point LU (CLU). Logon requests are sent from the Logon Manager to TPF on this session.
- The 3270 style terminal attempts to logon to the Logon Manager.

  The Logon Manager is actually known to VTAM by many names. One name for each TPF application. The user sends a logon request specifying the TPF application and VTAM passes it to the Logon Manager.
- The Logon Manager then sends a notification to TPF that this LU is requesting a session. The notification flows the a pre-established Logon Manager to TPF CLU session.
- TPF sends an extended BIND through the subarea network to the 3270 device.
- The NCP boundary function (BF) at the 3174 connection changes the BIND from "extended"to "business as usual" and forwards it to the 3174.

The VTAM Logon Manager is a complex program in that it allows for multiple channel connections to the same TPF and uses different LU names depending on the best route to the requesting SLU.

Note: The application name as specified by the 3270 at logon time *must* be different from the real application LU name (the PLU name which is sent in the BIND). This is because it is not possible to have multiple LU names defined in the same network. This does not matter because the 3174 does not examine the LU name in a BIND sent to it.

Of course, the Logon Manager is just a VTAM application and it could be accessed through NetView/Access and/or SAMON etc.

Other techniques of managing the logon are possible and may be implemented by a user.

The technique of ending the session and then passing it to the other application as used by products such as SAMON etc. will not work directly here. (A session cannot be passed directly to TPF.) This is because there is no SSCP to LU (TPF) session for the requested logon (CINIT) to be sent on. However, a session can be passed (CLSDST PASS) to the Logon Manager and the same effect achieved.

The user is free to write a VTAM application (or even use CICS, for example) to achieve the Logon Manager function for different applications. For example, another method might be to allow the 3270 to logon to a User Logon Manager (ULM). Such an application might then provide a menu of reachable applications (Independent PLUs) and require a password for security purposes. This application might work in the following way:

- The user logs on to the ULM application.
- The ULM then presents the user with a menu of accessible applications.
- The user enters an application name and (optionally) a password.
- The ULM then terminates its session with the terminal by issuing the CLSDEST macro (but without the PASS option - it cannot "PASS."
- The ULM then sends a notification to the Independent PLU that this LU is requesting a session. The notification flows on a pre-established ULM to Independent LU session.

---

[36] Logon Manager is supplied with MVS VTAM only and is intended to operate with channel-connected TPF only.

- The Independent PLU sends an extended BIND through the subarea network to the 3270 device.
- The NCP Boundary Function (BF) at the 3174 connection changes the BIND from extended to business as usual and forwards it to the 3174.

It may seem that, having replaced an old architecture with a newer, better, one we are now forced to re-build the old architecture through informal means just to get back the lost function. However, this is not the case.

All old devices still function as they always did in communication with old-style applications. The development direction however, is to have new devices, as far as possible obey the T2.1 node architecture and for all LUs to be independent. In this case, a 3270-like device in the future would not need to request a session with the application but could send a BIND itself directly to the intended application. In fact, IBM 5250 emulation on a PC or PS/2 uses this technique to communicate across a subarea network to an AS/400 system.

Network Interconnection

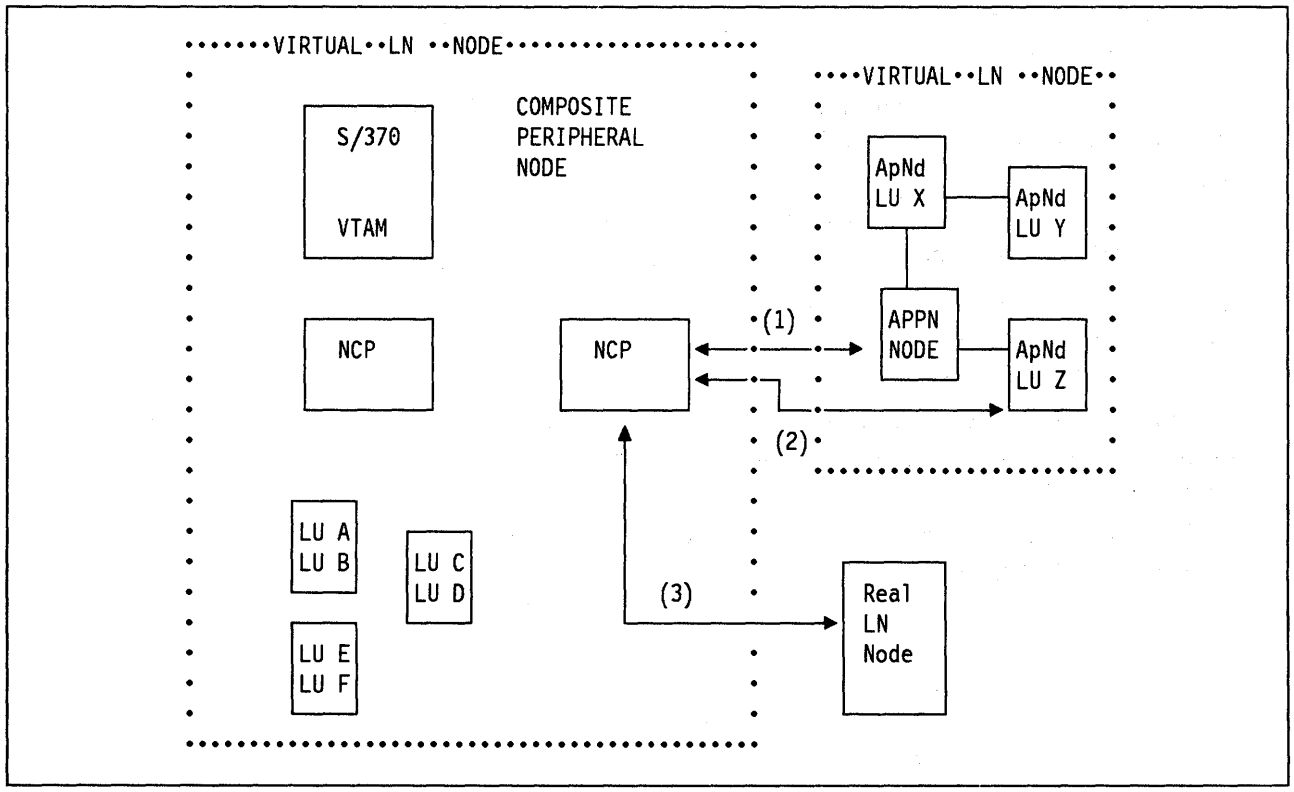


Figure 43. Subarea Network connected to an APPN network. On each of the illustrated connections, each network sees a virtual L.E.N. node consisting of a single node and a collection of LU accessed through that node.

When the subarea network connects to a T2.1 node or to an APPN network, the subarea network sees its partner as though that partner were a simple L.E.N. node. This applies regardless of whether there is in fact a L.E.N. node or an APPN node or a whole APPN network present. The attaching T2.1 node sees the subarea network also as a simple L.E.N. node containing a collection of LUs (even though these LUs may be anywhere in the subarea network). This L.E.N. view of the subarea network is called a Composite Peripheral Node.

Because of the L.E.N. node appearance of each network to the other there is no means of exchanging information between the control points. This means that each network is unaware of the structure of the other network.

In Figure 43 consider LU A in the subarea network wanting to set up a session with LU Y in the APPN network. LU A wishes to send a BIND to Y but neither the LU nor the network SSCP know where LU Y really is. Because there is no control point session from the subarea network to the APPN network, there is no way to search for LU Y and thus no way to set up the session.

The problem is solved in the simplest way by creating a node definition in the subarea network. This definition must reflect that a link exists from an NCP as shown and that connected to this link is a T2.1 node (defined as a PU 2.1). The definition must also show that LU Y is part of this node. This definition may be for a permanently connected (leased line) device of for a "switched" connection. Now when LU A tries to send a BIND the controlling SSCP will be able to set up a path to LU Y for the BIND to use.

Some consequences of this architecture are:

- There may be many links between the two networks but in practical terms (for reasons described below) this is very difficult to do effectively.
- There may be one and only one path used between two networks by any given LU from either network.
- The routing and priority schemes of one network stop at the junction of the two networks and must restart on the other side.
- Because of this structure there are limitations on the kinds of alternate routes, classes of service and transmission priority schemes which can be implemented for sessions using this gateway.

  These limitations only relate to a "gateway" environment such as the one illustrated. In the environment of direct T2.1 node attachment to the subarea network all the routing and priority characteristics of that network are maintained.
- The T2.1 node interface could be used as a gateway connection between two subarea networks. This is discussed later in "Casual Network Interconnection" on page 27 and in "Planning for a "Casual" Network Connection" on page 131.[37]
- Of course, the example of interconnected networks is a complex one. The simple case of a single L.E.N. node such as a PS/2 or a System/36 connected to the subarea network as shown attached to link (3) in the figure is more typical.

  In this case, the attached node may have multiple and/or parallel sessions with other LUs in the network no matter where they are situated.

[37] The connection of subarea networks to APPN networks using T2.1 node protocols was a feature of the previous T2.1 node support in VTAM V3 R2 and NCP V4 R3 and NCP V5 R2. The "casual" connection between two subarea networks using T2.1 node protocols is new with VTAM V3 R3. This connection applies both to IBM ES/9370 ICA connected devices and to NCP connected devices if NCP V5R3 is used.

Network Interconnection on a Single Link

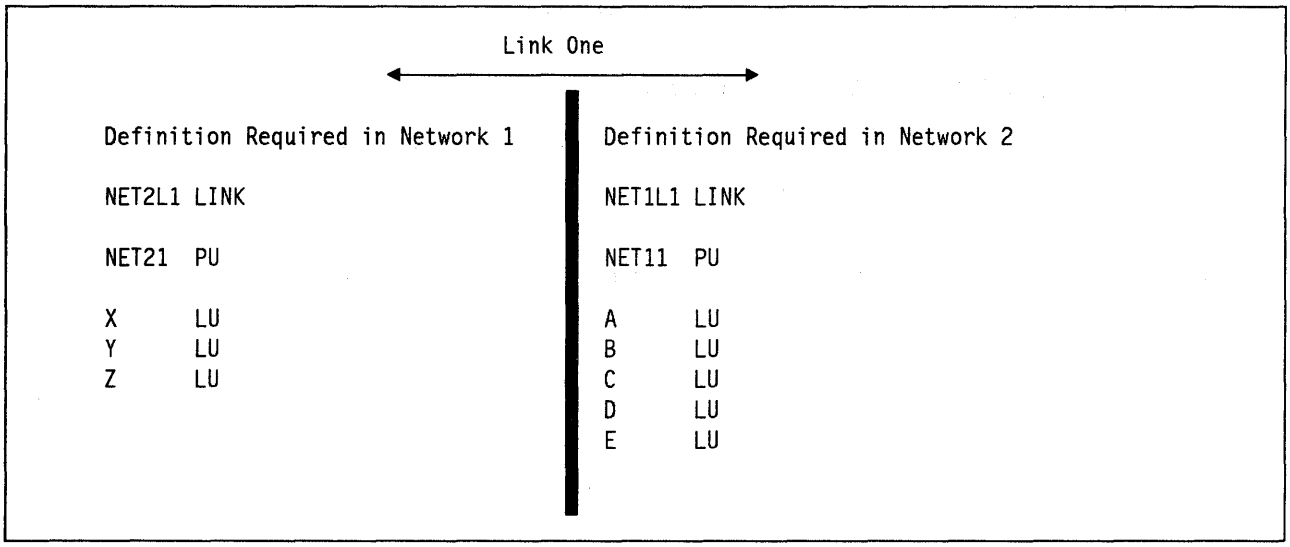


Figure 44. Conceptual Gateway Definitions - Single Link

Referring back to Figure 43 on page 78, and assuming that the only link between the two networks defined is "link (1)," the definitions needed to interconnect the two networks are shown in Figure 44.[38]

Note:

1. All LUs in a network that are to be accessed from the other must be defined in the network from which access is required.
2. All LUs that request access into either network from the other must be defined in the network into which access is requested.
3. In the diagram, LU F cannot access or be accessed from any LU in the APPN network.
4. Only LUs that need to participate in sessions which pass between networks require definition. There is no need to define all the LUs in each network.
5. **No name translation of any kind is performed by either network on session information using the connection.**

   Traffic that passes through an SNI gateway can undergo translation of LU names etc. to allow each network to have a unique name space. The T2.1 node connection is *not* a network to network interface (or gateway). Since each network "thinks" that the other is a single T2.1 node there is no possibility for a full-network interface.

For example:

- If LU A wants to send a BIND to LU X then the only way that VTAM has to find a route to X is via the definition of X on link 1. So X must be defined in the subarea network.
- When the BIND is received in the APPN network the APPN Network Node will check to make sure that the BIND arrives from an LU name that it knows as attached to this link. So if LU F sends a BIND to LU X then the APPN network will fail the BIND because LU F is unknown to it.

[38] The AS/400 uses "fixed" definitions for APPN definition but the style of definition shown here is the form of definition used in NCP/VTAM. AS/400 definitions have a similar effect but are expressed differently.

- If LU Y sends a BIND to LU D then the APPN network will route the BIND correctly to the subarea network over link (1) and it will be accepted by VTAM because LU Y is defined on link (1). If LU Y was not defined to VTAM, VTAM would *not* accept a BIND from it.

Network Interconnection on Multiple Links

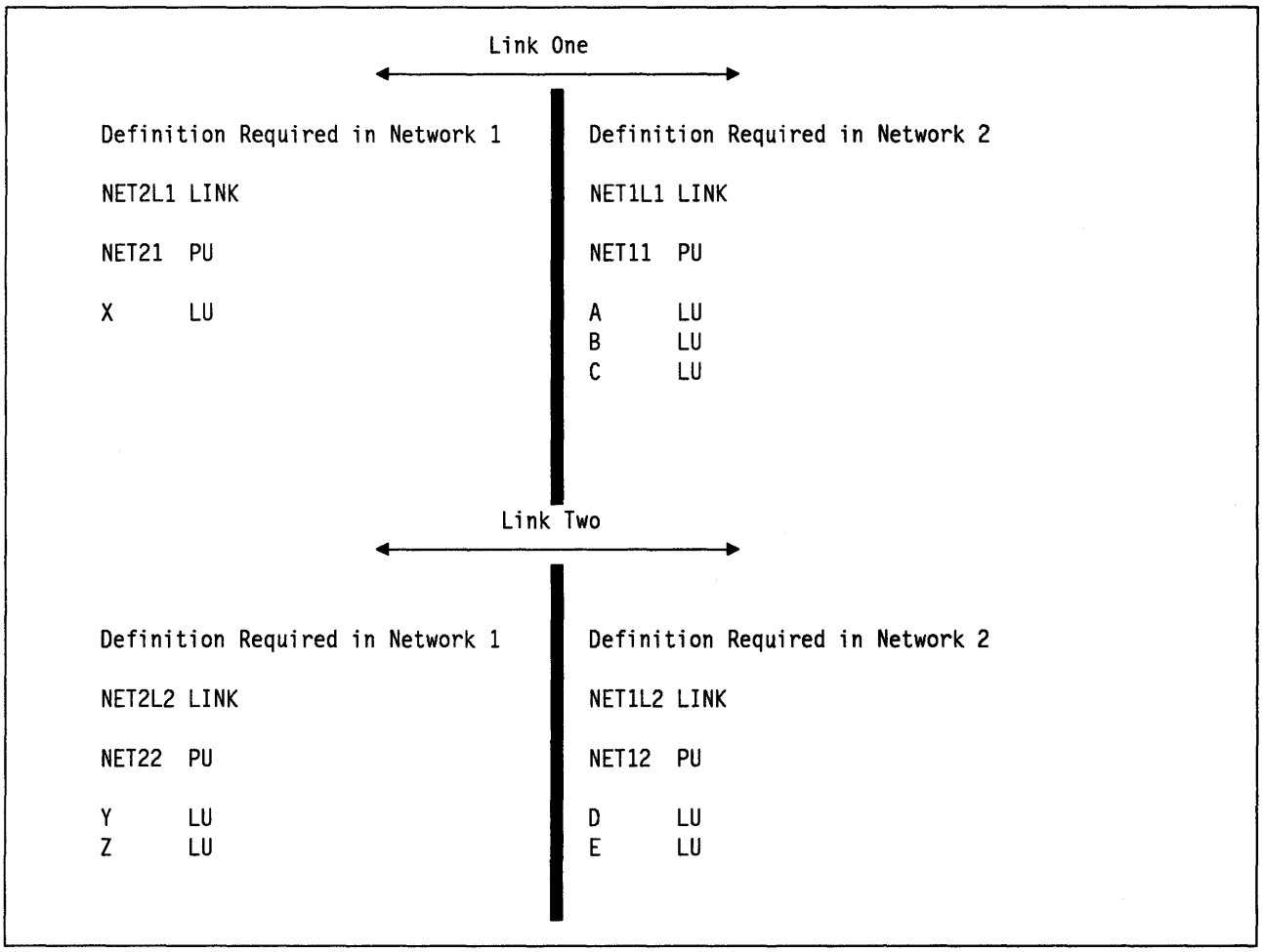


Figure 45. Conceptual gateway definitions - multiple links

The case where there are two active links between the networks is shown in Figure 43 on page 78. Figure 45 shows a conceptual view of possible definitions.

As described in the case above for a single link, LU X can communicate freely with LUs A, B and C. A, B and C can each send BINDs to X and X can send BINDs to them.

Likewise on link 2 LUs Y and Z can communicate freely with LUs D and E.

However, there can be no sessions between LUs defined in one network on one link and any LU in the other network defined as accessible through a different link.

For example, if LU X sends a BIND to LU D then the APPN network will route that BIND onto link 2 (where D is defined). The BIND will appear in the subarea network on link 2. VTAM will check and find that X is defined on link (1) and **VTAM will fail the BIND.**

The same is true in the other direction if D attempts to set up a session with X.

The conclusion that must be reached from the above is that multiple link connections between networks (subarea-APPN or subarea-subarea) have limited usefulness in practical situations.

Chapter 6. Token-Ring (TRN) Connection

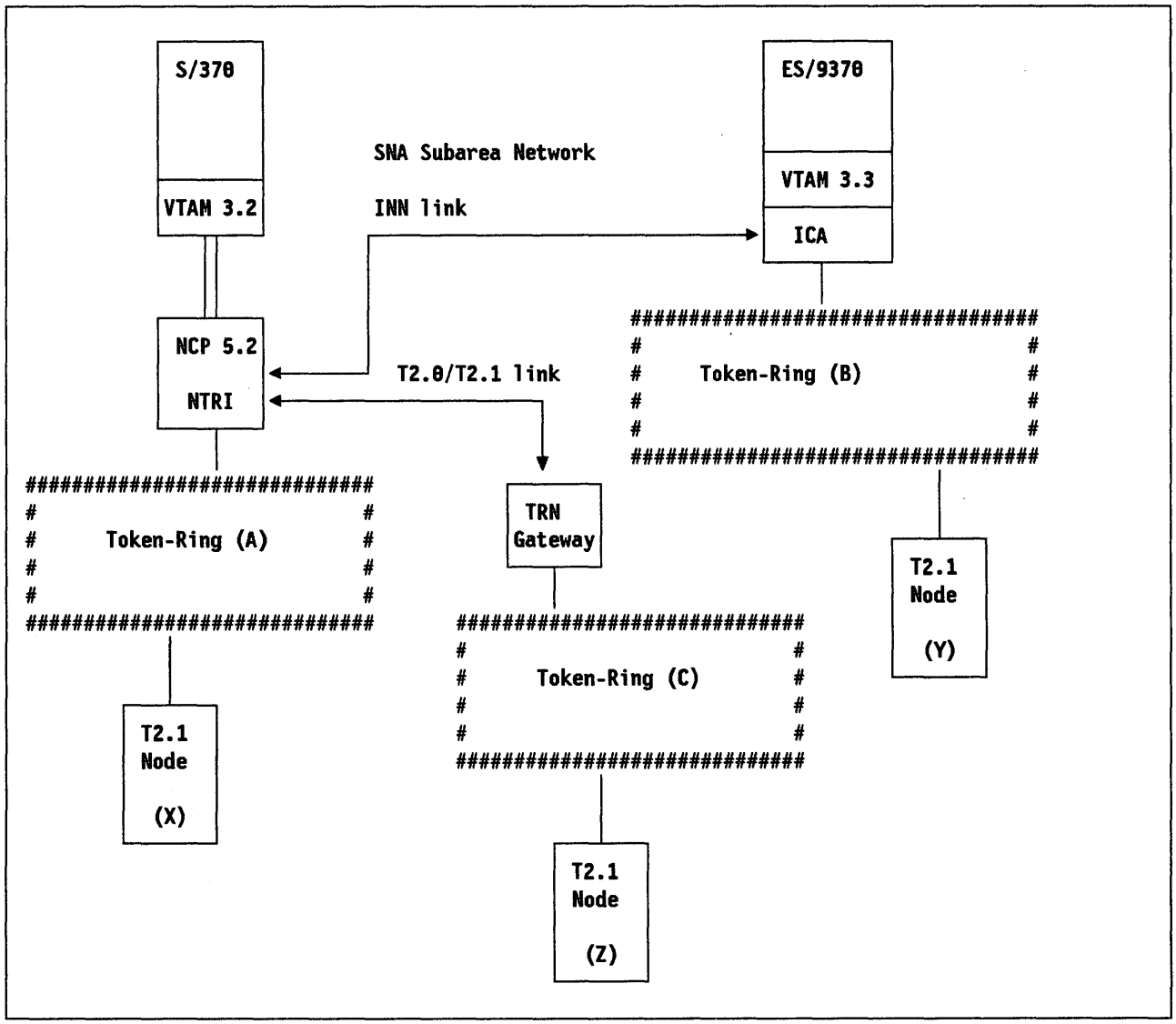


Figure 46. Subarea connection of Type 2.1 Node through TRN. LUs within either of the T2.1 nodes labeled X and Y may each communicate freely with each other and/or with host based LUs in either host. LUs within node Z may be restricted in function depending on the characteristics of the gateway.

TRN Direct Connection

In Figure 46, nodes X and Y are T2.1 nodes connected to token-rings attached to either the NCP or the VTAM boundary functions. For the sake of discussion it is necessary to ignore the other (peer-to-peer) communication functions that may be available to TRN-connected nodes through TRN communication not involving the subarea network. Node X is connected to an NCP via a TRN. Node Y is connected to VTAM using an integrated TRN adapter on an ES/9370 host processor.

Each of nodes X and Y may have full-function T2.1 node support as discussed elsewhere in this document. This means that the 37xx processors and the ES/9370 systems are all capable of providing all of the T2.1 node connection functions.

TRN Gateway Connection

In Figure 46 TRN (C) is connected to the network through a "gateway" function.[39] **In this, gateway connection case, the function available to Node Z could be considerably restricted depending on the characteristics of the gateway.**

At the time of writing there are four IBM token-ring gateway products that could fulfill the role specified as the "gateway" node in Figure 46 on page 83. These nodes and their functions are as follows:

AS/400

Described throughout this document, the AS/400 offers a nearly full-function gateway. The AS/400, configured as a Network Node, can treat the downstream T2.1 node as a node in its network. Therefore it can route traffic from node X to the subarea network NCP and back again.

This gives full function for LU 6.2 traffic. However, the AS/400 cannot allow traffic between the subarea network and dependent LUs connected to the TRN. This means that there is no access for 3270 style traffic from ring-connected PS/2's through an AS/400 (or an APPN network) and the S/370 host.

The AS/400 has a "pipeline" function which allows downstream SDLC link connected 3270 devices to "pass through" to the subarea network but it cannot "network" this type of traffic.

PC/DOS 3270 Emulation Program Version 3

This program appears to the subarea network as a PU 2.0 with dependent LUs only. Therefore it can have traffic from ring connected PS/2s doing 3270 emulation but **it cannot allow LU 6.2 traffic even as dependent LUs.**

OS/2 Extended Edition Version 1 Release 2

OS/2 EE Communications Manager (CM) offers much more function than the PC/DOS 3270 emulation program but it cannot allow transit traffic from independent LUs (unless implemented as a relay function by the user). OS/2 EE:

- Allows all types of dependent LU (including the older case of dependent LU 6.2) to use the gateway.
- Provides full-function gateway support for 3270 emulation traffic from nodes on the ring.
- Has the architectural limit of 255 dependent LU sessions using the gateway at any one time.
- Provides its full gateway functions for both Ethernet™[40] LANs and PC/Network LANs as well as for TRN.
- Allows full LU 6.2 independent LU support for LUs within the PS/2 performing the gateway function.
- Can co-reside within a PS/2 with the IBM LAN Manager Version 2.0 or the IBM LAN Manager Entry programs. However, if the LAN Manager is installed on a separate PS/2 on the LAN the LAN Manager's upstream connection may *not* go across the LAN and through the EE gateway.

  This is because the EE gateway cannot "pass through" SSCP to PU sessions. However, LAN Manager and EE can share an upstream SDLC link connection by multidropping on the same link.

---

[39] For the sake of this discussion, the 37xx processor family and the ES/9370 are not considered "gateways" but rather they are considered to be an integral part of the subarea network.

[40] Ethernet is a trademark of Xerox Corporation.

The IBM 3174 Remote Token Ring Gateway

The 3174 TRN gateway is different from the OS/2 EE gateway because it provides a routing function at the link level rather than at the session level. The upstream 37xx or VTAM processor "sees" each ring connected PS/2 as a separate physical unit in its own right. This approach allows for each attached PS/2 to be separately addressable and to retain its own characteristics when communicating with the upstream NCP or VTAM.

However, the 3174 cannot answer an XID3 on behalf of a ring-connected PS/2. This means that all ring-connected PS/2's appear to the upstream NCP or VTAM as T2.0 nodes. Therefore the independent LU facility is lost.

Operation of 3174 and OS/2 EE as Remote Token-Ring Gateways

OS/2 EE SNA LAN Gateway Support

The OS/2 EE SNA LAN Gateway provides a gateway function between a host (typically via an SDLC adapter) and a set of workstations on one of the IBM LANs. The gateway can also be an OS/2 workstation node. That is, it can contain LUs other than the gateway LUs, and can use other OS/2 EE-CM features.

The host connections include SDLC, Twinaxial, X.25 and Token-Ring. The gateway enables the SNA host to communicate with any LAN workstation implementing either the SNA LU 6.2 or the SNA LU 1/2/3 architectures. Thus APPC/PC and OS/2 APPC applications may use this gateway. In addition, OS/2 3270 terminal emulation, 3270 Emulation Version 3, and 3270 Workstation Program emulation sessions may access the host via the SNA LAN Gateway.

To the host, the LAN Gateway appears as an SNA T2.0 node control unit, supporting one or more LUs per workstation. To the network stations, the gateway will look like an SNA PU 4.0 Communications Controller. The network LUs are not aware of the gateway. They are configured as if they were directly attached to a host.

In reality, the gateway is a special flavor of T2.0 node. It implements the LU functions for dependent workstations, as a real T2.0 node would, but only as long as the corresponding workstation is offline. As soon as a workstation is on line, the gateway will leave to it the responsibility of implementing the LU functions. All LU-LU and LU-SSCP traffic exchanged with the host will simply be passed through the gateway transparently as if host and workstation communicated directly. In support of that illusion, the gateway remains a T2.0 node on the host side, but effectively behaves as a node Type 4/5[41] on the LAN side, forwarding such host messages as ACTLU and BIND.

It is important to understand the principle involved and to contrast it with the principle used in the IBM 3174 remote LAN Gateway. Reference **A** in Figure 47 on page 86 shows a remotely connected token-ring LAN. The box marked TR/GW shows a LAN gateway which could be an IBM 3174 or a PS/2 with OS/2 EE-CM installed.

[41] Neither OS/2 EE nor the IBM 3174 is a node type 4 or 5. What is happening here is that the gateway (OS/2 EE or IBM 3174) relays PIUs between a host connection and a T2.0 node. To the T2.0 node looking upstream it "sees" a Type 4/5 node but that view is through a mirror. The IBM 3174 merely relays PIUs. OS/2 EE relays PIUs also (but changes the headers to make all the LUs appear as if they were inside the gateway PS/2). Neither gateway provides node Type 4 or 5 functions.

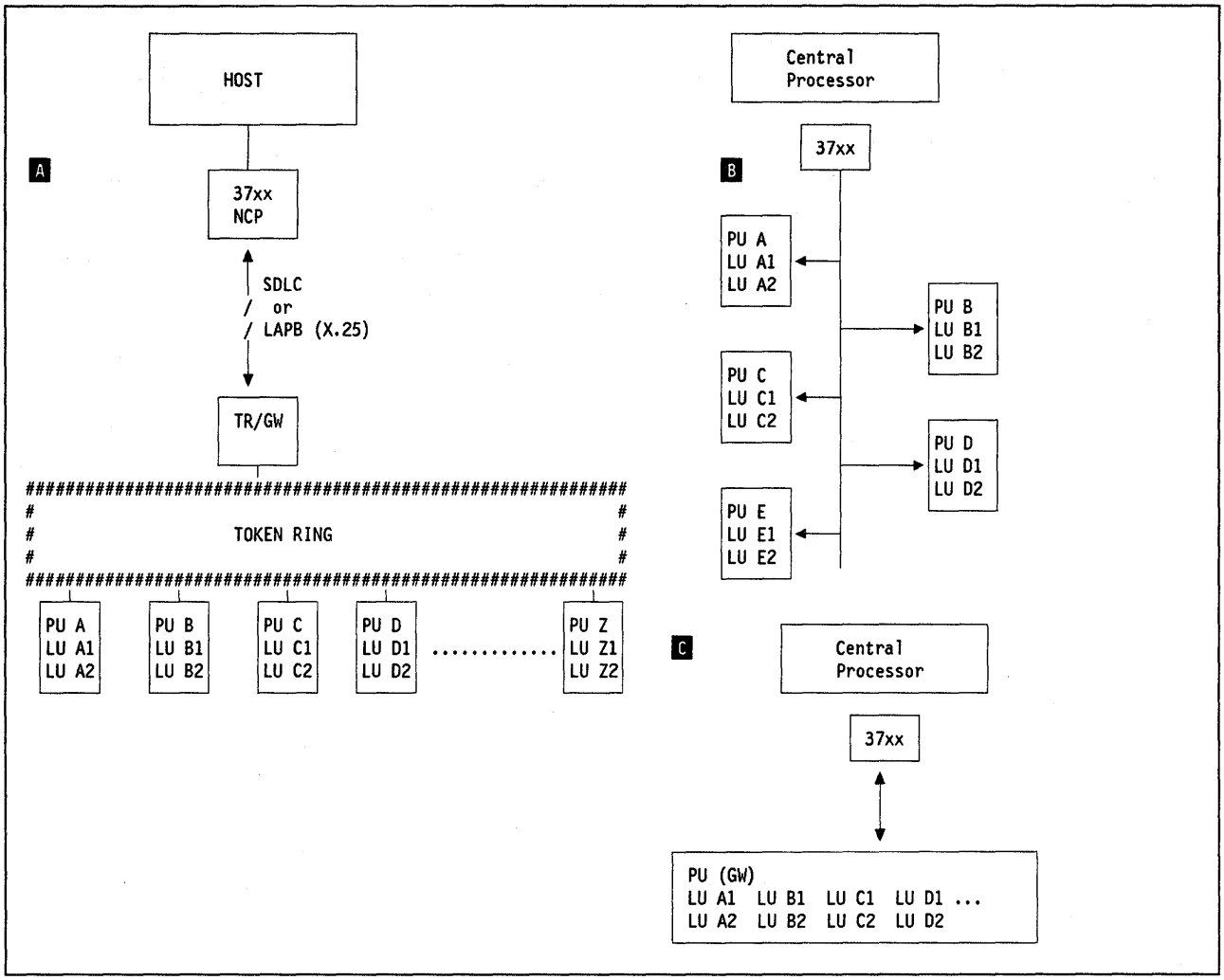


Figure 47. Token Ring Physical Configuration. Each physical device is seen as a separate PU and group of LUs attached to a multidrop link.

The 3174 TRN Gateway is shown in reference **B** in Figure 47. The figure shows the token-ring LAN as seen by the host and 37xx. Each PS/2 attached to the token ring is seen as a separate PU with its own LUs by the host system.

Note: With the 3174 support the upstream host link must be SDLC and cannot be X.25 LAPB.

The OS/2 EE-CM gateway is shown at reference **C** in Figure 47. This shows the host view of the token-ring using OS/2 EE-CM as a gateway. The "downstream PCs" are not seen by the host/37xx complex. The host system "sees" a single PU with a group of attached LUs.

There are many considerations apart from cost that could lead a user to choose one approach over the other.

- The advantage of the 3174 approach is that of flexibility. The 3174 just "passes through" link level blocks from the TRN connected devices and therefore those devices are free to use whatever LU protocols they like. Further each TRN-connected device may have up to 255 LUs in it while the OS/2 EE-CM approach limits the number of LUs sessions using the gateway to a total of 254 at any one time.

  The 3174 could potentially allow PCs connected through it to be either PU 2 or PU 2.1 and therefore allow for full-function LU 6.2 support (as independent LUs). **However, it does not do this.** The 3174 TRN gateway will not process XID3 on behalf of attached T2.1 nodes. Therefore all attached TRN devices can be T2.0 node only.

- The OS/2 EE-CM approach allows for *dependent LU* support only. This limits LU 6.2 sessions passing through the gateway to being single session and secondary (That is, they are able to communicate with a host but not with each other). For full-function LU 6.2 support OS/2 EE-CM would need to be an "APPN Network Node.." OS/2 EE-CM does not have this function; it is an SNA L.E.N. node.
- There are two releases of the 3174 support. The OS/2 EE-CM approach has a very significant advantage over 3174 Release 1 in performance. In the 3174 Release 1 approach, because each PC is seen by the host as a separate PU it must be polled individually by the 37xx.

  For example the 37xx may send a poll to the 3174 specifying PU A but there is no data available from that PU but some from PU D. In this case the 3174 must not send the available data to the host but must wait for a poll specifying PU D. This has the effect of clogging up the line with "unused polling" and places a severe limitation on the number of attached downstream PCs regardless of the transaction rates involved.
- In Release 2 of the 3174 support this "unused polling" problem is solved by instituting a "group poll" of the 3174 controller for all data it may have regardless of which downstream TRN device that data came from. (This operation is a little like the old BSC 3274 group poll.) Therefore in 3174 Release 2 the performance is much better.

There are many other considerations in the choice of a remote LAN connection device (OS/2 EE-CM, 3174, 3745, AS/400, ES/9370,...). Among these are maximum upstream link speed (PC with SDLC is 19.2kbps, 3174 is 64kbps and 3745 can allow much higher speeds and "Transmission Groups" for this link). Also a 3720, a 3745 or an ES/9370 used as a gateway offloads cycles (SNA BNN processing) from the upstream 37xx. The 3174 cannot use X.25 protocols for the upstream link in a LAN gateway. Of course, cost and flexibility for the purpose at hand is most important. Different users will find different solutions optimal.

ES/9370 with VTAM V3 R3 for Remote TRN Connection

As stated above, devices that offer full SNA network function should perhaps not be considered as "gateways" but as direct connections to the network. However, small ES/9370's and 3745's are now available at very low cost and should be considered in the place of gateways. The advantages are:

1. Full LU 6.2 connectivity function (independent LU) is available to all devices connected to the TRN.
2. Multiple upstream networked connections are possible from both ES/9370 and 3745 which are not available for the smaller gateway processors.
3. An ES/9370 may be a LAN file server (using "Extended Connectivity Facility"(ECF).) and provide host support for connected PS/2's. This may go to the extreme of allowing PS/2's without disk media at all to function by taking their files and programs from the ES/9370. This may not be appropriate for all PS/2's on the ring but the ES/9370 is able to be a server for some and network connection point for all.
4. Both ES/9370 and 3745 offer much greater performance than "gateway" boxes both in terms of the maximum number of devices connected and in throughput.
5. Of course, if load permits, the ES/9370 is a full 370 VM processor and may be used for local applications as well as for TRN connection. Such applications as PROFS (electronic mail) fit naturally into the many system designs here.

   In fact, with the VTAM V3 R3 support, the ES/9370 becomes a very attractive vehicle for the solution of many distributed processing problems.

Chapter 7. Technical Details

Node Type 2 Network Interface

The interface to the subarea network for Type 2 nodes has several variations or "flavors." These are summarised below:

Node Type	XID_3	PU-SSCP Session	Dependent LUs	Independent LUs
2.0	no	yes	yes	no
2.1 (a)	yes	yes	yes	no
2.1 (b)	yes	yes	yes	yes
2.1 (c)	yes	yes	no	yes
2.1 (d)	yes	no	no	yes

Figure 48. Node Type 2 Connections to the Subarea Network

The following points are important:

- A T2.1 node is defined by the ability to answer XID3.
- It is possible to have a T2.1 node (a) which has only the functions of a T 2.0 node.
- If there are dependent LUs in the T2.1 node which require sessions with LUs in the subarea network then there *must* be a PU-to-SSCP session from the subarea network SSCP to the T2.1 node.
- If dependent LUs are not present then the PU-to-SSCP session is optional. NetView information is communicated on the PU-to-SSCP session so for the attachment of a single "box" the connection is recommended.

When communicating with other T2.1 nodes, a T2.1 node may have different characteristics depending on the presence or absence of a control point and whether there is a control point session between the T2.1 nodes. Types of T2.1 nodes are introduced in "APPN Network Structure" on page 54. A comparison of node characteristics is shown in Figure 74 on page 161.

When communicating with non-subarea T2.1 nodes, a T2.1 node may never have a PU-to-SSCP session nor may it contain dependent LUs. These are subarea network entities only.

Boundary Function

When a T2.1 node or a T2.0 node is connected to a subarea network it is joined to the network by a logical processing entity called boundary function (BF) or Boundary Network Node (BNN). BF is needed to interface external nodes to the subarea network regardless of what physical medium is used to make the connection (SDLC link, Token Ring, X.25 Network or IBM S/370 Channel).

BF is the boundary of the SNA transport network. The important functions it performs are as follows:

1. Transforms network addresses by which LUs are known in the transport network to local addresses and LFSIDs on the link to the peripheral node.
2. Forms the end point of the routing structure within the SNA transport network. (BF is the junction of the virtual route (VR) and the route extension (REX).)
3. Acts as a staging point for flow control protocols (PACING).
4. Breaks data blocks (PIUs) up into smaller units (segments) to enable them to match the characteristics (such as buffer size) of the receiving peripheral node.

Boundary function exists within both NCP in the 37xx and within VTAM for local device connection (such as channel-connected 3174) and ICA connection to 9370 style processors.

BF is the major component of the SNA subarea network affected by the introduction of T2.1 node support. Of course, BF requires support from a control point and that control point must be its System Services Control Point within its owning VTAM.

BF support for T2.1 node is essentially the same for both NCP and VTAM connections. In this document therefore unless otherwise noted, the term "BF" is used to denote either VTAM_BF or NCP_BF.

End-to-End Communication

The T2.1 node interface is used for many purposes:

- To interface a single device or system to a subarea network.
- To interface different nodes to each other in an APPN network.
- To interface an APPN network to a subarea network.
- Under some circumstances, to interface two subarea networks to each other.

The subarea network has many internal mechanisms to control congestion, to effect resource optimisation and to allow route selection, etc. APPN networks have facilities which perform similar functions but in quite a different way. When the T2.1 node protocol is used as an interface between the two it is necessary to accommodate the differences between subarea SNA and APPN in order to perform these functions properly. The more important of these are:

Network Identification

The subarea network makes only limited use of the Network Identifier. APPN networks use network names extensively and are able to process all resource names with a network qualifier (see "Network Qualified Names" on page 99).

Session Identification

In the subarea network, the primary identifier of a session is the pair of origin and destination network addresses. In APPN there are no network addresses as such and sessions are identified by the FQPCID (see "FQPCID" on page 98).

Congestion Control

Because the APPN network has different kinds of congestion control from the subarea network, the mechanisms used in both kinds of network need to be joined in a seamless way to achieve effective control of congestion in the whole network. "New" mechanisms that need to be accepted by the subarea network are:

- BIND Pacing
- Adaptive Session Pacing
- Segmentation.

Logical Units (LUs)

LU Name

The name of an independent LU in a T2.1 node has end-to-end significance.

Dependent LUs can be identified solely by the index number in the transmission header (FID2) and thus it is not necessary for the dependent LU name as known by the network (SSCP) to be the same as the LU name known by the LU itself (if any).

Independent LUs are different. As described in "Operation of an SNA Type 2.1 Node" on page 68, there is no longer a fixed addressing relationship between LUs within the node and network addresses and LU names within the network.

When an independent LU sends a BIND the destination LU (secondary LU) for the session is identified by the LU name within the BIND. Likewise the only identification of the sending LU is its name, also present in the BIND.

For most purposes this LU name is qualified by the name of the network within which it exists. See "Network Qualified Names" on page 99.

Classes of LU

There are four different classes of LU distinguished by their mode of operation and location within a subarea network.

Host LUs are the traditional LU type assigned to S/370 host-based applications. These LUs are controlled by the SSCP and have a notional SSCP session so they should be considered dependent LUs. (They can send and receive control information to/from their SSCP.)

Host LUs may be primary or secondary. They may have multiple and parallel sessions with other LUs throughout the network.

With the T2.1 node support release of VTAM (V 3.2 and later), host LU support is extended such that all host LUs send "extended BIND" where appropriate. Thus they interface correctly and effectively with independent LUs in the network. Host LUs can always receive extended BIND.

The sending and receiving of extended BIND is masked from the application program interface (API) by VTAM.

Network resident dependent LUs are the "business as usual" LUs that have existed within T2.0 nodes for many years.

These LUs may only have one session and *must* be secondary. They may be present in either a T2.1 node or a T2.0 node.

Independent LUs exist within T2.1 nodes and have full network function. They may be primary or secondary and have multiple and/or parallel sessions. They are always capable of receiving an extended BIND but send either an LU 6.2 BIND or an extended bind.

Network resident dependent PLUs are a special case. This class of LU was introduced into SNA to cover application code (user or IBM) written to execute within the 37xx communication controllers under control of NCP.

The IBM products X.25 XI, NRF and NSI make use of this interface.

These LUs are capable of only a single session, and are dependent (have an LU to SSCP session) but can be primaries. In this way limited cross-network session capability has been available within SNA for some years.

LU Types supported over T2.1 node interface

LU-to-LU protocols within SNA are constructed from a range of protocol elements which are sent between communicating LUs without the involvement of the intermediate transport network. These protocol elements are carried in the RH (Request Response Header) and in the FMH (Function Management Header).

The way in which protocol elements are to be used by the communicating LUs is agreed at session setup time by passing "profiles" in the BIND command. These are called the Transmission Services (TS) profile, the Function Management (FM) profile and the Presentation Services (PS) profile.

"LU Types" on page 59 describes various "LU types." These LU types are in reality combinations of TS, FM and PS profiles appropriate to a given function.

In general, the transport network part of SNA is insensitive to the LU protocols in use over it. That is, T2.1 nodes may contain **any** LU type and **any** LU type may be an independent LU and may have direct sessions with any other compatible LU within the network.

The T2.1 node functions are not limited to LU 6.2 only. However, this must be qualified in two ways:

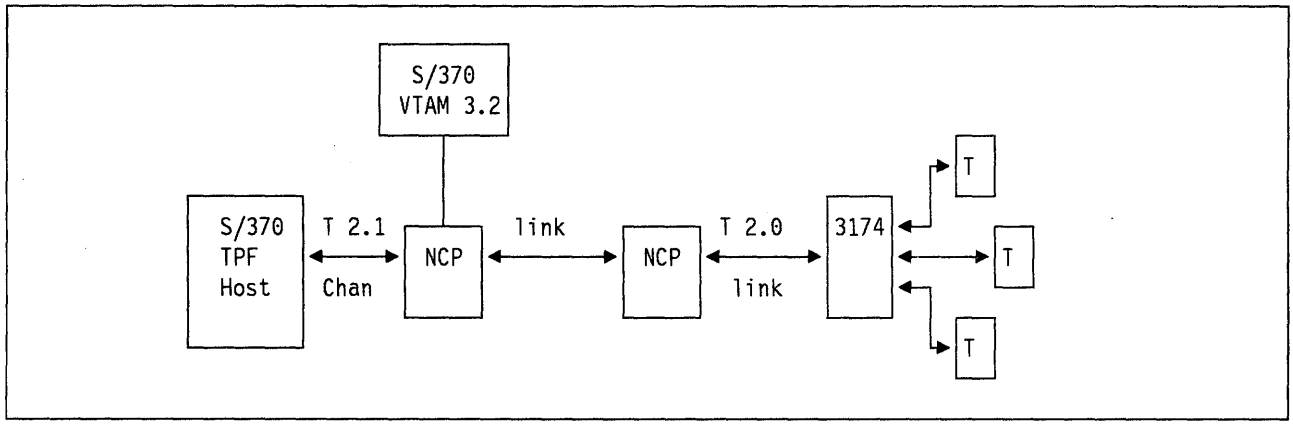
1. LU 6.2 is the "strategic" LU type for SNA communication and so the vast majority of new IBM development uses LU 6.2. At the present time, to the knowledge of the author, there is only one implementation of an independent LU within an IBM T2.1 node product that is not an LU 6.2. This is the case of the LU 2 implementation within TPF. TPF contains independent LU Type 2's and they send BIND directly to dependent LU Type 2's within the network.

   However, BF does support all LU types as independent PLUs within the T2.1 node. An example of this is shown in "LU Type 2 independent LU connection." on page 93 below.

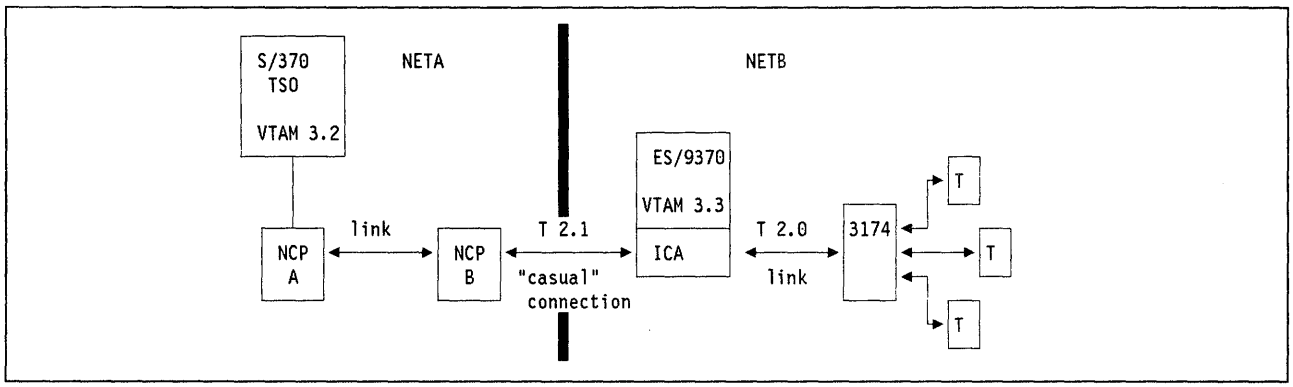
2. Independent LU 6.2s may send non-extended BINDs and have them extended and processed correctly by BF (with a little help from the SSCP).

   This is discussed in "LU 6.2 BIND" on page 96.

LU Type 2 independent LU connection. The subarea network (both using VTAM and NCP BFs), supports traffic from all LU types as independent LUs, not just LU 6.2. Consider the following example:



The TPF processor is seen by VTAM and by NCP as a channel-connected T2.1 node. This was discussed in "Example of 3270 Logon to a Type 2.1 Node (TPF)" on page 76. TPF sends a BIND into the subarea network as an independent LU acting as a PLU. This is an LU 2 extended BIND being processed by the subarea network. This configuration is supported. Another interesting variation is as follows:



In this figure the VTAM 3.2 processor containing TSO and the two NCPs (A and B) form a subarea network. The VTAM 3.3 processor with its attached 3174 forms another subarea network. As shown, the two networks are connected by a T2.1 node casual connection.

Provided something happens to initiate the session, (such as an operator typing "VARY LOGON... ") and provided an appropriate definition for the 3270 devices is present in NETA, then TSO will send a BIND to what it "sees" as an independent LU type 2. VTAM will build an extended BIND and send it.

It is an interesting side point that the 3174 connection to the ES/9370 **may even use BSC link control**. (It would normally be SDLC/SNA but in this case BSC will work because VTAM internally treats BSC and SNA devices alike.) This "trick" will not work with NCP.

Dependent LU 6.2s

LU 6.2 was designed to operate with parallel sessions between communicating LU 6.2 session partners. A subset of LU 6.2 was also foreseen in the architecture to allow for single sessions with "closed box" implementations such as the IBM 3820 printer. (A "closed box" implementation is one where no user programming interface is available.)

The distinguishing feature between single session and parallel session LU 6.2s is the ability to handle the

CNOS (change number of sessions) protocol. Of course, if an LU 6.2 is to be parallel session capable then the underlying network connection must also allow for parallel sessions.

There is however, the case of LU 6.2 dependent LUs where the LU can only have a single secondary session because of the nature of the underlying network support. These LUs may have a user programming interface.

When the session is started there is a bit in the BIND command (byte 24, bit 6) which requests that parallel session support be used. This bit is set by the PLU when sending the BIND. Of course, this bit must not be set if the SLU is a dependent LU. Various PLU implementations specify this request differently. If this bit is not set correctly, the following problems are possible.

When VTAM processes a received BIND, it looks in the PSERVIC entry of the MODETAB for a bit to say that the secondary LU is capable of parallel sessions. If the bit in the received BIND differs from that in the MODETAB then VTAM fails the BIND. This does not matter for host-based PLUs because VTAM builds the original BIND from the MODETAB entry before presenting it to the PLU for processing before it is sent.

However, if a BIND is received from a PLU in a T2.1 node then VTAM will check the SLU definition and fail the BIND if a difference is found. (If a null MODETAB entry is used with an entry only and without any definitions, VTAM will not fail the BIND.) If the MODETAB entry is wrong and a parallel session requesting BIND is accepted by a non-parallel session LU then the next command from the PLU will be CNOS and this will then fail.

Initiating Sessions

BIND Processing

When a BIND is received by a BF (either in NCP or in VTAM) there is no origin LU or destination LU network address present to enable the BIND to be routed to its destination. Other necessary information is missing from the BIND at this time.

When the BIND is received the information from within it is sent from the BF to the SSCP for processing. The SSCP must:

- Locate the destination LU in the network.
- Communicate with the "SSCP owner" of the destination LU within the network.
- Activate a route through the network for the new session to use.
- Allocate network addresses for the LUs involved in the session (or determine what they are if already allocated).
- Locate an appropriate COS table entry and allocate a COS and TPF.
- Calculate and assign an FQPCID for the session (if one is not already present).
- Extend the BIND as needed.

When the SSCP has completed these things it must send the information to BF so that BF can then send the BIND on towards its destination.

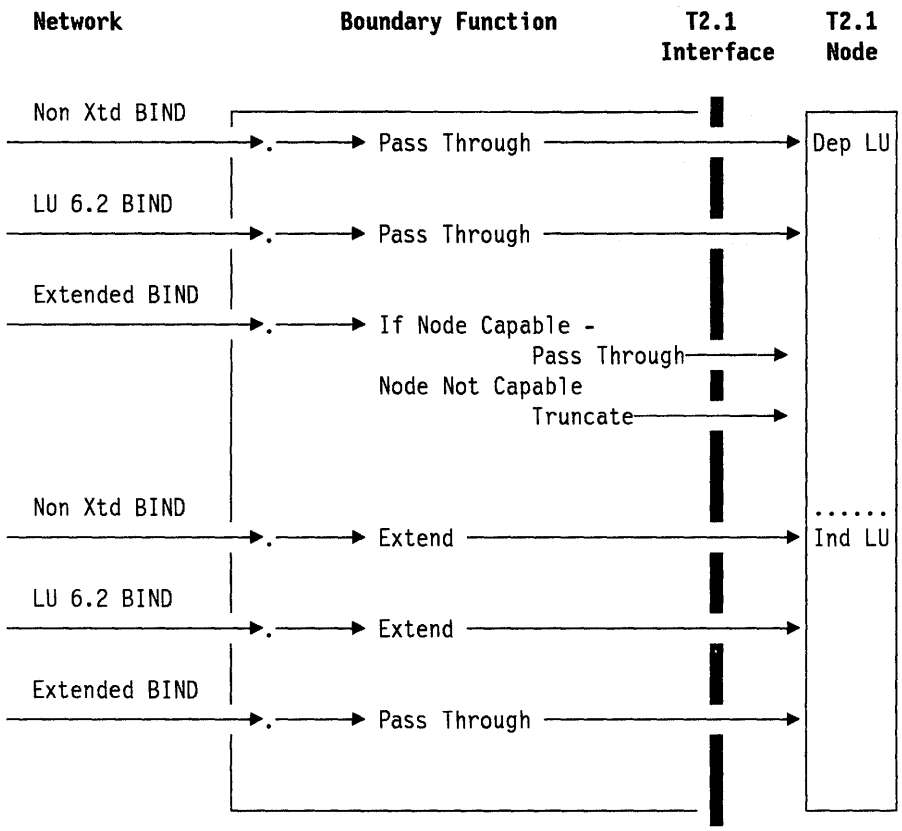


Figure 49. BF Processing of BIND Received from the Network

Extended BIND

In order for the session to be set up correctly when a BIND is received from or sent to an independent LU, additional information is needed in the BIND. Some of this information is essential for correct operation of the T2.1 node interface to the subarea network and some is accepted for compatibility with the BIND used by APPN and L.E.N. nodes.

An Extended BIND is a BIND which contains a Fully Qualified Procedure Correlation Identifier (FQPCID). (See "FQPCID" on page 98.) In addition, extended BIND may contain information to activate a session, to make use of features such as Adaptive Session Pacing, and to identify a particular session. An Extended BIND contains:

1. Both PLU and SLU name fields. These may be network qualified and now *must* be present.
2. A control vector (except for the existing XRF CV27 that exists in a current non-extended BIND) and the Control Vector Included Indicator (CVII).
3. A control vector containing the FQPCID.
4. An Adaptive Session Pacing Indicator (ASPI)
5. A Whole BIUs Required Indicator (WBRI)
6. The Network Name control vector (contains the real PLU Network Name (Network ID) if the PLU name field is uninterpreted).

   Note: An uninterpreted PLU name would usually come from an INITSELF command sent by a dependent (secondary) LU. Since it is unusual for independent LUs within T2.1 nodes to set up

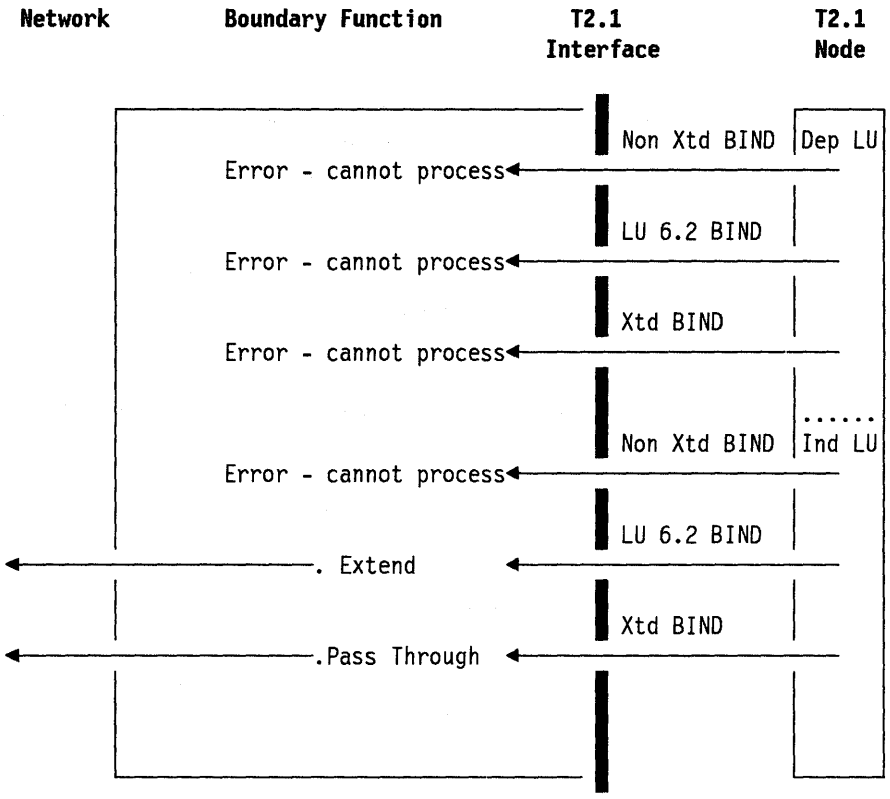


Figure 50. BIND Received by BF from Node 2.1 Device

sessions this way, (the technique is only used by TPF; See "Example of 3270 Logon to a Type 2.1 Node (TPF)" on page 76), it would be unusual for this field to be present.

7. The COS/TPF control vector
8. MODE control vector

An extended BIND may contain network qualified names (NQNs) in the NS fields identifying the origin and destination LUs. See "Network Qualified Names" on page 99. A BIND must be extended before it may contain NQNs. An extended BIND may or may not include NQNs. If an extended BIND does not contain NQNs in its NS fields then it **must** contain control vector x'0E' which contains the NQN of the initiating PLU.

LU 6.2 BIND

In some past implementations of T2.1 nodes, (L.E.N. nodes), it was not necessary to send an extended BIND. (In the peer environment an FQPCID hardly seemed necessary!) In order to get full networking function through the subarea network however, extended BIND is needed. It is desirable however, to allow these older types of L.E.N. nodes fully functional network attachment if at all possible.

LU 6.2 defines a "user field" in the BIND which contains a Network ID (NETID). There is therefore enough information in an LU 6.2 BIND but it is not in the correct fields for processing by the network.[42] When an LU 6.2 BIND is sent into the network (either subarea or APPN) it is changed into the Extended

[42] The principle of "layered architectures" calls for the routing information in the BIND to be placed in the structured NS fields. The process of extending the BIND is a method of extending support to older devices which could not be

BIND format (if required) by the owning control point (which will include its own name as the CPNAME in the NQN field).

BIND Queueing

In the APPN network use is made of a facility called "BIND queueing." This is not used in the subarea network. Within the APPN network, if a BIND is held up within the network for some reason it can be queued **indefinitely**. There is no response, the session setup just "hangs." It is up to the PLU to "timeout" the BIND and send an UNBIND if it takes too long and there is need to free up resources.

In the subarea network, since this cannot happen there is no standard support to allow for the relevant timeout process. Standard IBM subsystems will *never* set the queue bit in the BIND. (It is byte 6 bit 7.) User written code, however, may.

Care is necessary when sending BIND from a user-written VTAM program, for example, to avoid setting this bit as the session setup could be hung indefinitely.

BF processing of BIND

Figure 49 on page 95 and Figure 50 on page 96 show the way BF processes a BIND.

Note: BF processing of BIND is performed with the assistance of the SSCP.

The rule is that the BIND is extended if needed as soon as it is possible to do so. The discussion here assumes that all nodes are at the latest software levels. Complications in the diagrams (such as BF receiving a non-extended BIND from the network) are caused by the certainty of "back-level" nodes being present in the network.

Host LU BINDs

VTAM *always* extends a BIND from a host LU if the subarea containing the destination LU supports extended BIND. This means that the BIND is extended if it is to be sent to a host LU (controlled by VTAM 3.2 or later) or to a BF which supports extended BIND regardless of whether the destination node (or LU) supports the receipt of extended BIND.

The (host LU) BIND is extended according to the rules, regardless of the LU type in use. (It happens for all LU types - not just LU 6.2.)

When the BF receives an extended BIND it decides how to process it depending on the type of node and the characteristics of the destination LU.

When dependent LUs are activated they respond with an indication of whether they can accept extended BIND in the response to the ACTLU command. Independent LUs (when acting as secondaries) can always accept extended BINDs.

---

supported any other way. It should be regarded as a temporary mechanism which is present for migration purposes.

FQPCID

PCID

In the SNA subarea network, the primary way that a session is identified is by the pair of network addresses of the communicating LUs. (In the case of parallel sessions multiple network addresses are allocated for the same LU to preserve uniqueness of session identification.) In an APPN network, network addresses are not present as such and a session path consists of a sequence of nodes, links and "local form session identifiers." For routing purposes and for control purposes where the control signals flow on the session involved, this is quite sufficient. However, there are times when a session reference is needed that is independent of the routing mechanism.

In the past, SNA networks have used a "Procedure Correlation Identifier" for a number of purposes:[43]

- When information about a session is carried on another session.
- To provide a long-term identification of the session for customer problem determination purposes.
- Identification of the session for user accounting purposes.

In the subarea network, the PCID is essentially a hash (random) number which is qualified by (concatenated with) the network address of the control point that created it. The intention is simply to provide a unique number which can be used to identify a particular instance of the session.

All PCIDs are eight bytes in length.

PCID in APPN

In APPN networks a PCID is required for the same purposes as discussed above. Since network addresses do not exist in APPN, a hash of the CP name of the CP that created this PCID is used instead of its network address as a qualifier within the PCID structure.

Qualification of PCID

Because the interconnection of networks is becoming increasingly necessary, the architecture of PCID has been extended to include the network identifier and the control point name of the control point which generated it. (This applies to both APPN and to subarea networks). This is the reason that the NETID and SSCPNAME parameters are now required VTAM start parameters - so that VTAM has a name to put in the FQPCID.

The new PCID is called an FQPCID (Fully Qualified PCID). It is generated somewhat differently from the PCID but is still, conceptually, a random number preceded by the name of the control point (or SSCP) responsible for setting up the session. The name used in the FQPCID is the "Network Qualified Name" (NQN). A NQN is simply an LU name concatenated with the name of the network in which it exists. In the FQPCID the NQN is the name of a control point. So an FQPCID is essentially "netname.cpname.######."[44]

This provides a number of significant improvements:

- The FQPCID can be stored at each node through which a session passes and can be used for network management purposes.

---

[43] While the discussion here centers on VTAM and NCP in the subarea network and AS/400 in the APPN network it should be noted that PCIDs are generated and used by some older SNA SSCP implementations. These are TCAM and ACP/TPF.

[44] ###### represents a random number.

In subarea network SNA, when sessions pass through SNI gateways, LU names may be translated, network addresses always change.[45] But the FQPCID is *not* translated. This enables the correlation of instances of the same session crossing different subarea networks to be identified for problem resolution and diagnostic purposes.

- There are fewer chance of "collision" (two identical FQPCIDs being allocated in the network for different sessions) than with PCID.
- In releases of VTAM prior to VTAM level 3.2 every time a session crossed an SNI network gateway a new PCID was calculated and the PCIDs were "swapped." This was done to ensure a unique PCID within each network.

  FQPCIDs are not swapped (except when they are processed by "back-level VTAMs"). This improves performance somewhat and gives much improved problem resolution capability.
- There are a number of network internal functions that find the FQPCID a convenient way of identifying sessions.

Type 2.1 Node CP Name

With VTAM V3 R2, there is no need to specify IDBLK and IDNUM for Type 2.1 nodes. VTAM uses the CP name of a T2.1 node, if defined, to find the associated PU and LU definition statements. VTAM uses the CP name to locate switched node definitions, even if the boundary function does not support T2.1 nodes (such as VTAM boundary function and back-level NCP). IDBLK and IDNUM can continue to be used.

VTAM allows a T2.1 node CP to be defined as an LU. VTAM supports LU-LU sessions with the CP.

The VTAM SSCP does not participate as a CP in CP-CP session with T2.1 node.

Network Qualified Names

Ever since the invention of SNA, every resource (PU, LU or SSCP) in an SNA network has been identified by an eight character alphanumeric name. Network internals have changed radically over time but resource identification has been stable.

This single-level name structure has been found adequate for single network use even in very large networks. However, when networks are interconnected there exists a problem in the potential definition of duplicate names in the different but interconnected networks.

In SNI, this problem was solved by creating alias names and allowing for the (optional) translation of names from real names to alias names over a network boundary.

In APPN, a different approach is used for interconnection of APPN networks called network qualified names (NQN).

---

[45] When LU 6.2 sessions pass through SNI gateways, LU names **must not** be translated. To be pedantic, LU names may be translated but **only** to themselves. This is because in LU 6.2 the communicating LUs identify each other by the LU names in the USER fields of the BIND. SNI translates only the LU names in the NS fields of the BIND. When an LU 6.2 receives a BIND it may increase the number of sessions by sending another BIND back to the originating LU 6.2. But the BIND sent under these conditions will be directed to the LU name from the user (untranslated) field of the previously received BIND. Thus this BIND could not be routed to its correct destination.

A network qualified name consists of the resource name (usually LU name) prefixed by an eight character network ID (NETID) and separated by a period. Thus "NETID.LUNAME" is a network qualified name. NQNs can uniquely identify an LU over as many network interconnections as desired.

The use of NQNs removes the need for alias name translation over network boundaries. This gives significant advantages in processing requirements in gateway nodes and also removes the need for systems programmer involvement in alias definition. SNI uses alias translation because the use of NQNs universally throughout the SNA network would have meant changing the VTAM application program interface (API) in a way that would have forced all programs using that interface to make significant change.

LU 6.2 use of NQNs

In the architecture of LU 6.2 the LU names of session partners in different networks are NQNs. This usage allows the LU names to be assigned independently of one another. This applies to all implementations of LU 6.2 - even those in the subarea network.

In order to communicate NQNs between session partners over a network connection which did not allow for NQNs, LU 6.2 places the NQNs of the session partners in structured subfields in the "user" area of the BIND. The user field of the BIND is not (normally)[46] examined by the lower layers of SNA (path control etc.) and is merely a way for two LUs to exchange information at session start up time.

When an LU 6.2 session is bound through an SNI gateway, the origin and destination LU names in the formatted fields of the BIND may be translated (via the alias mechanism) but **alias translation is never performed on the NQNs in the user field of the BIND**. This brings the possibility of having different LU names (in a BIND) for the same real LU.

Note: For the above reason, alias name translation should **never** be performed for LU 6.2 sessions. Sometimes alias translation cannot be avoided and in this case the name must translate to itself.

1. LU 6.2 ignores the NETID from the NS fields (the origin NETID as known to VTAM) and identifies its partner by the NETID.LUNAME (NQN) in the user field.
2. VTAM identifies all LUs simply by their unqualified LU name. So when the LU 6.2 sends a BIND then VTAM just ignores the network name and processes the LU name as an LU within this subarea network. (Of course if the LU 6.2 is VTAM's or in a subsystem connected to the VTAM API, then there is no way for the LU to pass an NQN to VTAM anyway - since the VTAM API cannot recognise NQNs.)

This leads to the restriction that while a given LU 6.2 could potentially have simultaneous sessions with many LUs of the same name but in different networks, VTAM will not allow this to happen because it leads to an LU name conflict.

For example the NQNs "NET1.CICS" and "NET2.CICS" are different names as far as another communicating LU 6.2 is concerned. As far as VTAM is concerned they are the same name and cannot be known simultaneously within the same network.

---

[46] VTAM and BF use this information when extending a BIND from "LU 6.2 non-extended" to "LU 6.2 extended" format.

VTAM API Use of NQNs

There is no way for an application program to request a session by specifying an NQN as the partner LU name. The VTAM API does not recognise NQNs.

In general, when an NQN is present, VTAM will only use the LU name portion and will not use the NETID to identify the LU. VTAM will nevertheless check NETIDs under many conditions and refuse the session if errors are found. When VTAM builds a BIND on behalf of an API program, the DLU name will be qualified with the correct NETID of that LU as known to VTAM. (This applies to SNI connected networks and to T2.1 node connections which are not (PU type 4/5). BINDs addressed to a "casual" connection will not be extended.

However, VTAM will pass the user field of the BIND to a using LU and will accept a user field from that LU and this field may contain an NQN.

Note: This will happen for example when CICS is acting as an LU 6.2. In the VTAM internal implementation of LU 6.2, NQNs in the user field are processed by VTAM directly.

Also, when a BIND arrives from a T2.1 node to a host LU, if that BIND contains an NQN in the NS (structured) fields, VTAM will pass the NQN to the user in the LOGON exit.

VTAM Building of Extended BIND

As discussed in "Extended BIND" on page 95 and in "Host LU BINDs" on page 97, VTAM constructs extended BINDs wherever possible both for its owned host LUs and for BINDs arriving in BF from T2.1 nodes.

VTAM builds extended BINDs on behalf of owned host applications and extends BINDs received from T2.1 nodes as appropriate.

VTAM will put NQNs in the NS fields of the BIND if the SLU supports receipt of extended BIND and if all the SSCPs involved in setting up the session are post VTAM 3.2 level.

Rules for receipt of NQNs in extended BINDs are as follows:

1. Independent LUs are always capable of receiving NQNs in the NS fields of the BIND.
2. Host-based (VTAM API) LUs are never capable of identifying a communicating partner LU by its NQN. The VTAM API uses unqualified names only and there is no way for a host LU to specify an NQN as the identifier of an LU.

   VTAM will present the NQN of a BIND sender to a secondary host LU in the SCIP exit when a BIND is received with the NQN of the PLU in its NS fields.

   An LU 6.2 can identify the NQN of its partner from the NQNs in the "user data" fields of the BIND. This is not part of the VTAM API.
3. Network-based dependent LUs are sometimes capable and sometimes not capable of receiving NQNs in extended BIND. Some dependent LUs are not capable of receiving extended BINDs at all.

NQN Processing of Inbound BIND

An inbound (LU 6.2, extended) BIND from a T2.1 node to BF can contain five instances of NQNs.

1. The origin and destination LU names in the formatted (NS) fields of the BIND may be NQNs.

These are the fields that identify the origin and destination LU names to the network.[47]

2. If the LU type is LU 6.2 then there will be NQNs for the origin and destination LUs within the user field of the BIND as discussed above.
3. The FQPCID contains the NQN of the control point that created it.

The processing rule is that VTAM identifies LUs by their name regardless of their NETID qualifier.

The following rules apply to alias name translation of inbound BINDs received from T2.1 nodes.

- If the PLU name received in the BIND from the LU is network qualified, then the SLU name **must** also be network qualified.
- If the names are network qualified they must be the real names (That is, they must be the same as the names in the user field), or the session setup fails.
- A session cannot be concurrently established to different destination LUs with the same LU name and different NETIDs. (VTAM uses the LU name only as identifier, without regard for the network qualifier.)
- If the SLU name received in the BIND is network qualified, and the SLU is from a different network from the PLU, the SLU's alias name as known in the PLU's network must be the same as the SLU's real name.

VTAM 3.2 checking of NETID. Whenever VTAM 3.2 receives a BIND containing an NQN, it checks the NETID against its own ID **and fails the BIND if there is a difference.**[48] This means that all APPN nodes connected to a subarea network using VTAM 3.2 must have the same NETID.

VTAM 3.3. This release of VTAM allows for "casual" network connection as described throughout this document. It would be possible to assign the same NETID to all networks that want to casually connect to one another, however, in many situations that would be extremely difficult to administer. It is necessary to allow casual connection between networks of different names.: The AS/400 APPN network uses the NETID to identify an LU. (NETID is just a part of the LUNAME for identification purposes). To get proper communication between AS/400 APPN networks across the subarea network it is necessary to allow for a NETID different from the name of the subarea network.

Network ID in XID3

The NETID appears also in the XID3 received from a T2.1 node. In the initial release of VTAM 3.2 and NCP 4.3/5.2 VTAM checked this to ensure that the NETID was the same as itself. If it was not then the connection was refused. This was subsequently removed via a "temporary fix."

With the release of VTAM Version 3 Release 3 this checking has been changed again and is described in "Non-Native Network Connection (NNNC)" on page 103.

The AS/400 however, uses the NETID to establish the identity of an attached T2.1 node. The AS/400 treats the NETID as a part of the LU name for resource identification purposes. However, the AS/400 does not use the NETID as a way of delimiting the scope of a particular AS/400 network. Each AS/400 in an APPN network may have a different NETID. Nevertheless the differently named AS/400s may form a single, integrated, APPN network.

---

[47] The LU names referred to here are LU names in bytes K+1-m and p+1-r, the NS name fields, in BIND. (see "Requests RU" in *SNA Formats*, GA27-3136.)

[48] VTAM does not check the NQNs in the user field.

Non-Native Network Connection (NNNC)

In the initial support for connection of T2.1 nodes to the subarea network (in VTAM V3R2, NCP V4R3 and NCP V5R2), there was the stated requirement that all of the connecting T2.1 nodes must be part of the same SNA network. That is all connecting T2.1 nodes must have the same NETID as that of the subarea network. (All VTAMs in a subarea network must have the same NETID.) The NETID field was checked in the XID-3 exchange and the connection failed if the NETID of the attaching T2.1 node was found to be different from that of the network.

Many users, however, wanted the ability to attach devices like the As/400 to many different subarea networks. As a *temporary fix* a PTF was distributed for VTAM V3R2 which eliminated the XID-3 checking. This enabled users to have dependent LU sessions from T2.1 nodes to hosts in the subarea network. (For example, this allowed 3270 pass-through sessions from the AS/400.) However, AS/400-to-AS/400 sessions over the subarea network were not possible nor was any session involving independent LUs. (VTAM was unable to handle the presence of a NETID different from its own in the BIND.)

With the announcement of VTAM Version 3 Release 3 a new function called "Non-Native Network LU Association" has been introduced which allows the attachment of T2.1 nodes regardless of NETID differences. This NNNC feature is also implemented by PTF for VTAM Version 3 Release 2. This change applies for ES/9370 attached links as well as for NCP connected T2.1 nodes. For NCP connections NCP Version 5 Release 3 is required.

Concept

NNNC implementation is conceptually very simple. VTAM will process and route correctly BINDS including NQNs with NETIDs different from its own but *VTAM will identify each LU uniquely by using only its 8-character LUNAME.* VTAM will not allow two LUs of the same unqualified LUNAME to exist simultaneously within its directory.

If for example an LU is currently known by VTAM as NET3.LUXYZ and a BIND is received from a different LU directed to NET4.LUXYZ, VTAM will fail the BIND. In this case VTAM would recognise LUXYZ as a unique LU and the conflict in NETID would be regarded as an error.

Internally, VTAM is able to locate the LU by its unqualified LUNAME, its real network qualified LU name (for example NET2.LUABC) or its LU name qualified by VTAMs own NETID.

Whenever it builds a BIND, VTAM will use the real network ID associated with the LU.

This means that VTAM will route BINDS through its network correctly based on the LU name alone for routing. VTAM will check the NETID of the destination for consistency.

XID-3

The NNNC feature changes the way XID-3 is handled.[49] When a PU is defined to VTAM it may optionally have a NETID specification. (This is specified in the NETID parameter of the PU type 2 definition.)

When the XID-3 is processed, VTAM will check the NETID from the received XID-3 with the NETID specified by the user.

- If there is a difference then VTAM will refuse the connection. (This is different from the older design where VTAM ignores the mismatch.)

[49] This discussion applies to both NCP and to ES/9370 ICA connected T2.0 node and T2.1 nodes.

- If no NETID was specified by the user then VTAM will store the received NETID and use it for all LUs within the attached T2.1 node.

Casual Connection

When two subarea networks are connected using the casual connection feature, the NETIDs of the two interconnected networks will normally be different.

When the XID-3 is processed there is a node type identifier received in the XID-3. If this node type identified type 4 or 5 then VTAM recognises this as a casual connection.

In the case of casual connection, VTAM marks the PU as being incapable of handling extended BIND. When a BIND is directed from one network to the other, it will be "unextended" (if necessary) by the VTAM owning the connection and extended again by the receiving VTAM. This results in the LUNAMEs being qualified in each connecting network with the NETID of that network. NQNs are NOT passed across the casual connection.

Independent-to-Independent

In the case of T2.1 nodes which are not network connections, any form of BIND (non-extended, LU 6.2 and extended BIND) are allowed across the connection. BINDs are routed through the network based on LU name alone regardless of the NETID qualifier.

VTAM will include the correct NETID qualifier on every LUNAME in every BIND which it has to build or extend.

VTAM will check NQNs received to ensure the network names are consistent with the NETID of the T2.1 node in which the LU (OLU or DLU) resides. If a difference is found VTAM will refuse the connection (fail the BIND).

VTAM Messages

VTAM operator messages (and information passed to NetView) have been updated to include the real NETID of the LU in displays which contain the LUNAME.

Automatic Logon

Although an independent LU may be a primary LU, it is never possible for a secondary LU to log on to a primary, independent LU. This covers all cases where the SSCP would need to signal the PLU that the SLU requires a session.

Thus:

1. As discussed in "Dependent LU to Type 2.1 Node Networking" on page 75 and "Example of 3270 Logon to a Type 2.1 Node (TPF)" on page 76, a secondary LU cannot use "Initiate Self" or an unformatted (character coded) logon to request a session with an independent LU.
2. For exactly the same reason as quoted above (the absence of an LU to SSCP session), it is not possible for the operator to log a secondary LU on to an independent primary.
3. Again, no secondary LU (or secondary capable LU) may refer to an independent LU in the "LOGAPPL=" parameter.

However, independent LUs may be secondary LUs and as such may have sessions initiated either by the operator or at activation by use of the "LOGAPPL=" parameter on the LU definition in VTAM (As is usual with secondary LUs).[50]

Note: This may not always work since VTAM does not activate independent LUs. When a T2.1 node becomes active, VTAM will mark the state of all named LUs within it as active (if ISTATUS=ACTIVE was specified). But the LUs so named need not even be within that physical T2.1 node. They could be somewhere else in an attached network. In this case, if the named LU is inactive or inaccessible, the automatic LOGON will fail.

For dependent LUs (since they only have a single session), whenever a session terminates, VTAM will attempt to initiate a logon to the APPLID specified in the LU definition. Since Independent LUs are capable of having multiple sessions, there exists a possibility of VTAM setting up multiple (unwanted) sessions with a "controlling" primary LU.

To avoid this problem automatic logon processing is modified for independent LUs. VTAM does not automatically initiate a new session with a controlling LU if one is already established. A new session is initiated by VTAM if:

1. There is no session between an LU and its controlling PLU, or
2. The only existing session with the controlling PLU terminates.

If the attempt to initiate a controller session fails, VTAM will not attempt to initiate a session until an existing session with the LU terminates or the operator enters a command such as VARY NET,LOGON.

Information Transferred during Link Establishment

The primary distinction between a T2.0 node and a T2.1 node is that the T2.1 node always performs an XID format 3 (XID3) sequence during link establishment (for both leased and switched connections). (After the XID3 exchange the link is set up normally using the SDLC SNRM command.)

The T2.0 node does not understand the XID3. Connection establishment for T2.0 node on a leased line is performed using the SNRM SDLC command. On a switched line a T2.0 node exchanges a format 0 XID command, which contains some identifying information.

The link establishment sequence used by the T2.1 node is described in "DLC Activation" on page 136.

The information passed by the T2.1 node to the SNA BF is summarised as follows:

Node identification

These are the "IDBLK" and "IDNUM" fields. A T2.0 node passes these to BF in a format 0 XID to identify the node. When a T2.0 node connects to the network these fields are used as an index to find the network definition of the node.

In a T2.1 node these fields are passed in the XID3 and may be used for node identification if the user decides not to use the CPNAME as identifier.

BIND segmentation information

The Whole-BIND-PIUs Generated indicator bit is used to indicate whether BIND PIU segments can be generated by the sending node.

---

[50] When a secondary LU has a LOGAPPL= operand specified to VTAM, the application specified in that LOGAPPL operand is called a controlling LU. The secondary LU is considered to be controlled by the application.

With the additional features now supported within subarea and APPN networks, the BIND command may become quite long.[51] Historically, most T2.0 nodes have assumed in their code that a BIND will fit in a single receive buffer and hence have user appropriate addressing techniques to access the data. These techniques assume that the BIND information is contiguous in storage. In addition, some receiving nodes may have small buffer sizes and be unable to process long data blocks. Segmentation of BINDs is intended to allow the sending of long BINDs to such devices.

ACTPU suppression indicator

This is sent by a L.E.N. node to BF to indicate that it does not contain any dependent LUs and therefore does not need an SSCP-to-PU session.

Link Station characteristics

This has a different meaning and effect depending on the type of link being used for the connection.

For all connection types it is used to resolve the question of which end is primary and which is secondary. This link station role is used by the algorithm for allocating the local form session identifier (LFSID). The link station role determines the setting of the ODAI bit for sessions established from a given link station.

For SDLC connection it determines whether modulo 8 or modulo 128 is used for the SDLC sequence numbers.

For X.25 it specifies which form of LLC (Link Level Control) support is to be used (QLLC or ELLC). BF X.25 connection supports QLLC only.

SDLC window size (meaning the maximum number of I frames that will be received before sending an acknowledgement) is also communicated as a link station characteristic.

Network ID

The network ID (NETID) of the network to which the sending T2.1 node belongs is exchanged. Current releases of VTAM and NCP make no use of this information.

Product Set Identification

This vector contains the following information used for network management purposes:

- For hardware: machine type, machine model number, plant of manufacture and machine serial number.
- For software: Program name, Ver/Rel, link edit date and time and LOADLIB name.

This information can be used to determine the nature of the attaching T2.1 node Although VTAM makes no distinctions between types of T2.1 node implementations at the present time it offers a potential to allow VTAM to accommodate differences if needed in the future.

CP Name

This name (if present) is used by VTAM in place of the IDBLK and IDNUM fields to identify the node when making a switched network connection.

Maximum PIU size

This is the maximum length of a PIU segment that the attaching node is able to receive. (Normally this is related to buffer size available in the node.)

---

[51] A non-extended BIND may be up to 256 bytes in length. An extended BIND is limited to a maximum length of 512 bytes.

For T2.0 nodes this information is specified to BF using the MAXDATA= parameter of the PU statement. In the case of T2.1 nodes BF adjusts its transmitted segment size based on the information received in the XID3.

Congestion Control

SNA networks are able to utilise resources in a highly efficient way. That is, SNA networks are able to operate at very high resource utilisations. (In this context resources are 37xx controllers, links and processors containing SSCP code.)

The reason SNA networks are so resource-efficient is that they contain many very effective flow and congestion control mechanisms. In the subarea network the main mechanisms are:

Session Pacing

Session pacing is used to control the flow of data on a single session. It may be performed in stages.

Route Pacing

When a large number of sessions pass through the same node ordinary pacing control would require that a minimum number of buffers be (potentially) available for each session. In the case of 1000 sessions for example, at two buffers per session, a potential 2000 buffers would need to be available. Most implementers would allocate a smaller number based on probabilities but this means that pacing will have very little effect as a flow control mechanism.

Within the subarea network many sessions are carried on a smaller number of "routes" through the network. One of the reasons for having routes (VRs) within the subarea network is to enable flow control to be performed on a route basis rather than on a session basis. This dramatically reduces the buffer requirements within intermediate nodes and makes flow control very effective.

Link Pacing

This is a way of using the SDLC Receive Not Ready (RNR) command to stop input to a node in extreme cases of flooding. Since it does not distinguish between control traffic and data traffic (you need to continue to process control traffic even during congestion) this is not a preferred method of flow control.

Slowdown mode

Some products (such as 37xx NCP) contain a slowdown mechanism which allows for recovery of the resource **without** the loss of data or the ending of sessions. This mechanism is called "slowdown mode." It operates by shutting down non-essential functions in a last-ditch attempt to keep the resource operational in the case of extreme congestion.

Within L.E.N. and APPN networks the two main means of flow control are:

Adaptive Session Pacing

Adaptive session pacing is similar to session pacing but it is dynamic in operation. This gives the benefit of reducing the amount of network definition required but also solves many of the problems created by having a large number of sessions passing through a single node.

Adaptive BIND Pacing

In the absence of route pacing (there are no routes as such in APPN) and SSCP control of session setup requests a method of controlling network flooding caused by session setup traffic at node start time is needed.

BF T2.1 node support forms the boundary between the two kinds of SNA networking system and thus must accommodate the transition from one set of flow control mechanisms to the other for traffic crossing the boundary.

Adaptive Session Pacing

Session pacing is a means of controlling flow on a session. It may be used to restrict flow when the demands of the session exceed the resource capabilities. Session pacing may also be used to help prevent specific sessions from monopolizing resources (lines, VRs, buffers, etc.) which are shared among many sessions.

Pacing is implemented by means of windows. A PIU transmitter has a window size, which is the number of normal flow request RUs that may be sent before receiving a pacing response. If a pacing response is received before the entire window has been sent, the transmitter may send the next window or RU after completing the current window; otherwise, the transmitter must wait until the pacing response is received.

The previous method of session level pacing uses a fixed window size which is determined at BIND time. A T2.1 node, or the associated NCP, may have relatively limited buffer resources but still service a large amount of data. In order to help prevent flooding of the NCP and ensure efficient use of NCP storage, these nodes require a more controlled session pacing mechanism, a mechanism which varies the window sizes on a dynamic basis. Adaptive Session Pacing is such a mechanism. It allows a session pacing stage end-point to determine the transmit window size dynamically; the window size included in the pacing response is also sent to the session partner pacing stage end-point.

Whenever VTAM sends an extended BIND on behalf of an application program, Adaptive Session Pacing is used for the session. Identical logic applies to the NCP's boundary function. If a PN attached to the NCP does not support adaptive session pacing, NCP uses adaptive session within the subarea network and applies fixed pacing to the PN.

VTAM does not vary its own pacing window size dynamically. It uses the same receive window sizes for adaptive session pacing that it used for fixed pacing to the PN.

The VR and REX Stages

Session pacing is done in stages. A session path is broken into one or more sections, each of which is called a stage. The end-points of the session path and the points dividing the stages are called pacing end-points. Each end-point has a manager for controlling the pacing on its stage(s).

The two directions of flow operate completely independently, in NCP V4 R3, the NCP BF is a pacing end-point in both directions. The stages, of which the BF is an end-point called the REX stage (route extension stage) and the VR stage (virtual route stage). The REX stage extends from the BF onto the SDLC boundary link to the adjacent peripheral node. The VR stage extends from the BF onto the VR and ends at either LUS (LU Services) in VTAM or another NCP BF. PIUs received from the VR stage and forwarded onto the REX stage are directed outward. PIUs received from the REX stage and forwarded onto the VR are directed inward. A PIU is considered to have a direction only in the NCP BF; once a PIU has entered the subarea network (that is, flowing on a VR), indication of direction by inbound or outbound is meaningless. Pacing under NCP V4 R3 can be fixed or adaptive, the main difference between them is that the window size in adaptive session pacing is not fixed.

The Flow of Adaptive Session Pacing

The basic protocol for session pacing is little changed for Adaptive Session Pacing. The transmitter has a window. When the RPC (Residual Pace Count) reaches 0, and a PRR bit (pace response received) for the current window has not been received, then the transmitter is held and cannot transmit any more PIUs. The main change is that the window size is not fixed: rather, the receiver returns (in the pace response) the window size for the next window of PIUs. Thus, the mechanism for adaptive pacing operates as shown in the Figure 51 on page 110. NWS stands for next window size. When a session starts, the transmitter may send one PIU, so NWS = 1 and PRR = 1.

BIND Pacing

The session pacing discussed above applies to individual sessions and requires that a session be established before it comes into effect. Route pacing within the subarea network requires that a route be established before it can take effect.

When a new node is added to the network (or activated) it is quite possible that a very large number of sessions will be requested immediately by that node. In the subarea network, VTAM has ways of controlling the potential problem of flooding the network with this type of traffic. In any case, in the subarea network session control requests (INITSELF and LOGON etc.) and BINDs are sent on VRs which have a pacing flow control of their own.

In APPN however, there are no VRs and no SSCP-LU sessions. This means that a new APPN node (or L.E.N. node) becoming active could indeed flood its network partners with session setup (BIND) traffic.

The problem is solved in L.E.N. and APPN and now on the T2.1 node BF interface by a mechanism known as Adaptive BIND pacing.[52] Adaptive BIND pacing is similar to adaptive session pacing as described above with the following differences:

- Adaptive session pacing applies to session messages (which does not include BIND), while adaptive BIND pacing applies only to BINDs.
- Session pacing is performed separately for each session over a given link where adaptive BIND pacing applies to the whole box-to-box link.

Since adaptive BIND pacing is a part of the T2.1 node definition, it is also performed by the BF when connecting to a T2.1 node.

Segmentation

The Boundary Function (BF) splits the session path. On one side the BF interfaces with the VR; on the other side, the BF interfaces with the boundary link (referred to as the REX or route extension) on the REX link. PIUs received from the VR and transmitted on the REX are directed outward. PIUs received from the REX and transmitted onto the VR are directed inward.

[52] A good discussion of Adaptive BIND Pacing may be found in *Systems Network Architecture Type 2.1 Node Reference, SC30-3422.*

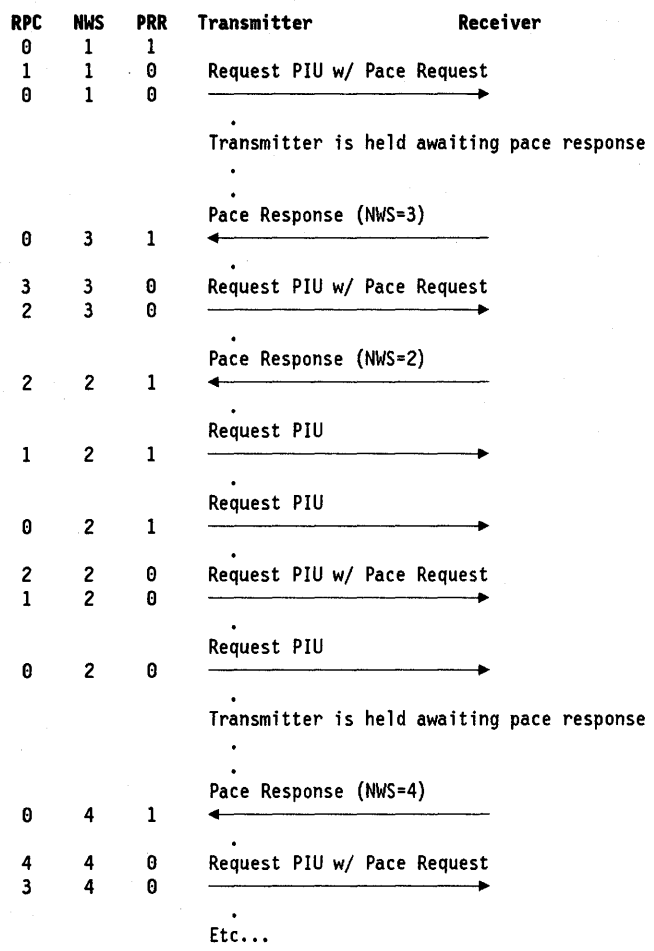


Figure 51. Adaptive Session Pacing Example

Previous Segmentation Support

- BF does not accept segmented PIUs from the VR. Any segmented PIUs received are rejected.
- BF may receive the segmented PIUs from the REX, which are not reassembled or checked for segment integrity; they are simply forwarded onto the VR. BF assumes that session control PIUs are not segmented. In some cases, this assumption is enforced by other NCP or VTAM code which checks the PIU prior to the BF processing.
- BF segments PIU transmitted onto the REX when necessary.

New Segmentation Support

PLUs are supported by the Boundary Function. The T2.1 node BF for a PLU may now receive segments from the VR, since the other end of the VR may be another T2.1 node BF which is sending segments. Also, the BF may not be able to send segments onto the VR, since the VR may end at a T2.1 node BF which does not accept segments.

The WBRI (Whole BIUs Required Indicator) is a bit defined in BIND and RSP(BIND), which are either extended or LU 6.2 (or both). When a node does not want to receive PIU segments from an adjacent node for a particular session, it sets the WBRI to 1 in the BIND sent to the adjacent node. When a BIND or RSP(BIND) is received from an adjacent node in which the WBRI is not defined, it is generally assumed that segments may be sent to that adjacent node. There is one exception: BF never accepts segments received from a VR and destined for an SLU unless the RSP(BIND) sent onto the VR was extended with the WBRI set to 0.

Although PIU segments may be generally transmitted onto the REX links, there must be some considerations as to the order in which segments from different sessions are transmitted. A PU may contain one of two types of re-assemblers: a station re-assembler or a session re-assembler. The following segmenting functions are supported:

- Receipt of segments from the VR destined for a PLU:

  Segments received from a VR destined for a peripheral PLU are accepted by BF if the BIND received by the NCP from the PLU has an WRBI equal to 0. If segments are not accepted from the VR and the WBRI is defined in the BIND sent onto the VR, BF sets the WRBI equal to 1 in the BIND sent onto the VR.

- Receipt of segments from the REX originating from an SLU:

  Segmented PIUs are accepted from the REX originating from SLUs unless the BIND received from the VR has the WBRI defined and set to 1.

- Receipt of segments from the VR destined for an SLU:

  Segments received from the VR destined for an SLU are accepted by BF if the adjacent node is a T2.1 node format and, in the RSP(BIND) received by BF from the SLU, the WRBI is defined and is set to 0. In this case, BF forwards the RSP(BIND) onto the VR with the WRBI equal to 0, indicating that segments are accepted.

- Receipt of segments from the REX originating from a PLU:

  For PLUs, PIU segments are only accepted from the REX when the RSP(BIND) received from the VR is extended and the WBRI in the RSP(BIND) is 0. If the SLU is connected to a NCP V4 R3, this occurs when the BF(SLU) accepts segments from the VR as described immediately above.

- SLU Maximum RU send size negotiation:

  If a BIND is received from a peripheral PLU with WBRI = 1, then BF will not send segments to the PLU. If the SLU maximum RU size in the BIND is greater than the segment size for the PLU's PU, then either (1) (if the BIND is negotiable), the SLU maximum RU send size is negotiated downward, or (2) (if the BIND is not negotiable), the session activation fails with an UNBIND type '0F', with an extended sense code of '0835000A'.

Transmission Priority and Route Selection

Transmission priority provides a mechanism for selecting on a per session basis the priority at which all data blocks on a session are to be transmitted. Higher priority messages (perhaps on interactive sessions) are transmitted before lower priority messages (perhaps batch traffic). Routes through the network may be selected according to specific desired characteristics. SNA subarea networks and SNA APPN/L.E.N. networks use different ways of selecting routes and of setting transmission priorities through the network.

Route Selection

In the APPN network routes are determined dynamically at the originating network node from an internal network map kept within the node. Each session is given an individual route (appended to the BIND at session setup) based on requested session characteristics.

As the Network Nodes (NNs) internal network map is constantly undergoing update, sessions can be started over many disparate paths through the network depending on the state of the network map at the time the session was started. Once a session is given a route that route never changes until the session is terminated and then perhaps started again on another route.

Note: If the session is terminated abnormally, (because of a link failure for example), the session is disrupted and data may be lost. When the session is started again, over a new route, it is up to the LUs to recover from any data loss that may have occurred.

As a session is started, a BIND containing routing information (in the form of a sequence of node/link pairs) and a priority indication (low, medium or high) is sent from node to node over the planned session route. The NNs along the route remember the priority information for this session and later this information is used to queue session data in priority order for the outbound link.

In the Subarea Network routes are statically defined when the network itself is defined.

When a session is set up between a S/370 host application and a secondary dependent LU within the network, a MODE name is associated with the session setup request. This association is made in a number of alternative ways depending on how the session is requested.

The MODE name is used to select an appropriate entry in a LOGMODE table **pointed to by the SLU definition**. If the primary and secondary LUs are in separate "domains" (ie. controlled by different VTAMs) then the VTAMs communicate with one another to find this entry from the secondary (destination) LU VTAM definition.

The LOGMODE entry contains a partial image of a BIND that may be used to setup a session with this SLU. The LOGMODE entry also contains a "Class of Service" (COS) name.

In the case of the PLU being a host application, this LOGMODE information is presented to the application in the LOGON exit and the host application (LU) has an opportunity to change it if required.

When the host application causes a BIND to be sent (by issuing the OPNDST macro), the BIND will contain the LOGMODE information (perhaps changed by the application).

The COS name that was associated with the SLU LOGMODE entry is used to search the COS table **in the domain of the primary LU** (That is, the VTAM owning the host application LU will search its COS table for a matching entry).

The selected COS table entry is an ordered list of Virtual Route Numbers (VRNs) and Transmission Priorities (TPFs).

VTAM will select an appropriate VR and TPF for the session.

Route Selection in the Subarea Network

When a BIND is received in the subarea network from a T2.1 node, route selection and priority determination is similar to that described above.

The MODE name is determined as follows:

- If the BIND is not an LU 6.2 BIND then it must be received in extended BIND format or it is an error. In this non-LU 6.2 extended BIND a vector containing the MODE name (CV X'2D') must be present. This entry is used as the MODE name.

  Note: The COS/TPF vector in the (received) extended BIND (CV X'2C') is ignored.

- If the BIND is an LU 6.2 BIND the MODE name is taken from the structured subfield X'02' (MODE name) *within the user area* of the BIND.

  Note: For LU 6.2 BINDs both the COS/TPF and MODE vectors of the (received) BIND (NS fields) are ignored.

For non-LU 6.2 BINDs, if the selected MODE name is unresolved by the SSCP owning the SLU then it defaults to the entry named "DLOGMOD" or if that is not present to the first entry of the MODE table. For LU 6.2 BINDs if the selected MODE name is unresolved by the SSCP owning the SLU then **the BIND is failed by VTAM**. (See "MODE Name in LU 6.2 BIND" on page 113.)

Selection of a MODE table entry and resolution of the COS name takes place as described in "Route Selection" on page 111. The COS information (VR/TPF list) is sent to the BF (that received the BIND from the T2.1 node PLU) and is used to select an appropriate route on which to establish the session.

The other BIND image information obtained from the MODE entry at the SLU is unused except:

1. The "Transmission Services" (TS) subfields may replace the TS subfields received in the BIND, before the BIND is sent on to the SLU.
2. The "parallel session requested" bit in the BIND (byte 42 bit 6) is checked against the "parallel sessions capable" bit specified in the PSERVIC entry in the MODETAB. If an incompatibility is found, the BIND is failed.

Note: There is no name translation of any kind performed at the T2.1 node interface. The MODE name is not translated. The name used is the one found in the BIND as described above.

The COS/TPF information determined from the MODE name as described above replaces the COS/TPF vector in the BIND before it is sent onward (placed on a VR) within the subarea network. This information is used within SNI gateways to select an appropriate onward routing in the attached network.

Route Selection in the APPN Network

When a subarea network tries to establish a session with an LU within an APPN network, the APPN network needs to select a route within it for the session to take. A conceptual description of how the AS/400 APPN network determines routes is given in "Route Selection" on page 149.

As with the subarea network, a MODE name is required when the session is started as a key to the route characteristics needed by the session.

Session setup is commenced when a BIND arrives from the subarea network (the APPN network sees this as coming from a L.E.N. node). The APPN NN to which the subarea network is connected finds its MODE name from the BIND. This is done in the same way as the subarea network does it:

- If the BIND is not an LU 6.2 BIND and it is addressed to an independent LU then it is an error.

  Note: AS/400 APPN networking supports only LU 6.2. Other LU types may exist within an AS/400 (LU type 2 for example) but these may only be dependent LUs. No APPN networking function is available for them. The AS/400 has a limited ability to "pipeline" (pass through) LU 2 sessions from downstream devices connected directly to the AS/400 having a connection to VTAM or NCP. However, this traffic cannot "network" through more than one AS/400.

- If the BIND is an LU 6.2 BIND the MODE name is taken from the structured subfield X'02' (MODE name) **within the user area** of the BIND.

  Note: For LU 6.2 BINDs both the COS/TPF and MODE vectors of the BIND (NS fields) are ignored.

The MODE name used by APPN is placed in the BIND by the LU 6.2 and not by VTAM (except when the LU is VTAMs LU 6.2 API). Therefore the manner of specification of this MODE name is different depending on the product implementation of the LU 6.2 BIND sender.

MODE Name in LU 6.2 BIND

In LU 6.2 the MODE name has a similar meaning to MODE name within the transmission network but it is used for additional purposes. When a transaction program requests a conversation, it cannot specify which session to use but rather it specifies the MODE name for the requested session.

Each mode name can be used by several sessions. These sessions form a group which can then be treated as a pool of sessions sharing the mode name characteristics.

When LU 6.2 requests a session the MODE name is placed in the user field of the BIND and is used in both the subarea network and in APPN networks to determine the route characteristics for the session.

In some situations when starting a session VTAM will take a default if the MODE name is incorrect or not present. This is not true for LU 6.2. It is a requirement for LU 6.2 support that the secondary LU MUST have a MODE name specification identical with that placed in the BIND by the PLU.

Note: The MODE name in LU 6.2 is carried in the user field of the BIND and is not translated by an SNI gateway.

Network Management

For T2.1 nodes connected individually to the subarea network backbone network management by NetView is essentially unchanged. As described below, session awareness was changed to provide for the new case of a session where both communicating LUs reside in the network and neither is in a VTAM application.

In the case of interconnected subarea networks ("casual connection") each network must be managed separately and independently from each other. NetView-to-NetView sessions through the T2.1 node gateway are not possible. Neither are VTAM-to-VTAM sessions. There is no PU-to-SSCP session possible from the T2.1 node gateway interface to either controlling SSCP. This is exactly what is needed in the casual connect situation. The T2.1 node connection is not intended to replace the normal cross domain function of SNA nor to perform the SNI role.

Network management for APPN networks connected to the subarea network is discussed in *Management of AS/400 in SNA Subarea Networks Using NetView Products.*

Session Awareness

Before the T2.1 node support, all sessions in an SNA subarea network terminated at one end in a host processor. This meant that network management tools that needed to refer to a session could trap the information as it passed through a host. As the session no longer needs to pass through a host[53] the older methods of trapping session information can no longer be used.

During session setup VTAM is notified about the session twice:

1. When the BIND arrives in the BF. (VTAM then needs to assist in session setup.)
2. When the positive response to the BIND is sent from the BF to the T2.1 node.

VTAM traps the BIND information at these points and passes it to the NetView Session Monitor as part of the the session awareness information.

The information passed to NetView Session Monitor is:

1. The BIND image sent by VTAM to the BF (PLU) when the BF (PLU) is establishing a session for a T2.1 node PLU. This is at the time that VTAM finishes its BIND processing and returns the BIND to the BF to be forwarded through the network.
2. The BIND image received by the BF (PLU) if one is received in a negotiable BIND response.
3. The BIND image received from the BF when the session is requested if session setup failure occurs.

Items 1 and 3 are sent to NetView when requested at the start of session awareness processing.

[53] In an NCP-based network session data does not need to transit a host at all and using the VTAM BF support the data does not transit the VTAM to application program interface

Security

Most of the usual techniques used to secure SNA networks apply unchanged to T2.1 nodes. Dependent LU operation is unaffected by the node type in which the LU resides. However, because of the absence of an LU-SSCP session, Independent LUs are somewhat different.

Independent LUs differ in their use of some encryption techniques and in the application of some access control techniques.

Encryption

There is no VTAM session cryptography key generation for sessions initiated by independent LUs acting as PLUs. These LUs must use other methods of cryptography key generation.

Access Control

In the past, the only[54] primary LUs in a network have been host-based (VTAM-controlled) LUs. Dependent LUs are secondary LUs and make access requests which are processed by VTAM before being presented to the application as a LOGON request in the LOGON exit.

When a LOGON request is presented to an application, that application has the opportunity to interact with the requesting LU to check if the user is entitled to have access to this application. Most applications require passwords before access is allowed for example.

In addition to the above password protection afforded when a remote LU is logged on to an application, many users apply other forms of access control. These access controls may use products such as NetView/Access and/or SAMON to present an application menu to the user and to *prevent* the user gaining access to (or even knowing about) applications for which that user is not authorised.

Access control applications typically operate by naming the security application as a "controlling application" for a particular SLU by specifying LOGAPPL= in the LU definition. Of itself, this specification will not usually be sufficient to prevent a user "breaking out" and entering a VTAM LOGON command. However, this latter event is easily prevented by specifying an interpret table or an interpret exit routine which is given control whenever the named LU attempts a LOGON and is able to force that LU back to its controlling application.

The above techniques still apply without change to dependent LUs situated within a T2.1 node. However, **they cannot operate in the way described for an independent LU** for several reasons.

1. Independent LUs may be primary LUs and, instead of requesting a session with another LU, they are able to send BIND.
2. Independent LUs may have a controlling application specified by LOGAPPL= but they can have multiple sessions and the controlling application need not know (cannot easily find out) about other sessions.

   VTAM processing of LOGAPPL= for independent SLUs is slightly different from that for PLUs as discussed in "Automatic Logon" on page 104.
3. The LU is "independent." There is no direct SSCP control and no SSCP-to-LU session.

Note: The VTAM Logon Manager discussed in "Example of 3270 Logon to a Type 2.1 Node (TPF)" on page 76 allows access requests and gives access control for dependent SLUs requesting LOGON to an independent PLU and is therefore not what is being discussed here.

---

[54] The trivial exception of PLUs residing within the NCP "applications" NRF, XI and NSI is irrelevant here and may be ignored.

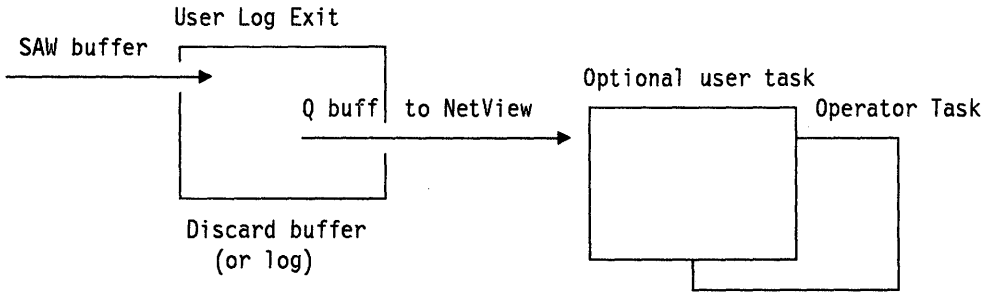
Access control of the type discussed above is not wholly relevant for independent PLUs since they are usually LU 6.2 and their sessions are "pipes" which many user transactions (conversations) may re-use at will. LU 6.2 has its own access security and control for end users.

However, it remains that an independent LU within an T2.1 node may send a BIND without surveillance by any controlling application. The following points should be considered:

1. When a BIND is received at an SLU, the PLU name (with network qualifier) is available and the SLU may refuse the BIND if the name is unacceptable.
2. The SLU may (as may any LU) enforce a security protocol on its session partner to determine the identity and authorisation of the user before allowing meaningful transaction processing to begin.
3. If the PLU is in one domain and the SLU is in another, the VTAM session management exit is driven and could be used by user coding to check for access authorisation.
4. Only LUs defined as present in this T2.1 node may send BINDs into the network. This is a useful first level of security for network to network (either subarea to subarea or subarea to APPN) connections. If a BIND arrives from an unknown LU then VTAM will refuse it immediately. Therefore on a casual connection, only LUs defined to VTAM as being present in this particular T2.1 node may participate in sessions in the subarea network.

   This characteristic was deliberately introduced for security. In the initial design, the BF was able to accept BINDs from any LU name and VTAM (in conjunction with the BF) would dynamically define it). It was thought that while this allowed ease of definition it also was too great a security risk.
5. If the session setup is in the same domain (controlled by the same VTAM) there is no available VTAM exit but a NetView technique is available which can accomplish the same thing.

A NetView Technique for Access Control. As stated above, when a session is started between two T2.1 nodes in the same VTAM domain, there is no exit available to check whether this LU has the right to start a session with the other. Most users will be satisfied to rely on standard logon procedures for this kind of security but some wish to have an additional level of security.: NetView can receive Session Awareness (SAW) data from VTAM whenever a session of this kind is started. The technique is simply to examine this SAW data to ensure that the session is authorised. If the session is not allowed then NetView can issue an appropriate VTAM operator command to terminate the session or (perhaps) deactivate the LU involved.



When NetView is started, definitions must be present to indicate that SAW data is required for all the LUs within the T2.1 node for which this form of access security is required.

The procedure is as follows:

1. When any of the named LUs starts (or ends) a session a SAW record is sent by VTAM to NetView.
2. NetView may then present the information to SMF to be recorded on the system SMF file or pass it to a user log exit routine.
3. In this case NetView will pass the SAW record to the user log exit routine.

It is difficult, in this routine, to access an external data base so the user log exit routine queues the data to a NetView "optional user task" or "operator task" using a user-defined message ID. Perhaps, in addition to refusing the session, the data could be sent to the operator console or to a security log file.

4. Either the optional user task or the operator task can be used to examine the origin and destination LU names perhaps with reference to an external data base. If the session is not authorised, the task can then take appropriate action to terminate the session.

Chapter 8. VTAM V3.3 Installation Planning

This chapter relates to the detailed specifications necessary to define T2.1 node support to VM VTAM V3 R3 for the IBM ES/9370 system. It is included to aid understanding of the functions but should not be considered a substitute for the official VTAM reference publications.

The functions introduced in VTAM V3 R3 (with the exception of the casual connect function) were introduced to MVS systems for VTAM V3 R2 and NCP V4 R3 and NCP V5 R2. Installation planning for the case of NCP operation is described in *VTAM V3 R2 and NCP V4 R3 Planning Guide for New Functions.*

In the following descriptions, all functions and operands that were introduced to VTAM for the purpose of supporting L.E.N. functions are discussed. Many of these operands were introduced in VTAM V3 R2 but are included here in order to facilitate understanding.

T2.1 nodes do not require any system generation process but are defined normally as T2.0 nodes. The main reasons for differences in VTAM definitions between T2.1 nodes and T2.0 nodes are the XID3 exchange and the presence of independent LUs within the T2.1 node.

VTAM Start Parameters

L.E.N. requires that all SSCPs supporting an attached Type 2.1 Node node have a NETID and SSCP Name. Prior to VTAM V3 R2, it only required that associated start parameter for gateway SSCPs. A new start parameter (GWSSCP) is defined so that NETID and SSCP Name may be specified for SSCPs that are not gateway SSCPs. The new and changed parameters are as follows.

NETID = cccccccc This is an existing start parameter which specifies the symbolic NETID of the network containing the SSCP. This was previously optional, and defaulted to a blank NETID. It is now a required start parameter.

XNETALS = YES XNETALS = YES indicates that this VTAM permits LUs which reside within an adjacent network to establish sessions through the use of the boundary function.

This operand enables the non-native network connection feature described in "Non-Native Network Connection (NNNC)" on page 103.

SSCPNAME = cccccccc This is an existing start parameter which specifies the symbolic name of SSCP. Previously, it was specified only if the SSCP was a gateway SSCP, and it was required then. It is now a required start parameter.

GWSSCP = YES|NO GWSSCP = YES is the default; it indicates that the SSCP is a gateway SSCP. GWSSCP = NO indicates that the SSCP is not a gateway SSCP, even though NETID is specified.

Note: The parameter is valid for all supported operating systems except VSE. VSE/VTAM is not gateway-capable.

procname This is the procedure name for the command and is unchanged for this release.

poolname = (baseno,bufsize,slowpt,F,xpanno,xpanpt) describes a buffer pool for new VTAM control blocks used with this release.

Coding for these parameters is described in *VTAM Installation and Resource Definition.*

VTAM Resource Definitions

VTAM dynamically determines whether a PU is PU T2.0 or Type 2.1 Node during activation of the PU. When coding the PU statement for a PU T2.0 or a Type 2.1 Node, specify PUTYPE = 2, as at present.

A new parameter, XID, is added to the PU statement. If XID = YES (the default is XID = NO) an XID exchange is used during activation of the PU to determine whether it is PU T2.0 or Type 2.1 Node. Most IBM PU T2.0s tolerate the protocol initiated by XID = YES and this should be used for all new definitions to facilitate change in the future. If there are any non-switched PU T2.0 nodes which do not tolerate an XID exchange, they must be coded for as XID = NO. When XID = NO is coded, VTAM assumes the node is PU T2.0.

Note: The only IBM Type 2.0 devices that *cannot* accept an XID are the: 3271-11, 3271-12, 3275-11, 3275-12, 3614, 3624, 3710 and 3791. These are the only IBM devices for which the coding of XID = NO is mandatory.

The VTAM Data Link Control supports receipt of full-length XID (up to 256 bytes) from communication adapter-attached peripheral nodes. Previous releases of VTAM did not allow an XID greater than 60 bytes, which would prevent attachment of switched Type 2.1 Node nodes to the VTAM boundary function.

Independent LUs are defined by coding LOCADDR = 0 on the LU statement. LOCADDR is an existing required parameter for LU statements.

Note: There can be as many LUs defined with LOCADDR = 0 as there are independent LUs to be accessed through this interface. LOCADDR = 0 simply means that the definition of the local address is dynamic. Definition of multiple LU names with the same (0) local address does not cause a conflict.

Dependent LUs continue to be defined by coding LOCADDR with a non-zero value.

PU Statements for Switched SNA Devices

CPNAME = cccccccc CPNAME is a new parameter. It specifies the CP name of an L.E.N. level Type 2.1 Node node. Either CPNAME or IDBLK and IDNUM must be specified on a switched PU definition statement. (Both may be specified.)

A switched L.E.N. level Type 2.1 Node node provides its CP name to VTAM in the XID exchange during the connection sequence. If a CP name is provided by a Type 2.1 Node node, VTAM uses it to locate the corresponding switched PU statement. If a CP name is not provided by the Type 2.1 Node node, or if there is no PU statement with the corresponding CP name, VTAM uses existing logic to locate the PU statement with IDBLK and IDNUM.

NETID = xxxxxxxx If NETID is specified on the PU definition, VTAM will assure that when the PU is active, the connecting resource is within the network specified by the NETID parameter. If the parameter is omitted, VTAM will dynamically learn the network ID during connection establishment.

If dynamic dial-out connections are required (for example if a session request initiates the DIAL), the PU network ID must be predefined.

IDBLK = xxx This operand is the identification block and is optional if the CPNAME parameter is specified.

IDNUM = xxxxx This operand is the identification number and is optional if the CPNAME parameter is specified.

LU Statements for Switched SNA Devices

The changes for the switched LU definition are identical to those for the VTAM LU definition statement. See "LU Statement" on page 122.

GROUP Statement

The following keywords are changed in the GROUP statement.

MAXLU = count MAXLU is ignored because it is no longer needed by VTAM. It is retained only for compatibility. In future releases it will cause an error.

The following keyword is added to the GROUP statement.

XMITDLY = interval|none Specifies the amount of time that VTAM delays its initial transmission after answering an incoming call. This delay is provided to allow the calling station to transmit first, and to allow the modems to complete any required equalization in the inbound direction, prior to the first VTAM transmission.

Specify the delay either as an integral number of seconds or in tenths of a second. If no delay is desired, specify XMITDLY = NONE.

The initial transmission delay is only used for incoming calls. Outgoing calls do not need an initial delay.

This keyword is valid only if DIAL = YES and LNCTL = SDLC are specified in this GROUP statement. If omitted the default is 2 seconds. The range is the same as that of the REPLYTO keyword.

LINE Statement

The following keywords are changed on the LINE statement.

MAXLU = count MAXLU is ignored because it is no longer needed by VTAM. It is retained only for compatibility. In future releases it will cause an error.

MODE = PRI|SEC (non-switched)

PRI This is the default. VTAM will be the primary for control of the link. This is required for all "leased line" type connections. This is exactly the way current T2.0 node support works.

SEC For "leased connection" (for example as a secondary on a multidrop link or as secondary on a point-to-point link) this operand specifies a fixed secondary role.

For switched connection (SDLC or Token-Ring) this operand specifies that link role negotiation should be performed. The primary/secondary role is then determined by negotiation at link setup time.

PU Statement

The following keywords are changed on the PU statement:

MAXLU = count MAXLU is ignored because it is no longer needed by VTAM. It is retained only for compatibility. In future releases it will cause an error.

MAXDATA = size Type 2.1 Node stations report the maximum Basic Transmission Unit (BTU) length they can receive in their XID 3 format. This value is equivalent to MAXDATA = size and is used in place of MAXDATA when NCP communicates with Type 2.1 Node stations.

MAXOUT = n Type 2.1 Node stations report the maximum number of I-frames they can receive before sending an acknowledgement in format 3 XID. This maximum number of I-frames value is equivalent to MAXOUT = n and is used in place of MAXOUT when VTAM is communicating with Type 2.1 Node stations. These values are in the XID VTAM sends to Type 2.1 Node stations, reflecting VTAM's capabilities.

PUTYPE = 1|2(1,2)|4 The description of this keyword is changed. The meaning of PUTYPE = 2 is changed to indicate that the generated PU is either PU T2.0 or Type 2.1 Node node.

The following keywords have been added to the PU statement:

XID = NO|YES Defines a PU's ability to receive and respond to an XID while in Normal Disconnected Mode (NDM). A YES value is suggested for all PUs that can process XID in NDM. (YES is mandatory if Type 2.1 Node nodes are to provide Type 2.1 Node function. If NO is specified, an XID is never sent to the PU.)

XID = NO is invalid for switched lines (DIAL = YES). **The default is NO.**

This keyword controls the use of XID during initial CONTACT processing. Its purpose is migration. There are devices designed for non-switched line operation that cannot accept an XID, since XID is defined as an optional function on top of the basic SDLC repertoire.

LU Statement

The following keyword is added to the LU statement.

EAS = n|10 EAS is the estimated number of active sessions with this LU. This operand has meaning only for Independent LUs (LOCADDR = 0).

It is used for performance tuning the size of VTAM "hashing" tables.

The following keywords are changed in the LU statement.

LOCADDR = n The valid range of the LOCADDR is changed from 1-255 to 0-255 if PUTYPE = 2 on PU statement. LOCADDR = 0 is a part of the definition of independent LU.

One LU macro with LOCADDR = 0 is necessary for every independent LU to be accessed through this connection. Multiple LU macros with LOCADDR = 0 may of course be specified for any T2.1 node definition.

Note: Unlike the definition for dependent LUs, it is essential that LUs defined with LOCADDR = 0 have the LU name defined in the attaching T2.1 node with exactly the same name as appears on the LU statement. This is the key to how VTAM identifies which LU is being referred to. With dependent LUs a LOCADDR≠0 is sufficient to identify the LU but for independent LUs this is not available because the LU name is used instead.

Defining a Type 2.1 Node Connection

SDLC Leased Line

This is defined as for T2.0 node connection with changes as described above.

The MODE= parameter on the LINE statement specifies whether the link is primary or secondary. The relationship is fixed and not changeable. MODE=PRI is the default.

On the PU statement XID works as described. Most T2.0 nodes will accept (but not react to) an XID3 and when that is unsuccessful, VTAM will retry using the SDLC SNRM command.

The XMITDLY= parameter is normally used for switched line connections only. However, it may be specified for leased line connection if modem equalisation is required.

SDLC Switched Line

SDLC switched lines are defined as for T2.0 nodes before with modifications as described for SDLC leased line connection above.

The XMITDLY= parameter must be specified.

X.25

X.25 connection specification is as before with modifications as described for SDLC connection above except that the MODE operand is not required for "leased line" PVCs under packet major node and XMITDLY is not valid for X.25. Switched virtual circuit connection is the same as SDLC switched connection and Permanent virtual circuit is the same as SDLC leased connection.

Token-Ring

Token-Ring specification changes are exactly as for switched SDLC connection described above except that the XMITDLY operand is not valid.

Channel Connection

Channel-attached T2.1 nodes (NCPs acting as T2.1 nodes) are defined to VTAM in a similar way to the current T2.0 node 3174 definition.

In the PU statement, XID=YES is coded to indicate the necessary change in the channel contact procedure. The default is for current T2.0 node operation.

The LU statement is changed to add independent LUs with LOCADDR=0 and the EAS parameter for tuning.

Definition Examples

The following examples are intended as an aid to understanding for those people who are familiar with VTAM definition. They should *not* be used as models for coding since they are based on early information.

Subarea Network to Subarea Network "Casual" Connection

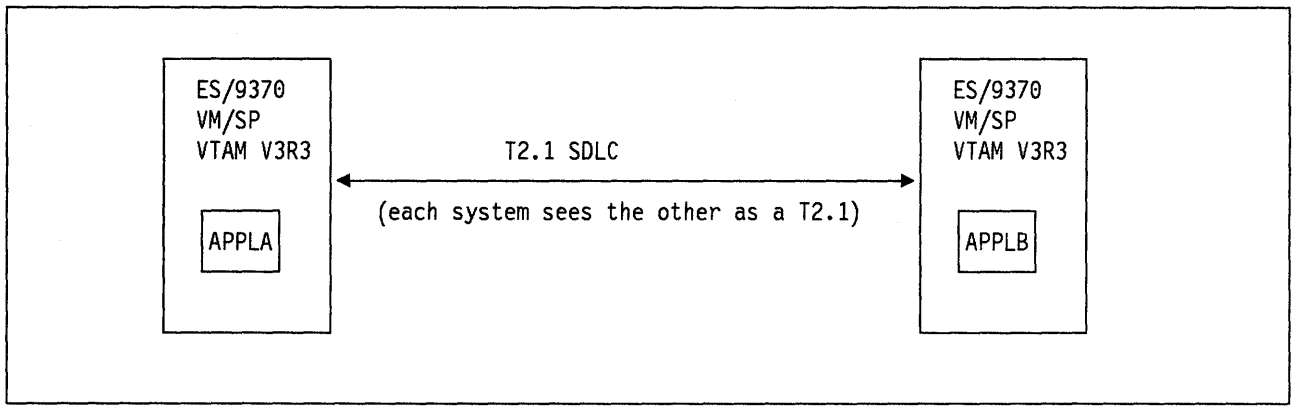


Figure 52. Appl-to-Appl "Casual" Subarea Network Interconnection

System Definitions

Definitions in Host A:

```
APPLA     VBUILD  TYPE=APPL
            .
            .
            .
SDLCATT   VBUILD  TYPE=CA
GROUP03   GROUP   LNCTL=SDLC,DIAL=NO
LINE3     LINE    ADDRESS=033,MODE=PRI
PU#B      PU      ADDR=A1,PUTYPE=2,XID=YES
APPLB     LU      LOCADDR=0
```

Definitions in Host B:

```
APPLB     VBUILD  TYPE=APPL
            .
            .
            .
SDLCATT   VBUILD  TYPE=CA
GROUP03   GROUP   LNCTL=SDLC,DIAL=NO
LINE3     LINE    ADDRESS=033,MODE=SEC
PU#A      PU      ADDR=A1,PUTYPE=2,XID=YES
APPLA     LU      LOCADDR=0
```

TRN Connected T2.1 Node Independent LU

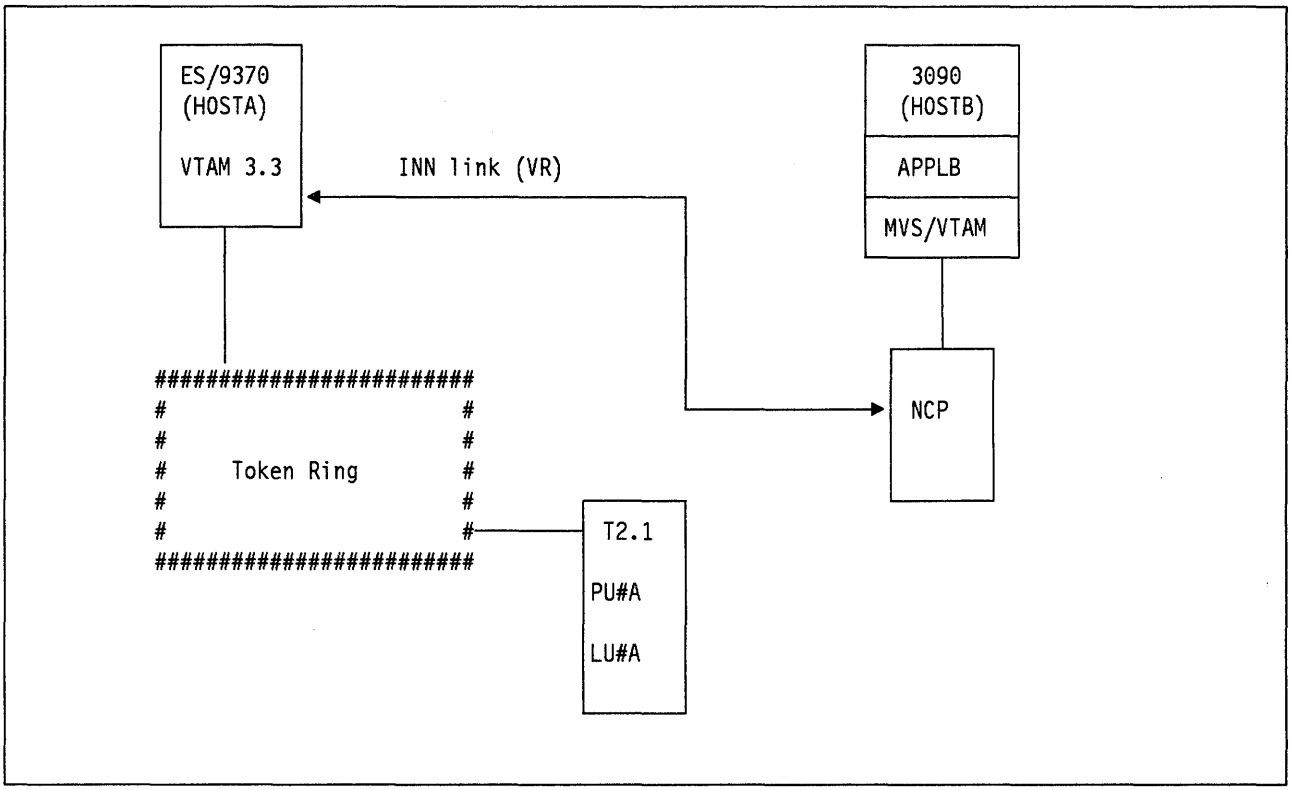


Figure 53. TRN T2.1 node LU in Session with a Cross-Domain Application

Figure 53 shows an ES/9370 and a T2.1 node connected to a Token-Ring LAN. An LU within the T2.1 node can have a cross-domain session with a host application in a different domain. The LU named LU#A is an independent LU (LOCADDR = 0) and is capable of multiple and parallel sessions with application APPLB as either the primary LU or the secondary LU or both.

System Definitions

Definitions in Host A:

```
LAN1      VBUILD  TYPE=LAN
PORT1     PORT    CUADDR=(040)
GRP1LAN   GROUP   DIAL=YES,LNCTL=SDLC
LN1LAN    LINE    CALL=INOUT
PU1LAN    PU

SWNODE    VBUILD  TYPE=SWNET
PU#A      PU      MACADDR=400000000013,CPNAME=PS2NAME
LANPATH1  PATH    GRPNM=GRP1LAN
LU#A      LU      LOCADDR=0

          VBUILD  TYPE=CDRSC
APPLB     CDRSC
```

Definitions in Host B:

```
APPLB     VBUILD  TYPE=APPL
            .
            .
            .

          VBUILD  TYPE=CDRSC
LU#A      CDRSC
```

Cross-Domain Connection

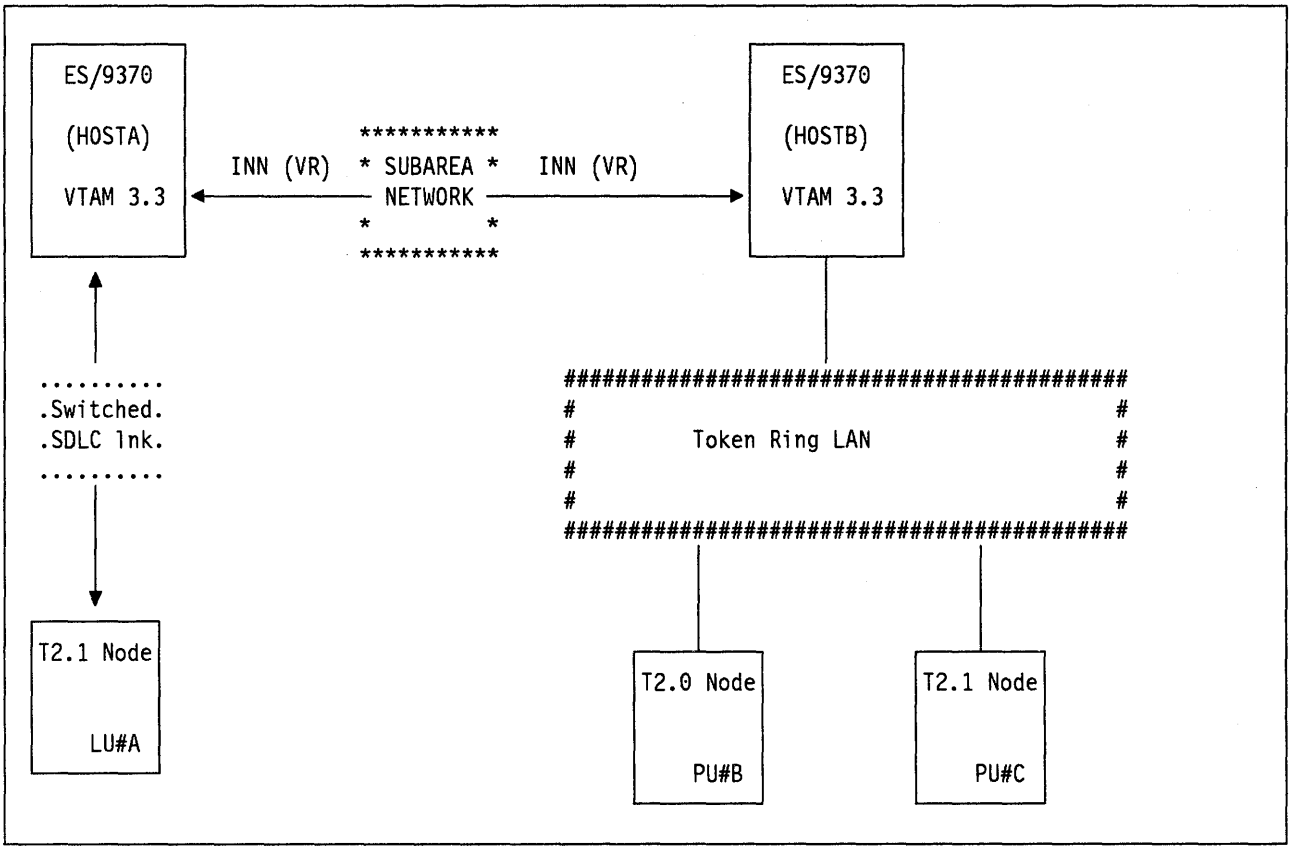


Figure 54. LU-to-LU Cross-Domain Connection

Figure 54 shows two ES/9370 systems with a T2.1 node and a T2.0 node attached to one ES/9370 communicating with a T2.1 node attached to the other ES/9370. One ES/9370 communicates with its T2.1 node via a switched SDLC/ICA T2.1 node attachment; the other communicates with its nodes via a token-ring attachment. The ES/9370's communicate with each other through an SNA subarea network connection (it could be direct or through other systems).

LU#A is an independent LU that is capable of multiple and parallel sessions. It can be in session with both LU#B and LU#C. LU#B is a dependent LU and LU#C is an independent LU. PU#B and PU#C are T2.0 and T2.1, respectively, not because of system definition but because of hardware capability determined during XID exchange.

System Definitions

Definitions in Host A:

```
SWCHATT   VBUILD  TYPE=CA
GROUP04   GROUP   LNCTL=SDLC,DIAL=YES
LINE4     LINE    ADDRESS=034,CALL=INOUT,XMITDLY=7
SWCHPU    PU

SWCHNET   VBUILD  TYPE=SWNET,MAXNO=25,MAXGRP=5
PU1       PU      ADDR=C1,IDBLK=012,IDNUM=00012,MAXPATH=1,MAXDATA=256,
                  PUTYPE=2
PATH4     PATH    DIALNO=45554,GID=1,PID=1,REDIAL=2,GRPNM=GROUP04
LU#A      LU      LOCADDR=0

          VBUILD  TYPE=CDRSC
LU#B      CDRSC
LU#C      CDRSC
```

Definitions in Host B:

```
LAN1      VBUILD  TYPE=LAN
PORT1     PORT    CUADDR=(040)
GRP1LAN   GROUP   DIAL=YES,LNCTL=SDLC
LN1LAN    LINE    CALL=INOUT
PU1LAN    PU
LN2LAN    LINE    CALL=INOUT
PU2LAN    PU

SWNODE    VBUILD  TYPE=SWNET
PU#B      PU      MACADDR=400000000013,CPNAME=NAMEB,XID=YES
LANPATHB  PATH    GRPNM=GRP1LAN
LU#B      LU      LOCADDR=1,PUTYPE=2
PU#C      PU      MACADDR=400000000014,CPNAME=NAMEC
LANPATHC  PATH    GRPNM=GRP1LAN
LU#C      LU      LOCADDR=0,PUTYPE=2

          VBUILD  TYPE=CDRSC
LU#A      CDRSC
```

Casual Connection APPL-to-LU through X.25 SVC

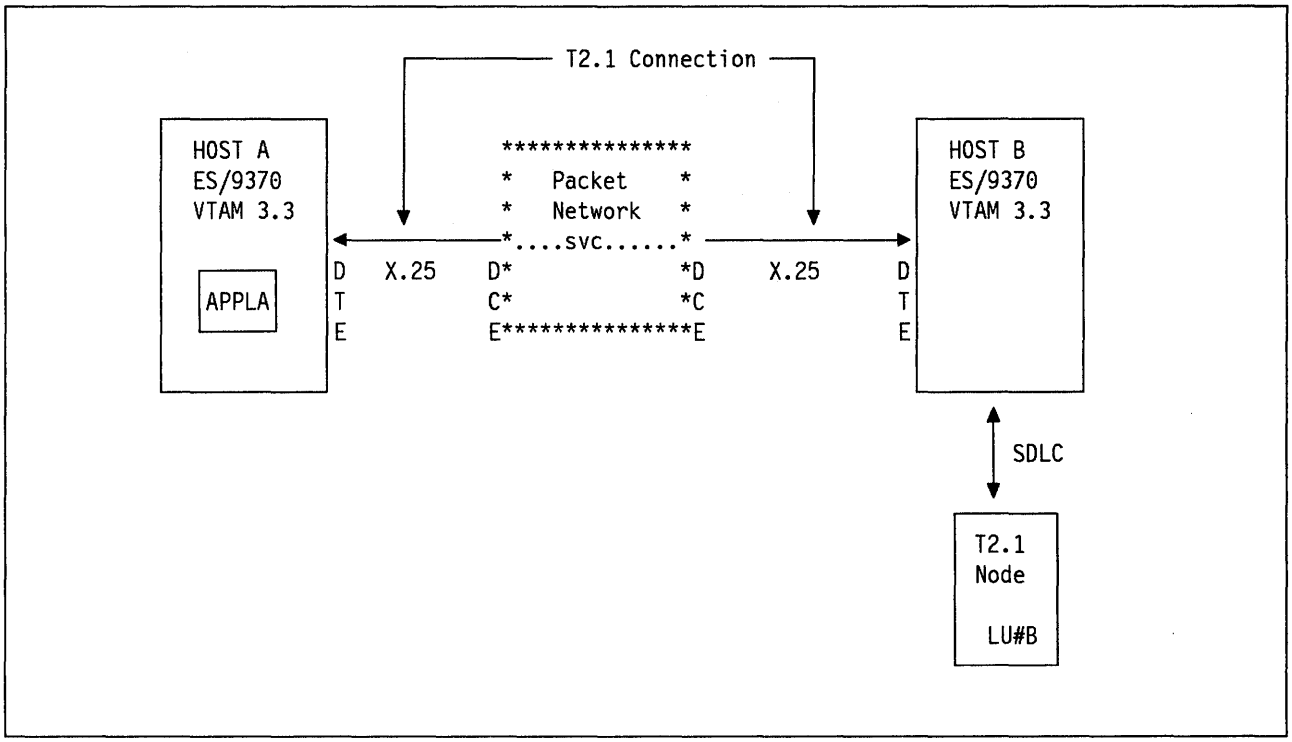


Figure 55. "Casual" Connection through X.25

Figure 55 shows two ES/9370s connected via an SVC in an X.25 network. One ES/9370 owns a T2.1 node that communicates with an application in the other ES/9370.

LU#B can request a session with APPLA by sending a BIND request to APPLA. APPLA can issue an OPNDEST ACQUIRE or a SIMLOGON to LU#B.

System Definitions

Definitions in Host A:

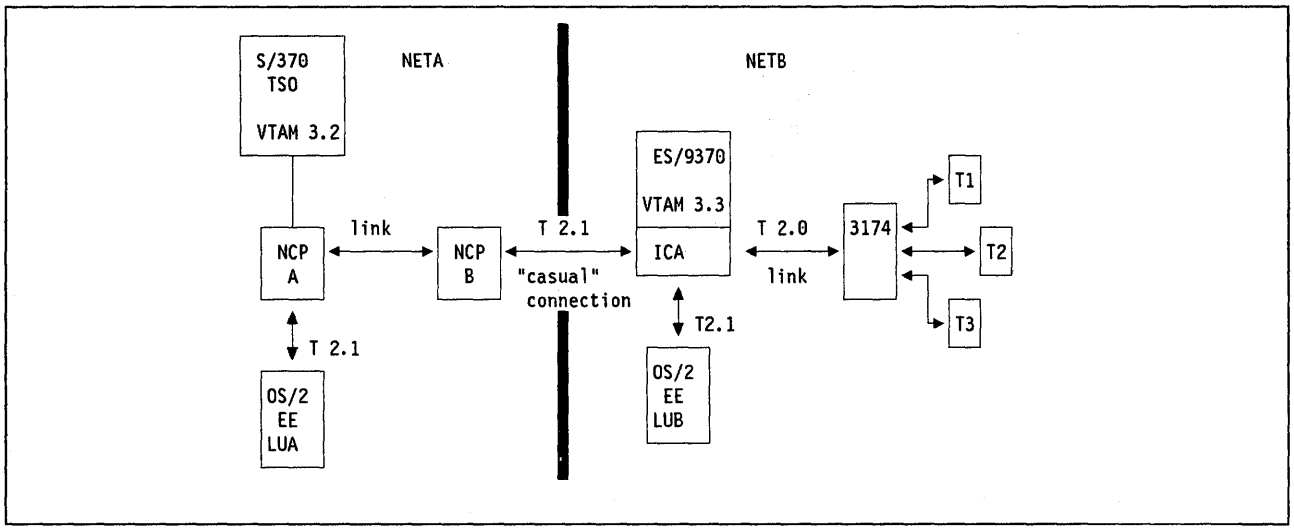
```
APPLA     VBUILD  TYPE=APPL
            .
            .
            .
PACKET04  VBUILD  TYPE=PACKET
PORT04    PORT    CUADDR=(031,039),NETTYPE=1,VCALLS=(,,004,007,,)
GRP04     GROUP   LNCTL=SDLC,DIAL=YES
LINE4     LINE    ADDRESS=004,CALL=INOUT
PU#D      PU

SWCHNET   VBUILD  TYPE=SWNET,MAXNO=25,MAXGRP=5
PU#B      PU      ADDR=C2,CPNAME=SSCPB,MAXPATH=1,MAXDATA=256,XID=YES   *
                  PUTYPE=2
PATH4     PATH    DIALNO=46016,GID=1,PID=1,REDIAL=2,GRPNM=GRP04
LU#B      LU      LOCADDR=0
```

Definitions in Host B:

```
PACKET04  VBUILD  TYPE=PACKET
PORT04    PORT    CUADDR=(031,039),NETTYPE=1,VCALLS=(,,004,007,,)
GRP04     GROUP   LNCTL=SDLC,DIAL=YES
LINE4     LINE    ADDRESS=004,CALL=INOUT
PU#E      PU

SWCHATT   VBUILD  TYPE=CA
GRP05     GROUP   LNCTL=SDLC,DIAL=YES
LINE5     LINE    ADDRESS=034,CALL=INOUT,XMITDLY=7
SWCHPU    PU

SWCHNET   VBUILD  TYPE=SWNET,MAXNO=25,MAXGRP=5
PU1       PU      ADDR=C2,IDBLK=012,IDNUM=00012,MAXDATA=256,MAXPATH=1   *
                  PUTYPE=2,XID=YES
PATH5     PATH    DIALNO=45554,GID=1,PID=1,REDIAL=2,GRPNM=GRP05
LU#B      LU      LOCADDR=0

PU#A      PU      ADDR=C2,CPNAME=SSCPA,MAXPATH=1,MAXDATA=256,PUTYPE=2,*
                  XID=YES
PATH4     PATH    DIALNO=46016,GID=1,PID=1,REDIAL=2,GRPNM=GRP04

APPLA     LU      LOCADDR=0
```

Planning for a "Casual" Network Connection

This subject was introduced in "Network Interconnection" on page 78. Casual connection is extremely easy to set up and define. Very little coordination between connecting networks is required. The connection does not have all the functions of MSNF or of SNI but that is not its purpose. The only things that require planning are LU name and MODE definitions for LUs involved in internetwork sessions.



In this figure the VTAM 3.2 processor containing TSO and the two NCPs (A and B) form a subarea network. The VTAM 3.3 processor with its attached 3174 forms another subarea network. Each network contains a PS/2 running OS/2 EE and has an independent LU 6.2. As shown, the two networks are connected by a T2.1 node casual connection.

Definitions are shown to allow the two PS/2s to communicate with each other using LU 6.2 parallel sessions. Definitions are shown to automatically log on terminal T1 in NETB to the TSO application in NETA. VTAM will build an extended BIND and send it.

For the LUA to LUB communication the following points are important:

- There must be a MODETAB and a DLOGMOD entry for the LU 6.2s as seen in the other network. Note however that the DLOGMOD is a dummy entry and MODEENT is located by using the LU 6.2 MODE specified by the LU 6.2 that starts the communication (sends the BIND). The MODE entry in the (LU 6.2) user data field of the BIND is used to search the specified MODETAB for a MODEENT. The example shows an incomplete MODEENT entry for an LU 6.2 MODE called "REALMOD."
- The LU 6.2 LOGMODE REALMOD will be included in the user data field of the BIND by the OS/2 EE communications manager. It has to be defined there at installation time of the application.
- Definitions in each network stand on their own, separately. When LUA sends a BIND to LUB then the MODETAB used in NETA will be the one specified in the definition for LUB in NETA. When the BIND arrives in NETB then the whole process is repeated and the MODETAB specified on the definition of LUB in NETB will be used for the NETB session initiation.
- Because the MODE name used is the one from the BIND, a MODEENT for the session must exist in BOTH networks.

For the TSO to 3270 session the following points are important:

- When the link with its associated resources between NETA and NETB becomes active, VTAM will drive TSO's LOGON exit and TSO will send a BIND (extended BIND) to T1.

Example T 2.1 Connection Definitions

Definitions in NETA:

```
TSO       VBUILD  TYPE=APPL
            .
            .
            .
*  Definitions for Casual Connection in NCP B
SDLCATT   VBUILD  TYPE=CA
GROUP03   GROUP   LNCTL=SDLC,DIAL=NO
LINE3     LINE    ADDRESS=033,MODE=PRI
PU#B      PU      ADDR=A1,PUTYPE=2,XID=YES
T1        LU      LOCADDR=0,MODETAB=MTGS3X,DLOGMOD=EM3277,            X
                  LOGAPPL=TSO
T2        LU      LOCADDR=0,MODETAB=MTGS3X,DLOGMOD=EM3277
T3        LU      LOCADDR=0,MODETAB=MTGS3X,DLOGMOD=EM3277
LUB       LU      LOCADDR=0,LOGMODE=MTGS3X,DLOGMOD=MODS361
            .
            .
            .
MTGS3X    MODETAB
            .
EM3277    MODEENT LOGMODE=EM3277,                                     X
                  FMPROF=X'03',TSPROF=X'03',                          X
                  PRIPROT=X'B1',SECPROT=X'90',                        X
                  COMPROT=X'3080',                                    X
                  RUSIZES=X'A8A8',                                    X
                  PSERVIC=X'02000000000000000000200'
            .
*-------------------------------------------------------------------
MODS361   MODEENT LOGMODE=MODS361
*-------------------------------------------------------------------
REALMOD   MODENT  LOGMODE=REALMOD  Entry reqd for LU 6.2 MODE
            .
          MODEEND
          END
```

Definitions in NETB:

```
            .
            .
            .
SDLCATT   VBUILD  TYPE=CA
GROUP03   GROUP   LNCTL=SDLC,DIAL=NO
LINE3     LINE    ADDRESS=033,MODE=PRI
PU#A      PU      ADDR=A1,PUTYPE=2,XID=YES
TSO       LU      LOCADDR=0
LUA       LU      LOCADDR=0,LOGMODE=MTGS3X,DLOGMOD=MODS361


      (MODETAB NOT SHOWN)
```

- If T1 is not active, the BIND will fail in NET2 and will not be retried by the system. An operator VARY LOGON command would then be necessary.
- VTAM will use the DLOGMOD in NETA defined for T1 (EM3277) to build its BIND.

- When the BIND arrives in NETB, the MODE name in the BIND will be used to retrieve a MODEENT from the MODETAB referenced in the NETB definition of T1. This information will be mostly ignored but the COS/TPF found will be used to determine routing within NETB. (Although COS/TPF is passed over the T2.1 node connection in the BIND it is replaced by a new one as soon as the BIND arrives in NETB.)
- The operator at T1 (or T2 or T3) has no way of requesting a session with the TSO application but the connection may be established as shown here.

Note: In each network, every LU in the other network which is to be involved in a session with any LU in this network **must be defined on the T2.1 node casual connection.** Other LUs in each network need not be defined on the casual connection if they are never to take part in sessions between networks. This applies regardless of which side initiates the session.

Appendix A. SNA Type 2.1 Node Overview

The Type 2.1 (abbreviated T2.1) node is a peer architecture for connecting small systems. IBM has announced SNA Low Entry Networking Architecture as an extension of the SNA T2.0 node that supports peer-to-peer communications. The architecture embodies the appropriate physical and session-level connectivity for support of LU 6.2. The T2.1 node will be a preferred implementation over the old T2.0 node, as well as the T1 node.

Link-Level Connectivity

The architecture of the T2.1 node is designed for *peer* attachment, as illustrated in Figure 56. (Note that the use of the term "peer" here refers to the communication protocols, not to the underlying product types.)

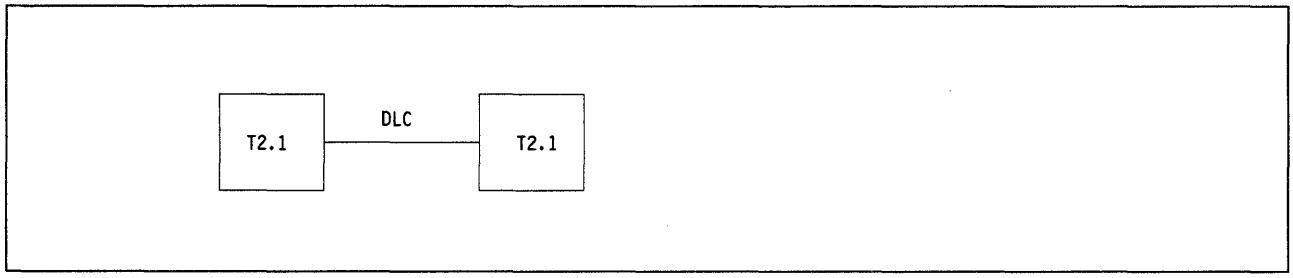


Figure 56. Peer Attachment of Type 2.1 Node

Note: A product that implements a Type 2.1 Node may also support attachment to subarea networks via boundary function nodes. In that case, the product appears as a T2.0 node as far as its boundary function attachment is concerned.

Data Link Controls

Two peer-coupled T2.1 nodes can be directly connected using a variety of supported DLCs, depending on particular product implementations. Supported DLCs include SDLC, IBM Token-Ring and X.25. The System/38, for example, supports SDLC and X.25, while the IBM Personal Computer and S/36 support SDLC and IBM Token-Ring.

Initial link-level contact between a T2.1 node node and a node of an unknown type uses the Null XID (that is, XID without an I-field), to initiate the exchange of information on partner node characteristics.

Between T2.1 node nodes, the Format 3 XID, or XID-3 (that is, XID with a Format 3 I-field), is used to exchange information on partner characteristics. This exchange occurs for both switched and non-switched connections.

---

[55] This appendix is abstracted from "An Introduction to Advanced Program-to-Program Communication (APPC)," (GG24-1584-1).

DLC Activation

With minor exceptions, T2.1 nodes that support SDLC can function as either primary or secondary stations. Some products, such as the Personal Computer and DisplayWriter, support a **role negotiation** protocol that allows certain aspects of their communication to be negotiated at initial link-level contact. This facility increases the flexibility of connecting T2.1 node nodes because it allows for dynamic determination of roles.

Figure 57 shows the overall activation flow for an SDLC connection (switched or non-switched) with a T2.1 as XID initiator.

Role negotiation is transacted using the I-field of the XID-3, with normal DLC protocols. The XID-3 contains information on the sender's characteristics, including link-station capability (Primary, Secondary, or Negotiable), node type, FID type supported, message size capability, and modulo (SDLC receive window) count. The receiver then transmits its characteristics in an XID response of similar format, according to certain negotiation rules discussed below.

There are two phases to the initial link-level contact procedure used by T2.1 nodes:

1. *Contact phase*, culminating in an initial successful XID exchange.

   During the contact phase it is possible for XID collisions to occur. This situation appears to the transmitting station as a timeout, and is handled by introducing a random delay before transmission of the XID. A new random value is created as long as timeouts recur, until eventually a successful exchange occurs.

2. *Negotiation phase*, resulting in the assignment of link station roles.

   a. *First-order* negotiation proceeds on the basis of the exchanged link-level protocol flags in the XID-3. It is successful unless both stations have specified the same value. The outcome is pictured in Figure 58.

   b. When both parties specify Negotiable roles, *second-order* negotiation proceeds by comparing a pair of **role negotiation values**. The field used for this purpose is the Node ID field of the XID-3. An iterative XID exchange is performed until the exchanged values of this field are unequal. At this point the node with the logically greater Node ID field (unsigned binary comparison) becomes the primary station.

      If both parties initially supply a unique Node ID, the first XID exchange will be conclusive. However, some nodes do not supply a unique Node ID. In that case, two Node ID subfields, **block number** (12 bits) and **ID number** (20 bits), are used to complete the negotiation. A value of either X'000' or X'FFF' in the block number subfield indicates that the Node ID does not contain a unique node-specific identifier, and that unique role negotiation values may be generated by storing random numbers in the ID number subfield. The node then generates new role negotiation values until the value sent does not equal the value received in the XID exchange. At that point, the value is "frozen" for the duration of the negotiation process. No more than two additional XID exchanges occur for the resolution of **block number** collisions, after that the XID fails.

   c. A final XID exchange takes place, with the role of XID sender set to the appropriate non-Negotiable value. This final exchange ensures that both nodes are in synchronism. The original (non-randomized) Node ID values may be sent on this occasion.

Note: The outcome of the role negotiation also determines the ODAI setting (see "Transmission Header Usage" on page 139.)

A possible negotiation sequence is shown in Figure 59.

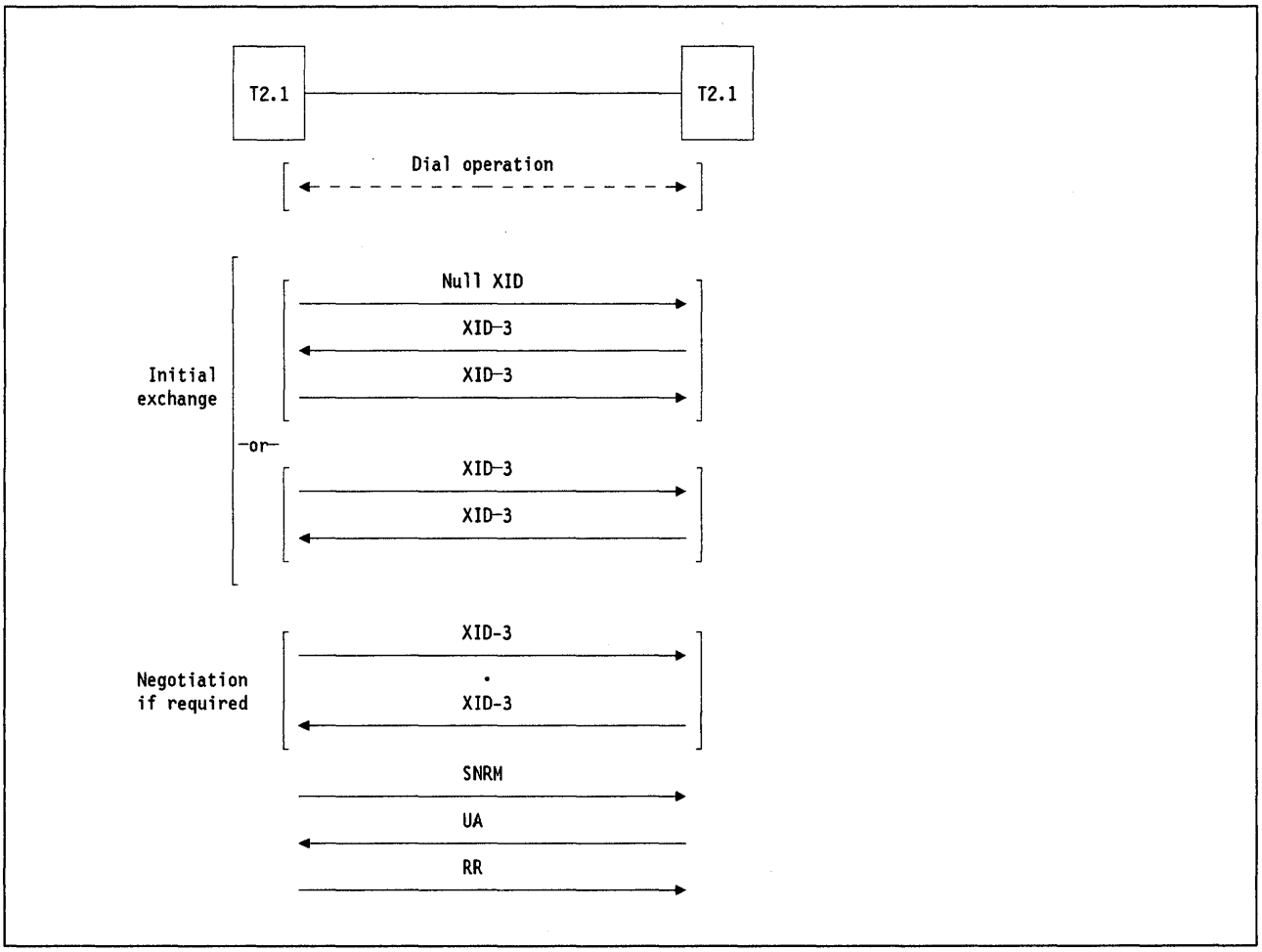


Figure 57. SDLC Activation Sequence for a T2.1-T2.1 Connection

Multiple Attachments

T2.1 node nodes may support multiple link attachments concurrently. One way of providing this support is illustrated in Figure 60, showing the case of a T2.1 node node with two link connections. The System/36 and System/38 are examples of T2.1 node nodes supporting multiple links.

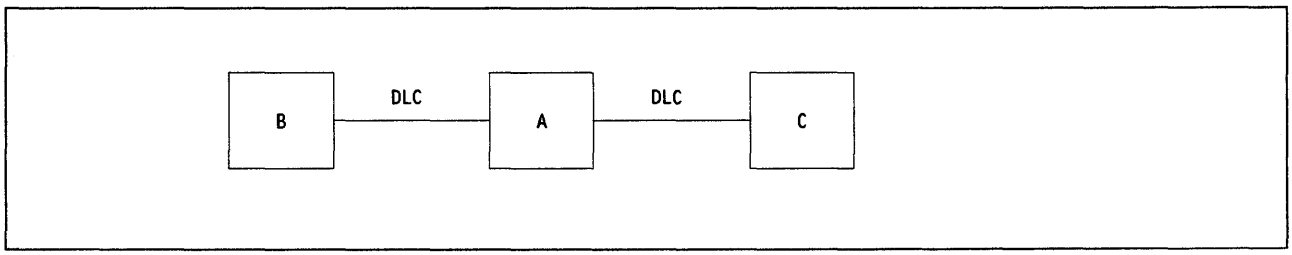


Figure 60. A Multiple-Link Type 2.1 Node.. In this example, node A has two link connections, allowing it to communicate in peer fashion with both B and C concurrently.

Note: Multiple links between two T2.1 nodes (*parallel* links) do not function like transmission groups in subarea networks, where the traffic for a given session may be split across different links. Rather, at session activation, the T2.1 node assigns the session to a specific link based on the combination of partner LU and mode name. This link, of course, may multiplex a number of sessions.

Another way of providing multiple attachments is through a multipoint SDLC link, as illustrated in Figure 61. VTAM, NCP and AS/400 all provide primary multipoint support.

SDLC link-station role sent	SDLC link-station role received		
	Primary	Secondary	Negotiable
Primary	Error (*)	PRIMARY	PRIMARY
Secondary	SECONDARY	System definition error	SECONDARY
Negotiable	SECONDARY	PRIMARY	Second-order negotiation follows

(*) A further XID is sent containing Control Vector X'22' (XID Negotiation Error)

Figure 58. Role Negotiation. First-order negotiation is attempted using link-station roles. The off-diagonal entries indicate the role assumed by the sending station.

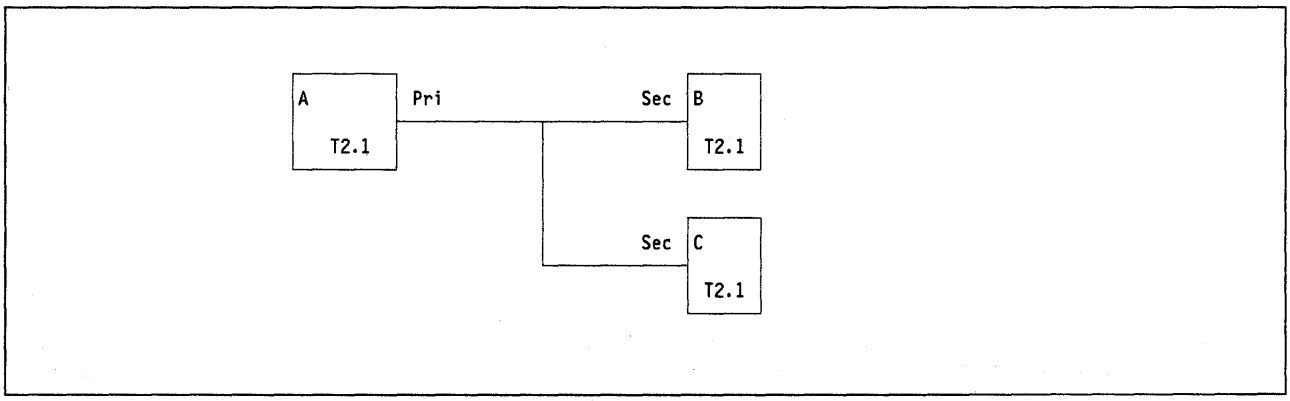


Figure 61. Multipoint Configuration of T2.1 node Nodes. In this example, node A is the primary station on a multi-point link, allowing it to communicate in peer fashion with both B and C concurrently.

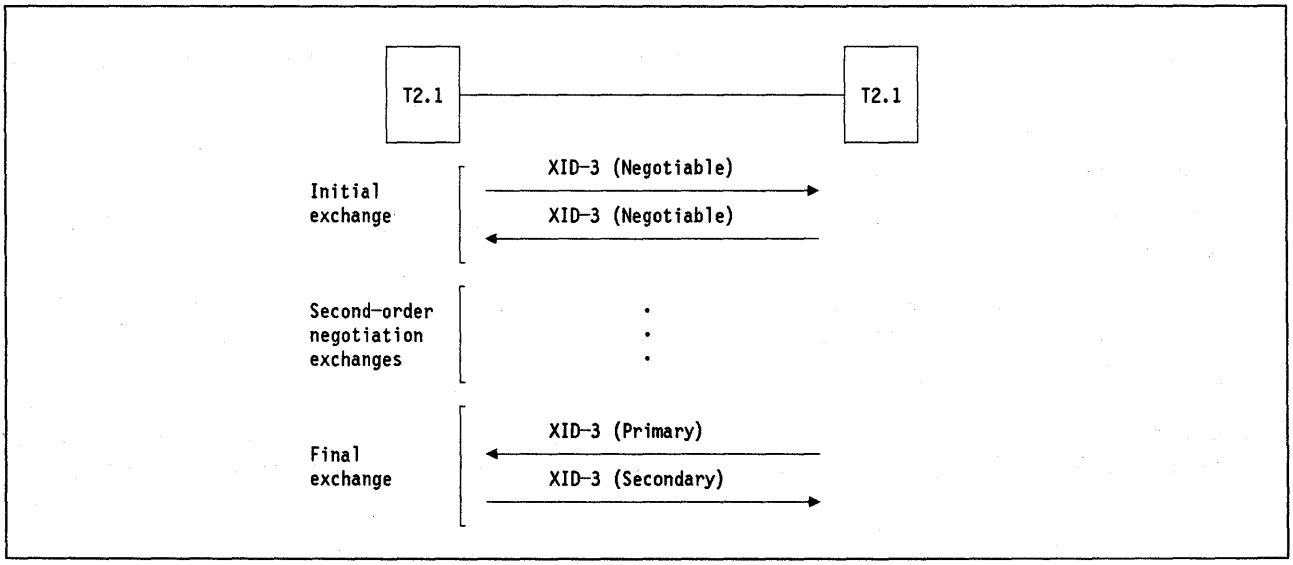


Figure 59. Example of SDLC Role Negotiation

LU-LU Session Support

This section describes the support provided for LU-LU sessions by the path-control layer of the T2.1 node.

Session Capabilities

The T2.1 node supports the following session capabilities:

- Primary LU (BIND sender) and Secondary LU (BIND receiver)
- Multiple and parallel sessions.

This is illustrated in Figure 62.

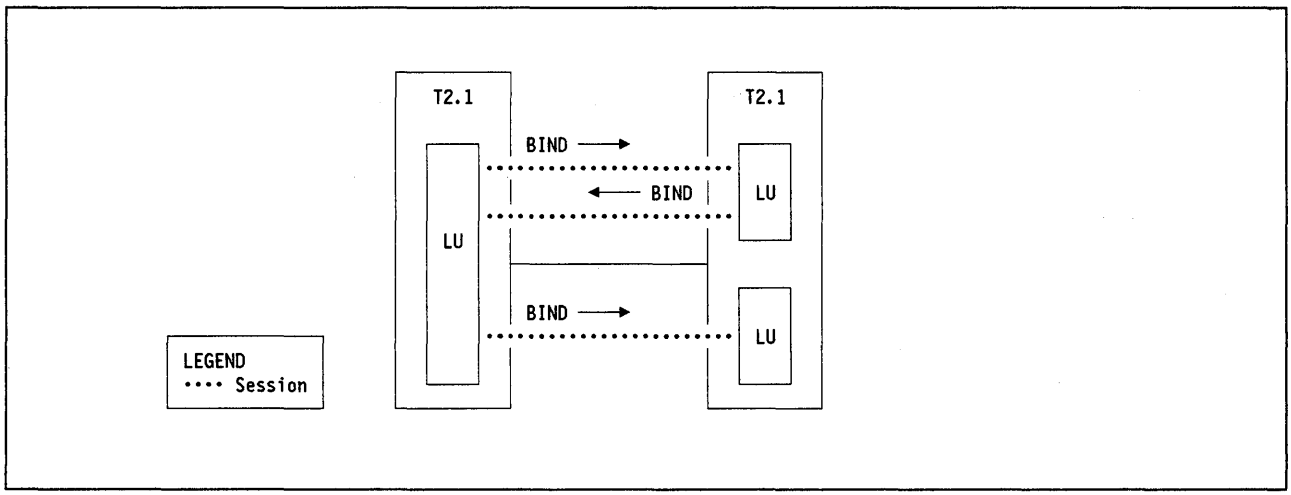


Figure 62. LU-LU Sessions Between Type 2.1 Nodes. In this example, the LU on the left has three concurrent sessions, two of which are parallel sessions with one of the partner LUs. As suggested by the labeling of the sessions, which LU is the primary (BIND sender) can vary from one session to another.

The only session protocols "seen" by the T2.1 node are the BIND and UNBIND. The target LU for an incoming BIND is identified by the *SLU Name* field. The basic session flow is shown in Figure 63.

Transmission Header Usage

Like the T2.0 node, the T2.1 node uses the six-byte FID2 transmission header, though with some differences:

- The OAF′ and DAF′ fields do not have individual significance, but function jointly to identify the particular session on which the associated RU is flowing.
- Bit 6 of byte 0, the **OAF′/DAF′ assignor indicator (ODAI)**, indicates the end of the link at which the OAF′ and DAF′ were assigned at session initiation:

  ODAI = 0 Used by the assigning node having the greater role negotiation value (carried in the XID-3, as described above under "DLC Activation" on page 136)

  ODAI = 1 Used by the other node

  For a given link, the combination of OAF′/DAF′ fields and the ODAI bit is known as the **local form session identifier** (LFSID). The manner in which it is assigned is described under "Addressing Mechanism."

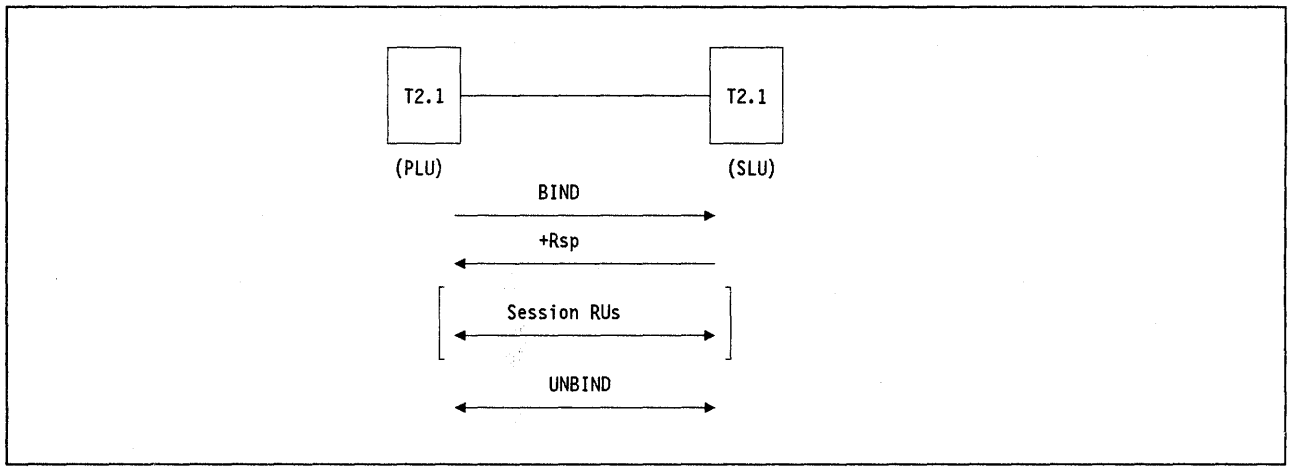


Figure 63. Session Activation and Deactivation Sequence. The BIND can flow after the initial link-level contact procedures are complete. The UNBIND may flow in either direction between the primary and secondary LUs.

Although the DAF′ and OAF′ fields are interchanged as the direction of transmission flips, the ODAI value retains its original value for the duration of the session.

Note: Contrast the above use of the FID2 with the situation on a link between a T2.0 node and a boundary function node, where one of the address fields (which one depending on the direction of transmission) contains the *local* (intra-node) *address* of the LU within the T2.0 node, while the other, the *session index*, merely identifies whether the session is an SSCP-LU or an LU-LU session, being set to 0 or non-zero respectively. (VTAM and NCP require an LU-LU session limit of 1 for each LU in a T2.0 node.)

Addressing Mechanism

The LFSID (Local Form Session Identifier) identifies a particular session on a given link, and is comprised of:

1. A one-byte **session identifier high (SIDH)**
2. A one-byte **session identifier low (SIDL)**
3. The ODAI for that session (see under "Transmission Header Usage").

   The LFSID is allocated separately and independently at each end of the link connection for sessions originating at that end. That is, when a BIND is to be sent on to the link a new LFSID is calculated at the originating end. This could lead to a conflict in that the same LFSID could be allocated independently at each end and therefore the same LFSID would refer to two sessions! To resolve the conflict, one end of the connection sets the ODAI bit in its LFSID to a "1" and the other end of the connection sets it to a "0". Thus LFSIDs are always kept unique on a particular connection.

The LFSID is specified by setting (SIDH,SIDL) to X′0101′ for the first session. It is subsequently incremented by one for each new session, with the provision that values released by deactivated sessions are reused, the lowest unused value being chosen each time. The algorithm is performed by the control point whenever a new session is needed. The concept is illustrated in Figure 64. In that figure, each descending chain of boxes represents a chain of session activations, in which the LU in the uppermost box is the primary LU, and those in the successive boxes the partner LUs, the number in each box being the hex value of the (SIDH,SIDL) part of the LFSID for that session. Node X is assumed to have the greater role negotiation value for the common underlying link and, therefore, ODAI of 0.

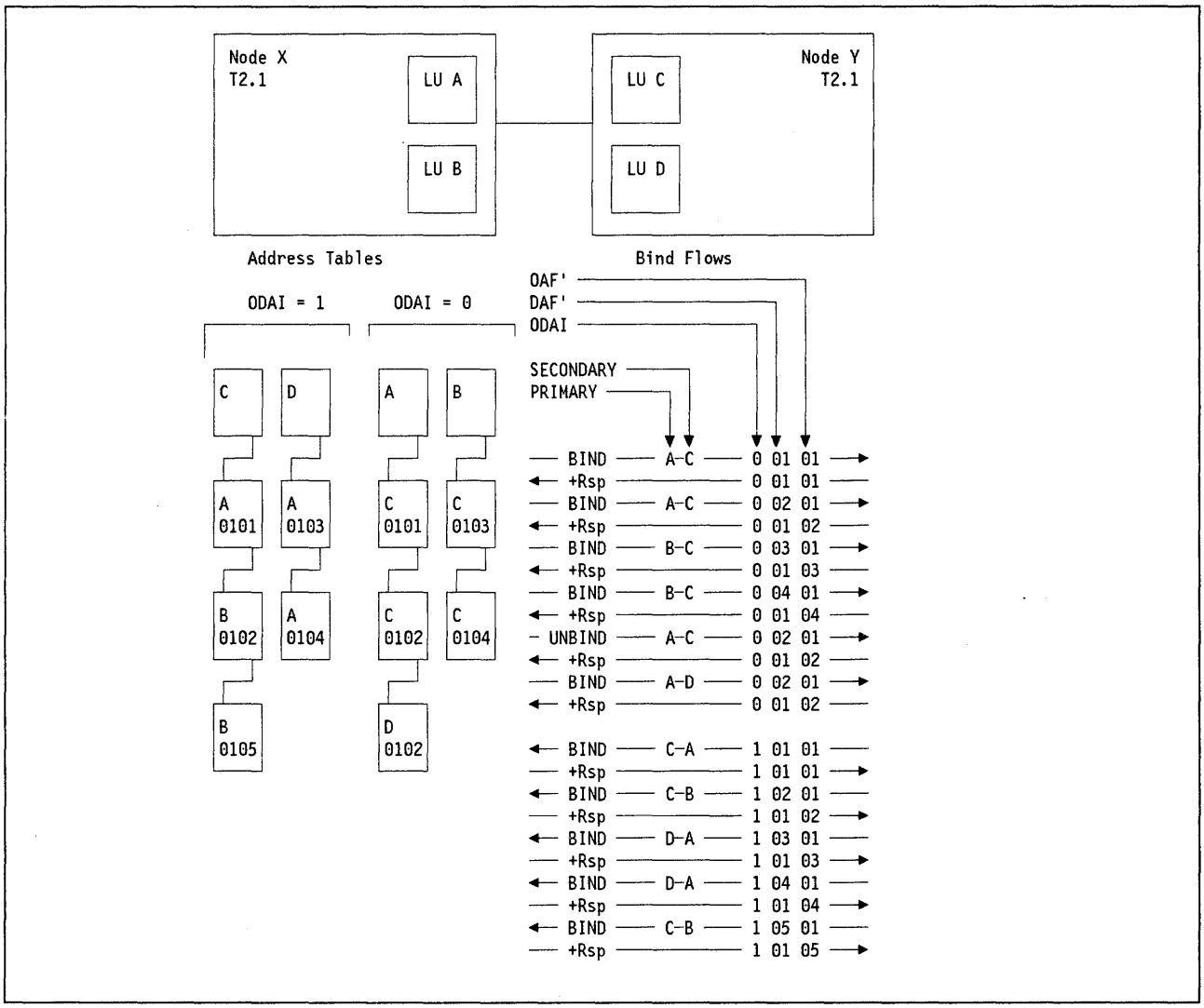


Figure 64. Conceptual Addressing Mechanism for T2.1-T2.1 Sessions

T2.1 Node Components

The T2.1 node contains a **control point** (CP) and one or more LUs.

The purpose of the CP is to coordinate the node resources. It performs link-level functions such as link activation and deactivation, initiating dial operations, and contacting and discontacting adjacent link-toptions.

The CP may also provide a **directory** of LU names, similar to that provided by an SSCP in a host (T5) node, to assist in session activation, and an interface to the node operator.

Additional information on T2.1 nodes can be found in *SNA Format and Protocol Reference Manual: Architecture Logic for Type 2.1 Nodes* (SC30-3422).

Appendix B. An Introduction to APPN Networking

Basic Functions of APPN

In this initial discussion, APPN function is explained. For the sake of clarity, there is no consideration given to different APPN node types. At this stage all nodes are assumed to provide total APPN functions (network node).[56]

"APPN in a Network Containing EN and L.E.N. Nodes" on page 157 discusses the APPN support for nodes which do not provide full APPN function: end nodes and low entry networking nodes. The part of the network which implements full APPN function is referred to as the **intermediate routing portion of the network** and does not include end nodes or low entry networking nodes.

There are some basic functions of APPN which provide the infrastructure to enable the other APPN functions to be performed. These basic functions of the control point are:

1. Control Point Manager Services (CPMGR).
2. Topology Routing Services (TRS).
3. Directory Services (DS).
4. Intermediate Session Routing.

These functions cover different levels of APPN.

1. **Control Point Manager Services**, provides services at a **coordination level**.
2. **Topology Routing Services**, provides services at a **route selection level**.
3. **Directory Services**, provides services at a **network searching (or location) level**.
4. **Intermediate Session Routing**, provides services at the **data transport level**.

Control Point Manager Services (CP)

Each node in an APPN network contains a **Control Point Manager (CPMGR)**. The Control Point Manager in each node controls the establishment and management of the control point session. This allows the topology and routing services and directory services functions to be performed over the control point sessions. It may also be considered the "general manager" of the node, responsible for managing APPN functions in that node and for communicating with other directly attached CPs. When a control point-to-control point (CP-CP) session is first established the capabilities of the two control points are exchanged. When this exchange is complete the TRS and DS functions are able to use this session.

Topology Routing Services (TRS)

There are three functions performed by Topology Routing Services (TRS). The first function (topology database) will be discussed here, while the other two functions (route selection services and class of service) is discussed in "APPN in a Network Containing Multiple Routes" on page 148.

[56] This appendix was abstracted from "IBM AS/400 Advanced Peer-to-Peer Networking (APPN)," (GG24-3287-0)

Each node in an APPN network contains a *topology database* (excluding Low Entry Networking nodes).[57] The topology of a network is the network shape or the network configuration; it describes the nodes in a network and how they are linked. Thus, a topology database contains information about the nodes in the network, the links between them and associated characteristics of these nodes and links.

When a *new* node or link is activated (or the characteristics of an existing node or link change) in an APPN network, the CP in that node uses a CP-CP session with adjacent nodes to communicate. These two CPs then exchange information regarding the network topology and update their respective topology databases. The information is then propagated around the network by an iterative process using CP-CP sessions throughout the network. Thus, all nodes in the (intermediate routing portion of the) network will update their respective topology databases with the information of the new node. There are two types of APPN nodes (as discussed in "APPN in a Network Containing EN and L.E.N. Nodes" on page 157) which will not obtain the entire topology information: APPN End Nodes (EN) and APPN L.E.N. Nodes (L.E.N.). A L.E.N. node does not contain a topology database at all and an end node contains a small topology database containing only information about adjacent links.

The AS/400 topology database is stored across IPLs; thus the database need not be rebuilt at each IPL. In addition, special sequence numbers, called flow reduction sequence numbers are maintained in order to reduce the amount of information that needs to be exchanged in order to keep the topology databases accurate in order to reduce the resources used after interrupted transmission.

Directory Services (DS)

A request for a session with a remote system may be made by using the CP name of the remote system, or, by using a nickname (remote location name) for the remote system. See "Multiple Location Names" on page 147.

Directory services keeps track of nicknames (or location names) which may be defined for nodes in an APPN network.

When a session is requested with a remote location, a search request is sent by directory services, using CP-CP sessions with adjacent nodes, to determine the control point that owns the remote location. The function of directory services is to:

1. **Search the local directory database**

   The local directory database contains:

   - All nicknames for the local system (local location names)
   - The location names of all End Nodes and L.E.N. Nodes in its domain
   - Other remote nicknames (remote location names) which have been previously defined or, as explained in point 3 below, added dynamically.

2. **Send a directed search through the network**

   If an entry is found in the directory database (the directory will specify which CP owns the remote location specified in the session request) then a **Directed Search** may be performed. A directed search is one in which the search request is sent directly to the CP owning the remote location name specified. A search must still be sent to ensure that the remote location name found in the local directory database is still owned by the same CP.

   The CP to which the search was directed will return the request with a positive response (if it still owns the specified location) or a negative response.

[57] End Nodes contain a limited subset of the full Network Node Topology Database.

3. **Send a broadcast search into the network**

   If the requested location nickname is not found in the local directory database (or a directed search was returned with a negative response), then a **broadcast search request** is sent to adjacent network nodes to which a CP-CP session is active.

   Adjacent nodes will pass the search request on to other nodes adjacent to them. This will occur even if the remote location requested resides in their directory database) because its current directory information may not be up to date. In the mean time the remote location may be controlled by another CP. If the remote location does reside in the adjacent nodes' directory database the broadcast search request is returned with a positive response (thus identifying itself as the CP owning the remote location requested) and the node will continue to forward the search request. The reason the search is continued, even when a the requested location is found, in order to check that there are no other control points that have the same location name defined as being local.

   In summary, a request for a session with a remote location will initiate:

   a. A directed search request if the requested location name is found in the local directory database and is not a (network node) CP name. The second request for a particular remote location will result in a directed search since the first request will have ensured that the directory database be updated to include it.

   b. A broadcast search if the requested location name is not found in the local directory database.

   c. No search request if the requested location name is also a network node CP name. (If the CP name of an End Node or L.E.N. node is used then a directed search is sent anyway). Since a search of the network must be made, the time to establish a session will be greater than if a network node CP name was specified as the remote location.

4. **Update the local directory database** with information about which CP owns the remote location nickname specified.

   The location directory database is built and maintained by:

   - Defining local location names in a **local location list**. A local location list may be accessed and updated by a user at any time by using the AS/400 command, WRKCFGL *LCL, described in the configuration section.
   - Defining remote locations in a **remote location list.** The AS/400 command is WRKCFGL *RMT. When entering a remote location name in the remote location list the remote control point name must equal the remote control point name in a controller description if the remote location is in an adjacent system.
   - Dynamic update when a session is established with a remote location not previously in the database.

   The IBM AS/400 directory database is stored across IPLs in the same way that the topology database is stored. Thus, like the topology database, it need not be rebuilt at IPL time.

Intermediate Session Routing

Intermediate routing is a function performed by an IBM AS/400 network node in allowing an LU6.2 (APPC) session to be routed through it; the session neither starts nor ends in the intermediate network node.

In the example in Figure 65 on page 146 system B may provide the intermediate routing for an APPC session from A to C (or from C to A). System B does not participate in the APPC session and thus need not provide any supporting APPC code. System B provides a routing not a relay function.

Intermediate Routing Portion of the Network

As discussed in "APPN in a Network Containing EN and L.E.N. Nodes" on page 157, there are some nodes in an APPN network which cannot perform intermediate routing. The intermediate routing portion of the network refers to that portion of the network in which all nodes and the TGs connecting these nodes may be used for intermediate session routing. In the network described up to this point, the entire network is within the intermediate routing portion of the network.

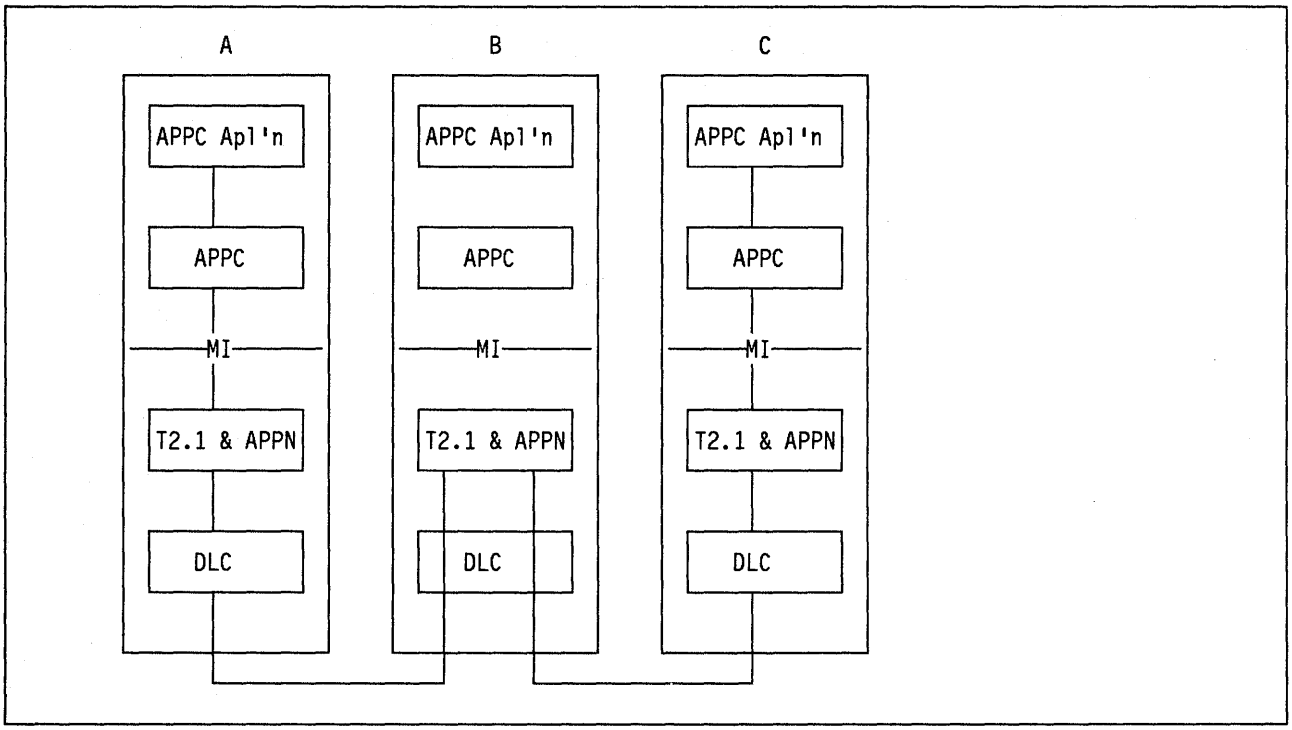


Figure 65. APPN Intermediate Routing

APPN in a Simple Three-Node Network

The basic functions of APPN provide each node in a network with information about the network and the means by which that information is distributed. Now, consider the following functions (or features supported by APPN) which are dependent upon the information and services provided by the basic functions:

- Session Activation
- Automatic Peer Device Creation and Activation
- Multiple Location Names.

Session Activation

An LU6.2 SNA session is established with a special set-up message called a BIND. The location names of the session source and target are specified in the BIND.

In an APPN network, a BIND also contains the names of source and target nodes. However, the target node need not be the adjacent system. Thus, an APPN BIND also contains information about where the intermediate system(s) should forward the BIND. This additional information sent with an APPN BIND is called a **Route Selection Control Vector (RSCV)**. Naturally, APPN provides the ability, within APPN

nodes, to recognize and forward BINDs which are not targeted for them; this is the intermediate routing function.

Consider the network in Figure 67 on page 149. System A may establish a session with System C by sending a BIND containing the names of both the session initiator (location A) and session target (location C). The BIND also contains a route description; in this case, of A to B to C which lets system B know where to forward the BIND.

As the BIND passes through node B along its path to node C, it also leaves behind a special routing marker called a **session connector**. A session connector will cause all subsequent data, belonging to that particular session, to travel along this route (that is, from A to B to C and back) without the need for an RSCV to be sent also. The RSCV also contains a TG number associated with every CP name in order to differentiate between parallel TGs between adjacent control points.

Route Selection Control Vector (RSCV)

An RSCV is a vector which is appended to an APPN BIND and provides information regarding the route over which a particular session is to be established. The routing information carried by an RSCV describes the total path from the origin to destination control point.

The RSCV is 255 bytes long and thus cannot contain an unlimited number of CP names. It is, then, the RSCV length and the number of CP names concatenated within the RSCV which determines the maximum number of intermediate hops along an APPN route. Naturally, the shorter the CP names (a CP name may be up to 8 characters in length), the more hops possible.

Automatic Peer Device Creation and Activation

In a non-APPN environment, APPC communications between any two peer systems, for example display station pass-through between IBM AS/400's not using APPN, requires a line description, an APPC controller description, and an APPC device description to be manually created and varied-on.

In an IBM AS/400 APPN environment, line and controller descriptions are still required (though for adjacent nodes only) but **there is no need to manually create or vary-on any device descriptions**. When a session is requested to another node in the network, two device descriptions are automatically created. On the local node, a device description is created to represent the remote (or target) location pairing; and, on the remote node, a device is created to represent the same local (or requesting) location pairing. Not only is the device automatically created but it is automatically attached to the correct controller description (the one which represents the route through which the session is established) and activated. The session is thus dynamically established.

Note: If no device description exists when a session is requested, it will be created, varied-on, and attached automatically to the correct controller description, both locally and remotely. However if manually created device descriptions exist for the selected location pair, they will be used and if necessary still activated automatically if attached to the correct controller description.

Multiple Location Names

The location names used for nodes in examples up to this point have been the CP names for each node; that is, A, B, C.

The topology database in each node is automatically updated with the CP name of each node in the network, as that node establishes a CP-CP session to another NN in the network and therefore joins the network. Thus, it makes sense to use the CP name as the remote location name to establish the session in some cases. However, there may be some instances where a number of different location names (nicknames)

are required or preferred for some remote nodes. (The use of location names is beneficial in the case of resource movement. This is explained further in the following paragraph.) These nicknames (as mentioned previously) are called *location names*; the CP name of a node is, by default, always a location name. Consider Figure 66. In this figure there are multiple locations for each of the nodes in our three-system network. The CP names remain the same; however, each node has been given a number of new location names which either may be more meaningful or simply associated with arbitrary resources such as files.

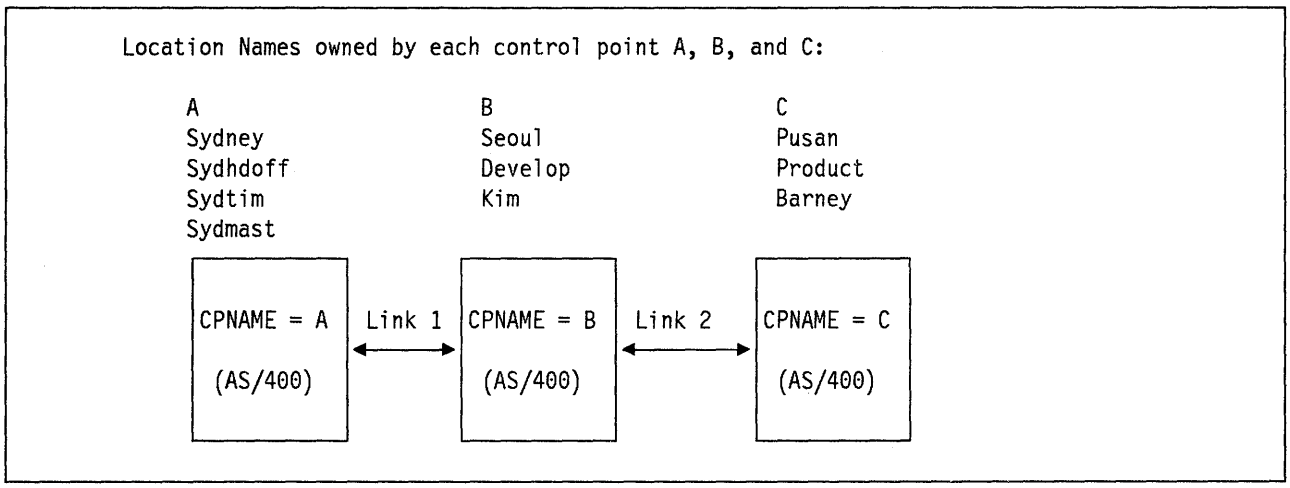


Figure 66. APPN Network of Three IBM AS/400s with Multiple Location Names. Location names are nicknames for a system; the CP name is automatically a location name. The other location names are user-defined and may represent a useful naming convention such as the node's physical location, location of a department, or associated with any arbitrary resource. Any arbitrary name will suffice that consists of between 1 and 8 symbol string type A character set.

Resource Moving

A major benefit of multiple location names is the flexibility which these names provide in an APPN network. This allows APPC applications (DDM, SNADS, Display Station Passthrough or another IBM or user-written program) to establish conversations with a remote node in an APPN network, identifying the remote node by any name (provided it is defined as a local location on the remote node). Thus applications may be written independently of the CP names (of remote nodes) to which they communicate (meaning that the location name used by an APPC application may be moved to any node in the network without affecting the APPC application itself). Moreover, a user or user application program may refer to a resource, for example a DDM file, by name without knowing the remote node in which that resource resides if the file name equals the remote location name.

APPN in a Network Containing Multiple Routes

Now suppose that the simple three node network becomes more complex. Consider Figure 67 on page 149, in which another possible route is now included between nodes A and C. In fact, the network now includes a fourth node, node D. Some local location names have been specified for each node (in this case, the city name) and the CP name remains as A, B, C, D.

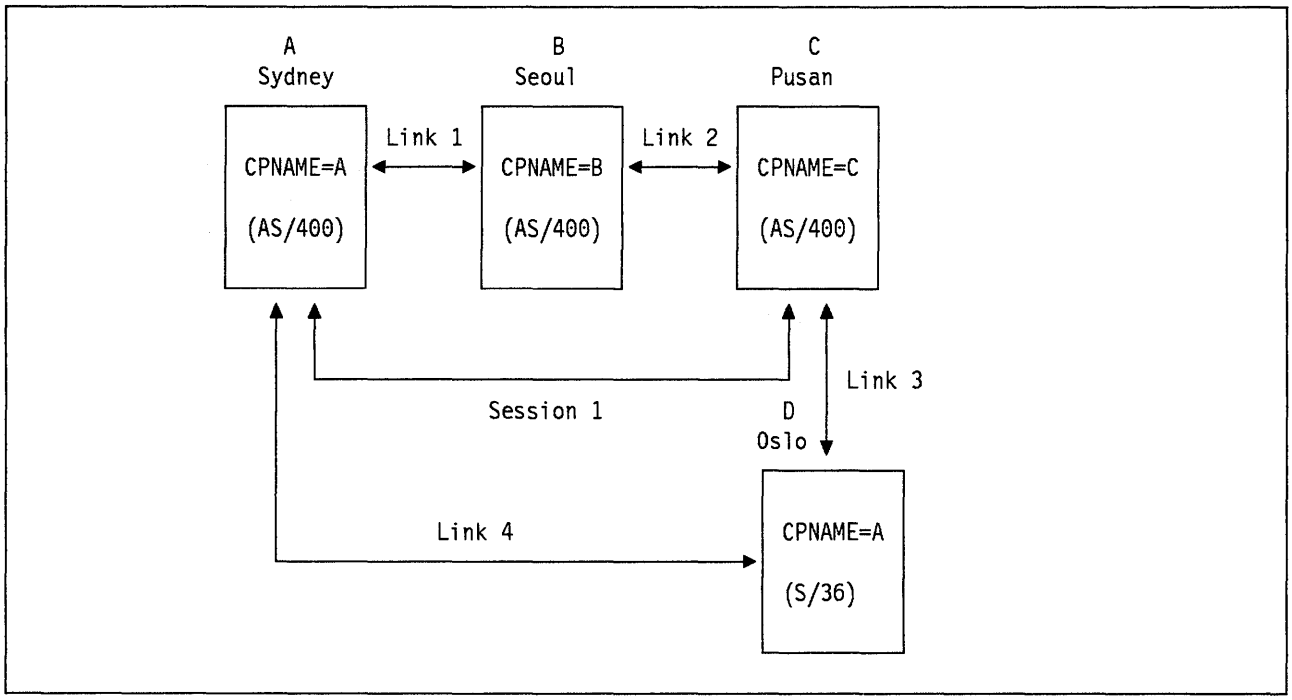


Figure 67. APPN Network with Multiple Routes

Route Selection

When Sydney (node A) establishes a session with Pusan (node C) it must decide by which route the session will be established. The choice is via node Seoul or via node Oslo. The best route to be taken depends on the requirements of the user; perhaps it is the cheapest route which is preferred or perhaps the fastest (or perhaps there is some other requirement). It is **Topology Routing Services (TRS)** in Sydney (the source or initiating system) which determines which route to follow. The source system will always determine the route (except when it is an end node, as discussed later).

There are three elements of Topology Routing Services (TRS).

1. Topology database.
2. Route selection.
3. Class of Service.

These services provide the means by which the most preferred route may be chosen between nodes in an APPN network.

In the IBM System/36 implementation of APPN, when a session was established, the route of the least number of hops was selected, regardless of the characteristics of the links between the nodes or the nodes themselves, which may not be the most preferred route.

In the IBM AS/400 implementation, Topology Routing Services (TRS) provides the flexibility to choose the *most preferred route* to be selected each time a session is established. In each node in an APPN network, a route is selected by comparing information in the topology database with two **user definitions:**

1. **Link Class of Service Table (TGCOS).**(Another name for a link is a Transmission Group (TG). Hence the name TGCOS.) The user may specify the link characteristics that are preferred in a **Link Class of Service Table**. The TGCOS table is ordered from the most to the least desirable link characteristics, according to user requirements for a particular session. The TG characteristics specified in the TGCOS table are the same as those found in the topology data base.

2. **Node Class of Service Table (NCOS).** The **node class of service table** enables a user to specify the relative preference for other nodes in the network to perform intermediate routing for the local node.

Once the most-preferred route has been calculated (by combining the preferences in both the TGCOS and NCOS tables), a **Route Selection Control Vector (RSCV)** is created. An RSCV is the mechanism by which best-route information is specified; it is attached to the BIND for the session and tells each node performing intermediate routing where the BIND should be forwarded next.

Class of Service Table

The Link Class of Service Table and Node Class of Service Table are in fact, **a single system object** of which there may be a number defined in any node. Both the NCOS and TGCOS tables (usually, and hereafter, referred to as a single **COS table**) are defined using the command CRTCOSD.

A different COS table may be specified any time a session establishment request is made, thus the user has the flexibility to establish a different session **over a different preferred route** at a particular point in time.

At session establishment time, the initiating system compares the physical characteristics of all the nodes and links in the network between the origin and destination control points (contained in the topology database) with those characteristics which have been defined to be desirable in the COS table. The route with the most desirable characteristics (as defined in the COS table) is the one selected for the session.

The physical characteristics for all links, found in the topology database, are extracted from the **line descriptions** at each node when the attached controller description is activated. The characteristics of a link are described later in this section.

The node characteristics, *route addition resistance (RAR)* and *congestion*, of all network nodes in the network, are also found in the topology database. The RAR is extracted from the **network attributes** of each node and propagated in topology database updates when a link is activated between two network nodes. See "Class of Service Table." Network node congestion is determined dynamically. Network nodes send out topology updates to notify other network nodes that they are congested when they reach 90% of their configured maximum intermediate sessions and will send out another update to indicate they are no longer congested when the number of active intermediate sessions goes below 80% of the maximum configured.

To summarize, in an APPN node there is a system object, representing two separate tables: a TGCOS table and an NCOS table. These tables define preferred characteristics of all the links and nodes in an APPN network. At session establishment time, these tables are compared with actual values in the topology database and a preferred route is thus selected.

In order to better understand how the best route is determined, consider the following illustrations: Figure 68 on page 151, Figure 69 on page 152 and Figure 70 on page 153 show a TGCOS table and an NCOS table. Figure 68 on page 151 is an example network for which a number of link and node characteristics have been defined.

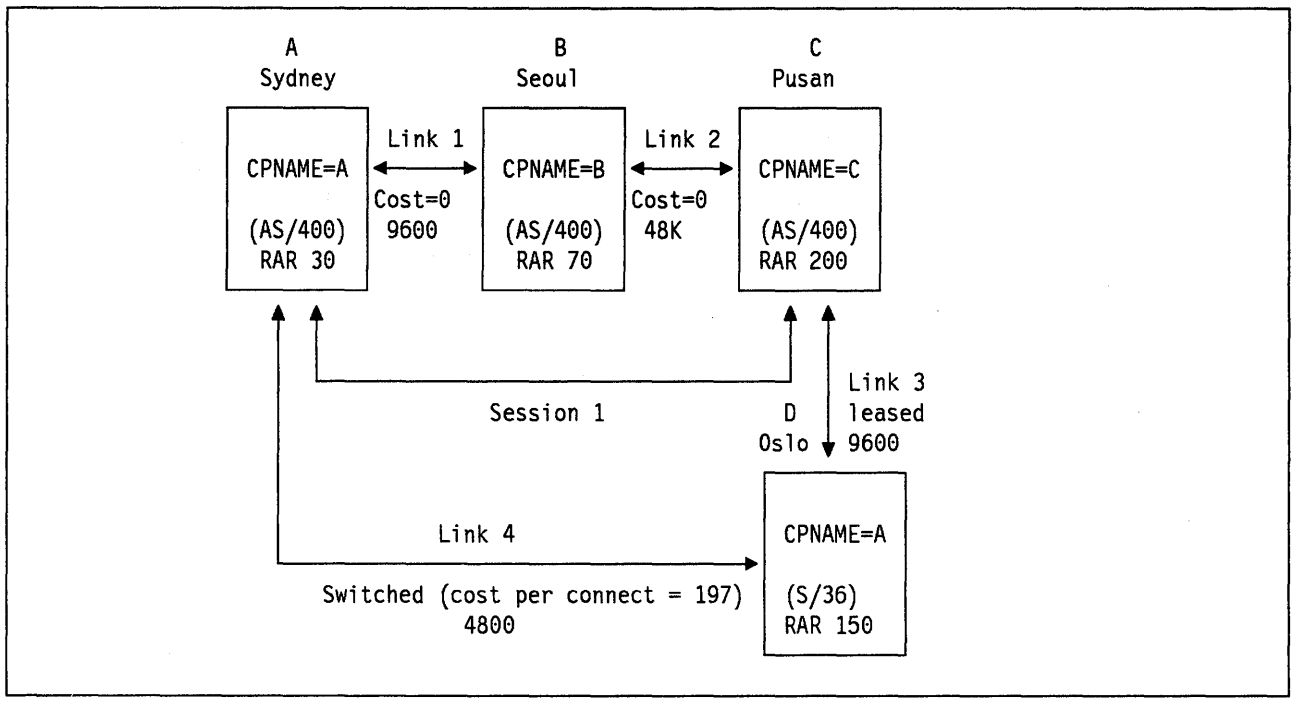


Figure 68. Example Network for Route Selection

For purposes of illustration, assume a session is requested from Sydney to Pusan.

Topology Routing Services (TRS) will perform the following:

1. Check the topology database in order to identify the characteristics of the links and the nodes in the possible routes.

   The topology database in node Sydney will show the information which is included in Figure 68; that is, RAR (defined in the network attributes) for each node (and whether or not each node is congested) and the following values defined in the line description:

   - Link speed
   - Security
   - Cost per connect time
   - Cost per byte
   - Propagation delay
   - Three user-defined fields.

2. Compare the values found in the topology database with the values specified in the COS table specified at session establishment.
3. Determine a "weight" to be assigned to each node (except the target node) and link. The weight assigned to a particular link or node is determined by using the values specified in the topology database and finding the first row of the COS table in which the value fits (each row of a COS table specifies a maximum and minimum value) The row chosen will then specify the weight for that link or node.
4. Sum the weights calculated for each "link and node" on each of the different routes.
5. Specify which route should be taken by determining which route has "the least weight." It is the set of nodes and TGs with least total weight which will be chosen.

The rules which need to be followed in using the table are as follows:

1. ALL conditions in a row must be satisfied before that row is chosen.

   That is, a row of the COS table will be chosen if:

   'Min' is less than or equal to 'value in topology database' is less than or equal to 'Max'

   for each and every column in the class of service table. If not, then no row will be chosen and that TG or node will not be chosen for this session route. There may be circumstances under which it is desirable for no session to be established. For example, if link 1 failed, the user may not wish a session to be established over link 4 since it is switched and thus more expensive.

2. If more than 1 row satisfies the conditions then the first row from the top (lowest number) will be used.

```
          Class-of-Service Description Line Information

        Link   Cost/   Cost/  Security    Propagation   User Defined
        Speed  Connect Byte   for Line    Delay          1    2    3

Min     4M      0       0     *NONSECURE  *MIN           0    0    0
Max     *MAX    0       0     *MAX        *LAN         255  255  255
Weight for Row 1 = 30

Min     56000   0       0     *NONSECURE  *MIN           0    0    0
Max     *MAX    0       0     *MAX        *TELEPHONE   255  255  255
Weight for Row 2 = 60

Min     19200   0       0     *NONSECURE  *MIN           0    0    0
Max     *MAX    0       0     *MAX        *TELEPHONE   255  255  255
Weight for Row 3 = 90

Min     9600    0       0     *NONSECURE  *MIN           0    0    0
Max     *MAX    0       0     *MAX        *TELEPHONE   255  255  255
Weight for Row 4 = 120

Min     19200   0       0     *NONSECURE  *MIN           0    0    0
Max     *MAX    0       0     *MAX        *PKTSWTNWT   255   55  255
Weight for Row 5 = 150

Min     9600    0       0     *NONSECURE  *MIN           0    0    0
Max     *MAX   128    128     *MAX        *PKTSWTNET   255  255  255
Weight for Row 6 = 180

Min     4800    0       0     *NONSECURE  *MIN           0    0    0
Max     *MAX   196    196     *MAX        *MAX         255  255  255
Weight for Row 7 = 210

Min     *MIN    0       0     *NONSECURE  *MIN           0    0    0
Max     *MAX   255    255     *MAX        *MAX         255  255  255
Weight for Row 8 = 240
```

Figure 69. The IBM-Supplied #Connect Class-of-Service Table. The figure shows the portion of the COS table which defines the weights to be used for links in the network. To change the likelihood of a session being established over a particular link, the weights assigned to a particular row may be changed or, alternatively, the values specified in each column may be changed. For example, changing the minimum line speed in each row to 9600 will mean that, by specifying this COS table at session initiation, no session is to be established unless every link has a line speed of 9600 or greater.

```
                    Node Class-of-Service Description
            Weight      Route Addition Resistance      Congestion
Min....                                 00                 *LOW
Max....       5                         31                 *LOW

Min....                                 00                 *LOW
Max....      10                         63                 *LOW

Minimum                                 00                 *LOW
Maximum      20                         95                 *LOW

Minimum                                 00                 *LOW
Maximum      40                        127                 *LOW

Minimum                                 00                 *LOW
Maximum      60                        159                 *LOW

Minimum                                 00                 *LOW
Maximum      80                        191                 *LOW

Minimum                                 00                 *LOW
Maximum     120                        223                 *HIGH

Minimum                                 00                 *LOW
Maximum     160                        255                 *HIGH
```

Figure 70. The IBM-Supplied #Connect Class-of-Service Table. The figure shows the portion of the COS table which defines the weights to be used for nodes in the network.

By using Figure 68 on page 151, Figure 69 on page 152 and Figure 70 and utilizing the following formula:

> Route(x) Weight = Sum of Link(i(x)) Weights + Sum of Node(j(x)) Weights; for all i in route x; for all (j ¬ = target node) in route x.

the following calculations would be made, the following weights determined, and the following route selected. Assume in each case that a node is not congested (and thus the column congestion in the NCOS table will always be '0'). Only two link characteristics are shown in this example for the sake of simplicity, but they are all used during route calculation.

Route 1 (Link 1 + Link 2 + Node B)

Link 1
Line Speed = 9600 (arbitrarily defined in Figure 68 on page 151)
Connection Cost = 0 (from Figure 68 on page 151)
First COS Table Row satisfying all conditions = 4
Weight for row 4 = 120
Therefore Weight for Link 1 = 120

Link 2
Line Speed = 48K (Figure 68 on page 151)
Connection Cost = 0 (Figure 68 on page 151)
First COS Table Row satisfying all conditions = 3
Weight for row 3 = 90 (notice this is a smaller weight than for 9600)
Therefore Weight for Link 1 = 90

Node B
RAR = 70 (arbitrarily defined in Figure 68 on page 151)
First COS/RAR Table Row satisfying all Conditions = 3
Weight for Row 3 = 20
Therefore Weight for node = 20

Route 2 (Link 4 + Link 3 + Node D)

Link 4
Line Speed = 4800
Connection Cost = 197 (defined in the line description)
COS Table Row satisfying all conditions = 8
Weight for row 8 = 240
Therefore Weight for Link 4 = 240

Link 3
Line Speed = 9600
Connection Cost = 0
First COS Table Row satisfying all conditions = 4
Weight for row 4 = 120
Therefore Weight for Link 3 = 120

Node D
RAR = 150
First COS/RAR Table Row = 5
Weight for Row 5 = 60
Therefore Weight for node = 60

Route (1) Weight = 120 + 90 + 20 = 230
Route (2) Weight = 240 + 120 + 60 = 420
Therefore, Route 1 will be chosen.
Route 1 has the least total weight and therefore a session between Sydney (A) and Pusan (C) (established specifying the COS table in Figure 69 on page 152 and Figure 70) will use route 1 (via Seoul).

Figure 71. Example of Preferred Route Calculation. Using the COS Table described in Figure 69 on page 152 and Figure 70.

User Defined Rows in the COS Table

As mentioned earlier, all the columns in a COS table correspond to values in the topology database which, in turn, are extracted from the line description.

There are some other parameters in the line description which correspond to columns in the COS table; they are the "user-defined" values. A relative number may be specified in one of the user-defined fields in the line description for each line, and then used as if it were another physical characteristic of the line. A user-defined field thus provides another condition which needs to be satisfied before a particular row in the COS table is selected.

Route Addition Resistance

The node characteristic Route Addition Resistance (RAR) can be used to identify network nodes that are more preferable for intermediate routing as compared to other network nodes. The value for RAR, specified in the network attributes (by using the CHGNETA command), is a relative value which is only meaningful when compared to other configured values defined around the network. A user may specify, in the NCOS table, which value for RAR is preferred; the default value is 128.

Node Congestion

Another node characteristic is *node congestion.* The maximum number of intermediate routing sessions supported by a network node may be defined using the CHGNETA command. Network nodes are said to be "congested" if 90% of that number, defined in the network attributes, is reached. The node becomes "uncongested" when the number of intermediate routing sessions through the node becomes less than 80%.

Note: It is also possible to define a NCOS table which will not choose a node which is congested.

When a node is congested (or becomes uncongested) the topology databases of other nodes are updated. Notice that the rows which show a "*high" in the "congested" column are the last in the NCOS table and thus will be the least preferred. If there is no other available route, however, then the route containing the congested node will be used up to 100%.

Modes

It has already been mentioned that a COS table (the term COS will be used to include both TGCOS and NCOS, from this point) is specified at session establishment time. The means by which a COS table is specified is, however, via a *mode* which is specified at session establishment time; the mode then points to a COS table to be used in route selection.

An IBM AS/400 mode is conceptually the same as a IBM System/38 mode or a IBM System/36 session group; it defines session characteristics such as:

- Maximum number of sessions.
- Maximum number of locally-controlled sessions.
- Number of pre-established sessions.

A mode is a system object. It is created by the CRTMODD command, and is associated with an APPN device at session establishment. In the IBM System/38 a mode is added to a device and can be used by only that device. An IBM AS/400 mode may be used by any location pair which specifies it at session establishment or is started remotely.

As already mentioned, a mode now also specifies a COS to be used for route determination.

The ability to specify a mode 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. Thus, the IBM AS/400 operating system is shipped with five predefined modes and corresponding COS tables. The user has the option to create other COS tables or modes according to specific requirements but this is not necessary in order to establish sessions in an APPN network. In fact, the user need not even consider modes and COS tables when using an APPN network since the system will use defaults provided in the line descriptions and COS tables. When using these defaults a session will always be able to be established if the #CONNECT, #BATCH, or #INTER COS is used.

The following predefined modes and corresponding COS tables are system objects for APPC/APPN communications that are shipped with the IBM AS/400 operating system:

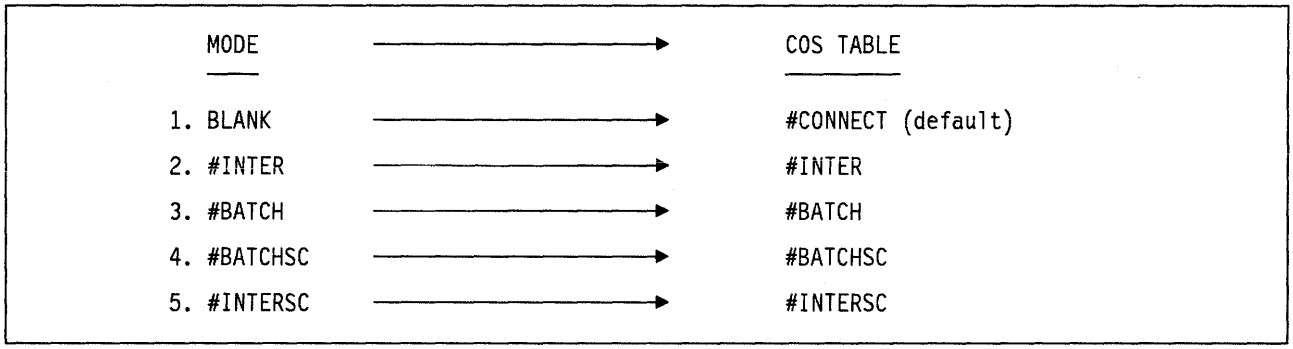


Figure 72. Predefined Modes and COS Tables for the IBM AS/400. The mode 'BLANK' is the same as *BLANK in the IBM System/38 and IBM System/36.

Note:

1. If left unmodified, the #CONNECT, #BATCH, and #INTER COS tables will be able to use any node or link.
2. When modifying the COS tables it is a good idea to copy the table first and then change the copy.
3. If any of the five COS tables and modes supplied by IBM are damaged or modified, then it may be recreated by deleting the object and re-IPLing the system.

Transmission Priority

Associated with a class of service table is a *transmission priority*. The three priorities which may be specified in a COS table are:

1. LOW (usually for batch traffic)
2. MEDIUM (normal traffic)
3. HIGH (interactive traffic).

Transmission priority is important if there are more than one session over a controller description; transmissions may then be prioritized by the transport network. The predefined COS tables shipped with the IBM AS/400 operating system specify a low-priority transmission (#BATCH), a medium-priority transmission (#CONNECT) and a high-priority transmission (#INTER). Thus, to achieve a high-priority transmission (for interactive passthrough users, for example), a user has the choice of using the #INTER mode and COS table or creating new ones that specify a transmission priority of high.

Parallel Transmission Groups

Between two AS/400 nodes in a network it is possible to have multiple logical links (that is, parallel TGs). When there is more than one controller description with the same remote network identifier and control point name between two nodes they represent *Parallel TGs*. Each TG is uniquely identified by the TG number that is either configured or negotiated. Since the default for TG numbers is set to "1" and TG numbers must be unique between control points then the second controller description must have the default changed to either *CALC or a number between 2 and 20. The corresponding controller description on the remote station must also be changed accordingly. If *CALC is specified then a number between 21 and 239 will be selected by the system.

APPN in a Network Containing EN and L.E.N. Nodes

Up until this point in the discussion, the assumption has been that each node would implement the full APPN function. However, there may not always be a need for all of the functions of APPN to be included in each of the nodes. In particular, it may only be necessary for intermediate nodes to utilize some functions such as intermediate routing. Moreover, it may be desirable for some systems which do not implement the full functions of APPN to participate in an APPN network (albeit in a limited capacity). Thus, different APPN node types were developed. There are three APPN node types:

- Network Nodes (NNs).
- End Nodes (ENs).
- Low Entry Networking (L.E.N.) Nodes.

Consider Figure 73 on page 158. The network now includes two more nodes: Bergen, an IBM AS/400 End Node(EN) and Melbourne, an IBM System/38 Low Entry Networking (L.E.N.) node. The other nodes in the network remain as NNs (nodes implementing the full suite of APPN functions) since they may be required to perform intermediate routing.

By allowing L.E.N.s and ENs to take part in what was an all network-node APPN network, some additional APPN functions become necessary and some qualifications may need to be made regarding the implementation of some functions in ENs and L.E.N.s rather than in NNs. The following discussion focuses on these additions and qualifications in the support provided by APPN for each of the three APPN node types.

Network Nodes

Network nodes effectively need no definition since they provide the full function of APPN. An NN can be:

1. An IBM AS/400 configured as an NN.
2. A IBM System/36 with the APPN features and configured to use APPN.

Network nodes provide not only APPN services to other NNs, but also may provide a number of APPN functions for End Nodes (EN) and, to a lesser extent, Low Entry Networking (L.E.N.) nodes in order to enable them to participate in an APPN network. An NN which provides APPN functions for an EN or L.E.N. node is called a *network node server* for that node. A NN which is a server for a node performs routing services and directory services for that node. Network servers are further discussed in the next section.

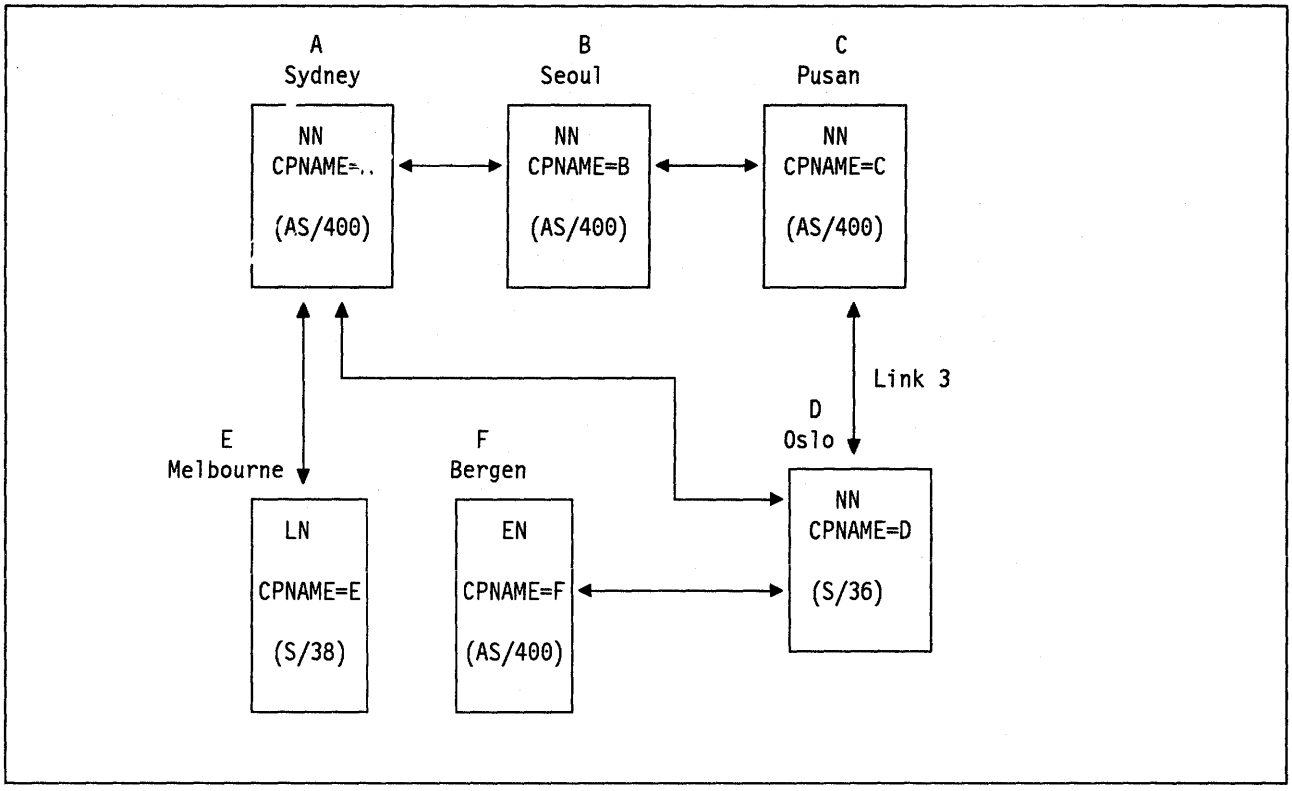


Figure 73. APPN Network including a L.E.N. and EN

End Node Support

An APPN End Node has the ability to request a CP-CP session with a network node. There are some special considerations for configuring adjacent nodes that are outside the scope of this introduction.

An APPN End Node (EN) may, as its name suggests, reside at the ends of an APPN network; it does not perform intermediate routing for other nodes in the network.

> An IBM AS/400 APPN End Node (EN) is a new concept and is **not** the same as what was called an end node in the IBM System/36 implementation of APPN (this is now called a L.E.N. See "Low Entry Network (L.E.N.) Node Support" on page 160. An EN can only be:
>
> 1. An IBM AS/400 configured to utilize CP-CP sessions.
> 2. An IBM AS/400 configured to not utilize CP-CP sessions.

The APPN EN enables an IBM AS/400 to participate in an APPN network without the resource overhead associated with the functions performed by an NN: It makes sense to exclude full topology, directory, and intermediate routing function, for example, from a node which does not perform intermediate routing. System tasks in a NN use the topology database to perform route selection and the directory database to perform network searches for remote locations (or nicknames). The tasks are minimized in an EN, because the majority of these services are performed by the adjacent network node server when needed.

The idea of a *network node server* was thus developed so that a server could provide these basic functions for the EN and thus relieve specified nodes (ENs) of some overhead. Consequently, an EN provides a small topology database which contains information about locally owned links only. Furthermore, L.E.N. nodes do not have a topology database at all. Importantly, updates to topology databases only occur between network nodes. The portion of the network which contains network nodes and links between network nodes only is called the intermediate routing portion of the network and it is only this portion in which topology updates are distributed

Network Node Server for an EN

The key support feature for ENs in an APPN network is that of a network node server. The support provided by the server to its end node depends on whether the network configuration and environment is conducive for the EN to establish a CP-CP session with an adjacent NN. Thus, the two cases are considered separately:

EN with CP-CP Session Capability: The functions performed by an NN server for an EN with a CP-CP session capability are:

1. Automatic registration, in the server's directory database, of all the location names defined at the EN as local location names.

   Every time an EN establishes a CP-CP session to obtain network node server functions, it will send all of its local location names to the NN server; the server registers this information in its directory database. When a CP-CP session is terminated between an EN and its server, the server deletes the registered local locations of the EN.

   Thus, a network node server need not configure any location names for an adjacent EN when the EN is going to establish a CP-CP session with any network node.

2. Directory services and preferred route selection.

   A NN server will perform directory searches in order to locate a remote location name specified by an adjacent EN (with a CP-CP session). Thus, an EN (with a CP-CP session to a NN server) need not manually configure any remote locations with which it may require a session.

   When an EN (with CP-CP session) requests a session with a remote location, the following process takes place:

   a. The EN sends a search request to the NN server for a session with a remote location, (along with information about the mode and COS table to be used for the session).

   b. Directory services on the NN server searches its directory database for the remote location specified. If the remote location is not found then the server forwards the search request (as a broadcast request) through the network to adjacent network nodes in order to find the specified remote location. Note that search requests are not sent to ENs unless the NN has previous knowledge of the requested location residing on the EN.

      If the remote location is found in the local directory database of the server then a directed search is sent to the remote location (to confirm that the directory information is still correct).

   c. The search request is returned to the network node server (with a positive response and information about the CP which owns the remote location), and the NN server also calculates the preferred route **from the EN** to the remote CP (based on the COS information specified by the EN when the search request was initiated).

   d. The server then returns the search request back to the EN, attaching to it an RSCV (which contains information about which route should be taken in order to establish a session between the local and destination locations).

   e. The EN then sends the BIND with the RSCV attached to the specified location via the preferred route as if it were an NN and had performed the work itself. Note that the first hop of the route does not have to be the network node server that calculated the route.

      Note: For an end node to use the services of an adjacent network node server it is essential that the network node server be defined in the end node's *network node server list.* A network node server list defines up to five possible network node servers for the local end node. The network node server list is defined in the network attributes of the end node. The server chosen by the end node will be the first node varied on, configured to support CP-CP sessions, in the network node server list. If

an adjacent network node is not added to the end node's server list then a CP-CP session cannot be established between the an EN and NN.

EN Without CP-CP Session Capability: An EN without CP-CP session capability receives less support by an NN server, since there is no means for transfer of certain APPN information (which can only occur via a CP-CP session). Thus an EN without a CP-CP session has some reduced capabilities in comparison with an EN with a CP-CP session. These are described as follows:

1. It cannot send its local location names to the NN server. Thus, the server cannot automatically register local location names of the adjacent ENs (without CP-CP sessions) and must configure them manually (those other than the CP name).
2. It cannot send (or receive) a search request to locate a remote location name.

   Thus, the EN (without a CP-CP session) must send a BIND to the NN server; it cannot send a search request if there is no CP-CP session because Directory Services (DS) requires a CP-CP session in order to exchange information. Once the NN server receives a BIND from the EN (without CP-CP session) it will then perform directory services in the same way as for an EN with CP-CP session described above. Once the NN server has found the remote location specified in the BIND, it calculates the preferred route, attaches the RSCV and forwards the BIND through the network transparently to the end node.

   If a remote location resides in an EN without a CP-CP session then that EN cannot receive a search request. In that case, the network node server must have (manually) configured the local locations of the EN, so that it can respond to the search request on behalf of the EN.

- Finally, the NN server will not return the RSCV to the EN (without CP-CP session) when the preferred route is calculated (as it will if the EN has a CP-CP session).

  An NN server will perform preferred route calculation for the EN but only in a limited capacity. When the server receives a BIND from the EN, it will calculate the preferred route **from the NN server** to the target node (based on the COS table specified by the EN when it sends the BIND). It will then attach the RSCV to the BIND and forward it to its destination.

  Thus, an EN without a CP-CP session **must** travel the route via its network node server, while an EN with a CP-CP session may travel via an alternative route which does not include the server which originally calculated the route.

Note: As is the case with an EN that wants to establish a CP-CP session with an NN, the EN that wants services from a NN but does not want a CP-CP session, also uses the network node server list in network attributes. In this scenario the list of five NN server names is still valid. The names should be entered in the order that they are to be preferred as a server. When a session request is issued on the EN for a remote location that the EN does not have knowledge of or the owning control point is not adjacent or active, the EN will forward the BIND to the NN that is both highest in the list and available for use. This means that the controller description that described the adjacent NN must be varied-on pending or higher in order to be used. For example, if the NN that is listed first is varied-off or is in a recovery-pending state, and the NN that is in the second position is varied-on pending, the controller that describes the NN in position two will be selected for activation. Furthermore, if the controller description that describes the NN in position one is made available for use then that one will be selected for activation on subsequent attempts.

Low Entry Network (L.E.N.) Node Support

A L.E.N. node may reside at the end of an APPN network and cannot perform intermediate routing.

Note: L.E.N. node is a new name for what was called End Node in the IBM System/36 implementation of APPN.

A L.E.N. node cannot specify an adjacent NN as a network server. However an adjacent NN will perform the role of a network node server in a way similar to that of an EN without CP-CP session capability. That

is, the network node server will perform route selection services for the L.E.N. node in the same way as for an EN without CP-CP capability.

A L.E.N. node must define all remote locations with which it is likely to communicate. The adjacent NN must manually configure the local location names of the L.E.N. node, other than the CP name. (A CP name, for a L.E.N. node, is simply the name which is used to identify it to the network. It does not imply a CP-CP capability.)

Comparison of the Capabilities of APPN Nodes

The following two tables compare the functions of different node types in an APPN network. The question to be asked in the table in Figure 74 is whether the function specified is performed in the node. An "L" in the table indicates that the function is provided but only to a limited extent.

	NODE TYPE			
FUNCTION	NN	EN(CP=Y)	EN(CP=N)	LN
Uses some NN Server Functions	N	Y	Y	Y
Intermediate Routing	Y	N	N	N
Multiple CP-CP Session Partners	Y	N	N	N
Directory Services	Y	Y	Y	N
Directory Database	Y	Y	Y	N
Perform Network Searches	Y	L	N	N
Auto Enter CP Name	Y	Y	Y	N
Register Locations with Server	N/A	Y	N	N
Topology Routing Services	Y	Y	Y	N
Topology Database	Y	L	L	N
Route Selection	Y	L	L	N
Class of Service	Y	Y	Y	N
Transmission Priority	Y	Y	Y	N
RSCV returned from Server to EN	N/A	Y	N	N
Auto device create and activate	Y	Y	Y	N
Auto Disconnect on Switched Lines	L	L	Y	Y
APPN Generic Routing	Y	Y	Y	N

Figure 74. Tables Comparing NNs, ENs and L.E.N.s. The first table illustrates the functions which each of the APPN node types may perform. An "L" in a column indicates that the node has a function but only in a limited way.

Appendix C. Independent (X.25) Packet Networks Compared to SNA

The debate about "X.25 networks" versus SNA networks is many years old and the issues have been often explored. However, the recent introduction of T2.1 node support into SNA combined with the availability of very reasonably priced ES/9370 processors and 3745 communication controllers changes many of the parameters of the debate. It is therefore appropriate to summarise the issues as they are now.

In many ways it is very difficult to debate this question because of the wide differences in concept and scope between the archetypical "X.25 packet network" and SNA networks. Nevertheless, an isolated packet network (X.25 network) can be used as a wide area network transport vehicle within an SNA network and, in many environments, SNA networks can directly replace isolated packet networks by performing equivalent (or enhanced) function for the user. It is reasonable that the question of which approach is appropriate in any particular situation should be asked by users.

Any discussion of this nature must start by emphasising that X.25 is an interface to a packet switched data network and not a specification of how such a network should function internally. The basic concepts of X.25 are reviewed in Appendix D, "An Introduction to X.25 Concepts" on page 169. In fact, by using the IBM product "X.25 Interconnect" (XI), SNA subarea networks are able to "be" X.25 networks. That is, they are able to carry traffic between X.25 interfaces and look to the attaching "DTE" like any other X.25 connection. XI is described in *Integrating X.25 Function into SNA Networks, GG24-3052-1.* Of course, SNA networks have for many years been able to use X.25 networks as a transport medium for SNA traffic and to interface to "X.25-non-SNA" devices for such purposes as OSI connection etc.

There is an enormous difference in scope between SNA on the one hand and an isolated packet network (IPN) on the other. Also, there are wide differences between the operational characteristics of networks provided by different suppliers and therefore it is difficult to compare characteristics in a general way.

Characteristics

Scope

The scope of SNA covers the whole range of user devices and programs (terminals, small computers, large computers, programs, links, packet switches, etc.). "X.25 networks" only address the wide area data transport component.

For example SNA includes the detailed specification of the interface an application program has to the communications system supporting it and the detailed specification of the end-to-end protocols necessary for program-to-program operation. X.25 is just a link interface to a wide area network.

Another way of saying this is that SNA has all of the functions described in the OSI model (all seven layers), and in addition covers network directory and network management functions, yet unspecified in OSI.

Recent Developments in SNA

Many people have commented that SNA is changing to become more like X.25. As described throughout this book, SNA now has an any-to-any capability called T2.1 node support which from some perspectives looks a lot like X.25. To begin with it provides a very similar function. In addition the dynamic allocation of LFSIDs on the T2.1 node interface is conceptually very close to the dynamic allocation of logical channels in X.25. (See "Addressing Mechanism" on page 140 and "Components of the X.25 Interface" on

page 170.) It is also true that SNA has evolved very quickly away from the strongly host-based architecture that it was originally to become much more autonomous and independent.

The Network Interface

The key difference between the T2.1 node interface used in SNA and the X.25 interface is that the X.25 interface was designed to isolate the end user device from an independent and separate transport network, the SNA interface was designed to integrate it. SNA specifies the internal operation of a T2.1 node in very great detail, in X.25 the internal operation of the DTE is unspecified. Many of the other differences flow from this basic design. In concept however, at the link level, the two interfaces are very similar.

The T2.1 node interface can operate over many different physical facilities such as:

- X.21, V.35 or V.24 links at any speed supported by the physical medium.
- LANs both Token-Ring and Ethernet™.
- The host channel interface.
- X.25 connections.

In addition, SNA is able to use multipoint SDLC link control which can significantly reduce the cost of attachment of distributed devices. X.25 can only use link connection and must be point-to-point and this increases the attachment cost.

X.25 is defined (recommended) by the CCITT and has been available in a usable form since 1980. Today, there is a substantial amount of equipment on the market that makes use of the X.25 network interface and it is accepted as one means of network attachment within OSI. In addition SNA networks can use X.25 for transport, attach to X.25 "native mode" for OSI connection and provide X.25 connections (via XI).

The T2.1 node interface is also "open" and is available for any vendor to use for attaching products. Many products are available which use this attachment. However, because the networking function of the T2.1 node interface is so new, there is little non-SNA equipment on the market which makes use of it.

Network Function

Even considered in isolation as a transport network, SNA has two major features not found in most X.25 packet networks.

1. SNA has a built-in network directory. Resources are known by their names regardless of their location within the network. Names are resolved into real addresses by a control point (in the subarea network this is VTAM), when communication is established.

   Even though some X.25 networks have re-routing of calls, etc. The directory function is a major advantage of SNA. If a resource fails in one location and is backed up in another location, an SNA network will automatically restart communications with the active resource regardless of where it is in the network. X.25 networks typically use a "telephone number" style of address which must be changed by the end user every time a resource moves to a new location on the network.

2. An integrated network management and security function which allows the network to be managed as a single logical entity. The effect of this is discussed later.

Philosophy

Internal Operation

Perhaps the most basic philosophy of the internal operation of an SNA network is to make the network as stable as possible. SNA has extensive congestion and flow controls and in addition it has a strong emphasis on error detection and recovery. Errors are retried at the logic level on which they are detected until it is reasonably certain that another retry will not be of use. Only then are higher layers informed about the error.

This is a cost trade-off issue. End devices in SNA do not have end-to-end recovery at the network level. This is the OSI layer 4 class 4 function. This function is very expensive to implement in an end device and is unnecessary if the network is stable. In the case of using some X.25 public networks within SNA it was found necessary to invent such a protocol (called ELLC) to accommodate for the effects of running SNA protocols over an unstable network.

Many, (but not all), packet networks available on the market use a philosophy which says "just send the packet, do not flow control, do not have any expensive internal stability mechanisms, if a problem such as link or node congestion occurs just throw away the data and tell the end user...." This is a perfectly legitimate way to run a network, but it means that the end user devices have to have additional functions to make communication stable (OSI layer 4 class 4).

End-User Isolation

The basic difference between SNA and "X.25 packet networks" is that SNA was designed as a private networking system for use within a single organisation or conglomerate. The X.25 interface was designed by the CCITT (a committee of telephone companies) as the interface to a public data network. In the public data network case, it is essential that the end user be unable to know about or affect the operation of the internals of the network and the network administration does not want to know about the operation of the end users' devices. In a private network the reverse is true. Central operation and management are often key to the successful implementation of a private network and it is essential that the network interface should not be a barrier to this.

Consequences

Scope of Cost Optimisation

When someone builds an isolated network, they attempt to optimise the cost of that network. In other words they try to minimise the network cost usually including the network nodes, a network management processor and the links over which it runs. When this happens, things that appear "too hard" tend to be "thrown over the wall." That is, functions that are difficult or costly to perform within the network itself are defined to be the responsibility of the end user. This is a characteristic of the X.25 CCITT specification and also an operational characteristic of many such networks. Cost is taken out of the network and added to the end user.

In SNA, it is assumed that the end user will evaluate the total cost of the whole network, including end devices and processors etc. The major reason for the higher (apparent) cost of the network part of SNA is that many functions are done in the network in order to reduce the cost of the attaching device. It is assumed that there will be many more end user devices than communications nodes and therefore building additional cost in the nodes to save cost in the end devices is justified.

If a cost comparison of the wide area component of an SNA network is made with almost any X.25 packet network, the cost of the packet network will be lower. However, if the total cost, including processor attachment, end user devices, LAN connections and network management is considered, SNA network will nearly always be substantially lower in cost.

Interface Cost

Processor attachment

X.25 only allows attachment through a serial communications link. Serial link attachment is a very inefficient method of attachment to a large host processor and therefore is not supported by most types of large hosts. If a separated packet network is used for example with an IBM 3090 mainframe, it is necessary to have a front end 3745 for attachment of the processor to the network. But right beside it in the machine room there will be one or more "X.25 packet switches." There is considerable duplication of function and redundant equipment in this configuration.

LAN connection

Although it is used for things like TCP/IP, X.25 does not cover the protocols to be used across a local area network. This means that all local area network attachment must be through a gateway process (translating addresses etc.) and network management and control stop and start again on the other side.

Multidrop link connection

One of the major sources of additional cost in X.25 is due to its inability to have a multipoint link attachment. This is discussed in *Integrating X.25 Function into SNA Networks, GG24-3052.* If the user want to have several devices in the same location then there must be several links to the network (and several interfaces to the network), one for each device. With a multipoint protocol such as SDLC one link can be sufficient.

In today's environment, multipoint links servicing multiple different locations are rapidly becoming unattractive as communications cost relationships change.

The X.25 interface

Three aspects of the X.25 interface design itself deserve comment.

1. Packet length. Most packet networks are designed to work with relatively short (128-byte) packets. But the load on a connecting processor is almost the same regardless of packet or block size. Thus 1000 byte blocks take about the same number of cycles to process as each 128 byte block within EDP equipment.
2. The process of packetisation. In X.25 when a logical block of data is broken up into packets for presentation to the network, the packets are formed into a logical block by setting the "more data" bit in the packet header of all packets except the last in each logical block. This is a similar process to "segmentation" in SNA.
3. Because it is better for the network, there is a rule that a packet with the more data bit set on MUST be full (be of the maximum size). This seems logical. However, within an attaching processor, buffers may be of many sizes for may reasons not associated with the packet size. When data is formed into packets in many products (such as the 3745) the data must be copied into blocks of just the right length. This copying in a high-speed logic processor such as the 3745 takes a considerable number of processor cycles.

   In SNA, a similar function called segmentation exists but it allows segments within a logical group to be of any length up to a maximum. This allows the processor to break up the data on logical break points such as buffer boundaries and thus saves considerable processor load.

   Some 30% of the "extra" cycles taken by the X.25 interface product NPSI within the 3745 can be attributed to data copying caused by the more data bit rule.

Network Management

When an isolated packet network is used as a vehicle for interconnection of a user's devices and processors

there are in fact two networks to consider.[58] First there is the packet network. Then there is the network consisting of the user's devices and connected together by virtual circuits within the packet network.

There are therefore two communications networks to manage. If X.25 communication is used within SNA, then the SNA network management task is usually NOT reduced. In fact in many ways it becomes more complex. In addition to the SNA network management task, then there is the management of the packet network underneath it. Further there is communication between the people who operate the packet network and those who operate the SNA network. These people are trained differently and have different technical language and different concepts.

When an SNA network is run "over the top" of an "X.25 network" the X.25 network component appears as a "black hole" within the SNA network management structure. Unless the user writes unique code (perhaps within NetView/PC) to integrate the network management of both networks, resolution of the causes of error is extremely difficult.

All this can be reduced to a cost, that of having two separate sets of network management people and systems. But not all of the cost is readily apparent. The additional delay in problem resolution caused by having separate and incompatible network management systems will show itself in network availability and end-user service.

Non-SNA Transport

SNA networks have extensive X.25 function as mentioned elsewhere in this book. However, many non-SNA end user processors and devices exist on the market which were designed to operate over X.25 packet networks. Many of these devices were designed to take account of the unstable nature of some of these networks.

If a user has predominantly non-SNA traffic and most of the devices are equipped with X.25 interfaces, then an SNA network will probably provide more function than that user is able to take advantage of. In this case an isolated packet network may be an economic alternative. But in the case where there is significant SNA traffic, the X.25 functions of the SNA network (XI, NPSI etc.) will make a single integrated SNA solution optimal.

Economics

The major force between the development of "packet" style wide area networks (including the WAN part of SNA) was the desire to optimise the use of long lines. In the early 1970's, long communications lines were seen as low in capacity and very high in cost. One of the main justifications for developing packet network technology was to allow multiple users to share the use of a high- cost, low-capacity, link.

But the same technology that produces economically priced packet switches also reduces the cost of long communications lines. The slow introduction of digital techniques to the internal operations of the world's telephone networks is finally bearing fruit. Digital exchanges and multiplexors and fibre optic transmission continue to bring down the cost of data transmission very rapidly. Long lines are no longer low in capacity and high in cost. They are high in capacity and the cost is reducing very rapidly.

Packet networks do other things than just share costs. For example, they allow for alternate routing and dynamic adjustment to network failures (but this is coming soon in the telephone network, too). However, many people believe that packet network technology is no longer justified on an economic basis. In some countries, it is true today that line cost saving alone cannot justify the installation of an isolated packet network.

---

[58] In fact the data processing network consisting of programs that process data and interconnected by logical connections can be considered as a third network running over the top of the other two.

The above remarks are true also for the WAN component of SNA. However, SNA integrates its WAN networking function with its LAN interface and with its interfaces to large computers. The changing telecommunications environment is causing an evolution in function placement within SNA devices. But even if line cost reduces to zero that would not remove the need for the SNA network (but it would change the placement of functions within SNA devices quite radically).

This is not a simple discussion. The traditional data network applications (30 characters in, 100 out for banking transactions, 50 characters in and 800 out for screen keyboard applications) are no longer the majority of the data traffic. Users are finding new applications which take advantage of the large data storage and transmission capacity now available have recently become economically justified. (The "paperless" office, which takes received mail, scans it, and then processes the document from then on as an image, is already a reality in some organisations.) User requirement for data transmission are changing very rapidly indeed and data networks need to be flexible enough to adjust to new user requirements as they arise.

Many people believe that the traditional isolated "X.25 packet network" was a technology of the 1970's, implemented in the 1980's but obsolete in the 1990's.[59] These people would say that in the 1990s, as the cost of communication bandwidth reduces and users implement applications requiring very high bandwidth and fast response times, ISDN and MAN (Metropolitan Area Network) technologies will dominate.

Conclusion

SNA networks offer a single, integrated communication solution for an entire enterprise. Private packet networks only address the wide area communication part.

If network cost is considered in isolation from the system costs, then the private packet network will nearly always appear to be lower. If the total system cost including network, end user devices, processor attachments, personnel etc. are considered the SNA solution will almost always be proven to be the most economic. In addition, the availability of a single integrated SNA network is very high and there is an intangible (but very real) cost associated with network availability.

[59] This appears in general to be true, but some applications of public packet switched networks (such as very infrequent access to remote hosts from ASCII devices) will remain economic for some years. This appears to be an exception.

Appendix D. An Introduction to X.25 Concepts

The CCITT recommendation X.25 describes an interface between a user and a packet-switched data network.[60] **It must be emphasized that the recommendation ONLY describes the interface between the user and the network. It describes the operation of the interface in great detail and it specifies what services should be available to a user device operating on such an interface. It DOES NOT say anything at all about how the network should operate internally.** Thus the phrase "X.25 network" cannot say anything meaningful about how the network operates, only that the network can support a certain type of connection between users.

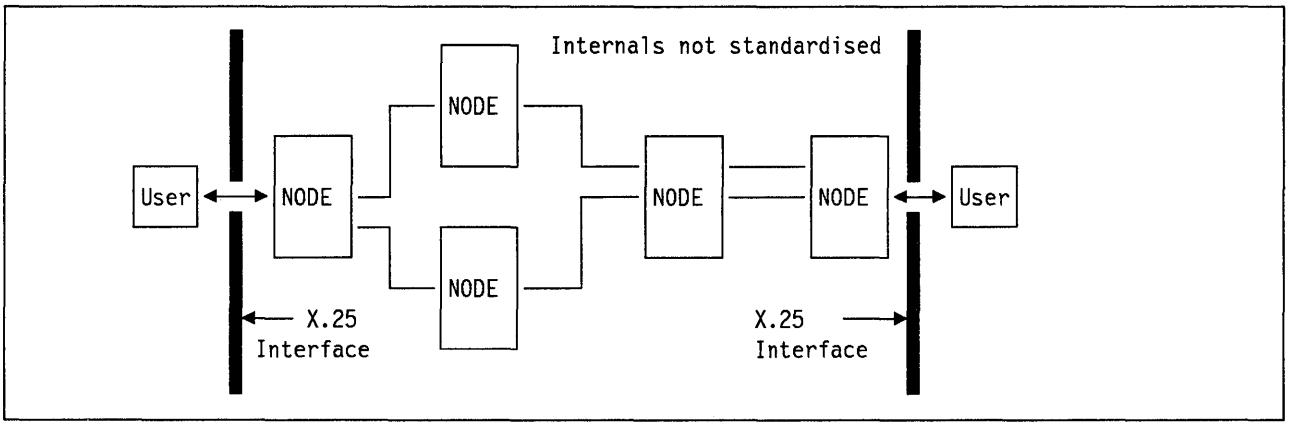


Figure 75. Schematic of a Packet Network. Each "node" (or "switch") receives a packet in full before routing it on towards it destination user. Not shown is the additional equipment necessary to manage the network.

Networks with X.25 capability are often represented as a "cloud". The "cloud" representation is useful since it emphasizes that the end users need have no concern about how the network operates internally.

The user presents data to the network in short blocks called packets. The function of the network is to deliver these packets to another user (destination) attached to the network. Packets are delivered, without change[61] to the data, in the same sequence as they were presented to the network and without being stored on any intermediate external storage medium (disk). Communication is synchronous in that both communicating users must be present at the same time for communication to take place.

The network is made up of nodes (also called "switches" or "Data Switching Exchanges" (DSEs)) that are connected together by data communication links (see Figure 75). In most networks each node is a specially designed computer but in addition there is almost always a larger network host (often a general purpose computer) to service the network nodes.

One of the parameters of network design in a packet network is the maximum length of a packet. Short packets make for fast transit times through the network. However they take considerably more resource both in the end user device and within the network. In X.25 the "universal compromise" packet size which every network must support is 128 bytes.

The central concept of X.25 is that of a "virtual circuit," that is, a circuit is completed between two communicating end users in the same way as a circuit connects two people who use a telephone. The circuit is

---

[60] This appendix was abstracted from "Integrating X.25 Function into Systems Network Architecture Networks" (GG24-3052)

[61] It is possible, albeit inefficient, for the length of a packet to be different at different ends of the network, but the data is unchanged.

called "virtual" because it does not use dedicated resource within the network, but the logical path between a pair of end users is nevertheless dedicated to communication between this pair of users and no others. Virtual circuits (VCs) can be "switched" or "permanent". A switched virtual circuit (SVC) is sometimes named a "Virtual Call" (VC) in analogy with the telephone system and to confuse the innocent. The abbreviation for "permanent virtual circuit" is "PVC."

Components of the X.25 Interface

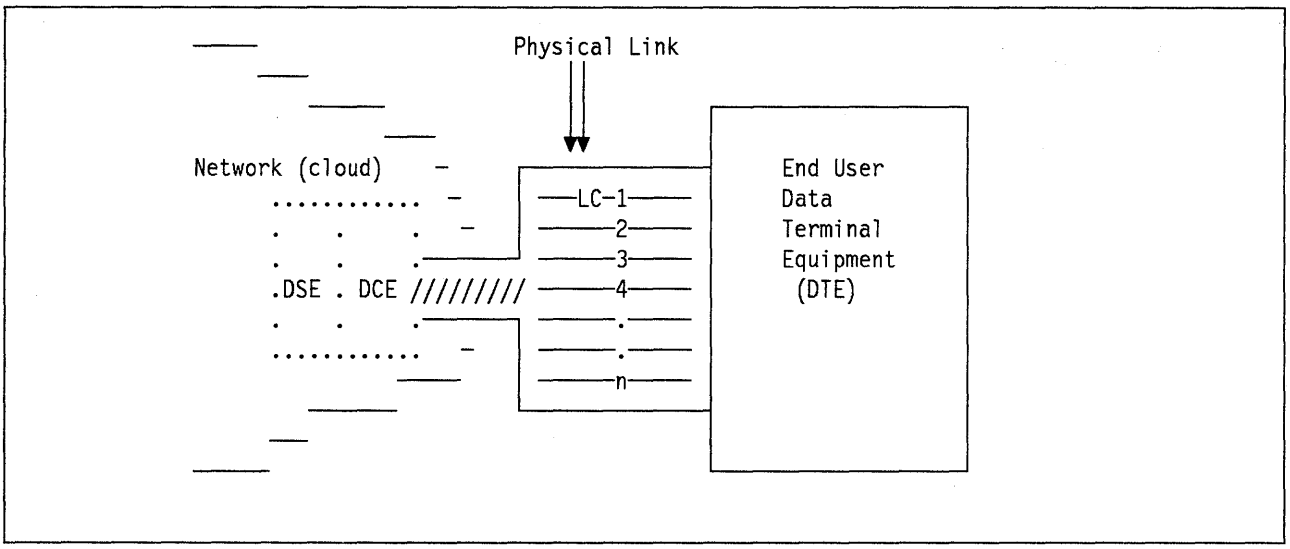


Figure 76. Elements of the X.25 Interface

The central concepts of the X.25 interface are illustrated in Figure 76. The more important of the concepts can be described as follows:

Packet The packet is the unit of data sent between the end user (called the DTE) and the network. Its maximum size is called the "packet size" for this interface. A packet that is not full is sent as a short packet. **It is never "padded out" to the full packet length.**

Packet Header Packets are transmitted with a three (or optionally four) byte header which is not included in the packet length. The header contains information about the packet type, a "more data" bit which allows for the grouping of packets into logical records and a 12-bit field which identifies which communication (virtual circuit) this packet belongs to.

Logical Channel Within the one physical link to the network the end user (DTE) may have many communications (virtual circuits) with other end users in the network. Since the packet header does not contain the network address of the destination, there needs to be some mechanism of identifying where this packet is to be sent. That method is the logical channel number (and the logical channel group number). A logical channel is simply a reference number to identify which virtual circuit this packet belongs to. Another way of saying it is that many logical communications can take place over a single physical circuit by multiplexing the physical circuit into many logical channels. The Logical Channel Number (LCN) is the identifier which is used to distinguish which virtual circuit a particular packet belongs to.

Virtual Circuit A Virtual Circuit is simply an association between two logical channels, one at each communication end point. See Figure 77 on page 171. A packet that is sent with a logical channel number of 3 by the user at interface A will be received at interface B with a logical channel number of 6 in its packet header. The detail of how the communication is achieved is left to the designers of the packet network.

DTE Data Terminal Equipment. This means anything that is a user of the X.25 interface. It could be a simple terminal, or a protocol conversion device or a large mainframe CPU. The interface is the same and it is treated in the same way.

DCE Data Circuit terminating Equipment. This is the network end of the link from the user. In various different contexts it can be the modem interface or the interface at the Network Node. In the X.25 context it normally refers to the Network node.

DSE Data Switching Exchange. This term is not often used since the process of data switching is hidden from users of the network. It refers to the logical switching process within a node.

Physical Link This is the link between the user and the network. In IBM "jargon" it is called the MCH (MultiChannel Link).

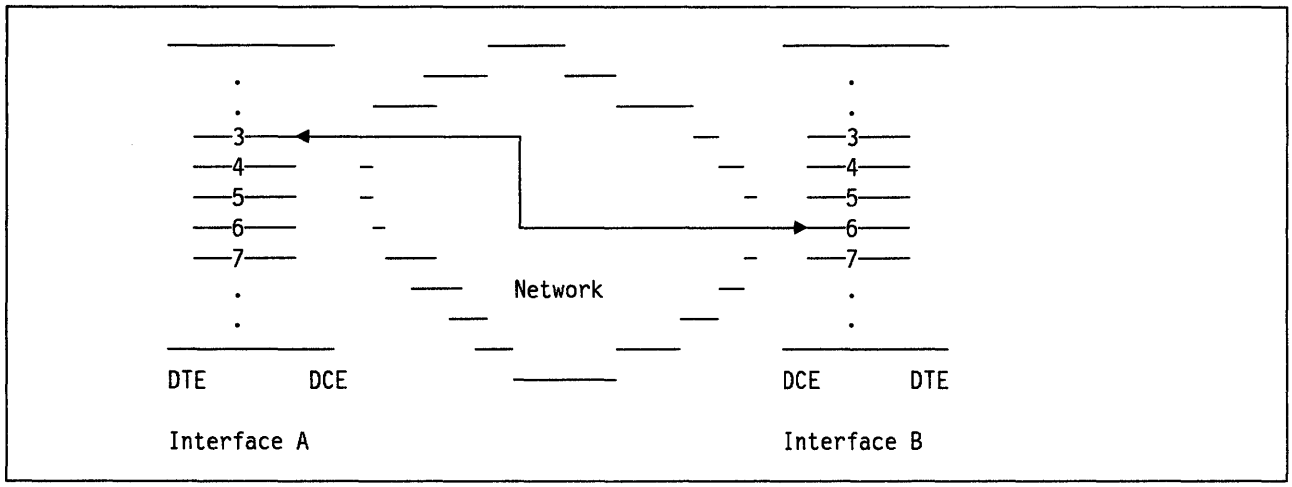


Figure 77. Virtual Circuit. Two logical channels (number 3 on interface A and number 6 on interface B) are communicating with one another. This pairing of logical channels is called a Virtual Circuit.

Logical Structure of the X.25 Interface

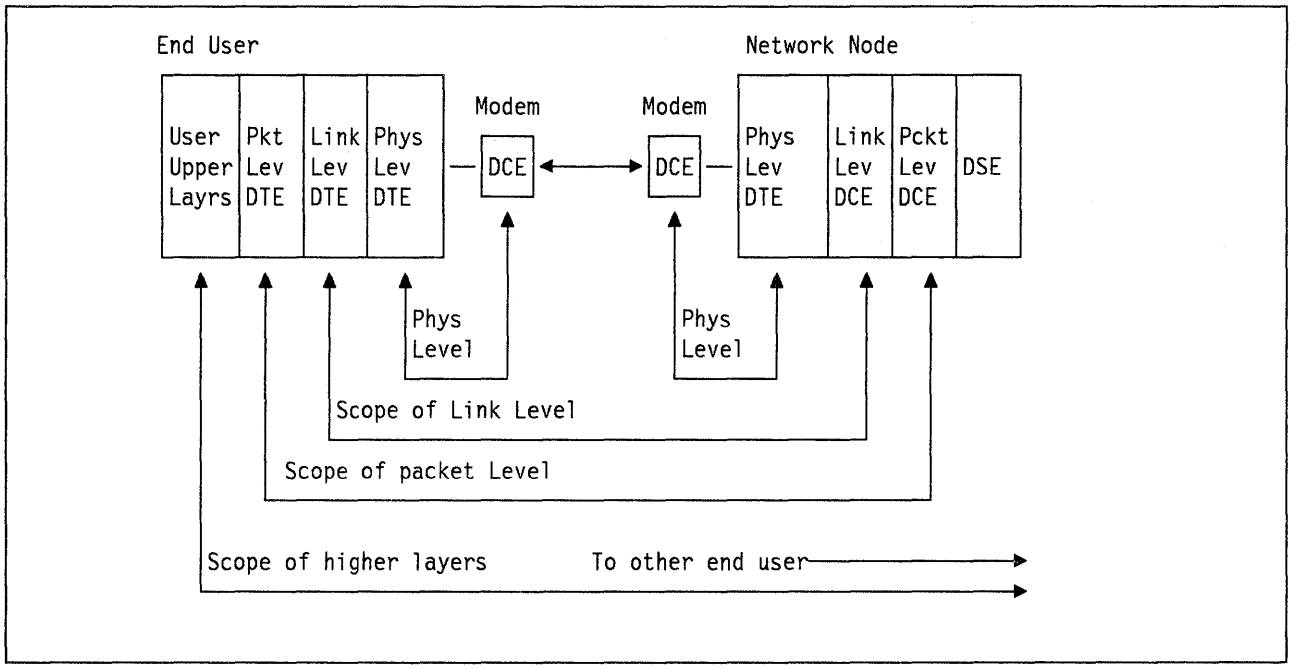


Figure 78. DTE and DCE Relationships. The scope of each layer is shown here. The relationship between DTE and DCE at each layer should be noted.

Logically, the interface operates in three "layers" called physical, link, and packet levels.

Physical Level. This is the lowest logical layer and provides electrical connection between the DTE and the link to the network. If a modem is used then the physical layer is specified by the CCITT recommendations V.24 or V.35.[62] If a "digital" link is to be used then recommendation X.21 is appropriate. This layer only provides for the sending and receiving of bits and for physical compatibility of plugs and cables and for voltage tolerances and signal timings etc.

Link Level. This is the "line control" or "link protocol" which passes frames of data to and from the network. Link control takes a queue of frames at each of the DTE and the DCE and is responsible for transferring them in as efficient a way as possible from one to the other. Link control takes frames from the packet level and delivers them to the packet level at the other side. Link level does not care about the content of the frames and does not know that the link is multiplexed among many virtual circuits. Link control is responsible for error detection and recovery (retransmission) between DTE and DCE. Inherent in the means of operation of the link control there is a "data flow control" enforced through a rotating "send window". However, it is not generally used for controlling flow except in conditions of extreme congestion because the DTE-DCE flow control is done at the packet level.

The link control protocol used in X.25 is a version (subset of options) of the international standard data link control "HDLC". The subset is the "Asynchronous Balanced Mode" and is called LAPB in the jargon of X.25.

[62] V.24 and V.35 are also sometimes called "X.21bis". There are minor differences in technical detail between these V-series recommendations and X.21bis but the terms are often used interchangeably. "A rose by any other name..."

Packet Level. The packet level protocol performs basically two functions. The first is to multiplex the physical channel provided by the link control into a number (up to 4096) of logical channels. The second is to provide a flow control between the DTE and DCE to provide an even delivery of data per logical connection (virtual circuit).

There is no error recovery in the packet level. (Though some network suppliers add one.) The scope of packet level is the same as for link level (from the nearest node in the network to the user) so error recovery is solely done at the link level.

Setting Up a Virtual Circuit

There are two kinds of virtual circuit:

Permanent Virtual Circuits are set up by the network administration and consist of a permanent relationship between a particular logical channel on one particular interface (or port) and another logical channel on a different interface somewhere else in the network. A permanent virtual circuit is always there (provided the network is operating) and the end user DTE requires nothing special to start using it.

Switched virtual circuits must be requested by the user and are then set up by the network (provided that the network has resources available). A special packet type (the "call" packet) is sent to the network on a logical channel that is not already in use. (There are strict rules for selection of the next logical channel to be used.) The call packet contains the address of the other user to which connection is requested. There is a standard format for addresses to be used by the network. The network then finds and sets up a path to the other user and selects a logical channel on which to notify the other user of the "Incoming Call". This incoming call (in reality the call packet sent by the other user with a different logical channel number and some fields changed) is presented to the other user which can then accept or reject the call. A "Call Accepted" packet is sent to accept the call or a "clear" packet is sent to refuse the call. "Clear" is the usual way of terminating an SVC connection (i.e., hanging up).

Packet Types

There are surprisingly few different packet types in X.25.

Data Packets. These are the units in which data is sent through the network. Packets can be of different (maximum) sizes and in fact different logical channels on the same physical link can use different packet sizes. In the X.25 recommendation, packet sizes of 32, 64, 128, 256, 512, 1024, 2048 and 4096 bytes (octets) are allowed. However, every X.25 network must support a packet size of 128 in addition to the other sizes it may optionally support. For example, the network is capable of changing the packet size during transit so that a DTE at one end may have a packet size of 128 bytes and at the other end a packet size of 256.

Qualified Data Packets. These are the same as data packets but have the "Q" bit in the packet header set on. This is simply a way of identifying a logically different kind of data from what is carried in a data packet and can be used by the DTEs in any way they like.

Interrupt Packets. These packets carry only a small amount of data (in the CCITT 1980 version 1 byte) and are given priority in transit through the network. Whereas Data and Q packets are always delivered to the destination DTE in the order that they were presented to the network, interrupt packets are presented as soon as possible. It is up to the user to decide the meaning of data (if any) in an interrupt packet.

Control Packets. These are packets like "Call", "Reset" and "Clear" that are used to communicate information between the DTEs and the network.

The PAD Function

While the recommendation X.25 deals with the attachment of synchronous link connected devices to a packet network, there are three other CCITT recommendations that are usually understood to be present when the phrase "X.25 network" is used. These relate to the attachment of "dumb" "ASCII TWX" devices to the packet network. Three Recommendations are involved. These are X.3, X.28 and X.29. They are often referred to as "the triple-x pad" or just the "PAD."

These asynchronous terminals typically send a character on the communications line every time a key on the keyboard is depressed. That is, transmission is one character at a time and the assembly of characters into logical records and blocks is done by the program to which the terminal is communicating. These types of terminal often do not use error checking at all but it is increasingly common to use a method of error control called "echo-plexing". In "echo-plexing", a character sent from the device to the computer is sent back in the opposite direction and the terminal compares the returned character with the character sent in. If they are the same then there has been no error.

For efficiency reasons it is obvious that sending characters across the packet network as one character per packet will not be very attractive. There is then a need for a function that assembles characters from ASCII devices into groups to be sent through the network in packets. "PAD" stands for "Packet Assembler/Disassembler" and the PAD device performs this function.

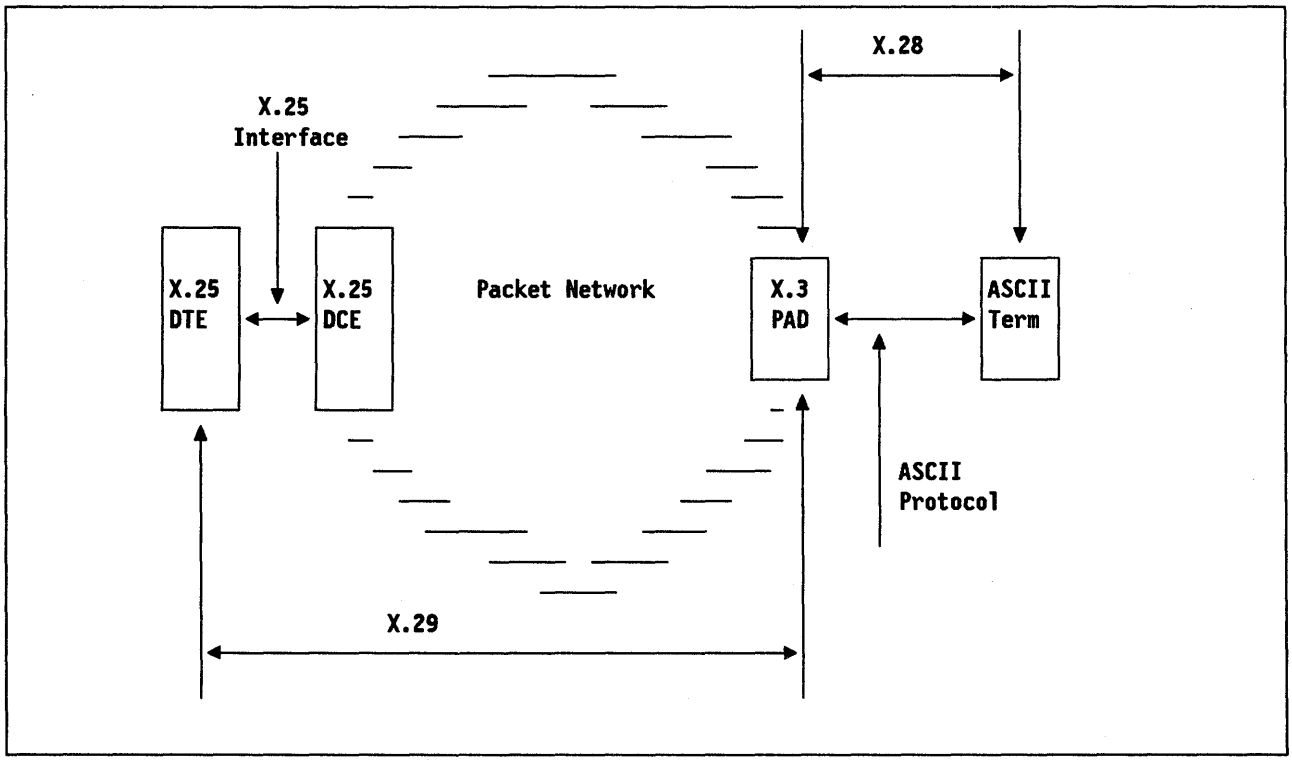


Figure 79. The ASCII "PAD". Scope of the three CCITT recommendations X.3, X.28 and X.29 is shown.

As characters are sent by the terminal, they are "echoed" (if needed) and assembled into a buffer. When certain criteria (such as a packet being filled or a pre-determined control character being entered) are met, then the PAD will forward the new packet on a virtual circuit to the partner on the other side of the network. The three recommendations cover the following functions:

X.3 This covers the internal operation of the PAD, the control parameters that can be entered from the terminal to customize the operation of the PAD and things like echo and flow control etc.

X.28 This covers the data link interface between the terminal and the PAD. Access can be via the "PSTN" (Public Switched Telephone Network), or via leased line, or through TELEX etc.

X.29 This covers the exchange of control messages between the P-DTE (Packet Mode DTE) and the PAD. These messages are used for example to set up PAD parameters in order to either relieve the terminal operator from the chore of setting up PAD parameters or to prevent the terminal user from changing parameters without the permission of the host application.

In the diagram Figure 79 on page 174 the PAD is shown as a function or a part of the network. This is not necessarily true. PADs are most often implemented as external devices as shown in the diagram Figure 80.

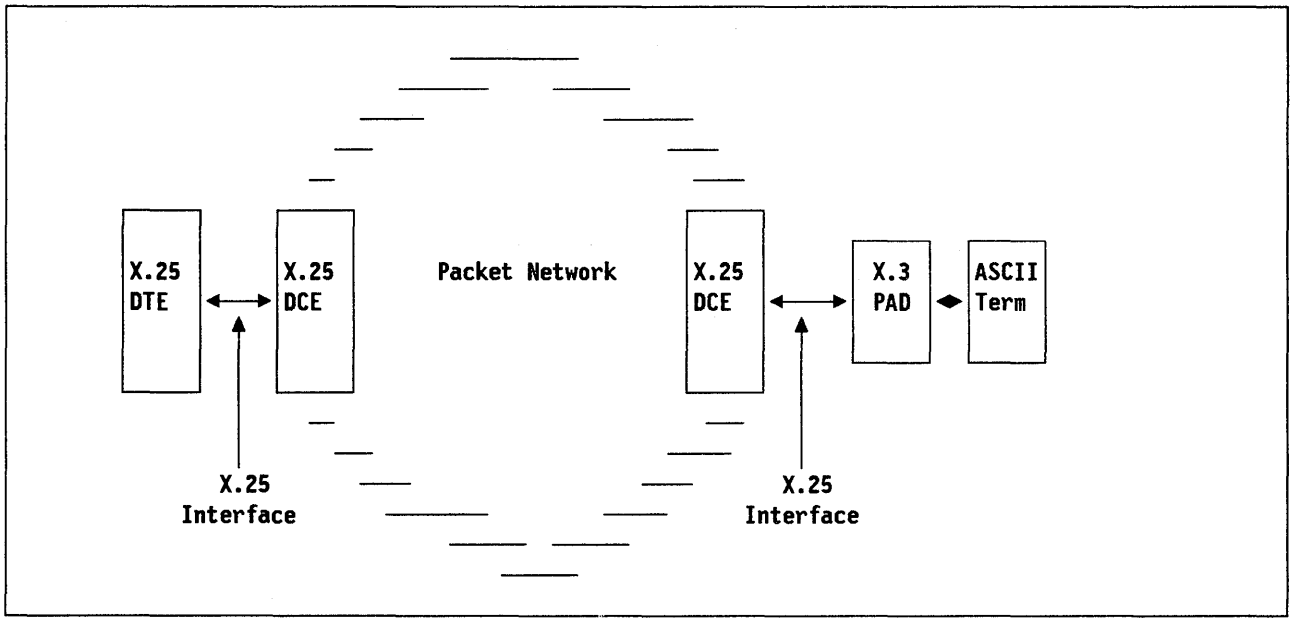


Figure 80. An "External" PAD. While described as a network function the PAD is most often implemented as an external piece of equipment.

Bibliography

Note: Manual numbers prefixed by the letter "L" refer to manuals which contain detailed internals information about the referenced product. This licensed material may only be ordered by licensees of the product concerned.

VTAM Publications

The following paragraphs briefly describe the library of manuals for VTAM V3R3.

VTAM Network Implementation Guide (SC31-6404)

This manual contains information about how to install VTAM, how to define a network to VTAM, and how to test your network definitions. This manual should be used in conjunction with the *VTAM Resource Definition Reference.*

VTAM Resource Definition Reference (SC31-6412)

This manual contains the VTAM definition statements and start options. It also has information on the operands of NCP definition statements that affect VTAM. To assist VM users, there is an appendix describing VSCS start options. This book should be used in conjunction with the *VTAM Installation and Resource Definition Guide.*

VTAM V3R3 Storage Estimates (SK2T-2025, a diskette)

These diskettes aid in estimating the storage requirements for VTAM. They contain an interactive program that guides the user step-by-step through the process for estimating storage.

VTAM Customization (LY43-0046)

This manual enables a system programmer to customize VTAM and tune it for better performance. It discusses modifying VTAM messages; modifying VTAM USS commands, installation exit routines, and replaceable modules; and tuning VTAM.

VTAM Operation (SC31-6408)

This manual enables a system programmer to prepare a "run book" for a VTAM network. This book also serves as a reference manual to programmers and operators requiring detailed information about specific operator commands.

VTAM Messages and Codes (SC31-6405)

This manual contains, in alphanumerical order, all messages and codes issued by VTAM. These messages include VTAM messages for network operators, TSO/VTAM messages for network operators, TSO/VTAM messages for terminal users, USS messages for terminal users, and VSCS messages. This manual can be inserted into the operating system messages manual, if desired, or used as a stand-alone manual.

VTAM Programming (SC31-6409)

This manual describes how to use VTAM macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain. Also included is a dictionary of VTAM macroinstructions.

VTAM Programming for LU 6.2 (SC31-6410)

This manual describes the VTAM LU 6.2 programming interface for host application programs. This manual applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this manual, however.)

VTAM Diagnosis (LY43-0042)

This manual assists system programmers in identifying a VTAM problem, classifying it, and collecting information about the problem in preparation for calling the IBM Support Center. The information collected includes traces, dumps, and other problem documentation.

VTAM V3R3 Data Areas for MVS (LY43-0043)

VTAM V3R3 Data Areas for VM (LY43-0045)

These manuals describe VTAM data areas and can be used to read a VTAM dump. They are intended for IBM programming service representatives and customer personnel who are diagnosing problems with VTAM.

VTAM Reference Summary (LY43-0047)

This manual is designed as a quick reference for system programmers. This manual contains selected reference information that includes VTAM and VSCS commands, VTAM definition statements, VTAM start options, VTAM macroinstructions, and VTAM and VSCS trace formats.

VTAM V3R2 Publications

The following manuals pertain to VTAM V3R2. Several of these manuals also contain information about VTAM V3R1.2 for VM and VSE, V3R1.1 for MVS and VM, and V3R1 for VSE.

VTAM Installation and Resource Definition (SC23-0111)

VTAM Customization (LY30-5614)

VTAM Directory of Programming Interfaces for Customers (GC31-6403)

VTAM Operation (SC23-0113)

VTAM Messages and Codes (SC23-0114)

VTAM Programming (SC23-0115)

VTAM Programming for LU 6.2 (SC30-3400)

VTAM Diagnosis (LY30-5601)

VTAM Data Areas for MVS (LY30-5592)

VTAM Data Areas for VM (LY30-5593)

VTAM Data Areas for VSE (LY30-5594)

VTAM Reference Summary (LY30-5600)

VTAM V3R2 Enhancements (LD35-0270)

VTAM Version 3 for VM/9370 (SD35-0271)

Other Network Program Products Publications

The following list shows the cross-product manuals for VTAM, NetView, NCP, SSP, and NetView/PC.

Network Program Products Planning and Reference (SC31-6811)

Network Program Products Bibliography and Master Index (GC31-6815)

NetView Publications

The following list shows the publications associated with Release 3 of the NetView program.

Learning about NetView (SK2T-0292)

NetView Installation and Administration Guide (SC31-6018)

NetView Administration Reference (SC31-6014)

NetView Tuning Guide (SC30-3481)

NetView Customization: Overview (SC31-6016)

NetView Customization: Using PL/I, C, and Assembler (SC31-6037)

NetView Customization: Writing Command Lists (SC31-6015)

NetView Directory of Programming Interfaces for Customers (GC31-6022)

NetView Operation Primer (SC31-6020)

NetView Operation (SC31-6019)

NetView Command Summary (SX75-0026)

NetView Problem Determination and Diagnosis (LY43-0001)

NetView Resource Alerts Reference (SC31-6024)

NetView Problem Determination Supplement for Management Services Major Vectors 0001 and 0025 (LD21-0023)

NCP Version 4 Publications

The following list shows the publications for NCP Version 4.

NCP, SSP, and EP Generation and Loading Guide (SC30-3348)

NCP Migration Guide (SC30-3252)

NCP, SSP, and EP Resource Definition Guide (SC30-3349)

NCP, SSP, and EP Resource Definition Reference (SC30-3254)

NCP Customization Guide (LY30-5571)

NCP Customization Reference (LY30-5612)

SSP Customization (LY43-0021)

NCP, SSP, and EP Messages and Codes (SC30-3169)

NCP, SSP, and EP Diagnosis Guide (LY30-5591)

NCP and EP Reference (LY30-5569)

NCP and EP Reference Summary and Data Areas (LY30-5570)

NCP Version 5 Publications

The following list shows the publications for NCP Version 5.

NCP, SSP, and EP Generation and Loading Guide (SC30-3348)

NCP Migration Guide (SC30-3440)

NCP, SSP, and EP Resource Definition Guide (SC30-3447)

NCP, SSP, and EP Resource Definition Reference (SC30-3448)

NCP Customization Guide (LY30-5606)

NCP Customization Reference (LY30-5607)

SSP Customization (LY43-0021)

NCP, SSP, and EP Messages and Codes (SC30-3169)

NCP, SSP, and EP Diagnosis Guide (LY30-5591)

NCP and EP Reference (LY30-5605)

NCP and EP Reference Summary and Data Areas (LY30-5603)

Glossary

This glossary defines important SNA, NCP, and VTAM abbreviations and terms.[63]

It includes information from the following sources:

- *IBM Vocabulary for Data Processing, Telecommunications, and Office Systems*, GC20-1699.
- *American National Dictionary for Information Processing*. These entries are identified by an asterisk (*).
- Draft proposals and working papers under development by the International Standards Organization, Technical Committee 97, Subcommittee 1. These are identified by the symbol (TC97).
- *CCITT Sixth Plenary Assembly Orange Book, Terms and Definitions* and working documents published by the Consultative Committee on International Telegraph and Telephone of the International Telecommunication Union, Geneva, 1980. These are preceded by the symbol (CCITT/ITU).

  Published sections of the *ISO Vocabulary of Data Processing*, developed by the International Standards Organization, Technical Committee 97, Subcommittee 1 and published sections of the *ISO Vocabulary of Office Machines*, developed by subcommittees of ISO Technical Committee 95. These are preceded by the symbol (ISO).

ACB. (1) In VTAM, application control block. (2) In NCP, adapter control block.

ACF/NCP. Advanced Communications Function for the Network Control Program. Synonym for *NCP*.

ACF/SSP. Advanced Communications Function for the System Support Programs. Synonym for *SSP*.

ACF/TAP. Advanced Communications Function for the Trace Analysis Program. Synonym for *TAP*.

ACF/TCAM. Advanced Communications Function for the Telecommunications Access Method. Synonym for *TCAM*.

ACF/VTAM. Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for *VTAM*.

ACF/VTAME. Advanced Communications Function for the Virtual Telecommunications Access Method Entry. Synonym for *VTAME*.

acquire. (1) For a VTAM application program, to initiate and establish a session with another logical unit (LU). The acquire process begins when the application program issues a macroinstruction. See also *accept*. (2) To take over resources that were formerly controlled by an access method in another domain, or to resume control of resources that were controlled by this domain but released. Contrast with *release*. See also *resource takeover*.

adaptive session pacing. Synonym for *adaptive session-level pacing*.

adaptive session-level pacing. A form of session-level pacing in which session components exchange pacing windows that may vary in size during the course of a session. This allows transmission to adapt dynamically to variations in availability and demand of buffers on a session by session basis. Session pacing occurs within independent stages along the session path according to local congestion at the intermediate nodes. Synonymous with *adaptive session pacing*. See *pacing*, *session-level pacing*, and *virtual route pacing*.

adjacent nodes. Two nodes that are connected by one or more data links with no intervening nodes.

Advanced Program-to-Program Communication (APPC). A synonym for logical unit (LU) 6.2 and its implementations.

alias name. A name defined in a host used to represent a logical unit name, logon mode table name, or class-of-service name in another network. This name is defined to a name translation program when the alias name does not match the real name. The alias name translation program is used to associate the real and alias names.

alias name translation facility. A function for converting logical unit names, logon mode table names, and class-of-service names used in one network into equivalent names to be used in another network. Available with NetView or NCCF licensed programs.

API. Application program interface.

APPC. Advanced Program-to-Program Communication.

application control block (ACB). A control block that links an application program to VSAM or VTAM.

[63] This glossary is abridged from the glossary in *VTAM Operation*, (SC23-0113-4).

application program. (1) A program written for or by a user that applies to the user's work. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

application program interface (API). (1) The formally defined programming language interface between an IBM system control program or licensed program and its user. (2) The interface through which an application program interacts with an access method. In VTAM, it is the language structure used in control blocks so that application programs can reference them and be identified to VTAM.

automatic logon. (1) A process by which VTAM automatically creates a session-initiation request to establish a session between two logical units (LUs). The session will be between a designated primary logical unit (PLU) and a secondary logical unit (SLU) that is neither queued for nor in session with another PLU. See also *controlling application program* and *controlling logical unit*. (2) In VM, a process by which a virtual machine is initiated by other than the user of that virtual machine. For example, the primary VM operator's virtual machine is activated automatically during VM initialization.

available. In VTAM, pertaining to a logical unit that is active, connected, enabled, and not at its session limit.

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

BIND. In SNA, a request to activate a session between two logical units (LUs). See also *session activation request*. Contrast with *UNBIND*.

BIU segment. In SNA, the portion of a basic information unit (BIU) that is contained within a path information unit (PIU). It consists of either a request/response header (RH) followed by all or a portion of a request/response unit (RU), or only a portion of an RU.

boundary function. (1) A capability of a subarea node to provide protocol support for attached peripheral nodes, such as: (a) interconnecting subarea path control and peripheral path control elements, (b) performing session sequence numbering for low-function peripheral nodes, and (c) providing session-level pacing support. (2) The component that provides these capabilities. See also *boundary node*, *network addressable unit (NAU)*, *peripheral path control*, *subarea node*, and *subarea path control*.

boundary node. (1) A subarea node with boundary function. See *subarea node* (including illustration). See also *boundary function*. (2) The programming component that performs FID2 (format identification type 2) conversion, channel data link control, pacing, and channel or device error recovery procedures for a locally attached station. These functions are similar to those performed by a network control program for an NCP-attached station.

bracket protocol. In SNA, a data flow control protocol in which exchanges between the two LU-LU half-sessions are achieved through the use of brackets, with one LU designated at session activation as the first speaker and the other as the bidder. The bracket protocol involves bracket initiation and termination rules. See also *bidder* and *first speaker*.

BSC. Binary synchronous communication.

CDRM. Cross-domain resource manager.

CDRSC. Cross-domain resource.

channel adapter. A communication controller hardware unit used to attach the controller to a System/360 or a System/370 channel.

channel-attached. Pertaining to the attachment of devices directly by System 370 input-output channels to a host processor.

channel-attachment major node. (1) A major node that includes an NCP that is channel-attached to a data host. (2) A major node that may include minor nodes that are the line groups and lines that represent a channel attachment to an adjacent (channel-attached) host. (3) In VM or VSE operating systems, a major node that may include minor nodes that are resources (host processors, NCPs, line groups, lines, SNA physical units and logical units, cluster controllers, and terminals) attached through a communication adapter.

CID. Communication identifier.

class of service (COS). In SNA, a designation of the path control network characteristics, such as path security, transmission priority, and bandwidth, that apply to a particular session. The end user designates class of service at session initiation by using a symbolic name that is mapped into a list of virtual routes, any one of which can be selected for the session to provide the requested level of service.

CLIST. Command list.

closedown. The deactivation of a device, program, or system. See *cancel closedown*, *orderly closedown*, and *quick closedown*.

cluster controller. A device that can control the input/output operations of more than one device connected to it. A cluster controller may be controlled by a program stored and executed in the unit; for example, the IBM 3601 Finance Communication Controller. Or

it may be controlled entirely by hardware; for example, the IBM 3272 Control Unit.

CMC. Communication management configuration.

CNM. Communication network management.

command. (1) A request from a terminal for the performance of an operation or the execution of a particular program. (2) In SNA, any field set in the transmission header (TH), request header (RH), and sometimes portions of a request unit (RU), that initiates an action or that begins a protocol; for example: (a) Bind Session (session-control request unit), a command that activates an LU-LU session, (b) the change-direction indicator in the RH of the last RU of a chain, (c) the virtual route reset window indicator in a FID4 transmission header. See also *VTAM operator command.*

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

communication controller. A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit; for example, the IBM 3725 Communication Controller. It manages the details of line control and the routing of data through a network.

composite end node (CEN). A group of nodes made up of a single type 5 node and its subordinate type 4 nodes that together support type 2.1 protocols. To a type 2.1 node, a CEN appears as one end node. For example, NCP and VTAM act as a composite end node.

configuration. (1) (TC97) The arrangement of a computer system or network as defined by the nature, number, and the chief characteristics of its functional units. The term may refer to a hardware or a software configuration. (2) The devices and programs that make up a system, subsystem, or network. (3) In CCP, the arrangement of controllers, lines, and terminals attached to an IBM 3710 Network Controller. Also, the collective set of item definitions that describe such a configuration.

contention. A situation in which two logical units (LUs) that are connected by an LU 6.2 session both attempt to allocate the session for a conversation at the same time. The control operator assigns "winner" and "loser" status to the LUs so that processing may continue on an orderly basis. The contention loser requests permission from the contention winner to allocate a conversation on the session, and the contention winner either grants or rejects the request. See also *bidder.*

control block. (ISO) A storage area used by a computer program to hold control information.

control point (CP). (1) A system services control point (SSCP) that provides hierarchical control of a group of nodes in a network. (2) A control point (CP) local to a specific node that provides control of that node, either in the absence of SSCP control (for type 2.1 nodes engaged in peer to peer communication) or to supplement SSCP control.

control program (CP). The VM operating system that manages the real processor's resources and is responsible for simulating System/370s for individual users.

controlling application program. In VTAM, an application program with which a secondary logical unit (other than an application program) is automatically put in session whenever the secondary logical unit is available. See also *automatic logon* and *controlling logical unit.*

controlling logical unit. In VTAM, a logical unit with which a secondary logical unit (other than an application program) is automatically put in session whenever the secondary logical unit is available. A controlling logical unit can be either an application program or a device-type logical unit. See also *automatic logon* and *controlling application program.*

COS. Class of service.

CP. (1) Control program. (2) Control point.

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

cross-domain link. (1) A subarea link connecting two subareas that are in different domains. (2) A link physically connecting two domains.

cross-domain resource (CDRSC). A resource owned by a cross-domain resource manager (CDRM) in another domain but known by the CDRM in this domain by network name and associated CDRM.

cross-domain resource manager (CDRM). In VTAM, the function in the system services control point (SSCP) that controls initiation and termination of cross-domain sessions.

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

cross-network session. An LU-LU or SSCP-SSCP session whose path traverses more than one SNA network.

cryptographic. Pertaining to the transformation of data to conceal its meaning. See also *encipher* and *decipher.*

data link control (DLC) layer. In SNA, the layer that consists of the link stations that schedule data transfer over a transmission medium connecting two nodes and perform error control for the link connection. Examples of data link control are SDLC for serial-by-bit link con-

nection and data link control for the System/370 channel.

definite response (DR). In SNA, a value in the form-of-response-requested field of the request header. The value directs the receiver of the request to return a response unconditionally, whether positive or negative, to that request. Contrast with *exception response* and *no response.*

definition statement. (1) In VTAM, the statement that describes an element of the network. (2) In NCP, a type of instruction that defines a resource to the NCP. See also *macroinstruction.*

dependent LU. Any logical unit (LU) that receives an ACTLU over a link. Such LUs can act only as secondary logical units (SLUs) and can have only one LU-LU session at a time. Contrast with *independent LU.*

destination logical unit (DLU). The logical unit to which data is to be sent. Contrast with *origin logical unit (OLU).*

device-type logical unit. In VTAM, a logical unit that has a session limit of 1 and usually acts as the secondary end of a session. It is typically a logical unit (LU) in an SNA terminal, such as a 3270. It could be the primary end of a session, for example, the logical unit representing the Network Routing Facility (NRF) logical unit.

direct activation. In VTAM, the activation of a resource as a result of an activation command specifically naming the resource. See *automatic activation.* Contrast with *indirect activation.*

direct deactivation. In VTAM, the deactivation of a resource as a result of a deactivation command specifically naming the resource. See also *automatic deactivation.* Contrast with *indirect deactivation.*

disabled. In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is temporarily not ready to establish LU-LU sessions. An initiate request for a session with a disabled logical unit (LU) can specify that the session be queued by the SSCP until the LU becomes enabled. The LU can separately indicate whether this applies to its ability to act as a primary logical unit (PLU) or a secondary logical unit (SLU). See also *enabled* and *inhibited.*

DLU. Destination logical unit.

domain. (1) An access method, its application programs, communication controllers, connecting lines, modems, and attached terminals. (2) In SNA, a system services control point (SSCP) and the physical units (PUs), logical units (LUs), links, link stations, and all the associated resources that the SSCP has the ability to control by means of activation requests and deactivation requests. See *system services control point domain* and *type 2.1 node control point domain.* See also *single-domain network* and *multiple-domain network*

DR. (1) In NCP and CCP, dynamic reconfiguration. (2) In SNA, definite response.

element. (1) A field in the network address. (2) The particular resource within a subarea identified by the element address. See also *subarea.*

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

enabled. In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is now ready to establish LU-LU sessions. The LU can separately indicate whether this prevents it from acting as a primary logical unit (PLU) or as a secondary logical unit (SLU). See also *disabled* and *inhibited.*

end node. A type 2.1 node that does not provide any intermediate routing or session services to any other node. For example, APPC/PC is an end node. See *composite end node*, *node*, and *type 2.1 node.*

ER. (1) Explicit route. (2) Exception response.

exception request (EXR). In SNA, a request that replaces another message unit in which an error has been detected.

exception response (ER). In SNA, a value in the form-of-response-requested field of a request header (RH). An exception response is sent only if a request is unacceptable as received or cannot be processed. Contrast with *definite response* and *no response.* See also *negative response.*

explicit route (ER). In SNA, the path control network elements, including a specific set of one or more transmission groups, that connect two subarea nodes. An explicit route is identified by an origin subarea address, a destination subarea address, an explicit route number, and a reverse explicit route number. Contrast with *virtual route (VR).* See also *path* and *route extension.*

EXR. Exception request.

extended network addressing. The network addressing system that splits the address into an 8-bit subarea and a 15-bit element portion. The subarea portion of the address is used to address host processors or communication controllers. The element portion is used to permit processors or controllers to address resources.

first speaker. In SNA, the LU-LU half-session defined at session activation as: (1) able to begin a bracket without requesting permission from the other LU-LU half-session to do so, and (2) winning contention if both

half-sessions attempt to begin a bracket simultaneously. Contrast with *bidder*. See also *bracket protocol*.

flow control. In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize the rate of flow of message units, with minimum congestion in the network; that is, to neither overflow the buffers at the receiver or at intermediate routing nodes, nor leave the receiver waiting for more message units. See also *adaptive session-level pacing*, *pacing*, *session-level pacing*, and *virtual route pacing*.

formatted system services. A portion of VTAM that provides certain system services as a result of receiving a field-formatted command, such as an Initiate or Terminate command. Contrast with *unformatted system services (USS)*. See also *field-formatted*.

frame. (1) The unit of transmission in some local area networks, including the IBM Token-Ring Network. It includes delimiters, control characters, information, and checking characters. (2) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures.

gateway. The combination of machines and programs that provide address translation, name translation, and system services control point (SSCP) rerouting between independent SNA networks to allow those networks to communicate. A gateway consists of one gateway NCP and at least one gateway SSCP.

gateway NCP. An NCP that performs address translation to allow cross-network session traffic. The gateway NCP connects two or more independent SNA networks. Synonymous with *gateway node*.

gateway node. Synonym for *gateway NCP*.

gateway SSCP. An SSCP that is capable of cross-network session initiation, termination, takedown, and session outage notification. A gateway SSCP is in session with the gateway NCP; it provides network name translation and assists the gateway NCP in setting up alias network addresses for cross-network sessions.

half-session. In SNA, a component that provides function management data (FMD) services, data flow control, and transmission control for one of the sessions of a network addressable unit (NAU). See also *primary half-session* and *secondary half-session*.

host node. A node providing an application program interface (API) and a common application interface. See *boundary node*, *node*, *peripheral node*, *subarea host node*, and *subarea node*. See also *boundary function* and *node type*.

host processor. (1) (TC97) A processor that controls all or part of a user application network. (2) In a network, the processing unit in which the data communication access method resides.

IMS. Information Management System/Virtual Storage. Synonymous with *IMS/VS*.

IMS/VS. Information Management System/Virtual Storage. Synonym for *IMS*.

inactive. Describes the state of a resource that has not been activated or for which the VARY INACT command has been issued. Contrast with *active*. See also *inoperative*.

independent LU. A logical unit (LU) that does not receive an ACTLU over a link. Such LUs can act as primary logical units (PLUs) or secondary logical units (SLUs) and can have one or more LU-LU sessions at a time. Contrast with *dependent LU*.

inhibited. In VTAM, pertaining to a logical unit (LU) that has indicated to its system services control point (SSCP) that it is not ready to establish LU-LU sessions. An initiate request for a session with an inhibited LU will be rejected by the SSCP. The LU can separately indicate whether this applies to its ability to act as a primary logical unit (PLU) or as a secondary logical unit (SLU). See also *enabled* and *disabled*.

initiate. A network services request sent from a logical unit (LU) to a system services control point (SSCP) requesting that an LU-LU session be established.

interconnected networks. SNA networks connected by gateways.

interconnection. See *SNA network interconnection*.

intermediate routing node (IRN). In SNA, a subarea node with intermediate routing function.

intermediate SSCP. An SSCP along a session initiation path that owns neither of the LUs involved in a cross-network LU-LU session.

interpret table. In VTAM, an installation-defined correlation list that translates an argument into a string of eight characters. Interpret tables can be used to translate logon data into the name of an application program for which the logon is intended.

IRN. Intermediate routing node.

ISTATUS. In VTAM and NCP, a definition specification method for indicating the initial status of resources. See also *indirect activation*.

link. In SNA, the combination of the link connection and the link stations joining network nodes; for example: (1) a System/370 channel and its associated protocols, (2) a serial-by-bit connection under the control of Synchronous Data Link Control (SDLC). A

link connection is the physical medium of transmission. A link, however, is both logical and physical. Synonymous with *data link*. See Figure 81 on page 187.

link station. (1) In SNA, the combination of hardware and software that allows a node to attach to and provide control for a link. (2) In VTAM, a named resource within a subarea node that represents another subarea node that is attached by a subarea link. In the resource hierarchy, the link station is subordinate to the subarea link.

local address. In SNA, an address used in a peripheral node in place of an SNA network address and transformed to or from an SNA network address by the boundary function in a subarea node.

local area network (LAN). (1) A network in which a set of devices are connected to one another for communication and that can be connected to a larger network. See also *token ring*. (2) A network in which communications are limited to a moderately sized geographic area such as a single office building, warehouse, or campus, and which do not generally extend across public rights-of-way. Contrast with *wide area network*.

logical unit (LU). In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points (SSCPs). An LU can support at least two sessions—one with an SSCP and one with another LU—and may be capable of supporting many sessions with other LUs. See also *network addressable unit (NAU)*, *peripheral LU*, *physical unit (PU)*, *system services control point (SSCP)*, *primary logical unit (PLU)*, and *secondary logical unit (SLU)*.

logical unit (LU) services. In SNA, capabilities in a logical unit to: (1) receive requests from an end user and, in turn, issue requests to the system services control point (SSCP) in order to perform the requested functions, typically for session initiation; (2) receive requests from the SSCP, for example to activate LU-LU sessions via Bind Session requests; and (3) provide session presentation and other services for LU-LU sessions. See also *physical unit (PU) services*.

logical unit (LU) 6.2. A type of logical unit that supports general communication between programs in a distributed processing environment. LU 6.2 is characterized by (1) a peer relationship between session partners, (2) efficient utilization of a session for multiple transactions, (3) comprehensive end-to-end error processing, and (4) a generic application program interface (API) consisting of structured verbs that are mapped into a product implementation.

logmode table. Synonym for *logon mode table*.

log off. To request that a session be terminated.

logoff. In VTAM, an unformatted session termination request.

log on. To initiate a session.

logon. In VTAM, an unformatted session initiation request for a session between two logical units. See *automatic logon* and *simulated logon*. See also *session-initiation request*.

logon mode. In VTAM, a subset of session parameters specified in a logon mode table for communication with a logical unit. See also *session parameters*.

logon mode table. In VTAM, a set of entries for one or more logon modes. Each logon mode is identified by a logon mode name. Synonymous with *logmode table*.

LU. Logical unit.

LU type. In SNA, the classification of an LU-LU session in terms of the specific subset of SNA protocols and options supported by the logical units (LUs) for that session, namely:

The mandatory and optional values allowed in the session activation request.

The usage of data stream controls, function management headers (FMHs), request unit (RU) parameters, and sense codes.

Presentation services protocols such as those associated with FMH usage.

LU types 0, 1, 2, 3, 4, 6.1, 6.2, and 7 are defined.

LU-LU session. In SNA, a session between two logical units (LUs) in an SNA network. It provides communication between two end users, or between an end user and an LU services component.

major node. In VTAM, a set of resources that can be activated and deactivated as a group. See *node* and *minor node*.

management services. In SNA, one of the types of network services in control points (CPs) and physical units (PUs). Management services are the services provided to assist in the management of SNA networks, such as problem management, performance and accounting management, configuration management and change management. See also *configuration services*, *maintenance services*, *network services*, and *session services*.

mode name. A symbolic name for a set of session characteristics. For LU 6.2, a mode name and a partner LU name together define a group of parallel sessions having the same characteristics.

multipoint link. A link or circuit interconnecting several link stations. Synonymous with *multidrop line*. Contrast with *point-to-point link*.

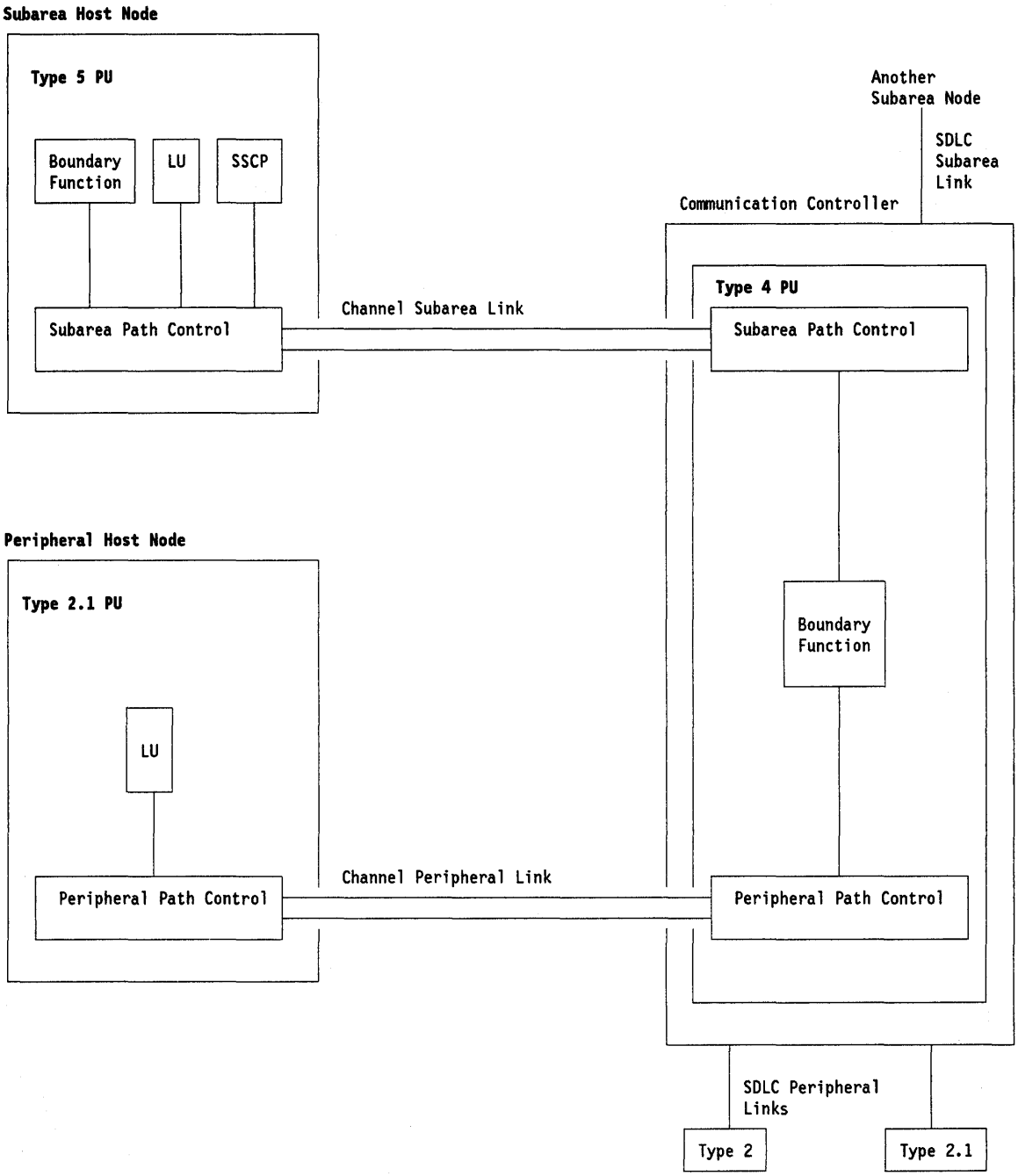


Figure 81. Links and Path Controls

name translation. In SNA network interconnection, converting logical unit names, logon mode table names, and class-of-service names used in one network into equivalent names to be used in another network. This function can be provided through NetView and invoked by a gateway system services control point (SSCP) when necessary. See also *alias name*.

native network. The network in which a gateway NCP's resources reside.

NAU. Network addressable unit.

NCP. (1) Network Control Program (IBM licensed program). Its full name is Advanced Communications Function for the Network Control Program. Synony-

mous with *ACF/NCP*. (2) Network control program (general term).

NCP major node. In VTAM, a set of minor nodes representing resources, such as lines and peripheral nodes, controlled by a network control program. See *major node*.

negative response (NR). In SNA, a response indicating that a request did not arrive successfully or was not processed successfully by the receiver. Contrast with *positive response*. See *exception response*.

NetView. A system 370-based IBM licensed program used to monitor a network, manage it, and diagnose its problems.

network address. In SNA, an address, consisting of subarea and element fields, that identifies a link, a link station, or a network addressable unit. Subarea nodes use network addresses; peripheral nodes use local addresses. The boundary function in the subarea node to which a peripheral node is attached transforms local addresses to network addresses and vice versa. See *local address*. See also *network name*.

network addressable unit (NAU). In SNA, a logical unit, a physical unit, or a system services control point. It is the origin or the destination of information transmitted by the path control network. Each NAU has a network address that represents it to the path control network. See also *network name*, *network address*, and *path control network*.

network control (NC). In SNA, an RU category used for requests and responses exchanged for such purposes as activating and deactivating explicit and virtual routes and sending load modules to adjacent peripheral nodes. See also *data flow control layer* and *session control*.

Network Control Program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Network Control Program.

network identifier (network ID). The network name defined to NCPs and hosts to indicate the name of the network in which they reside. It is unique across all communicating SNA networks.

network name. (1) In SNA, the symbolic identifier by which end users refer to a network addressable unit (NAU), a link, or a link station. See also *network address*. (2) In a multiple-domain network, the name of the APPL statement defining a VTAM application program is its network name and it must be unique across domains. Contrast with *ACB name*. See *uninterpreted name*.

Network Routing Facility (NRF). An IBM licensed program that resides in the NCP, which provides a path for messages between terminals, and routes messages over this path without going through the host processor.

network services (NS). In SNA, the services within network addressable units (NAUs) that control network operation through SSCP-SSCP, SSCP-PU, and SSCP-LU sessions. See *configuration services, maintenance services, management services*, and *session services*.

NLDM. Network Logical Data Manager.

node. (1) In SNA, an endpoint of a link or junction common to two or more links in a network. Nodes can be distributed to host processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities. See *boundary node*, *host node*, *peripheral node*, and *subarea node* (including illustration). (2) In VTAM, a point in a network defined by a symbolic name. See *major node* and *minor node*.

node name. In VTAM, the symbolic name assigned to a specific major or minor node during network definition.

node type. In SNA, a designation of a node according to the protocols it supports and the network addressable units (NAUs) that it can contain. Five types are defined: 1, 2.0, 2.1, 4, and 5. Type 1, type 2.0, and type 2.1 nodes are peripheral nodes; type 4 and type 5 nodes are subarea nodes. See also *type 2.1 node*.

non-native network. Any network attached to a gateway NCP that does not contain that NCP's resources.

no response. In SNA, a value in the form-of-response-requested field of the request header (RH) indicating that no response is to be returned to the request, whether or not the request is received and processed successfully. Contrast with *definite response* and *exception response*.

notify. A network services request that is sent by an SSCP to a logical unit (LU) to inform the LU of the status of a procedure requested by the LU.

NPDA. Network Problem Determination Application.

NRF. Network Routing Facility.

OLU. Origin logical unit.

origin logical unit (OLU). The logical unit from which data is sent. Contrast with *destination logical unit (DLU)*.

pacing. In SNA, a technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion. See

session-level pacing, send pacing, and *virtual route (VR) pacing*. See also *flow control*.

pacing group. In SNA, (1) The path information units (PIUs) that can be transmitted on a virtual route before a virtual-route pacing response is received, indicating that the virtual route receiver is ready to receive more PIUs on the route. Synonymous with *window*. (2) The requests that can be transmitted on the normal flow in one direction on a session before a session-level pacing response is received, indicating that the receiver is ready to accept the next group of requests.

packet switching. (TC97) The process of routing and transferring data by means of addressed packets so that a channel is occupied only during the transmission of a packet; upon completion of the transmission, the channel is made available for the transfer of other packets.

parallel sessions. In SNA, two or more concurrently active sessions between the same two logical units (LUs) using different pairs of network addresses. Each session can have independent session parameters.

path. (1) In SNA, the series of path control network components (path control and data link control) that are traversed by the information exchanged between two network addressable units (NAUs). See also *explicit route (ER)*, *route extension*, and *virtual route (VR)*. (2) In VTAM when defining a switched major node, a potential dial-out port that can be used to reach that node. (3) In the NetView/PC program, a complete line in a configuration that contains all of the resources in the service point command service (SPCS) query link configuration request list.

path control (PC). The function that routes message units between network addressable units (NAUs) in the network and provides the paths between them. It converts the BIUs from transmission control (possibly segmenting them) into path information units (PIUs) and exchanges basic transmission units (BTUs) and one or more PIUs with data link control. Path control differs for peripheral nodes, which use local addresses for routing, and subarea nodes, which use network addresses for routing. See *peripheral path control* and *subarea path control*. See also *link*, *peripheral node*, and *subarea node*.

path control (PC) layer. In SNA, the layer that manages the sharing of link resources of the SNA network and routes basic information units (BIUs) through it. See also *BIU segment*, *blocking of PIUs*, *data link control layer*, and *transmission control layer*.

path control (PC) network. In SNA, the part of the SNA network that includes the data link control and path control layers. See *SNA network* and *user application network*. See also *boundary function*.

path information unit (PIU). In SNA, a message unit consisting of a transmission header (TH) alone, or of a TH followed by a basic information unit (BIU) or a BIU segment. See also *transmission header*.

PC. (1) Path control. (2) Personal Computer. Its full name is the IBM Personal Computer.

peripheral host node. A node that provides an application program interface (API) for running application programs but does not provide SSCP functions and is not aware of the network configuration. The peripheral host node does not provide subarea node services. It has boundary function provided by its adjacent subarea. See *boundary node*, *host node*, *node*, *peripheral node*, *subarea host node*, and *subarea node*. See also *boundary function* and *node type*.

peripheral LU. In SNA, a logical unit representing a peripheral node.

peripheral node. In SNA, a node that uses local addresses for routing and therefore is not affected by changes in network addresses. A peripheral node requires boundary-function assistance from an adjacent subarea node. A peripheral node is a physical unit (PU) type 1, 2.0, or 2.1 node connected to a subarea node with boundary function within a subarea. See *boundary node*, *host node*, *node*, *peripheral host node*, *subarea host node*, and *subarea node*. See also *boundary function* and *node type*.

peripheral path control. The function in a peripheral node that routes message units between units with local addresses and provides the paths between them. See *path control* and *subarea path control*. See also *boundary function*, *peripheral node*, and *subarea node*.

peripheral PU. In SNA, a physical unit representing a peripheral node.

physical unit (PU). In SNA, a type of network addressable unit (NAU). A physical unit (PU) manages and monitors the resources (such as attached links) of a node, as requested by a system services control point (SSCP) through an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links. See also *peripheral PU* and *subarea PU*.

physical unit (PU) services. In SNA, the components within a physical unit (PU) that provide configuration services and maintenance services for SSCP-PU sessions. See also *logical unit (LU) services*.

PIU. Path information unit.

PLU. Primary logical unit.

POI. Programmed operator interface.

point-to-point link. A link that connects a single remote link station to a node; it may be either switched or non-switched. Contrast with *multipoint link.*

polling. (1) * Interrogation of devices for purposes such as to avoid contention, to determine operational status, or to determine readiness to send or receive data. (2) (TC97) The process whereby stations are invited, one at a time, to transmit.

positive response. A response indicating that a request was received and processed. Contrast with *negative response.*

primary half-session. In SNA, the half-session that sends the session activation request. See also *primary logical unit.* Contrast with *secondary half-session.*

primary logical unit (PLU). In SNA, the logical unit (LU) that contains the primary half-session for a particular LU-LU session. Each session must have a PLU and secondary logical unit (SLU). The PLU is the unit responsible for the bind and is the controlling LU for the session. A particular LU may contain both primary and secondary half-sessions for different active LU-LU sessions. Contrast with *secondary logical unit (SLU).*

protocol. (1) (CCITT/ITU) A specification for the format and relative timing of information exchanged between communicating parties. (2) (TC97) The set of rules governing the operation of functional units of a communication system that must be followed if communication is to be achieved. (3) In SNA, the meanings of, and the sequencing rules for, requests and responses used for managing the network, transferring data, and synchronizing the states of network components. See also *bracket protocol.* Synonymous with *line control discipline* and *line discipline.* See also *link protocol.*

PU. Physical unit.

PU-PU flow. In SNA, the exchange between physical units (PUs) of network control requests and responses.

queued session. In VTAM, pertaining to a requested LU-LU session that cannot be started because one of the logical units (LUs) is not available. If the session-initiation request specified queuing, the system services control points (SSCPs) will record the request and later continue with the session-establishment procedure when both LUs become available.

real name. The name by which a logical unit (LU), logon mode table, or class-of-service (COS) table is known within the SNA network in which it resides.

receive pacing. In SNA, the pacing of message units that the component is receiving. See also *send pacing.*

Recommendation X.21 (Geneva 1980). A Consultative Committee on International Telegraph and Telephone (CCITT) recommendation for a general purpose interface between data terminal equipment and data circuit equipment for synchronous operations on a public data network.

Recommendation X.25 (Geneva 1980). A Consultative Committee on International Telegraph and Telephone (CCITT) recommendation for the interface between data terminal equipment and packet-switched data networks. See also *packet switching.*

request parameter list (RPL). In VTAM, a control block that contains the parameters necessary for processing a request for data transfer, for establishing or terminating a session, or for some other operation.

request unit (RU). In SNA, a message unit that contains control information, end-user data, or both.

request/response unit (RU). In SNA, a generic term for a request unit or a response unit. See also *request unit (RU)* and *response unit.*

response unit (RU). In SNA, a message unit that acknowledges a request unit; it may contain prefix information received in a request unit. If positive, the response unit may contain additional information (such as session parameters in response to Bind Session), or if negative, contains sense data defining the exception condition.

route extension (REX). In SNA, the path control network components, including a peripheral link, that make up the portion of a path between a subarea node and a network addressable unit (NAU) in an adjacent peripheral node. See also *path, explicit route (ER),* and *virtual route (VR).*

RPL. Request parameter list.

RU. Request/response unit.

SDLC. Synchronous Data Link Control.

secondary half-session. In SNA, the half-session that receives the session-activation request. See also *secondary logical unit (SLU).* Contrast with *primary half-session.*

secondary logical unit (SLU). In SNA, the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. An LU may contain secondary and primary half-sessions for different active LU-LU sessions. Contrast with *primary logical unit (PLU).*

send pacing. In SNA, pacing of message units that a component is sending. See also *receive pacing.*

service point (SP). An entry point that supports applications that provide network management for resources not under the direct control of itself as an entry point. Each resource is either under the direct control of another entry point or not under the direct control of

any entry point. A service point accessing these resources is not required to use SNA sessions (unlike a focal point). A service point is needed when entry point support is not yet available for some network management function.

session. In SNA, a logical connection between two network addressable units (NAUs) that can be activated, tailored to provide various protocols, and deactivated, as requested. Each session is uniquely identified in a transmission header (TH) by a pair of network addresses, identifying the origin and destination NAUs of any transmissions exchanged during the session. See *half-session*, *LU-LU session*, *SSCP-LU session*, *SSCP-PU session*, and *SSCP-SSCP session*. See also *LU-LU session type* and *PU-PU flow*.

session activation request. In SNA, a request that activates a session between two network addressable units (NAUs) and specifies session parameters that control various protocols during session activity; for example, BIND and ACTPU. Contrast with *session deactivation request*.

session control (SC). In SNA, (1) One of the components of transmission control. Session control is used to purge data flowing in a session after an unrecoverable error occurs, to resynchronize the data flow after such an error, and to perform cryptographic verification. (2) A request unit (RU) category used for requests and responses exchanged between the session control components of a session and for session activation and deactivation requests and responses.

session deactivation request. In SNA, a request that deactivates a session between two network addressable units (NAUs); for example, UNBIND and DACTPU. Synonymous with *generic unbind*. Contrast with *session activation request*.

session-initiation request. In SNA, an Initiate or logon request from a logical unit (LU) to a control point (CP) that an LU-LU session be activated.

session-level pacing. In SNA, a flow control technique that permits a receiver to control the data transfer rate (the rate at which it receives request units) on the normal flow. It is used to prevent overloading a receiver with unprocessed requests when the sender can generate requests faster than the receiver can process them. See also *pacing* and *virtual route pacing*.

session limit. (1) In SNA, (a) the maximum number of concurrently active LU-LU sessions a particular logical unit can support; (b) the limit that determines how many sessions may be active between two logical units (LUs) that are using LU 6.2 protocols and a given mode name. Each partner LU is allocated a minimum share of contention-winner sessions within this limit. (2) In the network control program, the maximum number of concurrent line-scheduling sessions on a non-SDLC, multipoint line.

session parameters. In SNA, the parameters that specify or constrain the protocols (such as bracket protocol and pacing) for a session between two network addressable units. See also *logon mode*.

session partner. In SNA, one of the two network addressable units (NAUs) having an active session.

session services. In SNA, one of the types of network services in the control point (CP) and in the logical unit (LU). These services provide facilities for an LU or a network operator to request that the SSCP initiate or terminate sessions between logical units. See *configuration services*, *maintenance services*, and *management services*.

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

SLU. Secondary logical unit.

SNA. Systems Network Architecture.

SNA network. The part of a user-application network that conforms to the formats and protocols of Systems Network Architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

SNA network interconnection. The connection, by gateways, of two or more independent SNA networks to allow communication between logical units in those networks. The individual SNA networks retain their independence.

SP. Service point.

SS. Start-stop.

SSCP. System services control point.

SSCP ID. In SNA, a number that uniquely identifies a system services control point (SSCP). The SSCP ID is used in session activation requests sent to physical units (PUs) and other SSCPs.

SSCP-LU session. In SNA, a session between a system services control point (SSCP) and a logical unit (LU); the session enables the LU to request the SSCP to help initiate LU-LU sessions.

SSCP-PU session. In SNA, a session between a system services control point (SSCP) and a physical unit (PU); SSCP-PU sessions allow SSCPs to send requests to and receive status information from individual nodes in order to control the network configuration.

SSCP-SSCP session. In SNA, a session between the system services control point (SSCP) in one domain and the SSCP in another domain. An SSCP-SSCP session is used to initiate and terminate cross-domain LU-LU sessions.

start option. In VTAM, a user-specified or IBM-supplied option that determines certain conditions that are to exist during the time a VTAM system is operating. Start options can be predefined or specified when VTAM is started.

subarea. A portion of the SNA network consisting of a subarea node, any attached peripheral nodes, and their associated resources. Within a subarea node, all network addressable units, links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

subarea address. In SNA, a value in the subarea field of the network address that identifies a particular subarea. See also *element address.*

subarea/element address split. The division of a 16-bit network address into a subarea address and an element address.

subarea host node. A host node that provides both subarea function and an application program interface (API) for running application programs. It provides system services control point (SSCP) functions, subarea node services, and is aware of the network configuration. See *boundary node, communication management configuration host node, data host node, host node, node, peripheral node,* and *subarea node.* See also *boundary function* and *node type.*

subarea link. In SNA, a link that connects two subarea nodes. See *channel link* and *link.*

subarea node. In SNA, a node that uses network addresses for routing and whose routing tables are therefore affected by changes in the configuration of the network. Subarea nodes can provide gateway function, and boundary function support for peripheral nodes. Type 4 and type 5 nodes are subarea nodes. See *boundary node, host node, node, peripheral node,* and *subarea host node.* See also *boundary function* and *node type.*

subarea path control. The function in a subarea node that routes message units between network addressable units (NAUs) and provides the paths between them. See *path control* and *peripheral path control.* See also *boundary function, peripheral node,* and *subarea node.*

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

subsystem. A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system.

switched line. A communication line in which the connection between the communication controller and a remote link station is established by dialing.

switched major node. In VTAM, a major node whose minor nodes are physical units and logical units attached by switched SDLC links.

switched network backup (SNBU). An optional facility that allows a user to specify, for certain types of PUs, a switched line to be used as an alternate path if the primary line becomes unavailable or unusable.

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

system services control point (SSCP). In SNA, a central location point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

system services control point (SSCP) domain. The system services control point and the physical units (PUs), logical units (LUs), links, link stations and all the resources that the SSCP has the ability to control by means of activation requests and deactivation requests.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

TERMINATE. In SNA, a request unit that is sent by a logical unit (LU) to its system services control point (SSCP) to cause the SSCP to start a procedure to end one or more designated LU-LU sessions.

TG. Transmission group.

token. A sequence of bits passed from one device to another along the token ring. When the token has data appended to it, it becomes a frame.

token ring. A network with a ring topology that passes tokens from one attaching device to another. For example, the IBM Token-Ring Network.

transmission control (TC) layer. In SNA, the layer within a half-session that synchronizes and paces session-level data traffic, checks session sequence numbers of requests, and enciphers and deciphers end-user data. Transmission control has two components: the connection point manager and session control. See also *half-session.*

transmission group (TG). In SNA, a group of links between adjacent subarea nodes, appearing as a single logical link for routing of messages. A transmission group may consist of one or more SDLC links (parallel links) or of a single System/370 channel.

transmission header (TH). In SNA, control information, optionally followed by a basic information unit (BIU) or a BIU segment, that is created and used by path control to route message units and to control their flow within the network. See also *path information unit.*

transmission priority. In SNA, a rank assigned to a path information unit (PIU) that determines its precedence for being selected by the transmission group control component of path control for forwarding to the next subarea node of the route used by the PIU.

type 2.1 node (T2.1 node). A node that can attach to an SNA network as a peripheral node using the same protocols as type 2.0 nodes. Type 2.1 nodes can be directly attached to one another using peer-to-peer protocols. See *end node*, *node*, and *subarea node.* See also *node type.*

type 2.1 node (T2.1 node) control point domain. The CP, its logical units (LUs), links, link stations, and all resources that it activates and deactivates.

UNBIND. In SNA, a request to deactivate a session between two logical units (LUs). See also *session deactivation request.* Contrast with *BIND.*

unformatted. In VTAM, pertaining to commands (such as LOGON or LOGOFF) entered by an end user and sent by a logical unit in character form. The character-coded command must be in the syntax defined in the user's unformatted system services definition table. Synonymous with *character-coded.* Contrast with *field-formatted.*

unformatted system services (USS). In SNA products, a system services control point (SSCP) facility that translates a character-coded request, such as a logon or logoff request into a field-formatted request for processing by formatted system services and translates field-formatted replies and responses into character-coded requests for processing by a logical unit. Contrast with *formatted system services.* See also *converted command.*

uninterpreted name. In SNA, a character string that a system services control point (SSCP) is able to convert into the network name of a logical unit (LU). Typically, an uninterpreted name is used in a logon or Initiate request from a secondary logical unit (SLU) to identify the primary logical unit (PLU) with which the session is requested.

USS. Unformatted system services.

virtual route (VR). In SNA, a logical connection (1) between two subarea nodes that is physically realized as a particular explicit route, or (2) that is contained wholly within a subarea node for intranode sessions. A virtual route between distinct subarea nodes imposes a transmission priority on the underlying explicit route, provides flow control through virtual-route pacing, and provides data integrity through sequence numbering of path information units (PIUs). See also *explicit route (ER)*, *path*, and *route extension.*

virtual route pacing. In SNA, a flow control technique used by the virtual route control component of path control at each end of a virtual route to control the rate at which path information units (PIUs) flow over the virtual route. VR pacing can be adjusted according to traffic congestion in any of the nodes along the route. See also *pacing* and *session-level pacing.*

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

VR. Virtual route.

VTAM. Virtual Telecommunications Access Method (IBM licensed program). Its full name is Advanced Communications Function for the Virtual Telecommunications Access Method. Synonymous with *ACF/VTAM.*

VTAM application program. A program that has opened an ACB to identify itself to VTAM and can now issue VTAM macroinstructions.

VTAM definition. The process of defining the user application network to VTAM and modifying IBM-defined characteristics to suit the needs of the user.

VTAM operator command. A command used to monitor or control a VTAM domain. See also *definition statement.*

wide area network. A network that provides data communication capability in geographic areas larger than those serviced by local area networks. Wide area networks may extend across public rights-of-way. Contrast with *local area network.*

window. (1) In SNA, synonym for *pacing group.* (2) On a visual display terminal, a small amount of

information in a framed-in area on a panel that overlays part of the panel.

X.21. See *Recommendation X.21 (Geneva 1980)*.

X.25. See *Recommendation X.25 (Geneva 1980)*.

Index

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

X

Numerics

ENTERPRISE NETWORKING WITH
SNA TYPE 2.1 NODES
GG24-3433-00

READER'S
COMMENT
FORM

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

_____

_____

_____

_____

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

What is your occupation? _____

Reader's Comment Form

Cut or Fold Along Line

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 985A, Building 657
P.O. Box 12195
Research Triangle Park
Raleigh, North Carolina 27709
U.S.A.

Fold and tape | **Please Do Not Staple** | Fold and tape

IBM®

GG24-3433-00

ENTERPRISE NETWORKING WITH
SNA TYPE 2.1 NODES

GG24-3433-00

PRINTED IN THE U.S.A.

IBM®