IBM

International Technical Support Centers

IBM 3725 NETWORK CONTROL PROGRAM TOKEN-RING INTERFACE PLANNING AND IMPLEMENTATION

GG24-3110-01

IBM 3725 Network Control Program
Token-Ring Interface
Planning and Implementation

Document Number GG24-3110-01

March, 1987

International Technical Support Center Raleigh
Department 985-C, Building 657
P. O. Box 12195
Research Triangle Park, North Carolina

First Edition (March, 1987)

This edition applies to IBM 3725 Network Control Program.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this document is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

> The information contained in this document has not been submitted to any formal IBM test and is distributed on an 'As Is' basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, International Technical Support Center Raleigh, Dept. 985-C, B657, P. O. Box 12195, Research Triangle Park, North Carolina 27709. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

ABSTRACT

This document describes considerations and requirements for attaching an IBM 3725 to an IBM Token-Ring Network using the Network Control Program Token-Ring Interface (IBM 3725 NTRI) and supporting software. Although this version of the document only considers use of the IBM 3725 Communication Controller to connect the Token-Ring Network to the host, most of the content of the document will apply to the IBM 3720 Communication Controller as well.

The document is intended for personnel who need information and guidelines for attaching an IBM Token-Ring Network to a System/370 type host and for installing typical applications in such an environment.

CSYS WS (146 PAGEs)

ACKNOWLEDGEMENTS

- The author of this document is:

  Ulrich Loacker - IBM Germany

- Thanks are due to the project advisor:

  Joan McCormack

  Raleigh International Technical Support Center
  Raleigh, North Carolina

- Contributions to this document by the following are gratefully acknowledged:

  Stein Torget - IBM Norway

  Manuel Ibarguchi - ITSC, Raleigh

PREFACE

This publication is intended to provide information for personnel who need to attach an IBM Token-Ring Network to a System/370 type host and to get applications running in such a environment.

The material assumes some knowledge and experience with IBM Communication Products and Workstations. The subject matter in this document applies to the current releases of the following IBM Products:

- ACF/SSP Version 3 Release 2
- ACF/NCP Version 4 Release 2
- ACF/VTAM Version 3 Release 1.1
- Netview (NPDA Version 3 Release 3)

The document is organized as follows:

- INTRODUCTION

  The introduction provides a review of IBM Token-Ring Network concepts and an introduction to host attachment via the IBM 3725 Network Control Program Token-Ring Interface.

- IBM 3725 TOKEN-RING NETWORK HOST ATTACHMENT

  This section describes the implementation of the IBM 3725 Token-Ring Sub-System (TRSS) and the Network Control Program enhancements to support the Token-Ring.

- PLANNING and IMPLEMENTATION - HOST COMPONENTS

  Host software dependencies and customization planning considerations are discussed in this section together with examples of definitions used in tests conducted at the International Technical Support Centre - Raleigh.

  Topics discussed include:

  - Addressing and Naming Considerations
  - Configurations and Performance Considerations
  - VTAM and NCP Definitions and Suggested Parameters

- PLANNING - TOKEN-RING NETWORK ENVIRONMENT

  This chapter provides a description of System product dependencies for host communication in the Token-Ring Environment, and some considerations for selecting these products.

- WORKSTATION SOFTWARE FOR HOST COMMUNICATION

  This chapter describes the planning and implementation requirements for Personal Computer Workstation Products supporting host communications. Installation definitions are related to corresponding host system definitions to enable the user to understand the relationships and interdependence of specific parameters in the two environments.

- SAMPLE CONFIGURATIONS AND BACKUP CONSIDERATIONS

  This chapter introduces additional considerations for planning a Token-Ring Network with host attachment. Various configuration scenarios are described together with samples of backup procedures.

- NETWORK MANAGEMENT AND RECOVERY

  Concepts of error recovery in a Token-Ring Environment and the tools to support various network management capabilities are described in this chapter.

- PROBLEM DETERMINATION

  This chapter describes problem determination using system trace facilities. A trace example is provided together with guidelines for interpreting it.

- Appendix A - PLANNING and IMPLEMENTATION - LIST of TASKS

  This appendix supplements Chapters 3, 4, and 5 by summarizing and integrating the planning and installation tasks for both the host and Token-Ring environments.

- Appendix B - PLANNING AND IMPLEMENTATION - WORKSHEETS

  This appendix provides sample worksheets for the installation tasks for both the host and Token-Ring environments so that a record may be maintained and so users or systems programmers can ensure the consistency of their definitions.

- Appendix C - SAMPLE SOURCES FOR NCP, VTAM and CICS

  This appendix lists the definitions used in the tests conducted at the International Technical Support Centre - Raleigh.

- Appendix D - TRACE EXAMPLE

  This appendix provides a trace example obtained in the tests conducted at the International Technical Support Centre - Raleigh.

- Appendix E - SAMPLE SOURCE FOR APPC/PC and TRANSACTION

  This appendix provides a listing of the source code used in the Personal Computer to interface with CICS at the host. The host application used was the sample APPC program distributed with CICS/VS Rel 1.7.

- Appendix F - PUBLICATION REFERENCE

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 IBM TOKEN-RING NETWORK EVOLUTION

The IBM Token-Ring Network is a strategic IBM architecture and set of products to address the requirements for flexible communications between systems and workstations within an establishment. Early SNA networks were hierarchical in structure, consisting of one or more 'host' or primary communications systems and a number of terminal or secondary systems devices. Communication sessions in this environment involved a Master-Slave relationship. Increased availability of intelligent terminals and distributed systems has created a need for peer-to-peer as well as master-slave communications sessions within today's office buildings. Local Area Networks (LANs) have evolved to address the wiring and communications needs of offices and other buildings. The Token-Ring Network is a particularly flexible Local Area Network implementation, capable of supporting a wide variety of workstations, systems, and topologies.

The architecture is based upon requirements to support many types of devices and systems in environments ranging from a few network stations on a single ring to several interconnected rings. The use of token-passing rings with bridging capabilities provides an environment in which requirements for network growth, problem determination and network management can be accommodated with minimal disruption. In addition, the token-passing protocols permit a deterministic approach to capacity planning and performance analysis.

Such characteristics, combined with a detailed architecture definition, have led to the adoption of Token-Ring protocols as IEEE and ISO (OSI-related) Standards.

The product implementations are based upon the IBM Cabling System, which consists of a variety of cable types and accessories to permit the attachment of individual workstations to large systems. Simple networks (consisting of PC's on a single ring) to complex networks (consisting of multiple rings with intermediate systems or direct host system attachment) are possible.

1.2 TOKEN-RING NETWORK CONCEPTS

The IBM Token-Ring Network is a general purpose Local Area Network (LAN) with the topology of a star-wired ring, using baseband signalling and token-passing protocols, in conformance with the IEEE 802.5 standards for transmission control. Device attachments conforming to the IEEE 802.2 and 802.5 standards may communicate over an IBM Token-Ring Network.

The token passing technique for ring access control is based on a predefined bit pattern, called a Token, which continuously circles the ring. When a station has data to transmit, it waits until its station adapter detects a 'token' bit pattern (Token bit = 0).
When the station waiting to send receives a token, it starts transmission of the data as follows.

The transmitting station creates a 'frame' by setting the Token bit to '1' and inserting the destination and source addresses, other control information and the data to be sent to the destination address. During the time the frame is being transmitted, no token is available on the ring (Token bit = 1). and no other station can initiate a transmission. The frame is passed (received and retransmitted) by all stations on the ring until the station with a matching destination address receives it.

The destination station copies the data to its internal memory and retransmits the frame after setting control bits to indicate that it recognized the address and successfully copied the data.

When the frame returns to the originating station with control bits indicating successful transmission and receipt, it is removed from the ring. A new token is then created and transmitted, thereby permitting other stations to send data.

Successful interpretation and retransmission of the 24-bit token around a ring requires a minimum 24-bit delay for each circulation of a token or frame. However, this does not impact performance because the ring has a speed of 4 megabits per second.

1.3 TOKEN-RING NETWORK HOST ATTACHMENT

Like any other Network Station, the IBM 372X Communication Controller has an adapter interface to the Token-Ring which supports the Token-Ring protocols. In the 372X, this adapter is called the Token-Ring Interface Coupler (TIC).

The mapping of Token-Ring protocols to SNA protocols is implemented by the NCP Token-Ring Interface (NTRI) which is part of ACF/NCP V4R2. It provides a capability to exchange data between Subarea Networks and IBM Token-Ring Networks.

1.4 TOKEN-RING, SNA, OSI AND IEEE

The Token-Ring architecture is a layered communications architecture that spans two defined layers, the Data Link Control (DLC) layer and the Physical layer. The Data Link Control layer is further subdivided into two sub-layers, Logical Link Control (LLC) and Medium Access Control (MAC).

When an IBM 372X Communication Controller is attached to a Token-Ring, the SNA Physical Services and Data Link Control (DLC) layers are functionally replaced by the Token-Ring protocols to support the IEEE 802.5 and IEEE 802.2 standards. This support is provided by a Token-Ring Subsystem in the 372X and by new functions in ACF/NCP V4R2.

The following figure shows the relationship between the two lower layers of the OSI model, the SNA layer definitions and the IEEE 802.2 and 802.5 standards.

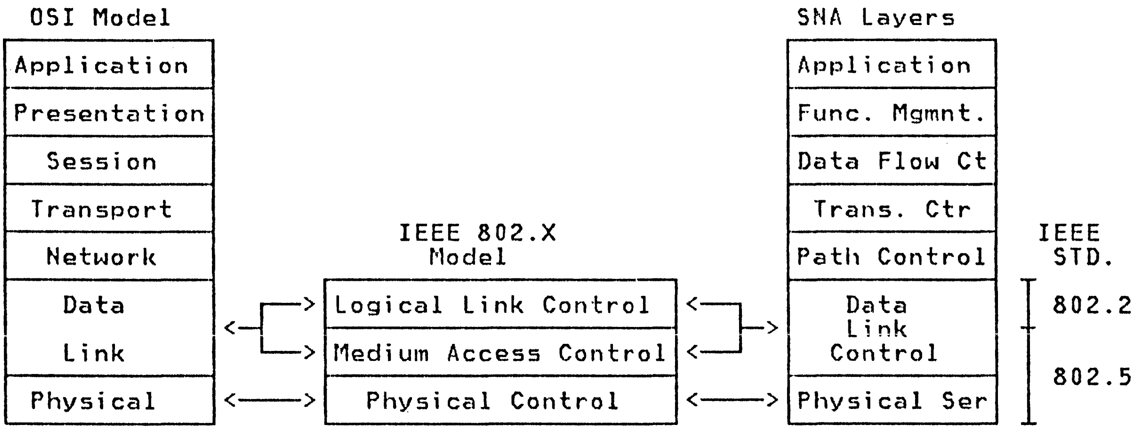


Figure 1. Layer relationships

Standard Token Ring IEEE 802.x Layer Definitions:

IEEE 802.2 - Logical Link Control Standard (LLC Sublayer)

IEEE 802.5 - Token Ring Standard (Physical Layer and MAC Sublayer)

1.5 TOKEN-RING HOST ATTACHMENT - VIEW FROM VTAM

Each physical connection to an IBM Token-Ring Network is viewed by VTAM as a leased full-duplex point-to-point line.

Each logical connection to an IBM Token-Ring Network Station is viewed by VTAM as a switched half-duplex point-to-point line and is defined in VTAM as a VTAM Switched Major Node.

With this approach, NTRI is transparent to ACF/VTAM and IBM Subsystems.

NTRI allows pre-definition of logical links for Token-Ring devices. Thus addition of new terminals does not require definition and generation of a new NCP module.

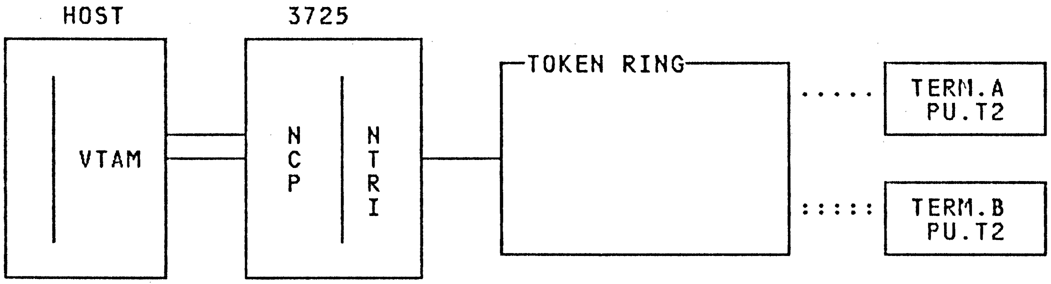


Figure 2. Physical Configuration

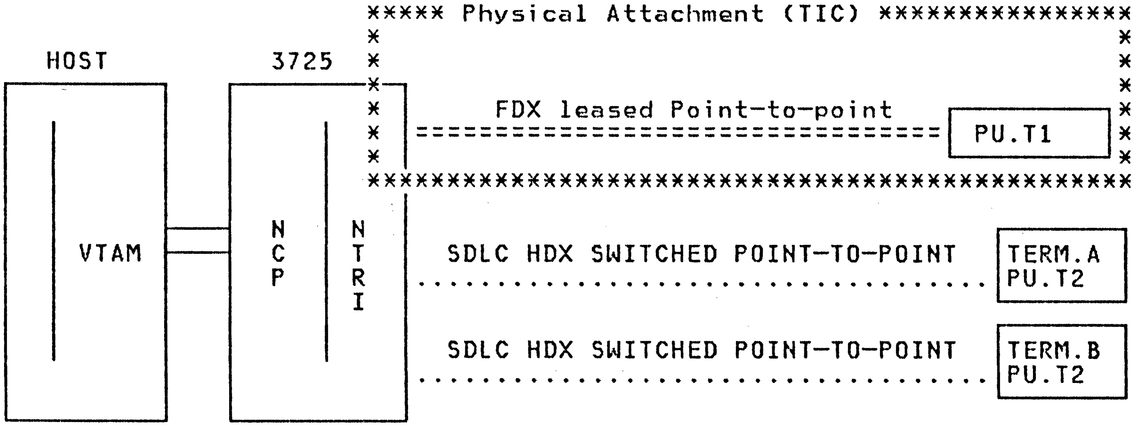

Physical Attachment (TIC)
HOST
3725
FDX leased Point-to-point
PU.T1
VTAM
NCP
NTRI
SDLC HDX SWITCHED POINT-TO-POINT
TERM.A
PU.T2
SDLC HDX SWITCHED POINT-TO-POINT
TERM.B
PU.T2

Figure 3. Corresponding logical Configuration

1.6 TOKEN-RING LAYERS AND FRAMES

Standards for Local Area Networks are based upon the two lower levels (Data Link Control and Physical Control) of layered systems architectures such as OSI and SNA.

Each node in a IBM Token-Ring Network contains a Data Link Control (DLC) layer. DLC includes manager functions (DLC.LAN.MGR) and covers two sublayers, Link Level Control (LLC) and Medium Access Control (MAC). The DLC.LAN.MGR supervises the operation and directs the flow of information through the MAC and LLC sub-layers. It also controls link activation and the attachment or removal of ring stations.

The physical layer provides the attachment to the medium (ring). This includes the cable from the adapter and the Multiple Station Access Unit (MSAU).

The unit of transmission in the IBM Token-Ring Network is called a frame. Such a frame is created when a node transforms a token and adds data to it. A frame is composed of the data to be sent, control information, application interface types, the source address and the destination address.

Each layer uses or adds its information to the frame and transfers a defined layout to the next higher or lower layer. The frame as it is transmitted around the ring contains all information added by the different layers and sub-layers.

The Token:

S D	A C	E D

SD = Starting Delimiter
AC = Access Control (bit 3=0 indicates Token)
ED = Ending Delimiter

The Token Ring Frame:

S D	A C	F C	Destination Address	Source Address	Rout. Info.	LLC Protocol Data Unit	Frame Chk.Seq.	E D	F S

DSAP Addr.	SSAP Addr.	Control field	Information field see field data types below:

SD = Starting Delimiter
AC = Access Control
FC = Frame Control Field
ED = Ending Delimiter
FS = Frame Status Field

Figure 4. Token and Token-Ring Frame Format

Information Field Data Types:

- SNA data -> PIU (TH/RH/RU - User data...)
- Control data
- Non-SNA data (e.g. NETBIOS data)

1.6.1 Interface between LLC and Higher Layers

The IBM Token-Ring Network uses Service Access Points (SAPs) designated in the architecture as code points through which an application may be defined to the LLC software. SAPs are the interface between the LLC and the Higher Communication Layers.

The IBM Token-Ring Network Architecture defines several SAPs, including SAPs for the SNA interface, the NETBIOS interface and non-SNA (OSI) interfaces. LLC software uses the SAP address (one byte) to determine whether it is using SNA or non-SNA connections.

The defined code point (SAP) between the IEEE 802.2 Link Level Control layer and the SNA Path Control layer is X'04'.

The data units exchanged between Path Control and LLC are called Path Information Units (PIUs).

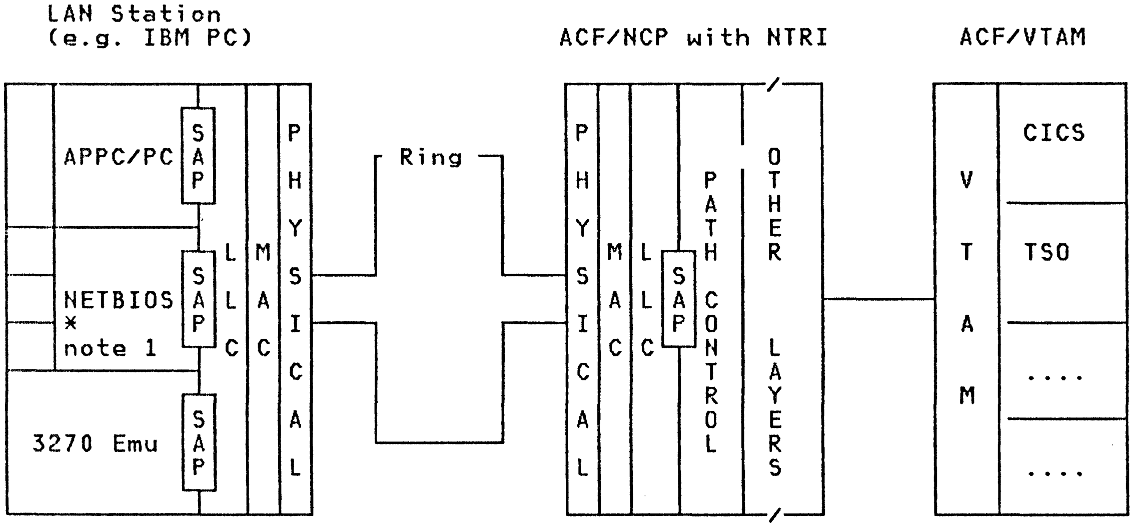


Figure 5. Token Ring Application Interfaces

Note 1: There is no NETBIOS SAP support within NTRI. Therefore, applications using the NETBIOS interface to access LLC cannot communicate with a subarea host.

1.7 TOKEN-RING NETWORK ADDRESSES

1.7.1 Universal or Locally Administered Adapter Addresses

Each Token-Ring Adapter has a unique adapter address. Multi-vendor assignment of these addresses is administered world-wide by the IEEE.
Several terms are used for this address: 'burned in' address, 'hard' address, or universal address. Using this universal address guarantees that the adapter address is unique.

The adapter can also be loaded with a locally administered, (or 'soft') address. The uniqueness of this 'soft' address must be locally administered.

If a Token-Ring Network is attached to a host, the usage of locally administered addresses is recommended because for a dial-out connection the adapter address must be equal to the dial number in the PATH statement of the VTAM switched major node. In this situation, a locally administered address allows portability of the network station, exchange of the adapter card in case of an error, and pre-definition of network stations in VTAM.

1.7.2 Token-Ring Network Addresses - Types

The Token-Ring adapter address field is 6 bytes (48 bits) in length and is, with minor variations in format, used as source address and destination address in frames.

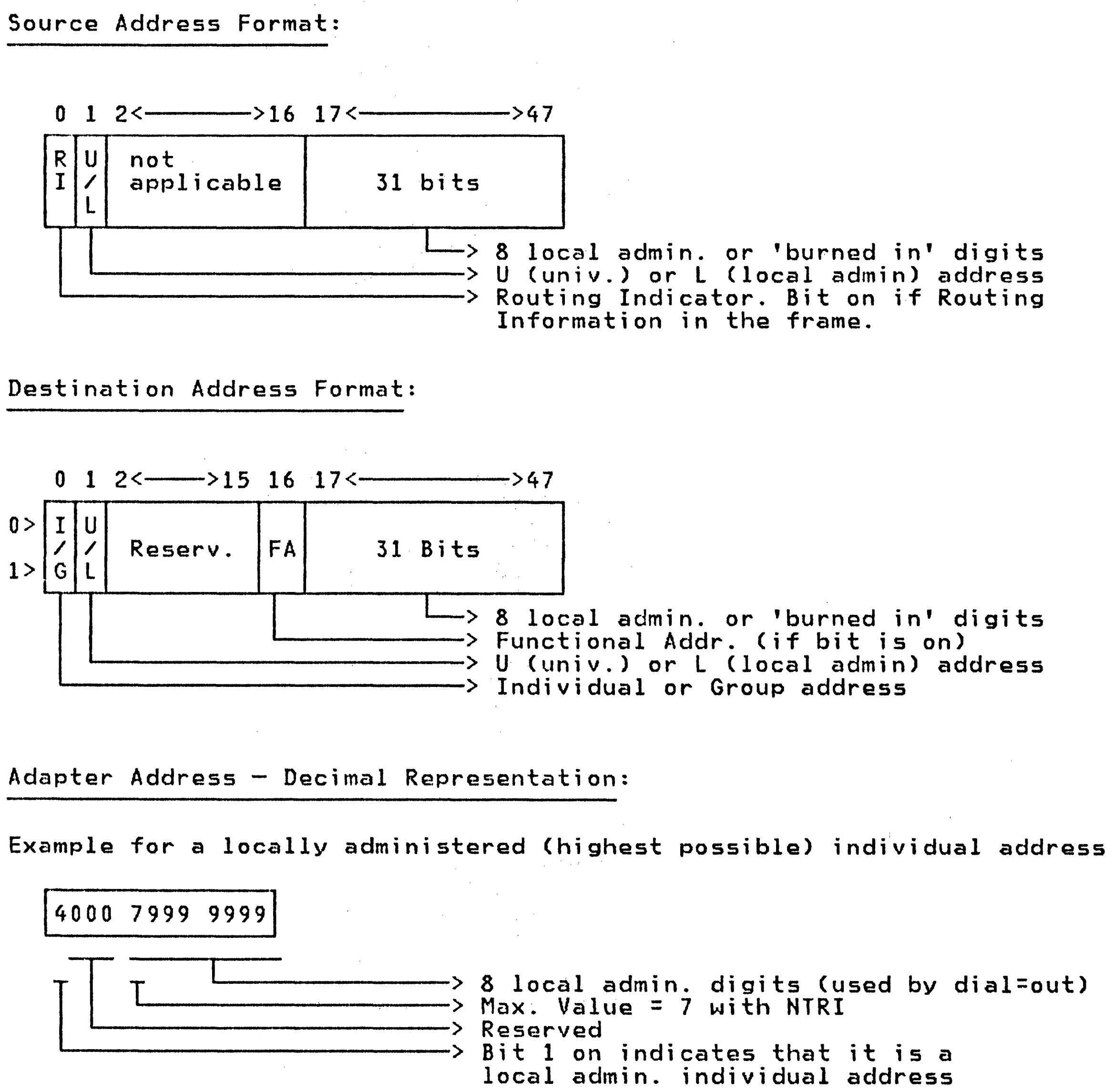
Each 372X (TIC) is also represented with an adapter address which must be a locally administered address. For example, the locally administered TIC address is used as the destination address when a workstation (PC) wants to communicate with a host using IBM PC 3270 Emulation or APPC.

A destination address can be either an individual address or a group address. An individual address addresses a specific destination where a group address is assigned to a collection of SAPs.

Special group addresses, called Functional Addresses, are defined to represent functions that are independent from a physical location because the destination address is not known by the station requesting the function. A typical example for such a functional address is the Monitor Function of a Token-Ring. Stations with defined functions will accept the functional address as the destination address.

Another special destination address is a Broadcast Address (all address bits on, x'FF') to denote all stations on a ring.

1.7.3 Source, Destination and Adapter Address Formats



Source Address Format:

> 8 local admin. or 'burned in' digits
> U (univ.) or L (local admin) address
> Routing Indicator. Bit on if Routing Information in the frame.

Destination Address Format:

> 8 local admin. or 'burned in' digits
> Functional Addr. (if bit is on)
> U (univ.) or L (local admin) address
> Individual or Group address

Adapter Address - Decimal Representation:

Example for a locally administered (highest possible) individual address

> 8 local admin. digits (used by dial=out)
> Max. Value = 7 with NTRI
> Reserved
> Bit 1 on indicates that it is a local admin. individual address

Figure 6. Address Formats and Layouts

1.8 TOKEN-RING NETWORK - ROUTING

When a Token-Ring Network consists of multiple rings connected by bridges, a station on the ring can communicate with another station on any other ring in the network as long as no more than seven bridges must be crossed by a frame traveling between the two stations.
The use of bridges is transparent to higher level applications running on the station and to connections between ring stations and subarea hosts. For applications written at the LLC interface, the use of bridges is no longer transparent.

The path a frame will follow through the network from a source to a destination station during a session is dynamically determined by the routing function. Imbedded in each transmitted frame is all the required information about the route (or path) to be followed through the Token-Ring Network.
There is no need for any centralized routing tables. There is also no limit, except perhaps for a limit imposed by the impact on performance, to the number of concurrent sessions which may use a bridge.

1.8.1 Routing Concept

When a session is initiated, a search is performed to acquire the required routing information. The source station can initiate a search by issuing a TEST or XID (Exchange Identification) LLC command on its local ring. If a response is not received, the destination station is not active on the local ring. The source station then broadcasts a TEST or XID command to all rings. This frame fans out through the network by being passed through all active bridges. As a frame passes a bridge, the bridge inserts its Ring ID into the frame's Routing Information field.

This may create multiple copies of the 'search' frame and each copy will continue to be passed through the network searching for the destination station. This process will continue until one or more copies of the frame reach the destination station. Each time a frame passes through a bridge is called a hop.

Search frames are deleted from the Network when they attempt to cross the same bridge a second time, when they attempt to cross the eighth bridge, or when a maximum 'hop count' has been reached.

Each frame reaching the destination station will have imbedded in it the ID's of all the bridges it crossed, in the sequence in it crossed them. The destination station sends the frame copies back to the source station. Each follows the same route it used to get to the destination station, but in the reverse direction. The first response to reach the source station is accepted as containing the best currently available route to the destination station. Any other responses received by the source station are discarded.

The destination station determines the preferred route (from the routing information field) when it receives the first non-broadcast frame from the source station. It will use that route for any subsequent transmissions to the source station. Therefore, all subsequent transmissions during the session will follow the same route.

TEST and XID Command Function

Both the TEST and the XID commands cause a search for a route to a destination station to be performed. The TEST command causes the destination station to return a TEST Response, and performs only a basic test of the transmission path. The XID command conveys the identification and characteristics of a source station. The response to a XID indicates the class of service supported and the maximum receive window size of the destination station. This denotes the maximum number of unacknowledged sequentially numbered frames that can be received.

Conclusion

The route used may change between sessions because of the traffic load existing on the ring and bridges when the route is established. This process tends to dynamically balance the overall network load.

The Token-Ring Network routing concept is also applicable for the 372X Token-Ring adapter (TIC). The routing is transparent to VTAM and no detailed path information is required to define a route in a Token-Ring Network.

1.9 INTRODUCTION TO SNA (FOR PC USERS)

Systems Network Architecture (SNA) is a definition of communications structures and protocols which provide functions that enable end-users to be relatively independent of the network's characteristics and operations. These functions are implemented in network 'nodes' (processors, controllers, or workstations) by network resources which fall into two basic categories: Network Addressable Units and the Path Control Network.

Network Addressable Units (NAUs) enable end-users to send data through a network and help network operators perform network control and management functions. Network Addressable Units provide functions to:

- Synchronize communication between end-users
- Manage the resources in a node
- Control and manage the network

Each NAU has an address that identifies it to other NAUs and to the Path Control Network. The Path Control Network uses this address to route data between NAUs. SNA defines three kinds of Network Addressable Units: Logical Units, Physical Units, and System Services Control Points.

Every end-user gains access to an SNA network through a Logical Unit (LU). Logical Units manage the exchange of data between end-users according to a common set of rules or protocols. The Logical Unit thus acts as an intermediary between the end-user and the network. (There doesn't have to be a one-to-one relationship between end-users and LUs. The number of end-users who can gain access to a network through the same LU is an implementation design option, and is expressed in terms of the LU profile or type.)

LU-LU Sessions: Before end-users can communicate with one another, their respective LUs must be connected in a mutual relationship called a session. Because this session connects two LUs, it is called an LU-LU session. Multiple concurrent sessions between the same two logical units are called parallel LU-LU sessions.

The architecture defines different kinds of logical units called LU Types. Since SNA was announced, IBM has developed a number of LU Types to handle the communications requirements of a variety of end-users.

Every node contains one Physical Unit (PU) to manage the links that connect the node to adjacent nodes. The PU represents the processor, controller, workstation, or printer to the network. Physical Units, like other NAUs, are implemented by a combination of hardware and software components within each node.

System Services Control Points (SSCPs) provide the capability to activate, control, and deactivate network resources in special nodes (called host subarea nodes) because of the way they define and control the network resources.

The Path Control Network routes and transmits data between network addressable units. The Path Control Network provides functions to:

- Transmit data across links between adjacent nodes
- Route data between nodes

For more information about Systems Network Architecture, refer to SNA Concepts and Products Reference - GC30-3072.

1.10 PU T2.0, PU T2.1 AND LU 6.2

- PU T2.0 is a hierarchical (primary-secondary) implementation of SNA that supports many host applications (from LU T0 to LU T6.2)
- PU T2.1 provides peer to peer connectivity. Devices supporting PU T2.1 include: PC, S/36, S/38, S/88, Series/1.
- LU 6.2 provides Application to Application Communication (APPC). Examples for Application Subsystems providing APPC support are CICS/VS (System/370) and APPC/PC (IBM PC). Other systems supporting APPC are S/36, S/38, S88, and Series/1.
- For peer PU T2.1 connections, only LU 6.2 sessions are supported. All other LU types (not 6.2) require an SNA SSCP.
- Currently all peripheral nodes attach to SNA subarea hosts as PU T2. Even those systems which implement PU T2.1 for peer communication with other systems (i.e. a PC with APPC/PC) still appear as PU T2.0 to the host system (VTAM/NCP).

In summary, an IBM PC attached to a Token Ring Network supports the following types of SNA communication:

- PU Type 2.1 for peer communication (using LU Type 6.2) with other PCs, S/36, S/38 and Series/1.
- PU Type 2.0 for 3270 Emulation (using LU Types 1,2,3) and for APPC (using LU Type 6.2) to communicate with a System/370 type host.

2.0 IBM 3725 TOKEN-RING NETWORK HOST ATTACHMENT

The IBM Token-Ring Network host attachment support is provided by two new components for the IBM 3725 Communication Controller:

- Hardware - The Token-Ring Subsystem (TRSS) - which includes the
  Token-Ring Multiplexer's (TRMs) and
  Token-Ring Interface Coupler's (TICs)
- Software - The NCP Token-Ring Interface (NTRI)

The TRSS can be installed in existing 3725 CCUs, and the supporting software is part of ACF/NCP V4R2. With these two components, the 3725 can be directly attached to an IBM Token-Ring Network. (The IBM 3720 Communication Controller will use the same facilities.)

There are no changes in VTAM because the NTRI support is transparent to ACF/VTAM. VTAM uses two types of connections. The first addresses the physical attachment (between the SSCP and the TIC) which is defined as a full-duplex leased line. The second defines one or more logical connections between LUs which are defined in VTAM as Switched Major Nodes on half-duplex switched lines.

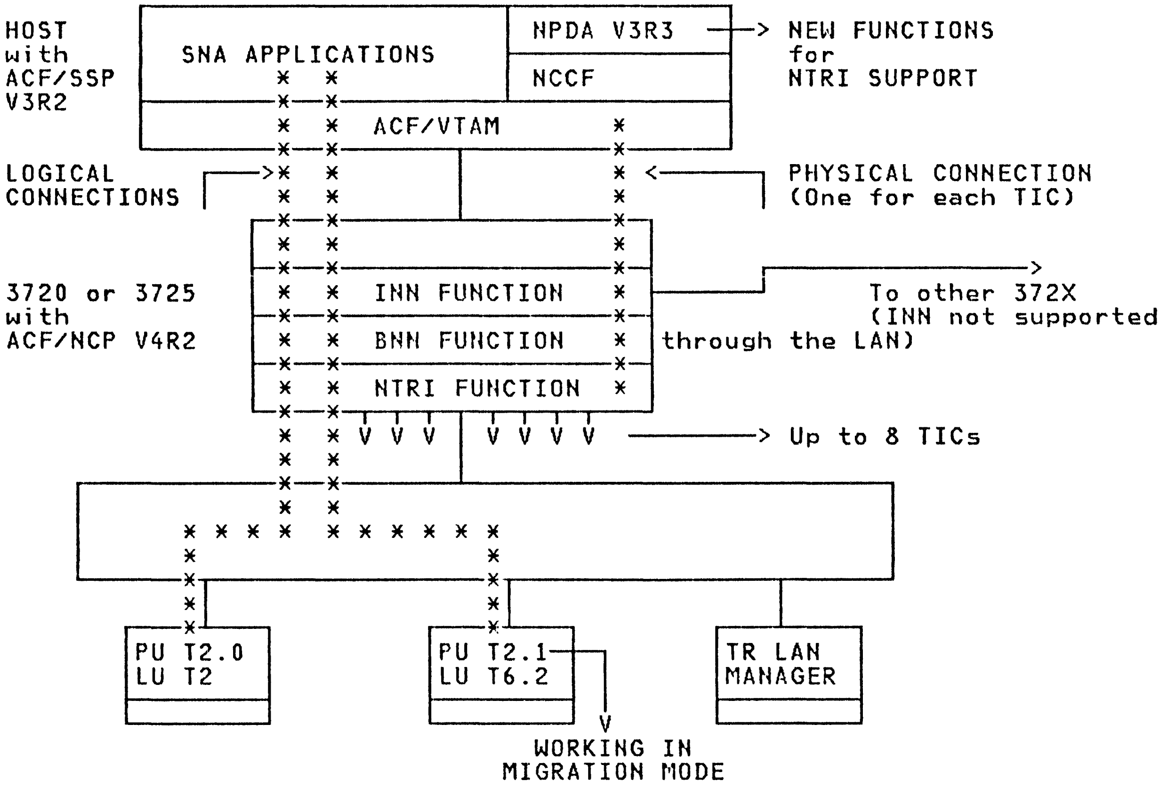


Figure 7. Host Attachment - Schematic View

2.1 THE TOKEN-RING SUB-SYSTEM (TRSS)

The IBM 3725 Communication Controller has been enhanced to support the IBM Token-Ring Network. Release Level 4 of the 3725 is required for this support. In addition, a new Communication Controller (IBM 3720) has been announced which will also provide this support.

The 3725 attachment to the IBM Token-Ring Network is made by: **Token-Ring Sub-system (TRSS)**, a new Transmission Sub-System, and by new functions for the: **Maintenance and Operator Sub-System (MOSS)**

2.1.1 TRSS Components

The following TRSS components are in the new Line Attachment Base Type C (LAB C):

- A Token-Ring Multiplexer (TRM)
- Token-Ring Interface Coupler's (TICs)

The term TRSS represents the entire Token-Ring Attachment implementation in a 3725 Communication Controller Complex (3725 and 3726). A TRSS may consist of up to two Token-Ring Adapters (TRA), one located in the 3725 and one in the 3726. If there is no TRA in a 3725, two TRAs can be installed in the 3726. (The term TRA should not be confused with a Token-Ring Adapter card.) The Token-Ring Adapter connection point to the ring is called a Token-Ring Interface Coupler (TIC) for the 372X.

The Line Attachment Base Type C (LAB C) designates the 3725 board which includes a Scanner, the Token-Ring Multiplexer Card (TRM) and the base for up to 16 regular communication lines.

A TRA consists of a Token-Ring Multiplexer (TRM) plus Token-Ring Interface Couplers (TICs). There may be up to four TICs per TRM.

The Token-Ring Multiplexer (TRM) is the 3725 Bus adapter card. Its main objective is to transform the TIC interface into the 3725 bus.

A TIC contains a microprocessor under control of resident microcode. The TIC communicates with the TRM card for data transfer and is controlled by NTRI.

Each TIC represents one physical Token-Ring connection. More than one TIC may be connected to the same Token-Ring. The data rate supported on the IBM Token-Ring is 4 Mbps (4 million bits per second).

The TIC-Interface meets all the requirements of the IEEE 802.5 standard.

2.1.2 TRSS Structure

This figure shows the structure of the Token Ring Sub-System components for a IBM 3725 Communication Controller. For a Controller Complex with both 3725s and 3726s, a second set of components (illustrated in the starred box) can be added.

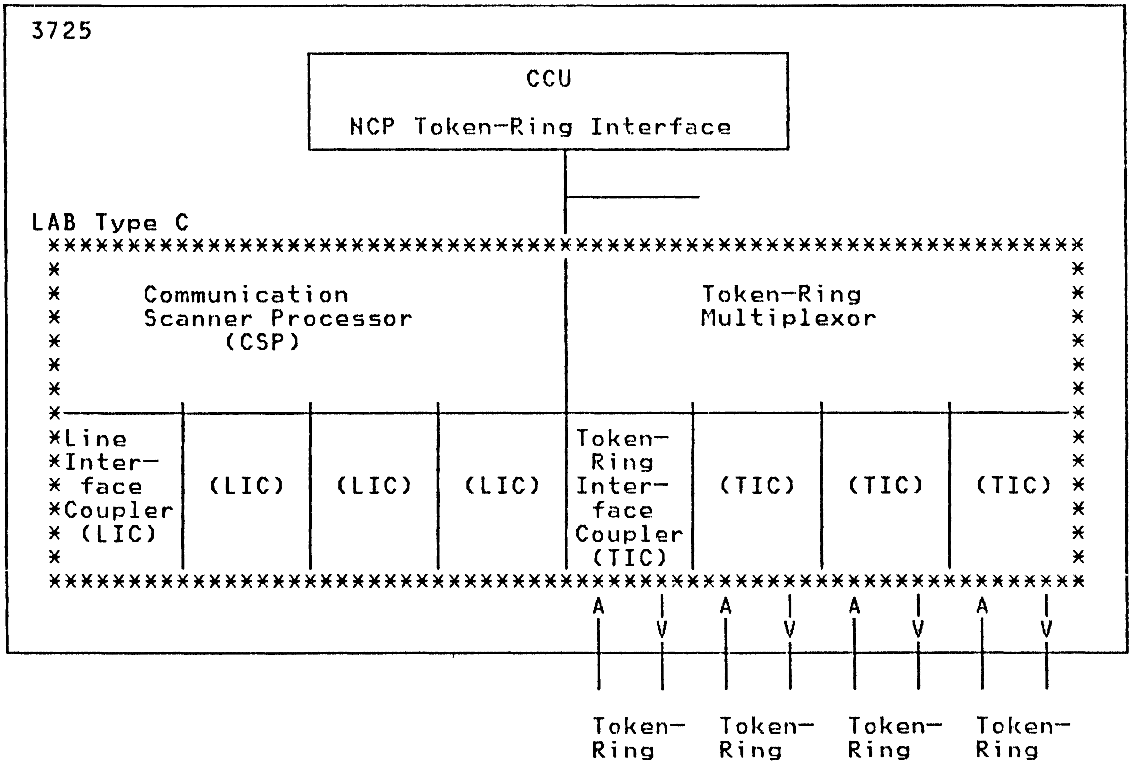


Figure 8. The Token-Ring Sub-System (TRSS)

2.2 THE NCP TOKEN-RING INTERFACE (NTRI)

2.2.1 Objectives of NTRI

NTRI is designed to provide users with a means to attach an NCP Communication Controller to the IBM Token-Ring Network using one physical medium, one attachment scheme and one communication protocol. NTRI permits attachment to IBM Token-Ring Networks through one or more physical links.

NTRI supports a IBM Token-Ring interface on the 372X so that the user can implement and carry forward the creation and installation of a Token-Ring Local Area Network and can start implementing data processing solutions in this environment.

NTRI provides a basic Boundary Network Node (BNN) interface to support SNA-to-SNA communication of devices attached to a Token Ring Network.
NTRI does not support the Intermediate Network Node (INN) facility of ACF/NCP through a Token-Ring Network which means that it is not possible to connect two ACF/NCPs using the Token-Ring Network.

2.2.2 NTRI Environment

Support of NTRI requires a number of new facilities inside ACF/NCP V4R2.

A 3725 with NTRI may be connected directly to a maximum of 8 token-rings, regardless of the number of interconnecting bridges between these token-rings. Several TICs may be connected to the same ring (e.g. to provide backup support).

NTRI uses the Boundary Network Node (BNN) of NCP and provides connectivity to PU T2.0 and PU T2.1. With the current implementation of the ACF/NCP Boundary Network Node (BNN) function, PU T2.1 is supported only in migration mode, which is like PU T2.0.

2.2.3 NTRI Components in ACF/NCP V4R2

The NTRI function is responsible for the physical connection of TICs, handling of SNA commands, Beacon and Medium Access Control (MAC) Frames and monitoring TIC operative conditions. NTRI also provides support for the 3725 error reporting mechanisms.

Logical Link Control (LLC) is that function which handles the establishment of logical connections with LAN devices and the exchange of data.

Medium Access Control (MAC) includes that function of NTRI which controls the TRM and TIC.

2.2.4 Related Software and Components with NTRI Support

ACF/SSP V3R2

Support of NTRI requires a number of new facilities within the ACF/SSP. The NTRI Generation process is part of the NCP/EP Definition Facility (NDF). This process allows the user to describe to NDF the attachment of an IBM Token-Ring Network to the 3725.

NetView - NPDA Component

NPDA V3R3 supports Network Management Vector Transport (NMVT) messages generated by NTRI.

NTRI generates three categories of NMVT messages.

- NMVT Alert messages
- NMVT Link Event messages
- NMVT PD Statistic messages

NetView - NLDM Component

- The Token-Ring Network is transparent to VTAM and therefore supported by NLDM across the NCP Token-Ring Interface.

MOSS

- Provides new facilities to support TRSS and NTRI including
  - Display of TIC addresses and status
  - TIC traces

2.3 NTRI - LOGICAL AND PHYSICAL CONNECTIONS WITH DATA FLOW

2.3.1 Logical Link Control (LLC)

LLC Connections

A logical link connection is defined by the Destination Address (6 bytes), the Source Address (6 bytes), the Destination Service Access Point (DSAP: 1 byte). and the Source Service Access Point (SSAP: 1 byte) Refer to the introduction for more details.

LLC Functions

The Link Level Control (LLC) layer is responsible for the establishment of the logical connection with a terminal and for the exchange and the integrity of the data. Data recovery is performed by the LLC.

The PIU's of each logical connection waiting to be transmitted are separately queued for output and mapped to a specific physical connection.
In the NCP generation a TIC is defined as PU Type 1 (like a 3767).

2.3.2 Physical Link Management

Physical Link Management includes the following functions:

- handling of the SNA commands and the internal requests (Inoperative conditions) to establish or cancel physical connections to the IBM Token-Ring LAN.

- handling of Medium Access Control (MAC) frames received from the TIC. Remember MAC frames are special frames exchanged between the adapters on the ring. Most of the MAC frames go up to the next higher layer, but Beacon and TIC internal MAC frames are processed by the PLM.

- handling of 'Ring Status Change' events reported by the TIC.

- handling of timer mechanisms which check that the TIC microcode is still running, thus avoiding a deadlocked or hung situation on the ring. It consists of sending an Interrupt request to the TIC every five seconds and monitoring for an Interrupt completion. If there is no Interrupt completion, NTRI will force removal of the TIC from the ring.

2.3.3 Physical Activation - Data Flow

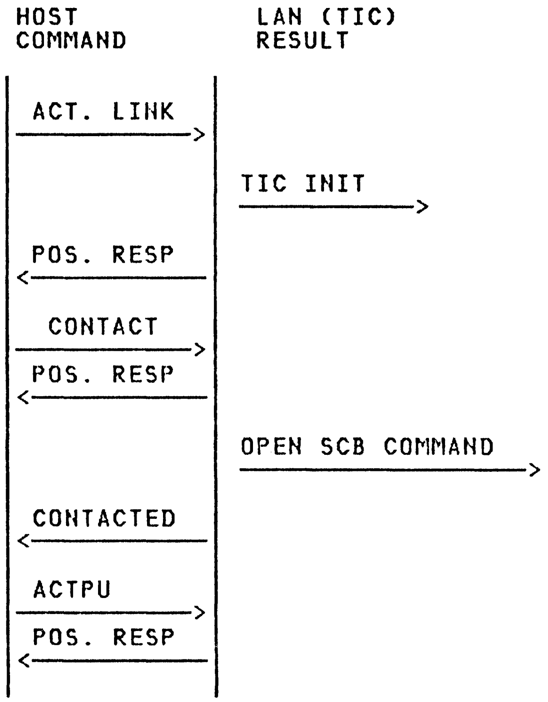


Figure 9. Generalized Representation of Physical Port Activation

Explanations:

TIC INIT - Initialization of the TIC Adapter

OPEN SCB - Open TIC Adapter - Status Control Block

(Detailed flow not shown)

2.3.4 Logical Connections - Outbound Call Data Flow

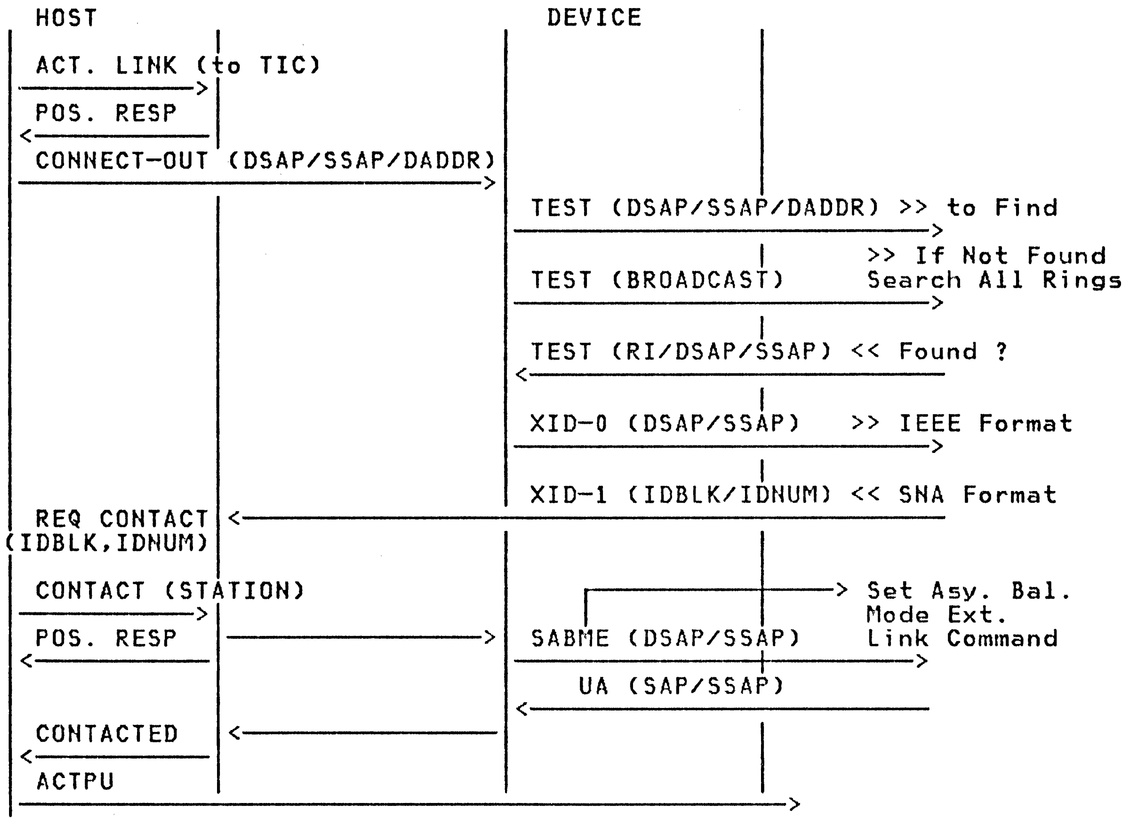


Figure 10. Generalized Representation of Host Initiated Calls

Notes re Figure 10.

- Outbound calls (DIAL=OUT) are initiated by the operator, VTAM, or a program for such stations as an IBM 3174 Control Unit or a Workstation PC executing APPC/PC. DIAL=OUT is normally not used for PC 3270 Emulation, except in restart situations, where it might be useful.
- An IEEE format XID is first exhanged to establish connectivity at the LAN level. If this is OK, then a traditional SNA XID is issued.
- The values for IDBLK and IDNUM and the Data Flow can vary depending on the software used at a workstation.
- If a connection already exists, the CONNECT-OUT will refuse a request for a new connection with a negative response.
- In a multi-host environment, the logical link over which the connection is made can be owned by a different VTAM than that which owns the physical port (i.e. the TIC).
- Detailed flow not shown.

2.3.5 Logical Connections - Inbound Call Data Flow

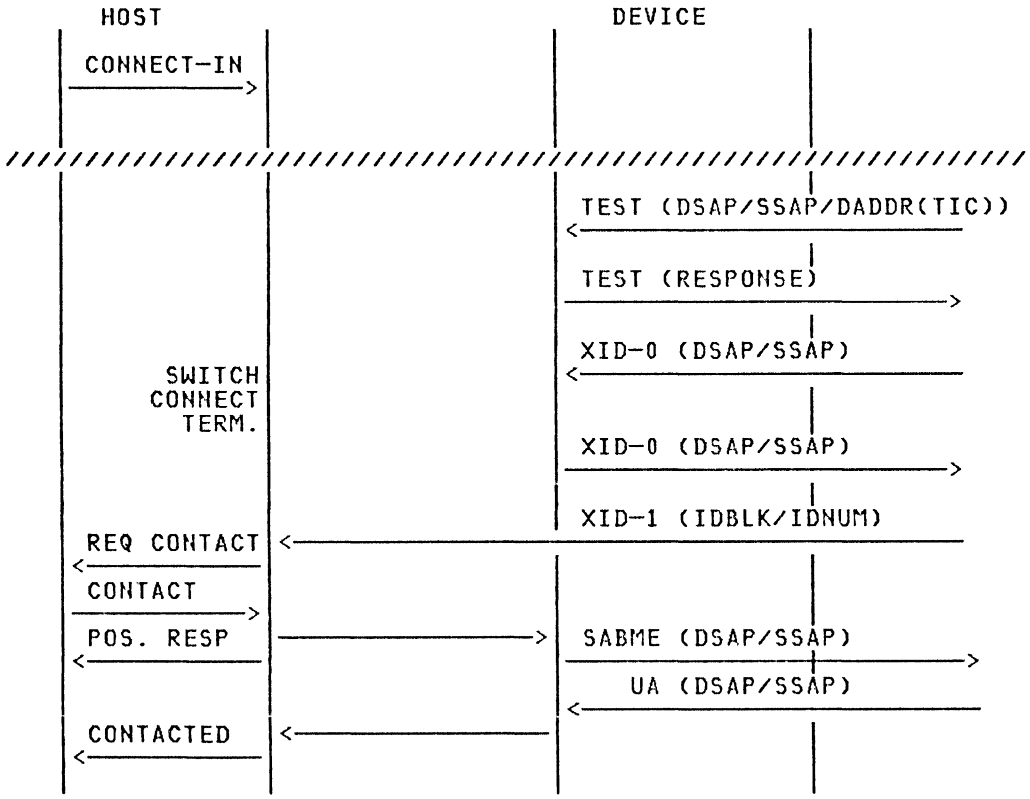


Figure 11. Generalized Representation of Inbound Calls

Note re Figure 11.

- Inbound calls (DIAL=IN) are initiated by the workstation. This can result from the user selecting 'COMMUNICATION' in the primary menu of the IBM PC 3270 Emulation program, or an APPC/PC transaction calling a host (CICS) transaction.
- In a multi-host environment, the logical link over which the connection is made can be owned by a different VTAM than that which owns the physical port (i.e. the TIC).
- The values for IDBLK, IDNUM and the Data Flow can vary depending on the software used at the workstation or LAN device.
- Detailed flows not shown.

3.0 PLANNING AND IMPLEMENTATION - HOST ENVIRONMENT

This chapter contains planning and preparation information for System/370 host components required for TRN host communication. In addition, naming and performance considerations are presented in this chapter.

Planning considerations for Token-Ring Products and Workstations are in separate chapters.

3.1 PLANNING HOST ENVIRONMENT PRODUCTS FOR NTRI

NTRI is a user definable option of ACF/NCP V4R2. ACF/VTAM has no specific code for NTRI because Token-Ring Network Stations are supported as regular SNA devices on switched lines.

The products containing new code to support NTRI are:

ACF/NCP V4R2
ACF/SSP V3R2
NetView (NPDA V3R3)

3.1.1 ACF/NCP V4R2 Product Requirements

- ACF/SSP V3R2 is required to generate ACF/NCP V4R2 with NTRI support
- ACF/NCP V4R2 requires EREP V3R2 or EREP V3R3
- Operating Systems supporting NCP V4R2 with NTRI:

  MVS/XA
  MVS/370
  VM/SP R4
  VM/SP HPO R4

- Versions of ACF/VTAM supporting communication with NCP V4R2 and NTRI:

  ACF/VTAM V3R1.1 --> only this version supports CNM for NTRI
  ACF/VTAM V3R1
  ACF/VTAM V2R2
  ACF/VTAM V2R1

- Versions of ACF/NCP with which NCP V4R2 can communicate:

  ACF/NCP V4R2
  ACF/NCP V4R1
  ACF/NCP V4 Subset
  ACF/NCP V3 for 3705 and for 3725
  ACF/NCP V2 for 3705 and for 3725

- Other Program Products operating with NCP V4R2:

  NTO R4
  NRF R3
  X.25 NPSI R4.3
  EP R4

- Other Program Products communicating with NCP V4R2 are:

  NetView (or NCCF V2R3, NPDA V3R3 and NLDM V2)
  NPM R1 or NPM R2

ACF/NCP V2 and V3 APARS must be applied to support coexistence with ACF/NCP V4R2.
The Program Directory for ACF/NCP V4R2 includes a list of required APARS. VTAM and NPM APARS are also required and listed.

3.1.2 ACF/SSP V3R2 Product Requirements

- ACF/SSP V3R2 is required to generate NCP V4R2 with NTRI support

- With ACF/SSP V3R2 the following levels of NCP can be generated, loaded and dumped:

ACF/NCP V4R2	-	3725,3720	-	MVS,VM	-> NTRI
ACF/NCP V4 Subset	-	3720	-	MVS,VM,VSE	-> NTRI- MVS,VM
ACF/NCP V4R1	-	3725,3720	-	MVS,VSE	-> no NTRI
ACF/NCP V3 for 3705	-	3705	-	MVS,VM,VSE	-> no NTRI
ACF/NCP V3 for 3725	-	3725	-	MVS,VM	-> no NTRI

Some APARS must be applied to ACF/SSP V3R2 to allow generation of different versions of ACF/NCP with NDF.
The Program Directory for ACF/SSP V3R2 contains a list of required APARS.

3.1.3 NetView Product Requirements

- ACF/VTAM V3R1.1 is required to support NetView R1

- ACF/NCP V4R2 is required to support all functions provided by NetView including NPDA support for NTRI.

- Operating System environments for NetView are:

  MVS/XA
  MVS/370
  VM/SP R4
  VM/SP HPO R4

3.2 PLANNING HOST PRODUCTS FOR TOKEN-RING COMMUNICATION

No special host products are required to support communications with workstations on a directly attached Token-Ring because VTAM views them as workstations on Switched SDLC links.

Products to be considered for application function include:

• CICS 1.7	- For APPC/PC support or 3270 Applications
• IMS/VS	- For 3270 Applications
• File Transfer Programs	- For file transfer to/from TSO,CMS,CICS
• DISOSS	- For Office System Communication - MVS
• PROFS	- For Office System Communication - VM
• TSO/E Servers	- For Enhanced Connectivity Facility -MVS
• VM/CMS Servers	- For Enhanced Connectivity Facility -VM

The Workstation Chapter of this document includes more information about APPC/PC, the File Transfer Program, and Host Server programs.

3.3 PLANNING - 3725 COMMUNICATION CONTROLLER

3.3.1 3725 Requirements to run NCP V4R2 with NTRI Support

- NCP V4R2 operates on 3725 and 3720 Communication Controllers
- NCP V4R2 requires 3725 Release 4
- 3725 will be automatically updated to Release 4 (EC 873055 or later) by CE.

NOTE: The NCP V4 SUBSET will support NTRI in appropriately configured 3720 Communication Controllers in an MVS or VM host environment.

3.3.2 3725 Storage Requirements for NTRI

NTRI program code requires approximately 85 Kbytes of memory.

In addition, a number of parameters must be defined and data areas reserved for each physical and logical connection. The storage requirements for these data areas are:

Requirements for each **physical** connection:

> Each physical connection (TIC) is viewed by VTAM as a full-duplex leased point-to-point line. In addition to the standard NCP data areas for such connections (488 bytes), NTRI requires about 1500 bytes for each TIC.

Requirements for each **logical** connection:

> Each logical connection (connection with a terminal) is viewed by VTAM as a switched point-to-point line. In addition to the standard NCP data areas for such a connection (380 bytes), NTRI requires about 276 bytes for each connection.

Publication Reference:

3.4 NAMING CONSIDERATIONS AND CONVENTIONS

3.4.1 Token-Ring Network Adapter Addresses

3.4.2 Universal Address or Locally Administered Adapter Address ?

In this document these are also referred to as 'Hard' and 'Soft' addresses respectively.

Each Token-Ring Adapter has a unique 'burned in' or 'hard' 6 byte address. The uniqueness of this address is administered by the IEEE among vendors, and internally by the vendor.

For a Token-Ring environment without a host connection, the IEEE 'Universal' address may be useful. The advantages of using the universal address are its guaranteed uniqueness and the avoidance of local address administration. Be sure to keep a list of stations, some topologic information, and the corresponding universal addresses for TRN Management.

As an alternate to the 'Universal' address, a locally administered address (soft address) can be used for a TR adapter.

For a Token-Ring environment with a host connection, the use of universal addresses has some disadvantages. In some cases, locally administered addresses are required (e.g. for host-initiated connections).

Restrictions of Universal Addresses for host-attached TRN:

- Pre-generation of Network Stations is not possible because the Adapter Address has to be defined in VTAM PATH statements for DIAL=OUT.
- If a defective adapter is replaced, VTAM has to be updated to reflect the new address before the user can re-establish connectivity.
- A locally administered or 'soft' address is required for the adapter address of the TIC.
- From an operational point of view (CNM and TRN Management) it may be useful to make the address informative, instead of a random number.

3.4.3 Administration of Adapter Addresses

There is only one requirement for the administration of 'soft' addresses: The address must be 'unique'. The last 4 bytes (8 digits) from the 6 byte adapter address (identified with 'nn') can be used to implement naming conventions. Considerations for these conventions can be based on topology, operational needs, a combination of both, other characteristics, or just a serial number.

The conventions we used for our tests are discussed later in this chapter.

Adapter Address Format

Adapter Address layout	4000 dddd dddd
'4' indicates a 'soft' address	4
Reserved	000
User administered section	dddd dddd
Max. value allowed	7999 9999
Part used in 'DIALNO=' (VTAM)	4000 dddd dddd

For administration, it may be useful to have a worksheet which contains information about addresses, status information and the corresponding network station location. A example of such a worksheet is in Appendix B.

3.4.4 Naming Convention Example for Adapter Addresses

In our example we considered mainly operational aspects for the naming conventions. The only topological information we used is a ring number and (for host attachment considerations) the subarea of the NCP which is most frequently used by the network station applications as the destination address and to which the PC 3270 Emulation program was customized. Another potentially useful piece of information could be the subarea of the SSCP in which the Switched Major Node is defined.
For operational considerations, we used one position to define the station type.

LOCALLY ADMINISTERED ADAPTER ADDRESS (12 DIGITS)	
FIXED (4 DIGITS)	VARIABLE (LAST 8 DIGITS)
4000	a bb cc ddd

a = Station type

0 - Not used (optional)
1 - 3725 or 3720
2 - 3174 L with LAN Attach
3 - PC Type Workstation
4 - 3174 R Attached to a TRN
5 - Printer
6 - ...
7 - ...
8 Not allowed, bit 0 used for
9 Functional Addr. Indicator

bb = Subarea number of NCP

cc = Ring number

ddd= Serial number

3.4.5 SNA Naming Considerations

Existing naming conventions for SNA resources can still be used for host attached Token-Ring Network resources.

One change which should be considered is to use one position of the naming conventions to identify a resource as a TRN resource. In our example in Appendix C, we used 'E' as TRN resource identification.

If there are no existing naming conventions for switched major nodes than new terms should be defined.

There is another situation which may be considered for naming conventions: specifying a difference between current SNA stations and TRN stations. A SNA station is normally defined as a 'fixed' PU Type, but a PC in a TRN environment can be used for multiple functions like PC 3270 Emulation and LU Type 6.2. Each of these functions requires a distinct PU/LU pair for the same station.

The function-related information is in the IDBLK specification of the PU statement, where '50' identifies APPC/PC and '17' identifies PC 3270 Emulation.

It may be useful to reflect this in the naming conventions. We consider this in the example on the next page.

Another field to consider for naming conventions is the IDNUM= value in the PU statement. This can be any value, but it must be unique, and the content must match the value specified during setup of the application subsystem in the workstation.

Naming conventions used for VTAM should take into consideration the requirements of application subsystems (such as CICS or IMS) which use these names, (for example, the NETNAME=LUname parameter of a CICS TCTTE).

3.4.5.1 SNA Naming Convention Example

PU/LU - Naming Conventions for VTAM Switched Major Node

```
PU/LU for PC with 3270 Emulation Standalone Configuration
--------------------------------------------------------------------
E13SW     VBUILD MAXGRP=5,..

E13PE01   PU     IDBLK=017,..        PU  -> 3270 Emulation
│ │ │ └─────────────────────────────> Sequ. Number  <───────────────┐
│ │ └───────────────────────────────> Application/PU identifier     │
│ │                                      - E = PU for 3270 Emulation │
│ │                                      - A = PU for APPC/PC        │
│ │                                      - C = PU for TRN Att. 3174  │
│ │ └───────────────────────────────> P = Physical Unit Switched    │
│ └─────────────────────────────────> NCP Subarea                   │
└───────────────────────────────────> E identifies NTRI Resource    │
                                                                    │
E13L0102 LU      LOCADDR=2,..        LU  -> 3270 Emulation          │
   │ │ └────────────────────────────> Sequ. Number (LOCADDR)        │
   │ └──────────────────────────────> Belongs to PU Sequ. Number <──┘
   └────────────────────────────────> L = Logical Unit Switched
--------------------------------------------------------------------
PU's/LU's for PC with 3270 Emulation and APPC/PC
--------------------------------------------------------------------
E13PE02   PU     IDBLK=017,..        PU  ->  3270 Emulation
E13L0202 LU      LOCADDR=2,..        LU  ->  3270 Emulation

E13PA02   PU     IDBLK=050,..        PU  ->  APPC/PC
E13L0203 LU      LOCADDR=3,..        LU  ->  APPC/PC
--------------------------------------------------------------------
PU/LU's for TRN Attached 3174
--------------------------------------------------------------------
E13PC03   PU     IDBLK=nnn,..        PU  ->  3174
E13L0302 LU      LOCADDR=2,..        LU  ->  Terminal 1
E13L0303 LU      LOCADDR=3,..        LU  ->  Terminal 2
E13L0304 LU      LOCADDR=4,..        LU  ->  Terminal 3
```

PATH - Naming Conventions for VTAM Swithed Major Node

```
PATH - only required if dial out is used
--------------------------------------------------------------------
E13D0201 PATH   DIALNO=nn044000abbccddd,GRPNM=nnnnn,...
   │ │ │                          └──────> see 'Soft' Address
   │ │ │                                   Conventions
   │ │ └────────────────────────────> First Path of several possible
   │ └──────────────────────────────> Path for PU id. 02
   └────────────────────────────────> D identifies PATH (Dial)
```

3.5 PERFORMANCE AND CAPACITY CONSIDERATIONS

3.5.1 General Token-Ring Performance Considerations

The Token-Ring Network architecture gives each ring station equal opportunity to transmit within a given priority. A major benefit of the token passing mechanism is that no collisions can occur.

Data is sent over the ring at a speed of 4 megabits per second (Mbps). Thus a Token or frame will be received at an adapter several thousand times per second and the user will be unaware of the 'token wait' time.

Also if an unusually large or heavily used 4 Mbps Token-Ring Network becomes overloaded, the flexibility of a Token-Ring Network and the IBM Cabling System allows easy reconfiguration of the ring to achieve a better load balance. For example, it is very easy to reconfigure two rings into three rings, interconnected with one or more additional bridges. This would give a potential total of 12 Mbps capacity instead of 8 Mbps. It is important, however, to evaluate the amount of traffic that may pass through bridges and the effect that distribution of Server and Gateway machines might have on a new configuration.

Only limited performance testing data was available during this study. Therefore the information given here is informal, and should be considered only as a reasonable estimate of Token-Ring Network performance.

Response times for the functions tested were shorter than those for similar functions using a 9600 bps SDLC link. However, the differences were not significant since, stress tests were not conducted in either situation. In each case, the application characteristics were more significant in determining response time than the type of connection.

The effective data rate (throughput) for an application depends on the complexity of the programming interface used for that application. The effective throughput may vary between 1.4 Mbps and 3.2 Mbps, depending on the application or system programming interface used, the size of the messages and the type (speed) of the workstation.

3.5.2 PC 3270 Emulation - Performance Considerations

One major consideration for 3270 Emulation would be to define each workstation as a Standalone configuration rather than to use Gateway and Network Station configurations. This reduces the potential bottleneck in the gateway and also improves availability. These benefits must be weighed against the offsetting requirement for additional switched major node (PU) definitions and corresponding virtual switched links. In addition, requirements for resource sharing (such as printer sharing) may be important.

3.5.3 IBM 3725 Capacity Planning

A configuration aid is available for the 3725 and 3720 Communication Controllers. This aid is normally used by IBM System Engineers, and its use is recommended prior to ordering TRSS components.

The following information is required for the Configurator:

- Planned Token-Ring Configuration
- Planned Host Attachment Configuration
- Type of Workstations
- Number of Workstations
- Applications used at Workstations
- Type of Messages for Host Communication
- Estimate of Message Traffic for Host Communication

3.6 NCP AND VTAM INSTALLATION AND PREPARATION

3.6.1 NEW NCP/EP Definition Facility (NDF) of ACF/SSP

Before preparing the new NCP source with NTRI support, familiarization with the new NCP/EP Definition Facility (NDF) of SSP V3 is recommended. This facility has a lot of advantages compared with ACF/SSP V2. The required SSP Version to generate ACF/NCP V4R2 with NTRI support is ACF/SSP V3R2.

NDF Functional Overview

NDF replaces the Stage 1 and Stage 2 generation process with a faster and more powerful process.

The NTRI Generation Process is part of NDF. It is not necessary to run a separate generation for NTRI resources. Several new NTRI keywords have been added to the NCP statements. The new statements are described in the next section of this document. NDF has also added a new OPTION keyword (NEWDEFN) which must be used for NTRI generation in order to generate the appropriate VTAMLST definitions for AUTOGENed lines. With NEWDEFN and the AUTOGEN facility of NDF, many new definitions will be created automatically by NDF.

3.6.2 Preparation of SSP and NCP

3.6.2.1 Preparation of ACF/NCP and ACF/SSP Installation Jobs

Follow the installation instructions published in the appropriate manuals and in the Program Directories of the products.

First, install ACF/SSP V3R2, then ACF/NCP V4R2.

- If SSP is installed first, the directories allocated (by using the sample jobs in the Program Directories) are large enough to run the APPLY Job for NCP.
- If NCP is installed first, the directories allocated are not large enough to run the APPLY Job for SSP.

3.6.2.2 NDF Definitions to generate NCP with NTRI Support

OPTIONS	NEWDEFN	= YES for NTRI. Generates output used to define NCP to VTAM. Must be first statement in source.
BUILD	MXRLINE	Defines the number of physical connections (TICs). Must equal the number of physical line definitions.
	MXVLINE	Defines the number of logical connections. Allow at least one for each PU to be defined with host connection.
	LOCALTO	Timeout value for attached (local) ring. Default of 1.5 seconds is satisfactory for IBM TRN's.
	REMOTTO	Timeout value for rings attached through bridges. Default is 2.5 seconds.
LUDRPOOL	NUMTYP2	Not a new parameter; adapt (increase) for SDLC switched links to reflect the maximum number of LU's that will be active at any one time.
GROUP	ECLTYPE	PHYSICAL/LOGICAL Required for NTRI. Specifies that this group defines a physical or logical connection. The physical groups must be defined first.
	AUTOGEN	Only for logical connections. Specifies the number of logical lines (LINE and PU) automatically generated by NDF. Can also be coded manually.
	PHYPORT	Only for logical connections. PHYPORT specifies the physical port address with which the lines in a logical group communicate. Value corresponds with the PORTADD parameter on the physical line. If PHYPORT=NONE (default) is specified, these Lines and PU's can be used with any port.
LINE		Only required if ECLTYPE = PHYSICAL The line definition references the physical line attached to the TIC; all parameters referenced apply to that TIC. The LINE definition is part of the PHYSICAL GROUP.
	ADDRESS	Corresponds to the physical position of the TRM. Valid address ranges for 3725 NTRI are 80-83, 112-115, 144-147,176-179, 208-211 and 240-243.
	PORTADD	Refers to the 'physical port' in the 372X. Used as reference for dial out. One port number required for each TIC. The number can be from 0 to 99.
	LOCADD	Specifies the 6 byte TRN adapter address. This address is used as the destination address if a TRN-Station is calling the host.
	RCVBUFC	Specifies the buffer capacity allocated to receive data from the ring. The value should be the largest frame plus 45 bytes, default is 1122, max.is 4095.
	MAXTSL	Specifies the maximum amount of data (PIU) that can be transmitted in one data transfer to a TRN-Station. The default is 265 bytes, max. is 1108. For APPC/PC, where larger frame sizes can be used, the MAXTSL value should be adapted.

3.6.2.3 NCP Definition Example

Definitions for OPTIONS,BUILD and LUDRPOOL

```
                                         Comments
           OPTIONS   NEWDEFN=YES           NTRI Generation
           BUILD     MXRLINE=3,            3 TIC's (in 2 diff.LAB's)
                     MXVLINE=60,           For 60  Stations for TRN
                     LOCALTO=2,
                     REMOTTO=6             Bridges in Network
                     ...                   Non NTRI BUILD Parameters
           LUDRPOOL  NUMTYP2=60            Max. 60 LU's concurrently
                                                active
```

Definitions for Physical Group

```
PGROUP     GROUP     ECLTYPE=PHYSICAL      Physical group for 3 TIC's
LAB1TIC1   LINE      ADDRESS=(80,FULL),    1st TIC
                     PORTADD=1,            Where LGROUP1 points to
                     MAXTSL=1108,          For larger (APPC/PC) frames
                     RCVBUFC=4095,         Recommended
                     LOCADD=400011301001   'Soft' Address for TIC 1
LAB1PU1    PU
LAB1LU1    LU        ISTATUS=INACTIVE

LAB1TIC2   LINE      ADDRESS=(81,FULL),    2nd TIC
                     PORTADD=2,            Where LGROUP2 points to
                     MAXTSL=1108,          For larger (APPC/PC) frames
                     RCVBUFC=4095,         Recommended
                     LOCADD=400011301002   'Soft' Address for TIC 2
LAB1PU2    PU
LAB1LU2    LU        ISTATUS=INACTIVE

LAB2TIC1   LINE      ADDRESS=(240,FULL),   3rd TIC (other LAB)
                     PORTADD=3,            Where LGROUP3 points to
                     RCVBUFC=4095,         Recommended
                     LOCADD=400011301003   'Soft' Address for TIC 3
LAB2PU1    PU
LAB2LU1    LU        ISTATUS=INACTIVE
```

Definitions for corresponding Logical Group's

```
LGROUP1    GROUP     ECLTYPE=LOGICAL,      Logical Group 1
                     AUTOGEN=10,           Autogen 10 log. Lines/PU's
                     CALL=INOUT,           Allow dial in and dial out
                     PHYPORT=1             Point to TIC 1 (PORTADD)
LGROUP2    GROUP     ECLTYPE=LOGICAL,      Logical Group 2
                     AUTOGEN=20,           Autogen 20 log. Lines/PU's
                     CALL=INOUT,           Allow dial-in and dial-out
                     PHYPORT=2             Point to TIC 2 (PORTADD)
LGROUP3    GROUP     ECLTYPE=LOGICAL,      Logical Group 3
                     AUTOGEN=30,           Autogen 30 log. Lines/PU's
                     CALL=INOUT,           Allow dial-in and dial-out
                     PHYPORT=3             Point to TIC 3 (PORTADD)
```

3.6.3 Preparation of VTAM Definitions

There are no new VTAM parameters for Token-Ring support. Each device attached to a Token-Ring is defined as if it were on a switched line. All definitions for TRN Stations will be placed in the Switched Major Node definition.

3.6.3.1 General Switched Major Node Considerations

A switched major node is used to define physical units which may communicate with the VTAM domain over a switched line or for the Token-Ring Network over NTRI.

For dial-out operation only, a PATH statement must be provided for each dial-out path associated with the physical unit, describing such information as the telephone number to be used. For devices attached to the Token- Ring Network, the locally administered adapter address is part of the dial-out number. The dial-out operation occurs when an application program requests a session with a logical unit whose switched physical unit is not already connected to VTAM.

Each PATH statement tells VTAM the line group and the dialing digits to be used. If the line group contains more than one dial-out line (TIC for Token-Ring), each line is tried in succession until contact is established. If contact cannot be established using the line group (TIC), VTAM tries again, using the line group identified by the next PATH statement.

In a multiple-domain network, a switched major node can be defined in one domain and treat its logical units as cross-domain resources in other domains.
A switched major node can also be defined in each domain that owns an NCP and has a dial port for a device.
If a switched major node is defined in only one domain, devices can dial into or be dialed out from only one host processor. If defined in more than one domain, they can dial into or be dialed out from each of the host processors in which the major node is defined.

VTAM XID exchange

VTAM constructs a 48-bit station ID that is used in XID exchange during the dial procedure. This station ID must be unique for each station within the network (not just within the major node).
The IDBLK and IDNUM values are used by VTAM to construct the station ID.

- IDBLK=identification block

  The identification block is a 12-bit string assigned to a specific device and is required. The number must be obtained from the component description for the device. For a PC which is able to emulate or support different functions, the IDBLK value is usually hard-coded in the appropriate PC application.

  For PC3270 Emulation, this value must be IDBLK=017.

  For APPC/PC, this value must be IDBLK=050.

- IDNUM=identification number

  The identification number is a 20 bit information field assigned to the station being defined and is also required. For NTRI the value for the number can be defined by the user but it must be unique. This number must be specified in the PU definition of the switched major node and must match the corresponding number specified during setup of the communication subsystem in the PC.

- The Station ID (XID) Structure:

Bits 0-3	Reserved
Bits 4-7	PUTYPE
Bits 8-15	'00'
Bits 16-27	IDBLK
Bits 28-47	IDNUM

3.6.3.2 VTAM Switched Major Node Definitions for TRN Devices

If the host is initiating the connection then it must know the locally administered address of the Token-Ring station. In addition the SAP of the destination device application and the physical port (TIC) are required for the DIALNO parameter.
For device initiated connections, no PATH statement is required or used in the VTAM switched major node. In this case the calling TRN station places the locally administered address of the TIC to be used in the destination address field of the Token-Ring frame.

If PC 3270 Emulation is running at a workstation, the session will normally be initiated by the device (user). For restart after an error it may also be useful to have the possibility of host initiated connections for 3270 Emulation. This is possible if a PATH statement is defined. The outgoing call will be executed after the operator activates the LU.

If APPC/PC is running at the workstation, connections can be initialized by the host application or by the workstation application. A PATH statement is required in the VTAM definitions for host-initiated APPC/PC connections,

Switched Major Node Definition Keypoints

- The ADDR parameter is required on the PU statement but the value is not used. For real switched lines this address refers to the SDLC station address.

- The MAXDATA parameter value is not used for switched PU's of NTRI resources and can be omitted.

- The DIALNO parameter in the PATH statement defines several things. The first byte is the port number (TIC) to which this station is connected. This number corresponds to the port number (PORTADD) in the NCP definitions. The second byte defines the SAP address of the terminal. This value is normally 04 for an SNA SAP. The last six bytes of the DIALNO correspond to the locally administered address of the Token-Ring station.

- The IDBLK and IDNUM values are used for the VTAM XID exchange.

- If both PC 3270 Emulation and APPC/PC are used on a PC, both PU's of the example must be defined for that network PC. The PATH statement for the APPC/PC PU is only required if dial-out has to be supported. The second PATH defines an alternate path (other TIC).

3.6.3.3 VTAM Switched Major Node Definition Examples

VBUILD for 60 Workstation PU's and 3 TIC's
PC with 3270 Emulation and only PC initiated connection

```
SWINOUT    VBUILD    MAXGRP=3,               3 GRPNM's (TIC's)
                     MAXNO=60,               60 DIALNO's
                     TYPE=SWNET              Required
```

```
PC3270PU   PU        ADDR=04,                Reqd. but value not used
                     IDBLK=017,              017 reqd. for 3270 Emul.
                     IDNUM=E0001,            As customized in 3270 Emul.
                     PUTYPE=2,
* ——————>            MAXDATA=nnn,            Not used for NTRI
                     DLOGMOD=QBITON..,       Query bit req'd for
                     ...                       File Transfer
                     ...

PC3270LU   LU        LOCADDR=02
```

PC with APPC/PC, host and PC initiated connections

```
SWINOUT    VBUILD    MAXGRP=...
                     ...
PCAPPCPU   PU        ADDR=nn,              Reqd. but value not used
                     IDBLK=050,            050 reqd. for APPC/PC
                     IDNUM=A0001,          As set in APPC/PC Sub-Sys.
                     PUTYPE=2,             PU 2.1 Emul.Mode > dest.PU
                     MAXPATH=2,            Alternate TIC for dial out
                     ...
PATH1      PATH      DIALNO=0104400031301001,
                            ── ── ────────────
                            │  │       └────> Local Admin. Station
                            │  │              Address
                            │  └────────────> Dest. SAP
                            └───────────────> Port (TIC)

                     GRPNM=LGROUP1,
                     ...
PATH2      PATH      DIALNO=0204400031301001,
                     GRPNM=LGROUP2,
                     ...
PCAPPCLU   LU        LOCADDR=02
```

PC with PC 3270 Emulation and APPC/PC

```
Specify both PC 3270 and APPC/PC definitions for this Station
--------------------------------------------------------------
SWINOUT    VBUILD    MAXGRP=...
PC3270PU   PU        ADDR=nn,              Required
                     ...
PATH1      PATH      DIALNO=0104400031301001,
                     ...
PC3270LU   LU        LOCADDR=02

PCAPPCPU   PU        ADDR=nn,              Required
                     ...
PATH1      PATH      DIALNO=0104400031301001,
                     ...
PCAPPCLU   LU        LOCADDR=02
```

4.0 PLANNING - TOKEN-RING NETWORK ENVIRONMENT

This chapter consists of a list of Token-Ring components and products with some planning considerations. It does not provide detailed Token-Ring planning information.

4.1 TOKEN-RING DEVICES AND PRODUCT SELECTION

IBM Devices with Token-Ring Attachment capability (Status August, 1986)

IBM PC's
IBM 3270 PC
IBM 3720 Communication Controller
IBM 3725 Communica.ion Controller
IBM 3174-3R Control Unit
IBM S/36 via Gateway PC
IBM S/1 via Gateway PC
IBM Token-Ring Bridge (IBM PC-AT is used for this function)

Selection of Token-Ring Adapter for IBM PC

- The Token-Ring Adapter II has an extra 8k of RAM. This is the only difference between it and Type I Adapter.
- Two Type II Adapters are required for a Bridge station
- A Type II Adapter is recommended for communication with heavy traffic where additional buffers can be utilized (e.g. for LAN Servers).

Selection of IBM PC Type

- The IBM PC-AT is required for the Bridge function
- The IBM PC-AT is recommended for server and gateway functions and for stations where improved performance for higher traffic is requested
- The current DOS limitation of 640 K might be a restriction for combining some functions in a single workstation.

Selection of PC Products with Host Communication

- PC 3270 Emulation - provides 3270 data stream emulation, Host Print support, File Transfer and a Application Program Interface (API).
- Application Program to Program Communication for PC (APPC/PC) - provides LU 6.2 support.
- PC Requestors provide functions for a PC user to request services or data from an IBM System/370 computer using Enhanced Connectivity Facilities.

See Chapter 5, Workstation Software, for more information.

4.2 TOKEN-RING NETWORK AND WORKSTATION COMPONENTS

4.2.1 Token-Ring Network - LAN Components

Refer to the Token-Ring Network Introduction and Planning Guide - GA27-3677 for more detailed information.

- Multi-Station Access Units (MSAU's) IBM 8228
  - Part number (p/n) 6091014
  - Minimum of one per ring
  - Eight TRN stations can be plugged in one MSAU
  - Use Installation Aid (comes with 8228) before attaching lobes
- Patch Cables between MSAU's
  - p/n 8642551 (8 FT), 8642552 (30 FT)
  - One per 8228 when using multiple MSAU's
  - 8218 and 8219 repeaters may be needed for long distances between MSAU's (e.g. distances greater than 600 meters)
- TRN Lobe Cables
  - IBM Cabling System Cables for establishment wiring to the wiring closet.
- Cables from PC Network Station adapters to Token-Ring (MSAU)
  - p/n 6339098
  - One per attached PC
- Cables from TIC's to Token-Ring (MSAU)
  - Cable group number 1666 - 3725 cable
  - One per attached TIC

Publication references for Token-Ring Network planning

Token-Ring Network Introduction and Planning Guide - GA27-3677

TIP Media Guide (use new version = suffix 2) - GA27-3714-2

Token-Ring Network Installation Guide - GA27-3678

4.2.2 Token-Ring Network Adapter for IBM PC and Adapter Software

- Token-Ring Network Adapter for PC >> p/n Type I = 6339100
  >> p/n Type II = 67X0438

- The product includes a TRN PC Adapter, a Diskette with TOKREUI (Adapter Support Program) and an Operator Guide.

- Adapter Switches must be set as explained in the Guide to Operations. In our installation, the default switch setting for TRN was satisfactory. See note below for more information.

- Requires 7K of storage

- Verify adapter operations and switch settings by running adapter diagnostics; use ring diagnostics to verify ring and neighbor stations.
  Both are part of TOKREUI diskette.

- Refer to the IBM Token-Ring Network Problem Determination Guide (SY27-0280) about how to run and interpret the diagnostic functions

- TOKREUI.COM must loaded prior to use of the adapter (except diagnostics)

- TOKREUI.COM cannot be loaded from a server disk (one copy is required for each user)

- If Locally Administered adapter addresses are to be used, the 6 byte (12 digit) Adapter Address must be entered in the TOKREUI load command.

- Adapter interrupt level settings

Adapter Type	Supporting Interrupt Levels
SDLC Adapter Card	3 and 4
IBM PC Network Adapter	2 and 3
IBM Token-Ring Netw. PC Adapter	2, 3 or 7
IBM 3278/79 Emulation Adapter	2

Note:

In case of conflicting interrupt levels, unpredictable results may occur. Workstations with an SDLC Adapter and a Token-Ring Adapter can have only one Network card present with jumpers set to use level 2.
If two Token-Ring Adapters exist in one Workstation, (as for a Bridge PC) the interrupt level must be switched according to the installation instructions.

4.2.3 Products for PC, TRN Communication and TRN Management

PC DOS 3.2 (p/n 6280057)

- Required for Token-Ring Network environments
- One copy per LAN user required
- Use the REPLACE command to update the current fixed disk
- Copy SHARE.EXE from DOS 3.2 into the NETWORK directory (only applicable if PC LAN program is used)
- Storage requirements 45-60K (see DOS publications for details)

NETBIOS Program Product (p/n 6467037)

- The product includes a diskette with NETBIOS and a Users Guide
- Required if programs written to NETBIOS interface are used
- Programs using NETBIOS - Examples
  - * PC 3270 Emulation for Gateway or Network stations
  - * PC LAN Program V1.1
- Programs not using NETBIOS - Examples
  - * APPC/PC
  - * PC 3270 Emulation if generated as Standalone station
- NETBIOS must be loaded after TOKREUI by the NETBEUI command
- Parameters required depend on type of application used
- Must be loaded prior to any NETBIOS application
- Requires 46K of storage
- Cannot be loaded from a server disk (one copy for each user)

PC LAN Program V1.1 (p/n 6280083)

- Required for LAN services like File Server, Print Server or Communication Server
- Includes 3 diskettes and Users Manual
- One copy required for each LAN user
- Storage requirement depends on functions selected during customizing. It is between 128K (Redirector) and 320K (Servers).

Token-Ring Network Bridge Program (p/n 6403831)
or
Bridge Installation Kit (Adapters + SW) (p/n 6476041)

- One copy for each bridge

Token-Ring Network Manager Program (p/n 6476046)

- One copy for each ring

4.2.4 PC Products for Host Communication

See workstation software in Chapter 5 for more information.

IBM PC 3270 Emulation V3 (p/n 59X9969)

- V3 is required for Token-Ring LAN environments.
- Required for most host communication services like 3270 emulation, file transfer and host printer support.
- Includes 2 diskettes and Users Manual

APPC/PC (p/n 6467038)

- Release 1 is currently available.

IBM PC Requestors (p/n 6316993)

4.2.5 Token-Ring Network Configurations and Management

Components to be considered for Token-Ring Environment

Naming Considerations

IBM Token-Ring Network Bridges

IBM Token-Ring Network Management

IBM Token-Ring Network Manager Program

IBM Token-Ring Network Routing Considerations

IBM Token-Ring Network Backup Considerations

Most of these subjects are discussed in the Publication:

- *IBM Token-Ring Network Bridges and Management* (GG24-3062)

Additional considerations for Token-Ring Networks attached to System/370 Hosts

Backup considerations	--> See Chapter 6 in this bulletin
CNM considerations	--> See Chapter 7 in this bulletin
Recovery considerations	--> See Chapters 6 and 7 in this bulletin
Naming considerations	--> See Chapter 3 in this bulletin
Routing considerations	--> See Chapters 1 and 6 in this bulletin

5.0 WORKSTATION SOFTWARE FOR HOST COMMUNICATION

In this chapter only workstation products with host communication capabilities are described. Products not communicating with a subarea host (e.g. IBM PC LAN program or NETBIOS) are not considered.

Workstation Products with host communication:

- IBM PC 3270 Emulation Program Version 3
- APPC/PC program

Workstation Products using PC3270 Emulation for host communication:

- Personal Services/PC (PSPC) V.1 R.2
- PROFS/PC Support
- PC Requesters (part of Enhanced Connectivity Facility products)

PS/PC and PROFS/PC are PC products used to access the subarea host office system products DISOSS/370 (MVS) or PROFS (VM). Through these facilities, document distribution and library services between subarea hosts and workstations can function.

The Enhanced Connectivity Facility (ECF) enhances the capability for resource sharing among IBM PC's and System/370 computers by providing PC users with a uniform structure for exchanging data and access resources on subarea hosts (MVS,VM). Highlights about this new product family can be found in this chapter.

Token-Ring Adapter Initialization

A PC program using the Token-Ring Adapter Support Interface cannot begin data transmission (including host communication) until a sequence of operations to load TOKREUI and to open the adapter (connect the PC to the Token-Ring) has been executed.
The sequence of operations includes:

- Loading TOKREUI: loading the adapter support code and optionally specifying a locally administered address to be used.
- Adapter initialization: resetting the adapter and performing of initial tests.
- Opening the Adapter: setting the adapter ready for ring communication. During this process, a 'click' may be heard in the MSAU indicating that the relay has switched.
- Opening the Service Access Point (SAP): The type of LLC interface will be defined through the SAP. It provides access to the LLC and lower level services provided by TOKREUI and the Token-Ring Adapter.
- Opening one or more Link Stations: to allocate resources used for protocols required by applications using data link or connection-oriented services.

It is possible to have multiple programs coexisting in the PC and communicating over the same Token-Ring Adapter. However, only one of these programs may perform the initialization and opening of the adapter.

5.1.1 Application Program Interfaces of TRN Stations

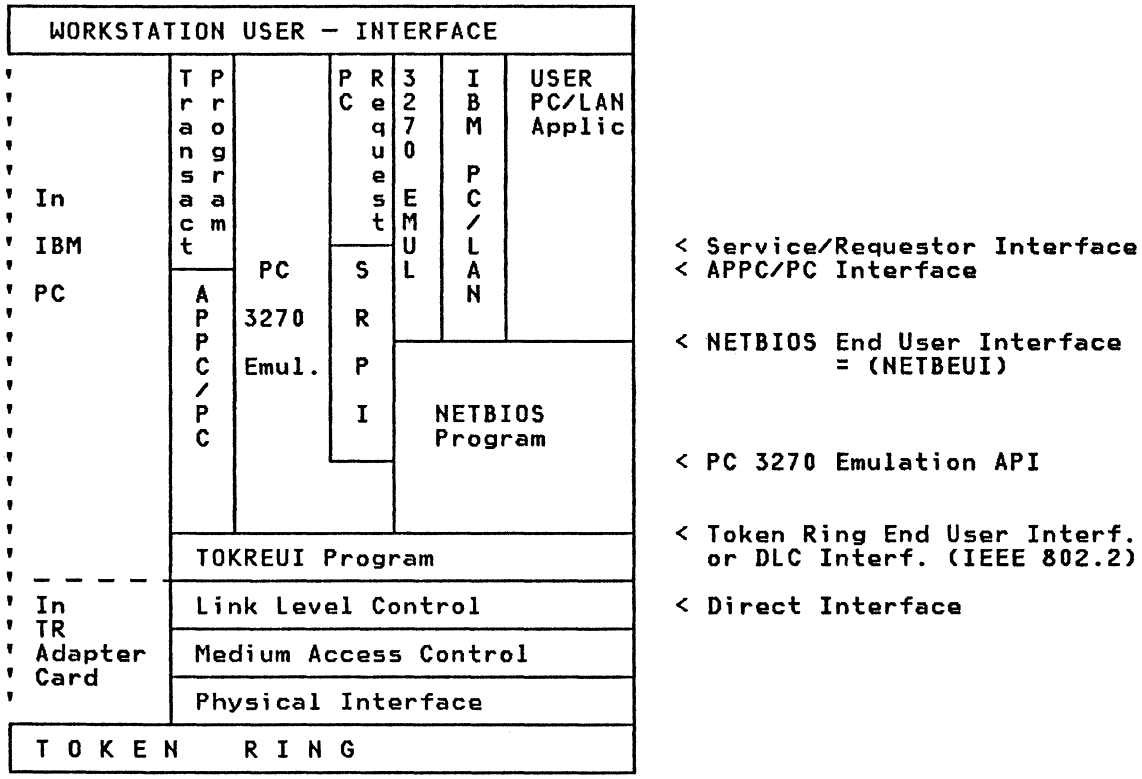


Figure 12. TRN - Application Interfaces for Workstation PC

Comments:

- PC 3270 Emulation: the Standalone Station configuration option of PC 3270 Emulation interfaces directly to the TOKREUI program and does not need the NETBIOS program, while the Gateway and Network Station configurations interface to NETBIOS.
- DLC Interface: The DLC code itself is implemented in the microcode which is resident in the Adapter Card. The TOKREUI program, shipped with the adapter, is the PC resident driver through which an application program passes requests to the DLC code.
- DIRECT and DLC Interfaces are low-level interfaces and are normally not used by a User.

5.2 IBM PC 3270 EMULATION

The main function of PC 3270 Emulation is to allow a PC user access to subarea host 3270 applications.
PC 3270 Emulation also provides other useful functions for PC Users:

- Downloading and Uploading of files

- Printing of subarea host files
- Interface for Office System and ECF products

PC 3270 Emulation provides LU Type 2 (for 327x Terminal Emulation support) and LU Types 1 or 3 (for printer support) for interfacing to SNA hosts.

5.2.1 Configurations and Functional Overview

5.2.1.1 Attachments and Configuration Types

PC 3270 Emulation supports the following attachments

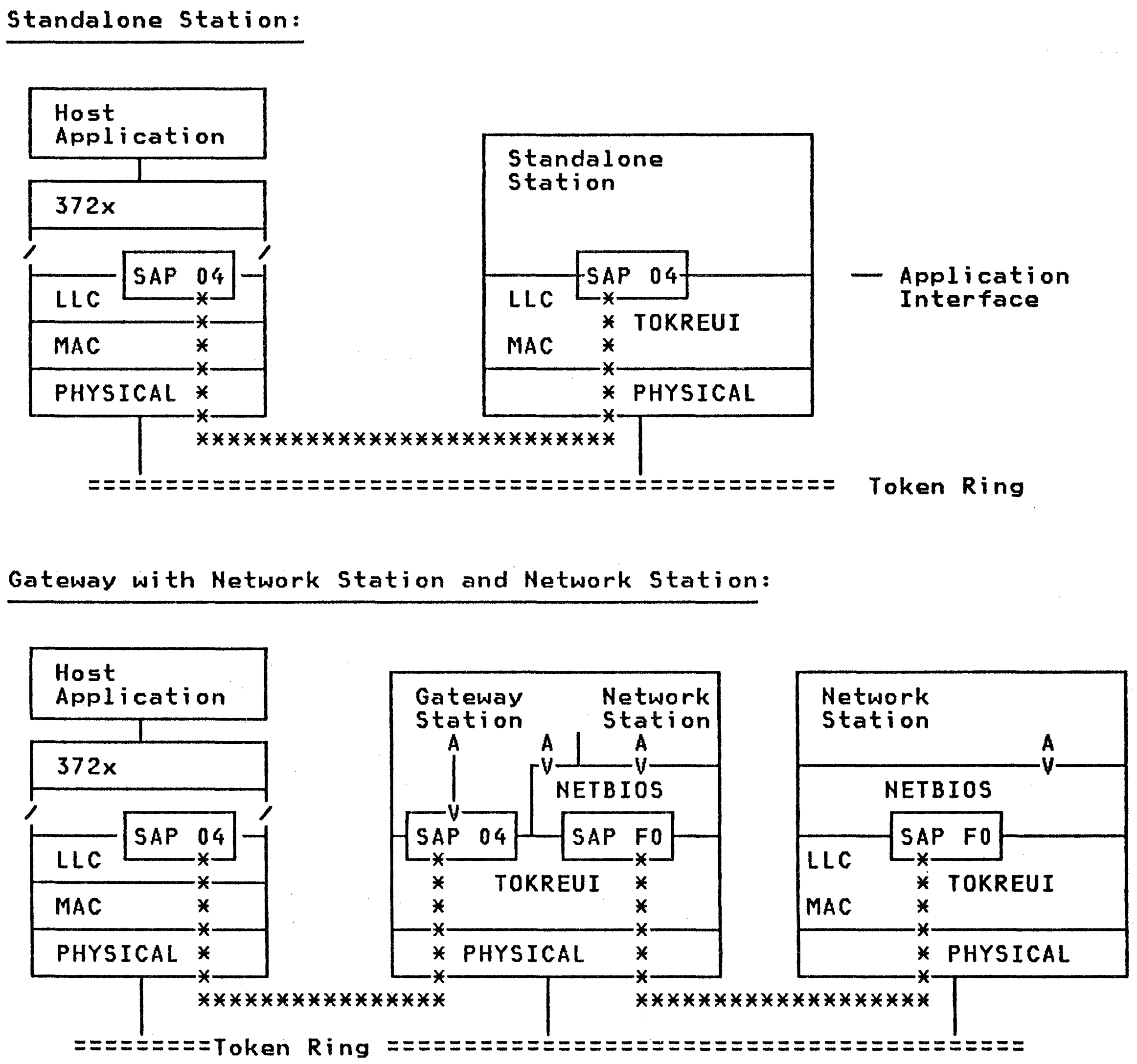
- SDLC Adapter: SNA Host Network Environment (Lines)
- DFT: Subarea Host Network Environment (3x74/Coax)
- Token-Ring: Token-Ring Network Environment

Note: Only Token-Ring Attachment is considered in this document

PC 3270 Emulation can be configured for the following station types:

- Standalone Station: Communicates directly with a subarea host
- Gateway Station: Acts as Gateway for Network Stations
- Network Station: Communicates via Gateway with a subarea host
- Gateway and Network Station: A Gateway Station can be configured to be used as a Network Station in addition to the Gateway function.

5.2.1.2 PC 3270 Emulation Interfaces to LLC


Standalone Station:

Gateway with Network Station and Network Station:

Figure 13. PC 3270 Emulation - Token Ring Interfaces

- SAP 04 is defined for communication with SNA (Path Control)
- SAP F0 is defined for NETBIOS and is used for communication between workstations

5.2.1.3 Request Tasks Facility

PC 3270 Emulation provides a facility to initiate or modify some program options. The Request Task Panel can be invoked by pressing the Request Key (normally, F10).

The functions provided by the Request Task Facility are:

ID	ITEM
* a	Name File for 3270 Screen Save
* b	Name File for 3270 Print to Disk
* c	3270 Printer Request
* d	File Transfer
* e	Display Status of Network Stations
f	Revise List of Network Stations
* g	Data Trace 1 = Start 2 = Stop
* h	Screen Mode 1 = Color 2 = Black and White

Function availability at a particular workstation depends upon specifications made during PC 3270 Emulation Product customization. Available Options are marked with an asterisk.

5.2.1.4 Application Program Interface

The Application Program Interface (API) of the PC 3270 Emulation program enables an application program in the PC to intercept and manipulate messages to or from the host.

The interface is a Presentation Space Application Program Interface (PS API) which provides a set of services. Using these services, a programmer can develop applications to improve the user communication interface by automating some operations, or distributing some processing. The primary API service is the 'string copy service', which copies data between the Presentation Space (PS) and the PC application. Data can be sent directly to a host by using the Write Keystroke service.

A PS API application program might perform the following functions:

- Analyze host output: copy the presentation space into a buffer area of the application. Use the String Copy Service, scan the screen copy and act on the data sent from the host.

- Send data to the host: copy a character string from an application buffer area into a presentation space by using the String Copy Services. Simulate hitting of the Enter key, and use Write Keystroke to send data to the sub-area host.

5.2.1.5 File Transfer Facility

- Supports Upload/Download of files to/from TSO, CMS and CICS
- An equivalent product is required at the host site
- File Transfer is an optional function. If customized, it can be invoked via the Request Task panel (F10).
- The basic format is: 'SEND/RECEIVE dosname hostname (options'
- The Presentation Services 'Query Bit' must be set on in the VTAM Logmode entry to support File Transfer

5.2.1.6 Host Session Key Definition and Keyboard Remapping

A 3270 keyboard has at least twelve PF keys and several keys for special functions (such as Clear, Enter, PA1).

The PC keyboard usually has ten function keys and several PC-specific keys, but not the all the keys available on a 3270 keyboard.

The standard keyboard layout used during 3270 host sessions is shown on the next page including the keyboard definitions of the Enhanced Keyboard.

A User can define his own layout (instead of using the standard ones) by defining the changes in a description data set.

The PC 3270 Emulation Operation section later in this chapter lists standard host session key definitions.

5.2.2 Planning

5.2.2.1 Coexistence and Dependencies on other Products

- PC 3270 Emulation V3 is required for host connection over the Token-Ring Network.

- DOS 3.2 is required for Token-Ring Network products.

- Subarea host software for File Transfer Program:

  For MVS/TSO: IBM 3270 PC File Transfer Program, Rel.1 (5665-311)
  For VM/SP: IBM 3270 PC File Transfer Program, Rel.1 (5664-281)
  For CICS: IBM CICS/VS PC File Transfer Program, Rel.1 (5798-DQH)

  Note: APARS are required in host products to use File Transfer facility

- PC 3270 Emulation V3 can coexist with the PC LAN Program Rel.1.1, even if PC LAN is configured as Redirector or as Server.

- The PC 3270 Emulation program cannot be used from a shared disk because one module is updated with the Communication profile parameters of the specific station. Each of the Directories must contain the complete code, and they cannot be connected (APPEND).

- The PC LAN program will not start if another application has already initialized the Token-Ring adapter; it therefore must be started before PC 3270 Emulation.

- PC 3270 Emulation checks the status of the Token-Ring Network adapter. If the adapter has been opened by another application, it will not try to reopen it, but will share the adapter (if possible).

A prerequisite for coexistence (multiple programs loaded at same time) is that sufficient storage is available in the workstation. Performance, product compatibility, and storage requirements must be considered for concurrent program execution.

5.2.2.2 Storage Requirements

The following memory is required to run PC 3270 Emulation V3 configured with a Token-Ring Communication Attachment. The values used are based upon early information, and may be slightly different in the program product when installed by users.

- Base Memory requirements:

- Standalone Station	165K
- Network Station	159K
- Gateway Station	188K
- Gateway with Network Station	217K

__Note__: To access Token-Ring DLC, the adapter software (TOKREUI) must be loaded. TOKREUI requires 7K.
For Gateway and Network stations, NETBIOS is required to access DLC. NETBIOS requires an additional 46K.

- Memory requirements for PC 3270 Emulation features:

- Alternate Tasks	5K
- Printer use/support	14K
- File Transfer	18K
- API	8K - 30K

5.2.2.3 Standalone Station vs Network Station

In a Token-Ring environment, a PC configured as a Standalone station has several advantages over a PC configured as a Network station. Standalone will normally be the configuration of choice.

Main advantages of a Standalone station:

- No additional station involved for the Gateway function, which will improve Performance and Availability, while eliminating the cost associated with a dedicated workstation

- Less workstation memory required because NETBIOS is not required for support of dependent Network Stations

A disadvantage of using Standalone stations could be that each PC is represented with a PU and LU in the IBM 372x and in NCP. There may be some situations where a configuration with Gateway and Network stations would be preferable. These include situations where the number of stations is very large, and thus the number of switched PU definitions is excessive. There may also be situations in which the sharing of such resources as high quality printers requires that the shared resource be defined on the same PU as the Display LU's which share it.

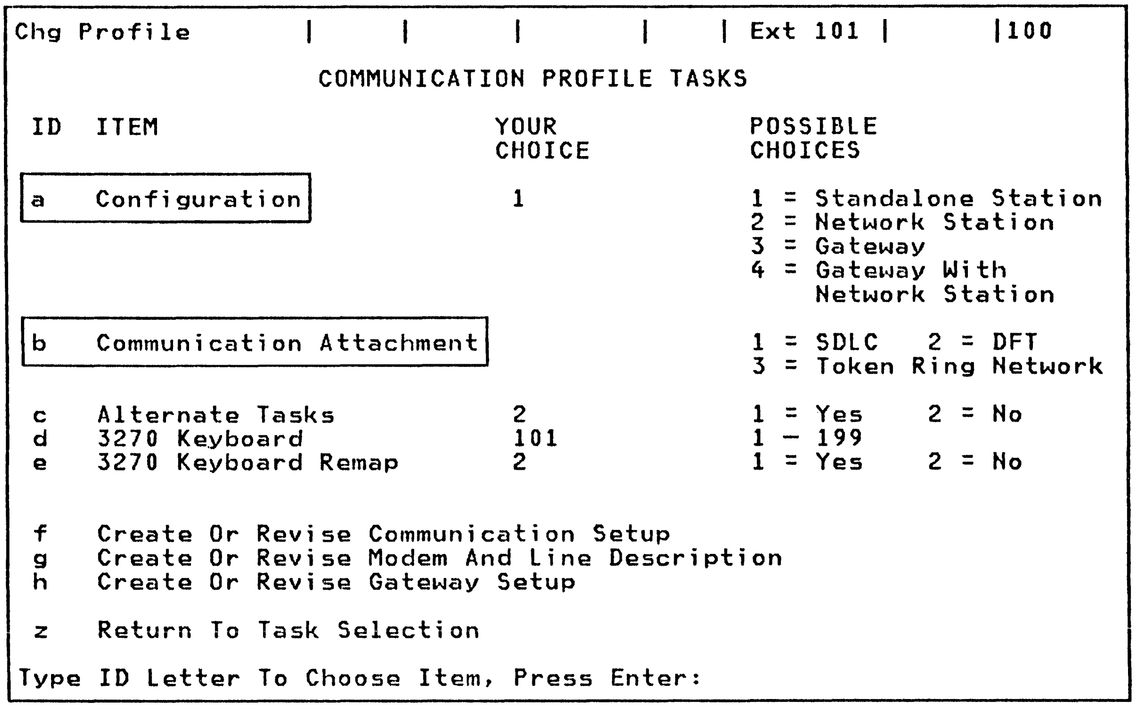
__Publication Reference__

__Guide to IBM PC 3270 Emulation Program Version 2__ GG24-3038

This Publication is based on PC 3270 Emulation V2; Token-Ring host attachment is not considered, but most of the other information in this guide is valid for PC3270 Emulation V3.

5.2.3 Preparation - Communication Profile Tasks

Communication Profile Task Panel — Example for Standalone Station



```
Chg Profile        |     |      |      |    | Ext 101 |      |100

                        COMMUNICATION PROFILE TASKS

 ID  ITEM                     YOUR           POSSIBLE
                              CHOICE         CHOICES

 a   Configuration            1              1 = Standalone Station
                                             2 = Network Station
                                             3 = Gateway
                                             4 = Gateway With
                                                 Network Station

 b   Communication Attachment                1 = SDLC    2 = DFT
                                             3 = Token Ring Network

 c   Alternate Tasks          2              1 = Yes     2 = No
 d   3270 Keyboard            101            1 — 199
 e   3270 Keyboard Remap      2              1 = Yes     2 = No

 f   Create Or Revise Communication Setup
 g   Create Or Revise Modem And Line Description
 h   Create Or Revise Gateway Setup

 z   Return To Task Selection

Type ID Letter To Choose Item, Press Enter:
```

This is the Communications Profile Task menu from which the station configuration and the communications attachment are specified.

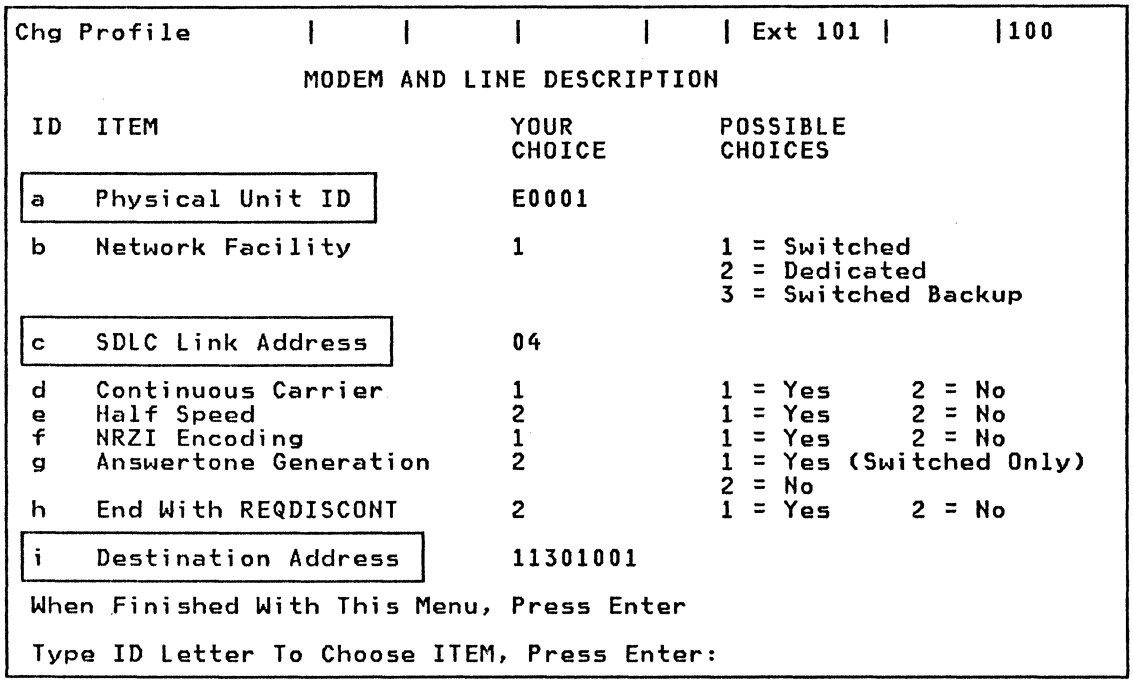
- There are two choices for display stations providing PU communications with the host, Standalone or Gateway.

- Use the Gateway configuration if that station is to act as a Gateway for other network stations on this or another ring.

- The Standalone configuration does not require NETBIOS.

- For the communications attachment, selection '3' is required for IBM Token-Ring Network.

Modem and Line Description Panel



```
Chg Profile        |     |      |       |     | Ext 101 |       |100

                   MODEM AND LINE DESCRIPTION

 ID  ITEM                        YOUR          POSSIBLE
                                 CHOICE        CHOICES

 a   Physical Unit ID            E0001

 b   Network Facility            1             1 = Switched
                                               2 = Dedicated
                                               3 = Switched Backup

 c   SDLC Link Address           04

 d   Continuous Carrier          1             1 = Yes      2 = No
 e   Half Speed                  2             1 = Yes      2 = No
 f   NRZI Encoding               1             1 = Yes      2 = No
 g   Answertone Generation       2             1 = Yes (Switched Only)
                                               2 = No
 h   End With REQDISCONT         2             1 = Yes      2 = No

 i   Destination Address         11301001

 When Finished With This Menu, Press Enter

 Type ID Letter To Choose ITEM, Press Enter:
```

The Modem And Line Description Menu is selected from item 'e' on the Communication Profile Tasks Menu (previous page).

- The Physical Unit ID must match with the IDNUM value in the VTAM Switched Major Node.

- The SDLC Station address must be the SAP value of the host which is X'04' for the NCP.

- The following parameters are not applicable in the Token-Ring Environment

  Network Facility
  Continuous Carrier
  Half Speed
  NRZI Coding
  Answertone Generation
  End With REQDISCONT

- The Destination Address is the last 4 bytes of the Token-Ring address (TIC) of the 3725. This value must match the last 4 bytes (8 digits) of the LOCADD parameter of the Physical Line definition in the NCP Generation. PC 3270 Emulation automatically adds the leading X'4000'.

5.2.4 PC 3270 Emulation - User Operation

5.2.4.1 Operating Scenarios

Commands to start a station configured as Standalone

```
- CD \TOKENV1             Change to TOKREUI Directory
- TOKREUI 4000nnnnnnnn    Load TRN Adapter with 'soft' address
- CD \PC3270V3            Change to PC 3270 Emulation Directory
- PSCPG                   Load 3270 Emulation to get primary panel
- a                       Select 'a' to start host communication
- F2                      F2 to get primary panel
- z                       Select 'z' to end 3270 Emulation
```

Commands to start a station configured as Gateway with Network Stations

```
- CD \TOKENV1             Change to TOKREUI Directory
- TOKREUI 4000nnnnnnnn    Load TRN Adapter with soft address
- CD \NETBIOS             Change to NETBIOS Directory
- NETBEUI  ,1,1           Load NETBIOS program with parameter
- CD \PC3270V3            Change to PC 3270 Emulation Directory
- PSCPG                   Load 3270 Emulation to get primary panel
- a                       Select a for host communication
```

Note: For fast path,'PSCPG ,F' can be used to start direct the host connection without selecting 'a'.

File Transfer Command

- Before entering the File Transfer command at the workstation, the related session in the host (TSO,CMS,CICS) must not be in a formatted screen or bracket state. For example, the 'ready' screen in TSO is required rather than ISPF panels.

- The File Transfer command can be selected at the PC3270 Emulation Request Task Panel which is invoked by pressing the Request (F10) key.

- The basic format of the File Transfer command is:

  'SEND/RECEIVE dosname hostname (options' where

```
- SEND (PC to host) or RECEIVE (host to PC) indicates the direction
- DOSNAME:   Is the PC data set name e.g. 'SAMPLE.FIL'
- HOSTNAME:  Is the host data set name e.g. 'USER1.PDS(MEMBER1)'
- OPTIONS:   For CMS and TSO
   ASCII  - used for translation (ASCII to EBCDIC or vice versa)
   CRLF   - insertion/removal of Carriage Return and Line Feed
            characters
   APPEND - add transmitted data to an existing data set
   RECFM  - Record Format of data set sent to a host
   LRECL  - Record Length for new host data set
- OPTIONS:   Especially for TSO
   BLKSIZE,SPACE AVBLOCKS, TRACKS and CYLINDERS give the necessary
   information to allocate a host data set for SEND operations.
   /PASSWORD is required for password-protected host data sets.
```

5.2.4.2 Standard Host Session Key Definition

Enhanced Keyboard	PC XT/AT Keyboard	3270 Emulation Keyboard Function
Enter	END	3270 Enter
Esc	Ctl & F7	Attn
Alt F8	F2	End Task - Exit 3270 Session
Alt F3	F3	DIR Key
N/A	F4	PF Key Select (F4 and 7 = PF7)
Alt F2	F5	Save
Alt F9	F6	Cursor Select
PAUSE	F7	Clear
End	F8	Erase EOF
Alt F12	F9	Reset
Alt F1	F10	Request 3270 Task Screen
Pg Up (H)	Ctl & F1	PA1
Pg Dn (H)	Ctl & F2	PA2
Sh Insert	Ctl & F3	Dup
Sh Home	Ctl & F4	Field Mark
Sh Prt Sc	Ctl & F5	Print
Alt Sc Lk	Ctl & F6	Sys Request
Alt End	Ctl & F8	Erase Input
Alt F11	Ctl & F9	Dev Cancel
Alt F4	Ctl & F10	Msg
Rt Cntl	ALT/ESC	Switches Between DOS And 3270

5.3 APPC/PC

The functions provided by the implementation of SNA LU 6.2 are referred to as Advanced Program-to-Program Communication (APPC). APPC is designed to provide enhanced SNA support for distributed transaction processing programs. This is achieved through a programming interface which allows applications to communicate with other applications on a transaction basis.

APPC/PC is the implementation of APPC for the IBM PC. It is a licensed product supporting conversations between applications running on an IBM PC and:

- Subarea host - CICS
- S/36
- S/38
- Series/1
- Another IBM PC with APPC/PC

APPC/PC provides a general purpose interface that is common throughout a large number of IBM products. It is a 'native' SNA programming interface for the PC, as opposed to a terminal emulation capability. Therefore, APPC/PC should be considered for use in data processing applications where distributed processing functions are required.

5.3.1 Key functions and considerations for using APPC/PC

- Requirement for conversational mode transaction processing
- Requirement for synchronized processing between remote SNA nodes
- Requirement for processing which is distributed across multiple nodes
- Requirement for Local or Remote transaction program initiation
- Importance of a standard set of verbs to establish and maintain the Transaction Program Conversations
- Importance of a high level of function to maintain integrity of data across the conversations

  A typical environment for such an application would have a need to perform synchronized transaction updates at a workstation and at other workstations or host data bases.

5.3.2 Supported PC Attachments

APPC/PC can be used with the following attachments:

- Token-Ring Adapter
- SDLC Adapter

The type of attachment has no impact on the APPC/PC programming interface which implies that a Transaction Program written for a SDLC connection can also be used for a Token-Ring connection, (either for a peer to peer or a host connection).

When a PC is equipped with both a Token-Ring adapter and an SDLC adapter, APPC/PC may use both facilities concurrently.

5.3.3 Implementation in SNA

APPC/PC supports peer-to-peer connections to other PU 2.1 nodes and host connections via a boundary function node such as a 372x/NCP. On peer-SDLC connections, APPC/PC can function as either a primary link station (point-to-point only) or a secondary link station. APPC/PC also supports role negotiation at link-level contact so that its link-station role need not to be predefined.

When attached to a subarea network, APPC/PC functions as a PU 2.0, necessitating a secondary SDLC station role. Parallel sessions are not supported in this environment, and only one session is possible at a time between an LU in the APPC/PC node and CICS. This session must be bound by CICS. However, as a substitute for parallel sessions, multiple LU's can be defined in APPC/PC, each having its own session with CICS concurrently.

5.3.3.1 Conversation and Session Operation

Two remote applications exchanging data using LU 6.2 constitute a 'conversation'. A conversation is supported by a 'session' available between two Logical Units.

Definitions from an SNA point of view:

- A session between two Logical Units is established when the Secondary Logical Unit accepts the BIND sent by the Primary Logical Unit.
- A conversation is a data exchange between brackets. The conversation begins with a 'begin bracket' and ends with the 'end bracket' command.

As only one bracket pair is possible at a time in a session, only one conversation can flow at a time, and a session will be serially occupied by a conversation.

5.3.3.2 Session initialization

A session is a connection between two logical units. One is the Primary Logical Unit (PLU) which sends the BIND request, the other is the Secondary Logical Unit (SLU) which receives the BIND.

A session between two Logical Units can be established in several ways:

- From the SLU, via the SNA command INIT-SELF. This means that APPC/PC can request the establishment of a session with CICS.
- From VTAM. The VTAM operator can request CICS to start a session with the PC SLU via a VARY NET command (e.g.'VARY NET ACT,ID=TRLU1PC1, LOGON=CICSA'). Instead of entering the command, the parameter LOGAPPL of the LU macro can be used to produce the same effect.
- From CICS. Several possibilities exist to start a session from CICS. One is to specify CONNECT=AUTO in the TCT TYPE=SYSTEM table of CICS. In this case, each time CICS is started it will try to establish a session with the workstation SLU. Another possibility is to enter a CEMT (CICS Master Terminal Function) command to acquire a session with the workstation SLU. A third possibility is to start a CICS user transaction which requests a conversation. CICS will try to start a session with the workstation SLU to allow the conversation to be processed.

Note that the SNA command Start Data Traffic (SDT) is not used with LU 6.2 protocols by the PLU (CICS). Instead, CICS will send an SNA LUSTAT command.

5.3.4 Elements of an APPC/PC System

A APPC/PC system contains three main components:

- APPC/PC itself: This is the executable code shipped by IBM. It must be loaded before any APPC/PC calls can be made. APPC/PC performs LU Type 6.2

defined functions and provides an interface for control and service functions.

- The Application Subsystem: This performs control functions for APPC and service functions for Transaction Programs. It sets up the communication base attaching the PU, attaching the LU, activating the DLC and preparing the sessions (CNOS).
- Transaction Programs: These are user application processing programs, typically cooperating with transaction programs at other APPC/PC nodes. A programmer uses APPC/PC verbs to communicate with APPC/PC. Each verb represents a service provided by APPC/PC, and many of them represent LU 6.2 functions defined in the architecture. Verbs are accompanied by parameters whose values are either supplied by the program or returned by APPC/PC.

5.3.5 Planning

5.3.5.1 Tasks and activities for APPC/PC Implementation

Several functions are involved, and tasks required, to establish APPC/PC communication.

- System Programming

  Configuring the Communication Profile for APPC/PC. This is a menu-driven procedure to set up the communication profile for a specific PC. There are three menus to specify system parameters and attachment information. This task is invoked by entering APPCONF (at the DOS prompt) and can also be performed by a skilled workstation user.

  Writing an Application Subsystem: The Application Subsystem has certain specific responsibilities (e.g. activating DLC and handling of incoming ALLOCATE requests). To write such an Application Subsystem, PC Assembler programming skill and communication architecture skills are required.

- Application Programming

  This is the task of writing user programs to perform distributed transaction processing. A transaction program uses certain APPC verbs to invoke APPC/PC functions. Currently these verbs must be specified through parameters passed to APPC/PC from Assembler language programs, but applications may be written in a higher level language, and assembler drivers can be used to interface with APPC/PC.

- Workstation User

  Initiation of a transaction program will be either local or remote. Prior to initiation of a transaction program, APPC/PC and the Application Subsystem must be loaded.

5.3.5.2 General planning considerations

Coexistence and Dependencies on other Products

- DOS 3.2 required for Token-Ring Network
- APPC/PC Release 1 supports Token-Ring Attachment
- CICS 1.7 is required for communication with APPC/PC Release 1
- APPC/PC can coexist with the PC LAN program 1.1, but only if the PC LAN program is configured as Re-director. A PC cannot support concurrent PC LAN Server processing with APPC/PC.
- APPC/PC cannot be used from a shared disk, because the communication profile is specific fo a PC.
- APPC/PC checks the Token-Ring Network adapter status. If the adapter has already been opened by another application, it will not reopen it, but will share the adapter if permitted to do so by the other application or subsystem.
- If any other application has opened the adapter (e.g. PC 3270 Emulation) then the locally administered address must match the address defined in the APPC/PC application (SEND_AS and RCV_AS).
- When an APPC/PC and a NETBIOS application run concurrently on a Token-Ring Network PC, they access the PC Token-Ring Network Adapter through different Service Access Points (SAP's). A parameter of the NETBIOS initialization command (NETBEUI) is used to reserve additional SAP's for such requirements as running APPC/PC. These SAP's are used to differentiate between NETBIOS requests and APPC/PC requests.

- APPC/PC deallocation does not close the Token-Ring adapter. APPC/PC closes the link station and resets the SAP, which terminates access to the ring for APPC/PC. Other applications can continue to use the adapter.

Storage requirement for APPC/PC

The minimum storage requirements for APPC/PC Release 1 are:

- Support for Token-Ring communications	-	202K
- Support for SDLC communications	-	192K
- Support for both type of communications	-	215K

Considerations for access to IMS data bases by using APPC

The following information is only applicable if IMS and CICS are both installed at the System/370 host.

IMS/VS does not provide APPC support, but it is possible to access IMS databases through CICS applications which support LU 6.2 and DL/1.

The IMS database can be used concurrently by the Communication facility of IMS/VS, since Block Level sharing of IMS databases is supported between IMS DC and CICS.

In addition, Inter-Systems Communication (ISC) can be used between IMS and CICS for distributed transaction processing.

5.3.6 Preparation

5.3.6.1 Tailor APPC/PC System and Communications profile

One of the following panels can be selected after typing APPCON at the DOS prompt:

1 Define/Update System Parameters

2 Define/Update IBM Token-Ring DLC Parameters

3 Define/Update SDLC DLC Parameters

On the next page is an example of the parameters we used for our test. Only selection 1 and 2 are considered because the SDLC DLC panel (selection 3) is not applicable for a Token-Ring environment.

System and DLC Parameters Panels for Token-Ring Environment

PANEL 1 - DEFINE/UPDATE SYSTEM PARAMETERS

ITEM	CHOICE	POSSIBLE CHOICES
Machine Type................	5170	0 - 9999
Machine Serial Number.......	0152578	7 Alphanumeric Uppercase char
Node ID	A0001	1 - 5 hex digits
Transl. Table File Name.....		DOS Filename
DOS Control Break...........	1	1 = yes 2 = no
Workspace Size..............	048	18-400 Kbytes

PANEL 2 - DEFINE/UPDATE IBM TOKEN RING DLC PARAMETERS

ITEM	CHOICE	POSSIBLE CHOICES
DLC Name	ITRN	
Load Option	1	1 = yes 2 = no
Percent Incoming Calls......	050	0 - 100
Congestion Tolerance	080	0 - 100
Receive Window Count	1	1 - 8
Send Window Count	2	1 - 8
Max. Number of Link Stations	02	1 - 32
Local Node Address..........	400031301 0001	12 hex digits (all 0's or starting with 4,5,6, or 7
Maximum RU Size.............	0256	256 - 1920
Adapter Number..............	0	0 = Primary 1 = Secondary
Free Unused Link............	1	1 = Link take-down 2 = No Link take-down

Comments about parameters affecting host communication

- NODE ID: This value is used during XID-3 exchange at initial link level contact and must match the value specified in the VTAM Switched Major Node (IDNUM= in PU Macro). The corresponding IDBLK (x'050' for APPC/PC) is hard-coded in APPC/PC and cannot be altered.

- DLC NAME: This parameter is used during activation of DLC to identify type of adapter in use. The value can be either 'ITRN' for Token-Ring Adapter or 'SDLC' for SDLC Adapter.

- MAXIMUM NUMBER OF LINK STATIONS: This value defines the extent to which multithreading may be supported by the Application Subsystem. To support locally and remotely initiated Transaction Programs, this value must be at least 2.

- LOCAL NODE ADDRESS: Specifies the Token-Ring adapter address. If a value of all 0's is defined, the universal (burned-in) adapter address will be used. The adapter address value is used in ATTACH_LU verbs as a parameter when defining PARTNER_LUs. If universal addresses are used, the Application Subsystem is tied to the physical adapter; use of locally administered adapter addresses may provide greater flexibility.

- Define/Update SDLC DLC Parameters - The SDLC Panel does not apply in Token-Ring environment.

5.3.6.2 Defining a Application Subsystem

An Application Subsystem must be developed prior to running APPC/PC applications. This subsystem defines the capabilities of the APPC/PC workstation, and can be developed as a 'generic' subsystem with predefined utility routines to execute attachment sequences for particular system or application configurations.

To communicate with a subarea host, the Token-Ring Destination Address is required to identify the 3725 TIC. This address has to be defined in the Application Subsystem.

Information about developing a Application Subsystem can be found in the APPC/PC Programming Guide (Part Number 61X3842) and an example can be found in Appendix E of this document.

5.3.7 Sample Operating Scenario

The following scenario was used during our tests. In these tests, a CICS host transaction was invoked by a transaction originating at the PC. The scenario shows all the required steps. In a production environment, this command sequence could be incorporated in a BAT file to simplify transaction processing for the end user.

- Allocate appropriate PC DOS subdirectory
- Load TOKREUI
- Load APPC/PC by entering APPC at the DOS prompt
- Invoke the Application Subsystem
- For a locally initiated transaction, enter the name of the Transaction Program.
- After the Allocation message shows a zero return code the transaction program starts execution and may prompt the user for transaction related data.
- Enter APPCUNLD at DOS prompt to unload APPC/PC when all APPC transactions are completed.

Publication Reference

More information about APPC/PC can be found in

An Introduction to Programming for APPC/PC GG24-3034

5.4 ENHANCED CONNECTIVITY FACILITIES (ECF)

These facilities were announced in July 1986. They were not implemented in our test environment and they are only highlighted in this document to complete the description of host communication facilities for Token-Ring Network stations provided via the 3725 NTRI.

5.4.1 Overview

The Enhanced Connectivity Facilities include a set of IBM-provided Requester and Server programs for IBM Personal Computers, MVS/XA, and VM/SP host computers. The IBM Requester/Server programs provide access to System/370 databases and files and extend the personal computing environment to include System/370 resources.

Also included is the Server-Requester Programming Interface (SRPI), a new programming interface for development of interconnected host and Personal Computer applications. The SRPI provides application programmers with a consistent means of issuing requests for services and receiving replies across a System/370 host to IBM Personal Computer connection. The SRPI is a program-to-program interface using SNA (LU Type 2 protocols) or non-SNA connections.

5.4.2 ECF Product Implementation

The following licensed programs support IBM System/370 to IBM Personal Computer Enhanced Connectivity Facilities:

IBM SYSTEM/370

- TSO Extensions (TSO/E) Release 3 with the MVS/XA feature (5665-285). The MVSSERV command in TSO/E includes the router and the SRPI.

- TSO/E Servers (5665-396) includes host data access and virtual services.

- VM/SP Release 4 (5664-167). The CMSSERV command being added to VM/SP Release 4 (via PUT) includes the router and the SRPI.

- VM/CMS Servers (5664-327) includes host data access and virtual services.

IBM PERSONAL COMPUTER

- IBM PC 3270 Emulation Program Version 3 (59X9969) includes the router and the SRPI.

- IBM Personal Computer Requesters (6316993).

5.4.3 Requesters and Servers

Two of the key elements of ECF are Requester and Server programs, operating in pairs. Requesters in the IBM Personal Computer are designed to provide the functions needed by a Personal Computer user to request services or data from an IBM System/370 computer. Servers in the IBM System/370 computer are designed to reply to requests for services or data from the IBM Personal Computer.

The IBM-provided Requesters/Servers are, as follows:

- IBM Personal Computer Requesters

- TSO/E Servers for MVS/XA host computers

- CMS Servers for VM/SP host computers

The IBM Requester/Server licensed programs provide the following set of functions:

- Host Data Access

  Dynamic or predefined query capability is provided for the Personal Computer user to directly access DB2 and SQL/DS. File definitions and query information may be stored for reuse.

  DXT allows the user to extract information from DB2, SQL/DS, DL/I and Fast Path databases, and VSAM and SAM files for use in Personal Computer applications. Extracted data is stored in DXT Integration Exchange Format. The user may also list and select jobs, and predefine those jobs to be run in batch mode.

- Virtual Disk

  Through the virtual disk, host disk space is used as personal computer virtual disk space. Data is stored in personal computer format. Virtual disk allows users to share data using existing host file facilities, and provides direct import/export from host files.

- Virtual File

  Virtual file allows the user to use host files as local Personal Computer files. Translation from host data types to Personal Computer data types and field-level transformations are provided. VM CMS files and MVS sequential and partitioned datasets are supported.

  Personal computer access to host files is controlled by host security facilities (such as RACF). Host files can be shared among host and Personal Computer applications.

- Virtual Print

  Using virtual print, Personal Computer printer output may be directed to the host printer. Virtual print transforms Personal Computer print datastreams to IBM 3800 Model 1 or IBM 1403 host printer datastreams.

- Execution of Host Procedures, Commands, or Programs

  The user may initiate real-time execution of host procedures, commands, or programs, including VM EXECs, TSO/E CLISTs, and TSO/E or CMS host commands.

- Full-Screen User Interface

  An easy-to-use, full-screen interface provides Personal Computer user access to host data and resources, as well as an extensive online help facility.

- Personal Computer Command Interface

  A Personal Computer Command Interface is provided for programmers and experienced users. Commands may be entered on the DOS command line or in a BAT file, or they may be invoked from an application program through a subroutine call.

- File Transfer

  - File transfer allows files to be transferred between the personal computer and the host, or copied between a virtual disk and a host file. Field-level transformations and an exit for user- supplied data mappings is provided.

  - The following Personal Computer interchange formats are supported

    - Data Interchange Format (DIF) (1)
    - LOTUS Corporation SYMPHONY(1) WRK Format
    - LOTUS 1-2-3 (1) WKS Format
    - dBASEII (2) and dBASEIII (2) SDF Delimited Format
    - Comma separated variable (CSV)
    - Multiplan (3) Input Format (SYLK)

(1) DIF, WRK, WKS, LOTUS 1-2-3, and SYMPHONY are registered trademarks of Lotus Development Corp.

(2) dBASEII and dBASEIII are registered trademarks of Ashton-Tate Corp.

(3) Multiplan is a registered trademark of Microsoft Corp.

5.4.4 Communication Support

A wide range of connectivity options are available through IBM PC 3270 Emulation Program Version 3:

- Direct attachment to the IBM 372X Communication Controller via the IBM Token-Ring Network
- Distributed Function Terminal (DFT) Mode attachment to an IBM 3174 or 3274 Control Unit.
- Remote communications attachment to an IBM 372X via SDLC links.

5.4.5 Software Requirements

For MVS/XA TSO/E:

- MVS/SP Version 2 Release 1.2 (MVS/XA), JES2 (5740-XYS) or JES3 (5665-291) and either MVS/XA Data Facility Product Version 1 Release 1.2 (5665-284) or MVS/XA Data Facility Product Version 2 Release 1 (5665-XA2)
- TSO/E Release 3 with MVS/XA feature (5665-285) which includes the SRPI Support
- ACF/VTAM Version 2 or a higher level
- Interactive System Productivity Facility (ISPF) Version 2 Release 2 (5665-319) if using IBM TSO/E Servers. When using functions that require IBM DB 2 and/or IBM Data Extraction Program (DXT), IBM DB 2 Release 1 (5740-XYR) and DXT Version 2 (5668-788) are required.

For VM:

- VM/System Product Release 4 (5664-167) which includes SRPI support (via PUT), with or without High Performance Option (HPO) (5664-173)
- ACF/VTAM Version 3 (5664-280) (for SNA/SDLC connection)
- Interactive System Productivity Facility Version 2 Release 2 (5665-282) if using IBM CMS Servers.
- When using functions that require SQL/DS and/or DXT, SQL/DS Release 3.5 (5748-XXJ) and DXT Version 2 (5668-788) are required.

For IBM Personal Computers:

- IBM Personal Computer DOS 3.1 or 3.2. DOS 3.2 is required for the IBM Token-Ring Network.
- IBM PC 3270 Emulation Program Version 3 (59X9969) on the IBM Personal Computer, Personal Computer XT, or Personal Computer AT
- EZ-VU II Run Time Facility (6317025) if using the IBM Personal Computer Requester Full Screen Interface.

IBM Requester/Server Licensed Programs:

- IBM TSO/E Servers (5665-396) or
- IBM CMS Servers (5664-327), and
- IBM Personal Computer Requesters (6316993)

Publication Reference

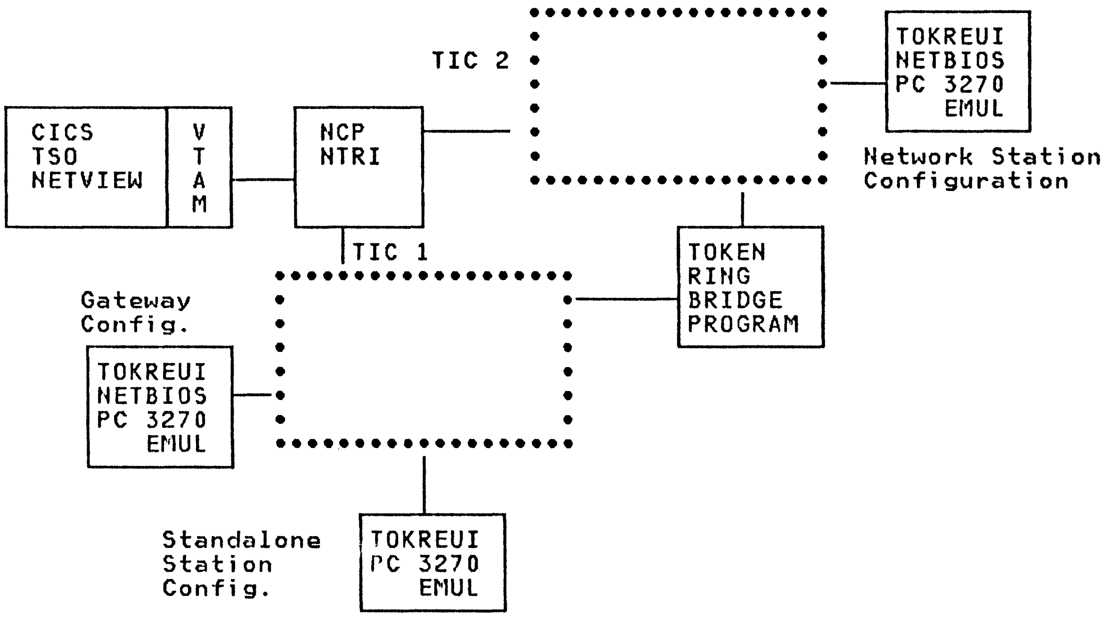
Introduction to IBM System/370 to IBM Personal Computer Enhanced Connectivity Facilities (GC23-0957), provides more information about Enhanced Connectivity Facilities and the IBM Requester/Server licensed programs.

5.5 TEST ENVIRONMENT

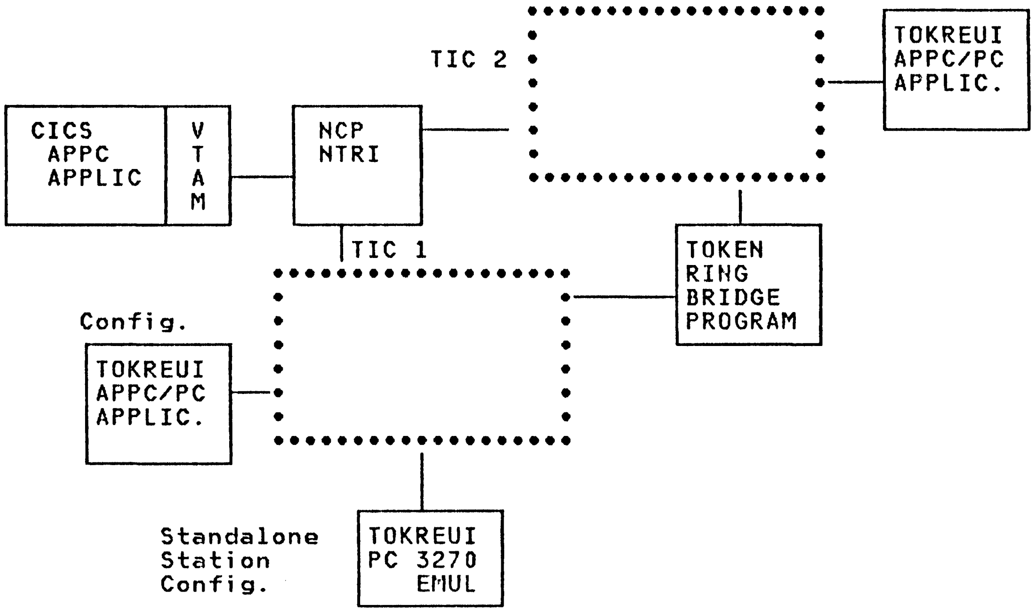
The workstations used in our host communications tests were IBM PC/XT's and PC/AT's with standard products for communicating with each other over the Token-Ring Network or with a /370 host via an SDLC link. The following products and levels were tested:

- PC/DOS V3.2
- TOKREUI - Token-Ring adapter code supplied with the PC Adapter Card.
- NETBIOS V1
- PC 3270 Emulation V3 - in both standalone and gateway configurations
- APPC/PC - with an application written to communicate with CICS
- Token-Ring Network Manager
- Token-Ring Bridge Program

5.5.1 PC 3270 Emulation Configurations



5.5.2 APPC/PC Configuration



6.0 SAMPLE CONFIGURATIONS AND BACKUP CONSIDERATIONS

3725 NTRI and Token-Ring Network capabilities provide backup flexibility to handle nearly all situations. The Token-Ring Network with bridges supports many configuration possibilities and the routing facility of Token-Ring Network makes it unnecessary to predefine backup paths. Each TIC is represented on the Token-Ring Network like any other station.

Existing VTAM and NCP backup facilities are valid for the logical (switched) connections of Token-Ring Network devices.

6.1 BACKUP CONSIDERATIONS

From the host's point of view the rules for backup of switched resources are valid for the Token-Ring Network environment. A new facility in NCP V4R2 provides enhanced switched connection support.

6.1.1 Enhanced Switched Connection Support of ACF/NCP V4R2

In previous versions of ACF/NCP, switched lines could only be defined as ANS=STOP, with the result that switched lines were always disconnected during ANS processing. With NPSI and NTRI, switched connection support and receovery become more important. ACF/NCP V4R2 allows a user to define a switched connection with ANS=CONTINUE. With this type of connection, the Switched Line and PU remain active when the owning SSCP (host) is lost. The LU session between the terminal and the non-owning host remains active until the session terminates normally or the network operator intervenes. Takeover is not provided for and any attempt to activate a switched link that is already active is rejected by NCP with a sense code of X'0801'.

Since no VTAM support is required for this enhancement, the user must still specify ANS=STOP in the Switched Major Node PU definition. ACF/NCP V4R2 checks the PU to see if ANS=CONTINUE was specified. If during ANS-Processing the ANS=CONTINUE indicator is set for a Switched Line PU, NCP does not issue the disconnect for the PU and all LU-LU sessions remain active. The switched line, PU's and LU's do not have any owner, and the switched connection remains active even if all LU-LU sessions have ended. The only way to disconnect the link is with 'forced deactivation'.

If the owning VTAM is restarted, it assumes that the switched connections are broken but that they are still active. VTAM and NCP do not support activation of a switched line if the line is already active. If VTAM issues ACTLINK for an already active link, the NCP will send a response with sense code X'0801' and indicate that the activation failed.

If another SSCP (host) wants to get (take over) ownership of the link, a DACTLINK has to be issued before reactivating the link for the new SSCP.

6.2 BACKUP CONSIDERATIONS FOR THE TOKEN-RING HOST ATTACH

From the network station point of view, the main question is how to survive in case of a fault in one of the components and how to restart using a different path to bypass the failing component.

The situations where sessions with application hosts remain active if the owning host fails are covered in the previous section. If a network component fails, the sessions using this component will be disconnected.

Token-Ring Network facilities exist to recover from many such situations. See "Network Management and Recovery" on page 77 for more details. The remaining network component which can fail is the 3725. If this is the case an alternate path and resource takeover is required. Automatic takeover is not available for switched major nodes but a manual takeover is possible when the resources are predefined in a second subarea node.

Alternate path to the Host by using a backup TIC

Backup network components are required for an alternate path. This can be a second 372X Token-Ring Network attachment, directly attached or accessible over bridges to a ring where an alternate path is configured.

Workstation-initiated request:

The communication profile for a host communication application of a network PC usually specifies the locally administered address of the TIC to be used by a particular application. To change this destination address in subsystems (such as the PC 3270 Emulation program) involves a very short task which can be pre-defined by creation of two 3270 Emulation start modules, one with the primary TIC address and one with the alternate TIC address. If the emulator is always loaded by BAT file commands, a second BAT file can be used to start communication using an alternate TIC. This makes normal startup easier for the end-user, and makes backup relatively transparent.

SCENARIO to create BAT files for 3270 Emulation

```
    Execute Communication Profile Task with TIC A as Destination Address
    Rename resultant module PSCAnnnn.PG1 to TICA
    (Where 'nnnn' is the product version and level number e.g.  PSCA0300.PG1)
    Execute Communication Profile Task with TIC B as Destination Address
    Rename resultant module, PSCAnnnn.PG1, to TICB
    Create BAT File (name=3270.BAT) which copies TICA as PSCAnnnn.PG1
    Create BAT File (name=3270B.BAT) which copies TICB as PSCAnnnn.PG1
```

If gateway and network stations are used, this procedure is only required for the gateway station.

A disadvantage of this implementation is that the end user must know when to use the backup command.

Subarea host-initiated request:

A procedure not involving the end user is to have a alternate TIC pre-generated with the same locally-administered address as the primary TIC. The physical line for the alternate TIC is normally not activated. If the 372X with the primary TIC fails, now the physical Line and PU with the original destination address can be activated and the end user can reestablish host communication using the normal production command or BAT file.

6.3 SAMPLE CONFIGURATIONS

6.3.1 Single Host / Multiple Ring

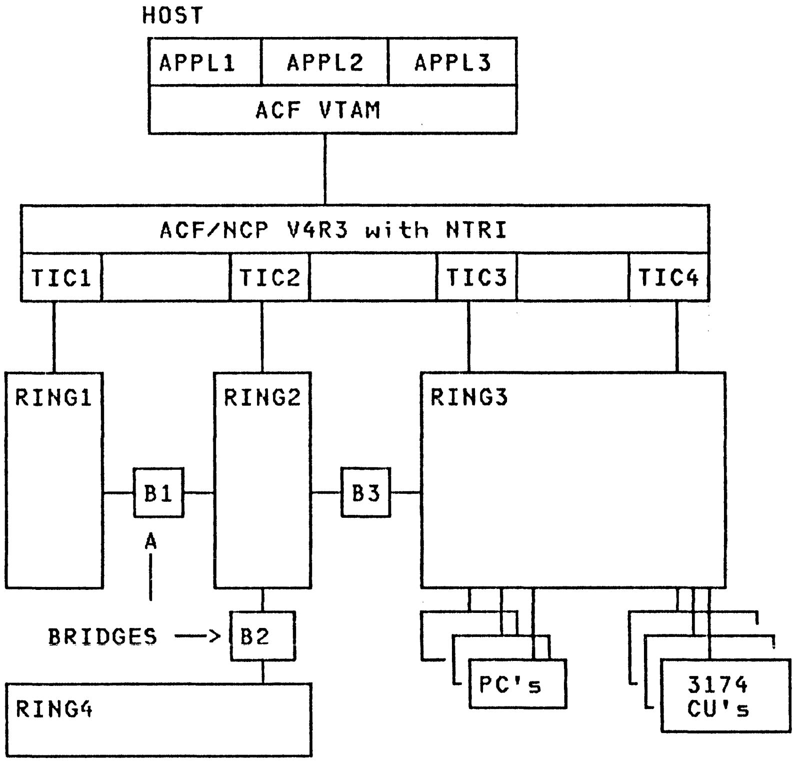


Figure 14. Configuration with Single Host and Multiple Rings

Comments

- Up to 4 rings can be directly connected to a 3725
- Up to 8 rings can be directly connected to a 3725 with 3726
- Other rings can be connected through bridges
- Several TIC's can be connected to the same ring
- Each TIC could be backup for any other TIC's (change of destination address in communication profile required).
- A TIC (e.g. TIC4) could be predefined as backup TIC (not used as destination). In a situation where backup is requested, a NCP must be loaded with the locally administered address of the failing TIC used in TIC4.

6.3.2 Multiple Host / Single Ring

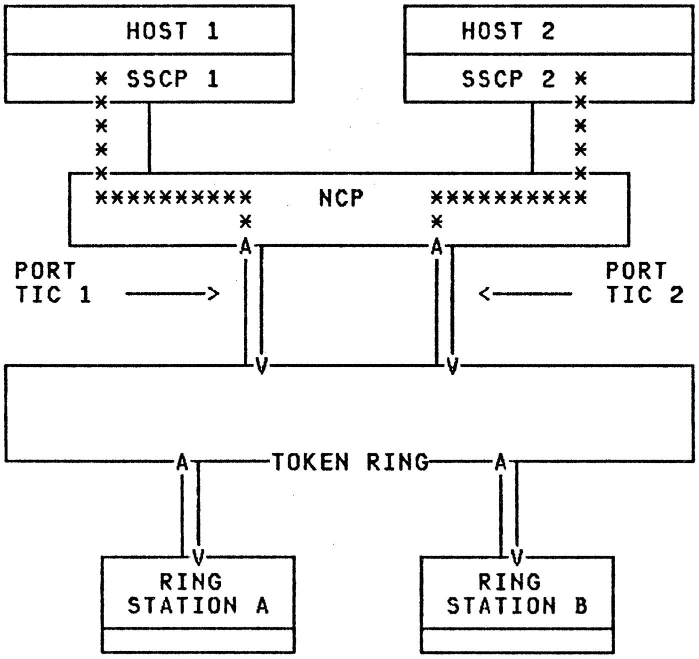


Figure 15. Sample Configuration - Routing of Incoming Calls

ROUTING OF INCOMING CALLS

This figure indicates that NTRI permits routing of incoming calls to a predefined SSCP. This can be achieved by using two different ports from the same token ring.

With dial lines, any incoming call is routed to the owning SSCP of the dial line. On reception of an incoming call,NTRI has to choose an available dial line. A network station calling a host uses the TIC locally administered address as destination for the dial in request,but TIC's can be owned by different SSCP's (e.g. by OWNER Parameter in LINE).

If no owner is specified, the SSCP activating the line first will assume ownership. Therefore, a station may direct its incoming call towards either Host 1 or Host 2 by sending its incoming call to Port 1 or Port 2.

6.3.3 Multiple Host / Backbone Ring

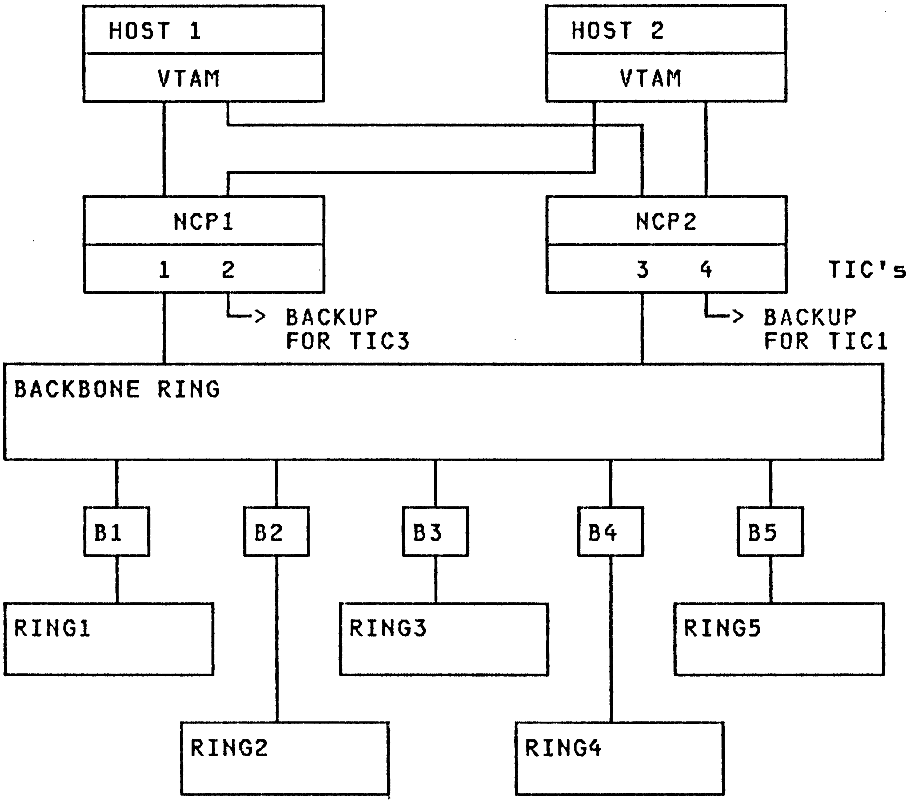


Figure 16. Configuration with Multiple Hosts and Backbone Ring

Backup Considerations for Failing Components:

Host 1 fails:

- Application sessions with Host 1 become inactive
- All sessions with Host 2 are still active (independent of ownership)
- Manual (predefined) takeover is possible to enable Logon to applications on Host 2.

3725 with NCP1 fails:

- Application sessions using TIC1 become inactive
- All sessions using TIC3 are still active
- If the locally-administered address defined in NCP for TIC4 is identical with the locally-administered address of TIC1, new sessions can be started after activating the physical Line and PU for TIC4.
- TIC4 can be loaded with the locally administered address of TIC1. After opening TIC4 (ACT PU), normal processing continues. Loading of the Backup 3725 (NCP2) has to be considered.

6.3.4 Multiple Host / Multiple Backbone Ring

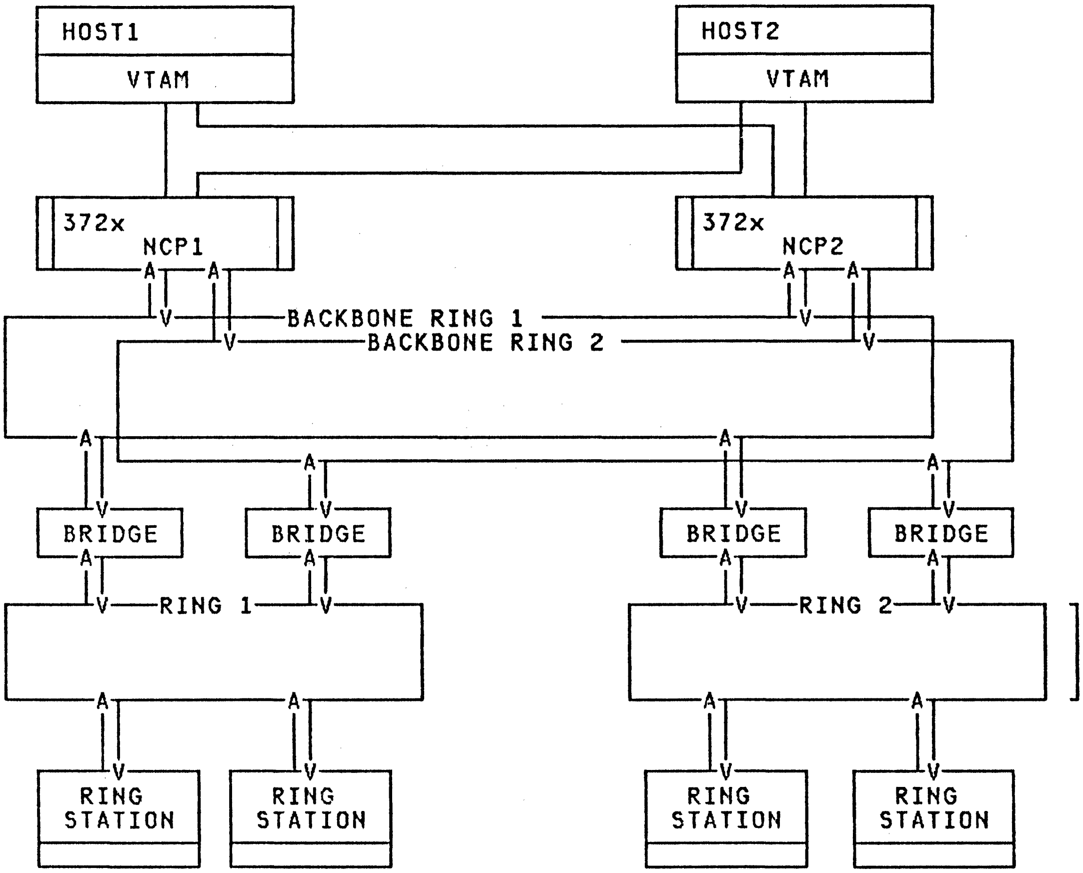


Figure 17. Sample Multiple Backbone Configuration

COMMENTS:

- This configuration offers high availability
- This configuration offers additional capacity
- The utilization of the two backbone rings is automatically balanced through the routing concept of IBM Token-Ring Network
- Host communication sessions normally pass only 1 bridge

7.0 NETWORK MANAGEMENT AND RECOVERY

7.1 NETWORK MANAGEMENT

Token-Ring Network stations require Switched Major Node definitions in VTAM to support requests for host communication. Using this approach, the fact that the stations reside on a Token-Ring Network is transparent to VTAM and to NetView. However it is not transparent for NPM, which cannot support the measurement of the line utilization of a Token-Ring.

The Token-Ring Network itself may be a host-independent environment, and therefore requires host-independent Network Management components and functions in addition to those typically available at a host. To minimize redundancy, the host components are complementary to these capabilities.

If a Token-Ring Network is attached to a host, NTRI provides Network Management support through Network Management Vector Transport (NMVT) messages. NTRI does not manage the Token-Ring Network but notifies the host about inoperative conditions on the ring. The Token-Ring Network Management products are still required and responsible for network management of the Token-Ring.

7.1.1 Token-Ring Network Management Products - Highlights

- Adapter Diagnostic Program:
  - Is part of the PC Token-Ring Adapter package
  - Included in the Adapter Support Program Diskette
  - Performs adapter test functions
- Ring Diagnostic Program
  - Is part of the PC Token-Ring Adapter package
  - Performs ring diagnostic functions
  - Monitors ring status and indicates changes
  - Can be used as a continually running program
- Token-Ring Network Manager Program:
  - Is a licensed program
  - Provides improved functions to manage the ring
  - Provides ring configuration information
  - Supports allocation of symbolic names for ring stations
  - Performs Alerts and event logging
- Token-Ring Network Bridge Program:
  - Supports Bridge Management functions
  - Provides Bridge Configuration information
  - Reports beaconing conditions
  - Provides a path trace function

Selecting Token-Ring Management Support

Both the Ring Diagnostic Program and the Token-Ring Network Manager Program support network management. The Token-Ring Network Manager provides more functions and is recommended for large rings or when high availability and reliability is important.

It is difficult to give precise guidelines for the number of workstations for which the Token-Ring Network Manager program is recommended. When the number of workstations gets above 30-40 for a single ring or (there is sufficient traffic on the ring,) it would be useful to use the Token-Ring Network Manager. Another criterion might be the use of devices such as 3174 Control Units, because the availability requirement in such an environment is traditionally very high.

Publication References

- Token-Ring Network - Problem Determination Guide SY27-0280
- Token-Ring Network - Bridges and Management GG24-3062

7.1.2 Host Products for Token-Ring Network Management

- NetView support - NPDA Component
  - NTRI notifies the NPDA operator with an Alert message when the Token-Ring becomes inoperative
  - NTRI generates the following types of Network Management Vector Transport (NMVT) messages for NetView:

= Alert messages	Major Vector = X'0000'
= Link Event messages	Major Vector = X'0001'
= PD Statistics messages	Major Vector = X'0025'

  - Alert and Link Event messages are used for error reporting
  - PD Statistics messages are for Statistical data reporting
  - New NPDA screens are available to support the new Subvectors
  - NMVT messages used by NetView for operator attention and logging
  - NPDA cannot issue diagnostic request for the ring
- NetView support - NLDM Component
  - No changes were required in NetView to support NLDM for Token-Ring network stations.
  - The NLDM functions are session-related and therefore transparent to Token-Ring stations.
  - The Response Time Monitoring (RTM) facility is not supported by the PC 3270 Emulation Program.
  - The Configuration support functions of NLDM are supported for SNA resources of the Token-Ring.
- Network Performance Monitor (NPM)
  - Ring utilization cannot be measured by NPM because the 'NPACOLL' and the 'NPARSC' keywords in the NCP definitions are invalid if ECLTYPE has been specified.
  - NPM functions which are transparent to VTAM and not based on NPACOLL data information should work for Token-Ring Network stations (e. g., the Response Time Measurement Facility of NPM).

7.2 TOKEN-RING NETWORK AND NTRI RECOVERY PROCEDURE

From the host's point of view, ring recovery is part of the ring function and no host components are involved other than the TIC, which is part of the ring and is thus subject to the same ring recovery procedures as other station adapters. In case of errors, the TIC will inform NTRI, and appropriate recovery actions will be taken and alerts raised as necessary.

7.2.1 Ring Station Insertion and Removal

The Token-Ring Network operates with a single token protocol and unidirectional transmission. The token is continually received and retransmitted by every active station on the ring. Insertion or removal of network stations does not halt or disrupt transmission and causes automatic update of the ring topology through the Nearest Active Upstream Neighbor (NAUN) Notification process.

The Multi Station Access Unit (MSAU) is designed so that if a ring station is disconnected or a lobe wire is broken, the lobe is automatically removed from the ring at the MSAU.

If a ring adapter determines that it is failing and interrupting the ring, it will remove itself from the ring at the MSAU.

Ring Station Insertion: The act of connecting a Ring Station lobe into the MSAU does not make it a member of the ring. An electromagnetic relay at each of the eight ports of the MSAU keeps the circuit between the ring adapter and the ring media open.

In order to connect to the ring the ring adapter must send a low voltage DC signal to the MSAU port. This is an induced DC voltage and therefore does not affect the digital signal on the lobe cable. This signal closes the relay in the MSAU and makes the ring station (lobe) part of the ring. This allows the ring station to identify itself and to participate in ring operations.

Ring Station Removal: If the ring adapter determines that it is the source of a ring lobe problem then the DC signal is dropped and the MSAU relay opens. This breaks the ring circuit for that lobe and the ring station is thus bypassed. If the lobe cable is unplugged from the MSAU, the relay will again open. This is also true if the lobe cable is unplugged at the ring station.

7.2.2 Type of Ring Errors and Beaconing Process

Two Types of Error may occur on the Token-Ring Network - Soft Errors or Hard Errors.

7.2.2.1 Soft Errors

Soft errors are defined as intermittent errors which temporarily disrupt normal operation of the ring. Normally soft errors are caused when data must be transmitted on the ring more than once to be received correctly.

Soft errors are handled by error recovery procedures and are not usually noticed by the user. Each ring station maintains counts of the frequency of occurrences of the most critical soft errors and periodically sends these counts to the Ring Error Monitor (REM).

Typical soft errors include:

- Line Errors: First detection of a Frame Check Sequence error or a Code Violation
- Internal Errors: A ring station detects a recoverable internal error
- Lost Frame Errors: Detected when a ring station's physical trailer timer expires
- Token Errors: Detected by the active monitor when its token timer expires

7.2.2.2 Hard Errors

Hard errors are permanent faults, usually in equipment, which cause the ring to stop operating within the normal Token-Ring protocols.

Equipment causing hard errors could be:

- Multi-Station Access Units (MSAU's) (e.g., Relay)
- Cables between MSAU's
- Transmitter side of station adapter
- Receiver side of station adapter

If a hard error occurs on a ring, the next downstream station will recognize the fault (missing incoming signal) and automatically start a recovery procedure. This recovery procedure is called a beaconing process.

7.2.2.3 Beaconing Process

When a hard failure is detected, the cause of this failure must be isolated and bypassed in order to restore proper operation of the Token-Ring Network. The ring station that has detected this hard error transmits a 'Beacon' MAC frame with an all-stations address to its ring and is called the 'beaconing' station. All other stations that receive the Beacon MAC frame enter 'Beacon repeat mode'. The Beacon message circulates one ring only; it does not cross bridges onto other rings.

The beaconing station assumes that the hard failure (broken situation) must be somewhere between its receiver and the next upstream station. This station is called the Nearest Active Upstream Neighbor (NAUN). Consequently, the Beacon MAC frame identifies the address of the Nearest Active Upstream Neighbor (NAUN). When the NAUN of the beaconing station has copied eight of these Beacon MAC frames, the NAUN removes itself from the ring and tests itself. If the test is successful, the NAUN reattaches to the ring, otherwise the NAUN remains unattached.

If the ring does not recover after the NAUN tests itself and is back on the ring, the beaconing station removes itself from the ring and tests itself. If that test is successful, the beaconing station reattaches to the ring, otherwise it remains unattached.

If the ring does not recover after both the NAUN and the beaconing station have self-tested, the error cannot be repaired using automatic recovery, and manual intervention is required. Such a non-recoverable hard error situation could occur if the main ring path between the NAUN and the beaconing station is broken. This situation is only recoverable if the ring is configured with an alternate backup path.

Up to 30 seconds of the beaconing process may be required before the error can be declared an unrecoverable hard error. During this time, the host sessions are interrupted, but they are still alive. If the hard error can be recovered by removing a station from the ring the users can continue working without a session restart.

Beaconing situations are immediately recorded and displayed by:

- Ring Diagnostic Program
- Token-Ring Network Manager Program
- Bridge Program
- NetView Program (host)
- NetView/PC Program
- 372X MOSS Console

7.2.3 NTRI Process related to Token-Ring Error Recovery

The ring adapter (TIC) notifies NTRI about beaconing via the Ring Status interrupt with the Hard Error bit on. When NTRI services such an interrupt, it starts a 30-second timer.

At the end of the 30-second timer, the ring status bits are checked to identify the type of hard error:

- Auto-Remove Error: This is a detected failure of the adapter or wiring. NTRI remains de-inserted and reports an ALERT message, 'Permanent Error'.
- Hard error: If there is a Hard Error status after 30 seconds, then NTRI sends an ALERT message, 'Permanent Error'.
- No Error: If there is no Hard Error and no Auto-Remove error status after 30 seconds, then the error has been recovered and an ALERT message, 'Temporary Error' is reported.

When NTRI generates an ALERT message, it specifies the Beacon fault domain.

If the 'Transmit Beacon' indicator is OFF in the Ring Status, it indicates that the fault domain is the Beaconing station, the NAUN of the last received Beacon MAC frame or the lobes and wiring between that two stations.

If the 'Transmit Beacon' indicator is ON in the Ring Status, it indicates that the TIC is beaconing. Therefore the fault domain consists of the TIC itself and its NAUN.

When the VTAM operator activates the physical line to the TIC, the TIC is initialized but not opened. The TIC will be opened and become a part of the ring after the operator activates the physical PU for the TIC. If the TIC is inserted into a ring which is beaconing or begins to beacon during insertion, the attempt to open the TIC will fail. In this situation, the VTAM operator should not try to reactivate the Line and PU before the 30-second interval (Beaconing Process) has ended.

7.3 CENTRALIZED NETWORK MANAGEMENT

Management of a Token-Ring Network from a central location may be advantageous. Larger Token-Ring Networks, particularly those with 3174 Control Units will generally require high availability of the host interfaces as well as local server functions. The skill to manage and improve the availability of such resources is more likely to be found at the host site.

It is important for the central site network management to designate a person to be responsible for managing the Token-Ring Network resources. This function may also be the focal point for end users to contact when problems arise (Help Desk).

Netview/PC provides a capability for the host Netview operator to get network management alerts and data from the Token-Ring Network Manager Program.

NetView/PC also provides Remote Console Facility (RCF). It enables NetView/PC and Token-Ring Network Manager operating in a Local PC to be controlled from a Remote PC over an asynchronous communication link. The Remote operator may control any NetView/PC or Token-Ring Network Manager Function of the Local PC. When the RCF session is active, all keystrokes from the Remote keyboard are passed to the Local PC (attached to Token-Ring). The local operator's keyboard is locked, giving the Remote operator control over the Local NetView/PC unless the local operator breaks the session.

Using Remote NetView/PC is currently the only way to control a Token-Ring Network Manager from another site such as a Central Network Control Center. The remote NetView/PC operator in the NCC could, for example, use the menus provided by IBM Token-Ring Network Manager program to perform problem determination or execute ring tests.

7.3.1 Centralized Network Control Center - Example

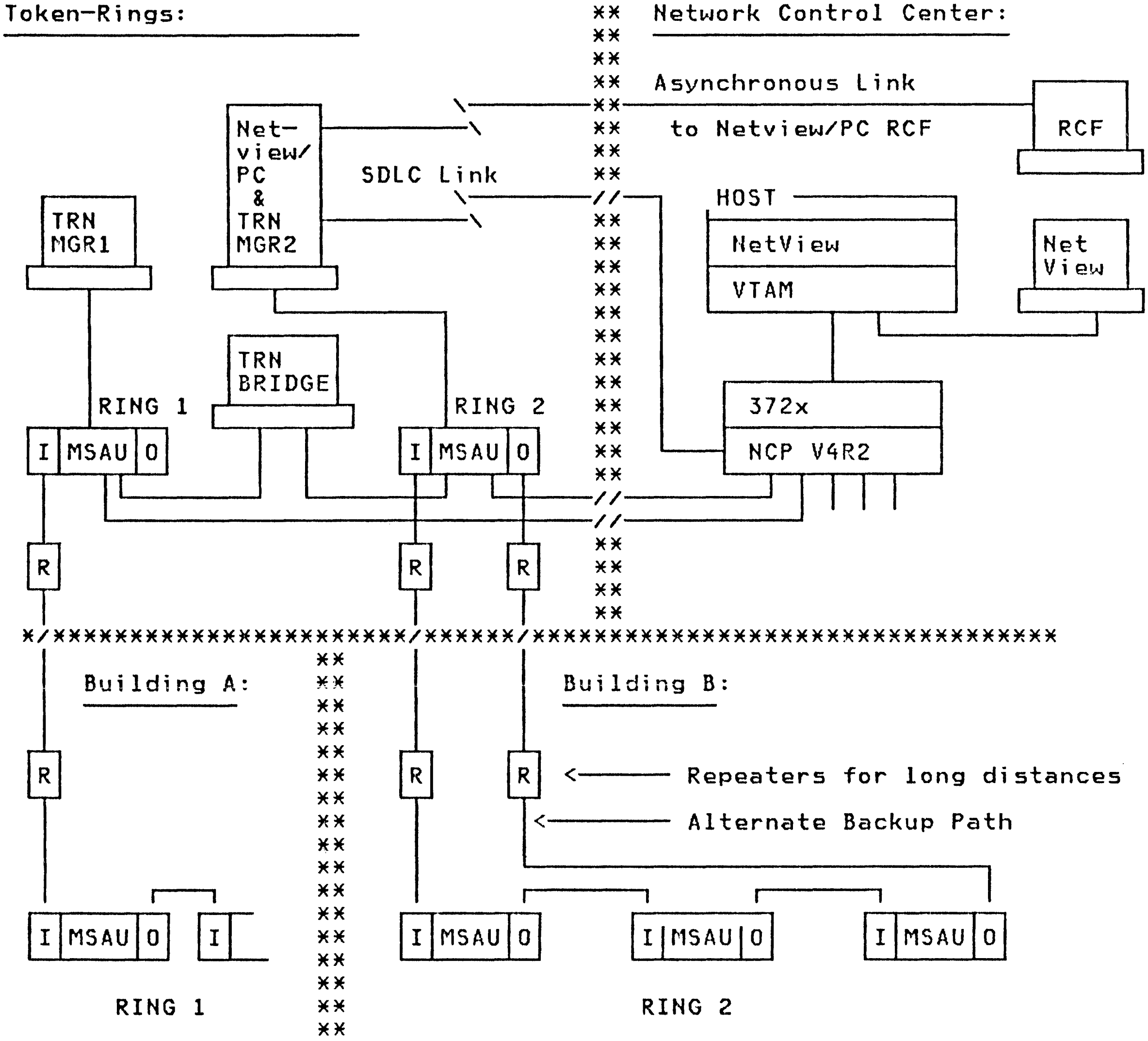


Figure 18. Centralized Network Management - Configuration Example

COMMENTS:

- All TRN Management functions can be performed from the NCC. with a combination of Netview and Netview/PC.
- Each Multi-Station Access Unit (MSAU) implements an internal ring without the need to close the path connecting the 'I' and 'O' of the same. In this figure, an optional second Path has has been defined for Ring 2, which automatically provides a backup path if one of the cables between NCC and building is defective.
- For Backbone Ring configurations, the considerations are a bit different.

8.0 PROBLEM DETERMINATION

8.1 PD TOOLS FOR TECHNICAL SUPPORT

8.1.1 NTRI Trace Facilities

NTRI provides three types of trace facilities:

- LINE Trace
- IOH Trace
- TIC Internal Trace

8.1.1.1 The TIC Internal Trace Facility

The TIC internal trace reports information to the host relative to the internal process of the TIC and is activated from the host as a Scanner Interface Trace (SIT).

The TIC internal trace facility is normally not used by system programmers or system engineers. To interpret it, HW and TIC internals skills are required, and information is not generally available in the field to support interpretation.

8.1.1.2 The IOH Trace Facility

The IOH Trace Facility supports a trace of all PIU's sent to a Token-Ring Multiplexer (TRM) or a TIC. This trace is activated, deactivated and reported to the host with the GTF line trace.

8.1.1.3 The Line Trace Facility

The Line trace is the trace most often used for debugging. Data related to the Line trace is transferred to the host, where it is retrieved and processed by ACF/TAP.

The activate/deactivate procedure for the NTRI Line trace is the same as for other Line traces. The only line for which a Line trace can be started is the physical link, but all logical link events are traced by the physical link.

Line Trace Key Points

- The Line trace entries are taken from MAC software functions in the NCP. The data at this level is not exactly the data sent/received to/from the ring.
- Since the Line trace is supported only for Physical link, DA, SA, DSAP and SSAP information must be used to determine the logical link.
- For performance reasons, NTRI allows only one Line trace to run at a time.
- Not all data received on the ring are traced for performance reasons.
- The order of frames shown in the trace is the order that was transmitted to and received from the TIC. The frames are not necessarily in the order that they were processed by NTRI.
- When a frame is received from the TIC, the Routing Information field has been padded to a full 18 bytes by the TIC.
- The Line trace and the IOH traces run together, and their elements appear mixed in the trace records.
- The Line trace process is designed to impact the path length of the normal process as little as possible.
- A RECEIVE frame containing only the LIT entry represents a TIC MAC frame

8.1.2 Line Trace Example

This is a Line Trace Example for a workstation initiated connection. The workstation application was PC 3270 emulation which ended with a 'COMMUNICATION CHECK 510' after attempting to invoke host communication.

The reason for the hung communication was an invalid IDBLK parameter value in the VTAM switched major node definition. (03A instead of the 017 required for 3270 Emulation) After changing the value to '017' the problem was solved.

1.Trace Record: Receive IEEE Standard XID '8103FE'

```
REC 40072DC8 0007D078 C1072DCA 70001006             ---> LIT Entry
    1040 400001130000 C00000313000                  |-->
    ---- ------------ ------------                  |
     F1       F2           F3                       |
                                                    |
    0200 0000 0000 0000 0000 0000 0000 0000 0000    |--> LLC Frame
    ---- ---------------------------------------    |
     F4                  F5                         |
                                                    |
    00 04 BF 8103FE                                 |-->
    -- -- -- ------
    F6 F7 F8   F9
```

2.Trace Record: Response XID

```
TRA 4007D078 00044000 C1072FC6 78CC0404             ---> LIT Entry
    0040 400000313000 C00001130000                  |-->
    0280                                            |--> LLC Frame
    04 01 BF 8103FE                                 |-->
```

3.Trace Record: Receive XID - 0

```
REC 40072DEC 0007D204 00072DEE 70001006             ---> LIT Entry
    1040 400001130000 C00000313000                  |-->
    0200 0000 0000 0000 0000 0000 0000 0000 0000    |--> LLC Frame
    04 04 BF                                        |-->
```

4.Trace Record: Response XID - 0

```
TRA 4007E6A4 00044000 00072FEA 78CC0404             ---> LIT Entry
    0040 400000313000 C00001130000                  |-->
    0280                                            |--> LLC Frame
    04 05 BF                                        |-->
```

5.Trace Record: Receive SNA XID (IDBLK,IDNUM)

```
REC C0072E10 0007D390 00072E12 70001006             ---> LIT Entry
    1040 400001130000 C00000313000                  |-->
    0200 0000 0000 0000 0000 0000 0000 0000 0000    |--> LLC Frame
    04 04 BF 0200017E0001                           |-->
```

The explanation of the different trace information and the fields F1 -> F9 follows.

Explanation for Line Trace Example

See also 'Frame Format' in the introductory chapter of this document for information about the frame layout.

- LIT Entry: Only the last 2 bytes are important for debugging. Explanations can be found in the updated version (Rel.4) of the 3725 Communication Controller Guide (GG24-1653).

- Field 1 - AC and FC field of Token-Ring Frame:

  The first part (SD) of the MAC frame is not shown in the trace because it is only available on the physical layer.

  AC '10' - Token bit on indicates that it is a frame. In a Transmit frame (from the 372X to the ring) the token bit cannot be on because it is inserted by the physical layer.

  The Frame Control byte (FC) is the same in all trace entries (40), which indicates that it is a LLC frame.

- Field 2 - Destination Address:

  The destination address identifies the ring station that is to copy the frame, and is always 6 bytes in length. For a transmit frame, the address identifies the station (PC) which has to copy the frame; for a receive, it contains the TIC address. X'40' at the beginning indicates that it is a locally-administered address.

- Field 3 - Source Address:

  The source address identifies the ring station that originated the frame. The first byte contains 'C0', which indicates that it is a locally-administered address and that Routing Information is available.

- Field 4 - Routing Information

  This field is optional and variable in length. The first two bytes contain Routing Control Information and the rest are up to 8 2-byte segment numbers. For NTRI the routing information is always padded to the full length, but the interpretation is done as shown in the trace example.

- Field 5 - Routing Segments

  For an explanation, see Field 4 - Routing Information

- Field 6 - DSAP

  The Destination Service Access Point (DSAP) address field identifies the Service Access Point (SAP) for which the LLC Protocol Data Unit (LPDU) is intended. The SAP's are defined by IEEE 802.2; X'04' is the SNA Path Control default. X'20' defines a LLC Sublayer management function.

- Field 7 - SSAP

  The Source Service Access Point (SSAP) address field identifies the Service Access Point (SAP) that originated the LPDU. The definitions are the same as for DSAP except that bit 7 is used to identify it as either a command or response function.

- Field 8 - Control Field (8 or 16 bits)

  The Control field of the LPDU can have three different formats. In the trace example it is an unnumbered PDU which is only 1 byte and used to provide additional control functions.

  Unnumbered Format Control Field Bits for Link Stations:

0	1	2	3	4	5	6	7	Commands and Responses
0	1	1	P	1	1	1	1	SABME Command
0	1	0	P	0	0	1	1	DISC Command
0	0	0	P	0	1	1	1	SIM Command
1	0	1	P/F	1	1	1	1	XID Command/Response
1	1	1	P/F	0	0	1	1	TEST Command/Response
0	1	1	F	0	0	1	1	UA Response
0	0	0	F	1	1	1	1	DM Response
1	0	0	F	0	1	1	1	FRMR Response

  Bits 6 and 7 identify unnumbered format. P/F indicates Command/Response.
  'BF' in the trace example represents an XID Command/Response.

- Field 8 - XID Information Field

  The XID information fields in Record 1 and 2 are IEEE 802.2 Standard. The XID field in Record 5 is SNA standard as described in the SNA Architecture Reference Summary (GA27-3136). In the trace, the ring station (PC 3270 Emulation) has sent an XID '0200017E0001' where '017' is the IDBLK and E0001 is the IDNUM.

Publication References about trace interpretation

- Token-Ring Network Architecture Reference - SH19-6558
  (In some countries available from IBM Direct as p/n 6165877)

- SNA Architecture Reference Summary - GA27-3236

8.1.3 New 3725 MOSS Functions

The MOSS has been upgraded to support NTRI. New MOSS services permit the MOSS user to look at the status of a particular Token-Ring.

A new TIC dump support function is similar to the existing CSP dump support that exists today. Up to 4 TIC dumps can be stored on the diskette. Dumps can be requested by the operator, or occur automatically in case of an error. If the operator asks for a CSP dump, it is deleted after being sent to the host. The dumps can be formatted with ACF/SSP V3R2.

Several new diagnostics and alarms have been defined for NTRI.

8.2 TRACES FOR TECHNICAL SUPPORT AND APPLICATION PROGRAMMING

8.2.1 PC 3270 Emulation Trace Facility

The PC 3270 Emulation Program provides a Host Line Trace facility. The start procedure for the trace is on the Request Tasks Panel, which can be invoked by pressing the Request key (F10). To be able to get the Request Task Panel, the Communication task must first be started. Thus, problems occurring during initialization of the Communication task (e. g. Communication Check 510) cannot be traced by this trace facility and a Host Line trace is recommended.

For problems occurring after the Communication task has been started successfully, the 3270 Emulation trace is useful, because it produces much less output than a line trace. The Trace is also very useful during development of PC applications using the Application Program Interface (API) which allow a PC application program to communicate with the host.

During the start procedure of the trace, the user is asked for the name of the trace data set. When all trace information is captured, the trace must be turned off to close the data set properly. The trace data can then be formatted with the BASIC program TRACE.BAS to get an interpretable trace listing.

8.2.2 APPC/PC Trace Facility

The APPC/PC program has a built-in trace facility that can be enabled/disabled by an application program using the verb 'Trace'. This trace facility allows tracing of the flow at two possible levels:

- The Application Program Interface level

- The DLC Interface = message level

The Application Program Interface trace shows the verb exchange between the Transaction Program and APPC/PC.

The DLC Interface trace shows the flow of Path Information Units (PIU's) sent/received to or from another APPC/PC station or host.

More information about the APPC/PC trace including an example can be found in the following Publications:

- APPC/PC Programming Guide - p/n 61X3842 or - SX27-3757
- Introduction to Programming for APPC/PC - GG24-3034

8.3 PD FOR NETWORK OPERATING (NETWORK CONTROL CENTER)

8.3.1 Network Management Products

Host Products:

- NetView

Token-Ring Products:

- TR Network Management program
- TR Bridge Program
- Netview/PC

8.3.2 Operating - Token-Ring Network Error Scenario

From the Network Operator's point of view, the recovery and restart activities are nearly the same as for the existing VTAM environment. New considerations are required for the TR Network components and management products.

Example for a hard error on a TRN

To produce a hard error we simulated a broken cable between MSAU's by using a special plug on the RI or RO side of the MSAU.

- Indication on MOSS Console:
  - Hard error = on
  - Ring error recovery = on
  - TIC status is still = open
  - After calling 30 seconds, TIC status = frozen
- Indication on NetView Console (NPDA):
  - Alert message = Token-Ring failure
    Communication Controller and Physical Line Names
  - Detail = Hard error on Token-Ring which has not been recovered
    by the Beaconing process
  - Probable cause = Token-Ring failure. Contact Token-Ring Administrator
- Indication on NetView Console (NCCF):
  - INOP received for Physical Line Code = 1
  - Recovery in progress
  - Unrecoverable error
  - INOP received for Physical PU Code = 1
  - .... After beaconing for 30 seconds ....
  - Termination of Logical Connections and INACT Physical Line
- Indication on TR Network Manager
  - Status area changed immediately from NORMAL to BEACONING
  - The following messages appear at the Event Log Report:
    --> Ring not working: Data items displayed are reporting ring station, NAUN, ring number and error type.
    .... After beaconing for 30 seconds ....
    --> Ring recovery failed: Same information as above.
    --> Beep and Alert Message issued at the Network Manager
- Indication at the network station (PC)
  - PC 3270 Emulation: Communication Check 510 at bottom line

After the Beaconing condition was solved by manual intervention, the Token-Ring was ready to run and the host connections (physical) could be reactivated by:
- ACT LINE = MOSS TIC status change to INITIALIZED
- ACT PU = MOSS TIC status change to OPEN

After the Logical connections were reactivated, host communication sessions could be started.

If a PC where 3270 Emulation was running still indicated 'Communication Check 510' status, a host-initiated connection (dial out) could be established by activating the Logical Unit. The host- initiated connection can be very important if 3174 Control Units are attached.

If the Beaconing condition was recovered (in our test by manual intervention) within 30 seconds, the 3270 sessions were not deactivated and the end user could continue working at the same point at which he was interrupted.

8.3.3 Allocation and Management of Logical Lines

Logical Line Groups are defined in NCP; one possible way to define them is in a separate group for each TIC. The term 'Virtual Line' can also be used (e.g. for X.25) because the allocation of such a line is dynamically mapped onto a real physical resource.

For an incoming call, a free virtual line from the top of the group will be allocated. For an outgoing call, the allocation will start from the bottom of that group.

Separate Groups can be specified for DIAL=IN, =OUT and =INOUT. In our example we used only INOUT because we always wanted to have both possibilities for TR Network devices.

The virtual lines and their names may be automatically generated by NDF. See Appendix (NCP Source - Output from NDF) about what is generated and which virtual line names are automatically defined (e.g. J000D001).

The allocated name of the virtual line can be necessary information for problem determination but VTAM display commands don't show which virtual line has been allocated for a specific connection. If NetView is installed, the name of the allocated virtual line will be displayed by using the Session Configuration function of the NLDM component.

8.4 PD FOR TOKEN-RING NETWORK ADMINISTRATION

The following publication is available for assistance in planning and executing Problem Determination procedures.

- Token-Ring Network Problem Determination Guide
  SY27-0280

This publication describes the following activities:

- Preparing for Problem Determination
- The Problem Determination Procedures
- Preparing a Diskette for Ring Diagnostic
- Using the Ring Diagnostic
- Ring Recovery Procedure for experienced user
- Testing a pair of MSAU's with RI to RO Connection

8.5 PD FOR LOCAL SUPPORT FUNCTION AND PC USERS

The above publication can also be used by Local Support Functions or by experienced users.

Finally, the Network Station or its Software may be the reason for a problem. If the problem occurs during first-time use of a specific application, a definition either in the host or in the network station may be the reason and technical support is normally required to solve such a problem.

If the problem occurs during ongoing work, some tests and activities can be executed by the Local Support Function or by the user before the help desk or technical support is called. If support has to be called, some information should be available before calling.

8.5.1 Sample Procedure

Information required before starting or calling support

- Soft adapter address: If adapter address specified during 'TOKREUI' opened with soft address. Note: TOKREUI does not open adapter.
- Ring number or equivalent topologic information.
- Type of failing workstation application
- Error information: Symptom or message number

Problem determination activities

Are there other stations with problems ?

- If yes, call help desk (if not already done)

Test adapter and ring

- Start the adapter diagnostic program
- Start the ring diagnostic program if the Token-Ring Network Manager is not available

Test Workstation

- Are other (without host communication) applications running ?
- Are other (with host communication) applications running ?
- Check PC disk environment and directory allocations
- Power off/on the PC: A RE-IPL (ALT CTR BRK) of the PC may not be enough to recover from a workstation software problem because the storage above 512k may only be reset during power off/on.
- Restart host communication application
- If not successful, call support.

9.0 APPENDIX A: PLANNING AND IMPLEMENTATION - LIST OF TASKS

Worksheets to record data required to implement the outcome of these planning tasks have been provided in Appendix B.

9.1.1 Planning Tasks

Identify user requirements	
Identify management requirements	
Define objectives	
Define applications with TRN communication / what protocol	
Define applications with Host communication / what protocol	
Define start configuration	
Define target configuration	
Define backup and targets for performance & Availability	
Define changes for VTAM naming conventions	
Define conventions for local administered adapter addresses	
Define administration for adapter addresses	

9.1.2 Ordering Tasks

Configure 3725 (new or upgrade) with TRSS components	
If upgrade, verify that microcode of 3725 is Rel.4 level	
Order 3725 with TRSS components	
If upgrade, order MES for TRSS components (LAB-C's and TIC's)	
Upgrade 3725 storage if required	
Order ACF/NCP V4R2, ACF/SSP V3R2 and opt. NetView (NPDA V3R3)	
Order other products with dependencies on above products	
Order Token-Ring Network components (MSAU's Cables,..)	
Order Token-Ring Network SW products (NETBIOS, APPC/PC,...)	
Order CICS 1.7 if APPC/PC is planned to be used	
Order Host Products for Communication (File Transfer, DISOSS)	

9.1.3 Preparation Tasks

Develop test plan	
Define NTRI and TRN network resources	
Complete worksheets	
Complete administration sheets	
Get familiar with the new NDF function of ACF/SSP V3R2	
Create NCP source with new NTRI parameters	

9.1.4 Installation Tasks - Token Ring Network

Install TRN (test) environment (MSAU's and cables)	
Install TRN adapter cards into workstations	
Install TR Network station software (e.g.TOKREUI)	
Attach TR Network stations to the ring	
Install (opt.) TRN Management Program	
Install TRN software for LAN communication (e.g.PC-LAN PGM)	
Test TRN software for LAN communication	
Get familiar with TRN, test facilities and management	
Install TRN software for host communication (e.g.APPC/PC)	
Setup TRN software for host communication	

9.1.5 Installation Tasks - Host

Install ACF/NCP V4R2, ACF/SSP V3R2 and NetView (or NPDA V3R3)	
Install CICS 1.7 if APPC/PC is planned	
Generate ACF/NCP with NTRI and TRN resources	
Attach Token Ring Network to 3725	
Load new NCP containing NTRI definitions	
Test 3725 MOSS functions for the TRSS	
Activate and test NTRI Physical Link	

9.1.6 Implementation Tasks - Token Ring Network and Host

Get familiar with NTRI and TRN problem determination tools	
Activate and test NTRI logical link	
Prepare and test TRN software with host communication	
Get familiar with NTRI and TRN recovery and beaconing	
Test backup situations	
Test recovery situations	
Develop/customize End-user Interfaces/Procedures	

9.1.7 Post Installation Tasks

Add/establish Token Ring Network Administration	
Establish administration of TRN station addresses	
Add/establish NTRI and TRN management	
Add/establish NTRI operating procedures	
Skill transfer to help desk, operating, info center, ...	
Skill transfer to application department and end-users	
Advice for help desk, operating and info center	
Advice for end-users	

9.1.8 Allocation of Functions

Token-Ring Network Management needs, and the design or development of new integrated applications for host-attached workstations require consideration of additional tasks.

ENVIRONMENT	INVOLVED FUNCTIONS		
Host and 372x	Host Netw. System Programmer Network Operation Network Administration Information Center Application Planning Application Development		
Token-Ring Network	LAN Administration LAN Operation		
Network Station	End-User Local Support Function Information Center Application Planning Application Development		
Independent Functions	Help Desk Technical Support Administration		

10.0 APPENDIX B: PLANNING AND ADMINISTRATION WORKSHEETS

10.1 LIST OF HARDWARE AND SOFTWARE PRODUCTS

Product summary - Software

Product			Avail. Date	Ordered	Shipped
ACF/NCP V4R2					
ACF/SSP V3R2					
ACF/VTAM V3R1.1					
NetView Rel.1					
IBM PC 3270 EMULATION V3					
APPC/PC Rel.1					
NETBIOS Rel.1					
PC LAN PROGRAM V1.1					
TRN MANAGER PROGRAM					
TRN BRIDGE PROGRAM					
PC DOS 3.2					
CICS 1.7 for APPC/PC Comm.					
File Transf.Progr.TSO/CMS/CICS					
ECF - TSO/E, VM/CMS Servers					
ECF - IBM PC Requesters					
DISOSS, PS/PC - MVS, PC					
PROFS, PROFS/PC - VM/SP, PC					

Product summary - Hardware

Product			Avail. Date	Ordered	Shipped
TRN PC ADAPTER I					
TRN PC ADAPTER II					
TRSS for IBM 372x					
IBM 3174-3R					
8228 MULTI-STATION ACCESS UNIT					
82xx TRN REPEATERs					
PATCH CABLES for MSAU's					
CABLES for TRN STATIONS					
OTHER TRN EQUIPMENT					

10.2 USER REQUEST AND STATION ADDRESS ADMINISTRATION - WORKSHEETS

User Request - Worksheet

NAME:	STATION TYPE:
DEPARTMENT:	STATION ADDRESS (HARD):
BUILDING:	APPLICATIONS REQUESTED: 3270 EMULATION: APPC/PC: - -
ROOM NUMBER: TEL:	
COMMENTS:	
INFORMATION GIVEN BACK TO THE USER FOR SETUP WORKSTATION ENVIRONMENT	
ALLOCATED SOFT ADDRESS: ALLOCATED LU-NAME's: ALLOCATED RING NUMBER: ALLOCATED ID-VALUES: IDBLK's: IDNUM's: ALLOCATED TIC NUMBER's PRIM: ALTERN.:	

Station Address Administration - Worksheet

					STATUS			
USER NAME	ST.TYPE	SUBA.	RING NR.	SOFT ADDRESS	A	B	C	D
/	/	/	/	/	/	/	/	/

STATUS A = IN USE
STATUS B = PREGENERATED BUT NOT USED
STATUS C = PREGENERATION REQUESTED
STATUS D = ...

10.3 NETWORK TOPOLOGIES - WORKSHEET

RING NUMBER:		COMMENT:				
HOST ATTACH PORTS:						
BRIDGE NUMBERs:						
BUILDING NUMBERs:						

BUILDING NUMBER:										
WIRING CLOSET NUMBER:										
MSAU NUMBER:	FROM	Used		.	.	.	.	.	.	TO
MSAU NUMBER:	FROM	.	.	Free		.	.	.	.	TO
WIRING CLOSET NUMBER:										
MSAU NUMBER:	FROM	Used		.	.	.	.	.	.	TO
MSAU NUMBER:	FROM	.	.	Free		.	.	.	.	TO
BUILDING NUMBER:										
WIRING CLOSET NUMBER:										
MSAU NUMBER:	FROM	Used		.	.	.	.	.	.	TO
MSAU NUMBER:	FROM	.	.	Free		.	.	.	.	TO
WIRING CLOSET NUMBER:										
MSAU NUMBER:	FROM	Used		.	.	.	.	.	.	TO
MSAU NUMBER:	FROM	.	.	Free		.	.	.	.	TO

10.4 NETWORK DEFINITION - WORKSHEET

SOFT ADDRESS:		COMMENTS:
HARD ADDRESS:		
LAN MNGR. ID:		
STATION TYP:		

NCP NAME:			ALTERNATE:
PHYSICAL LINE NAME:		PORT:	ALTERNATE:
PHYSICAL PU NAME:			

SWITCHED MAJOR NODE NAME:			ALTERNATE:	
PU-NAMEs:				
IDNUM:				
LU-NAMEs:				

CICS APPLICATION NAME:			
'TRMIDNT' NAMEs:			
'TRMTYPE' NAMEs:			
'NETNAME' NAMEs:			

One Page per Token Ring Network Station.
Used by Sysprog, LAN and Host Operation.

11.0 APPENDIX C: SAMPLE SOURCE FOR NCP, VTAM AND CICS

11.1 NCP SOURCE - INPUT TO NDF

```
*********************************************************************** 00012000
*    NEWNAME = NTRINCP      *         *    UNITSZ   = 152             * 00020003
*****************************         *    MAXBFRU  = 34              * 00030000
*                           * NETVIEW *    MAXSUBA  = 63              * 00040000
* ACF/NCP V4R2 FOR 3725     *         *    SUBAREA  = 13              * 00050003
*  (07/16/86)               *         *                               * 00060017
*********************************************************************** 00070000
* MODIFICATIONS:                                                      * 00080000
*                SEVERAL FOR NTRI SUPPORT                             * 00090004
*********************************************************************** 00160000
*                                                                       01240000
*     LAB POSITION 3 (TYPE C )                                          01250004
* ____________________________________________________                  01421004
*|____|________|_____|__________|___|___|_______|_____|                 01400004
*|LINE|LINE I/F| HEX |NCP LINE  |TIC|TRM|TAILG. |SPEED|                 01401004
*|ADDR|ADDR    | ADDR|NAME      |POS|#  |  POS. |     |                 01402004
*|____|________|_____|__________|___|___|_______|_____|                 01403004
*| 80 |160/161 |A0/A1| L13080   | 5 | 1 |  N2   |4 MEG|                 01410004
*| 81 |162/163 |A2/A3| L13081   | 6 | 1 |  P2   |4 MEG|                 01420004
*|____|________|_____|__________|___|___|___ ___|_____|                 01421004
*                                                                       01460004
*                                                                       01510000
*********************************************************************** 01511008
*       OPTIONS DEFINITION STATEMENT                                  * 01512008
*********************************************************************** 01520000
NCPOPT   OPTIONS NEWDEFN=YES       >>> Required for NTRI                01520111
*********************************************************************** 01521008
*       PCCU'S    MACRO SPECIFICATIONS                                * 01530000
*********************************************************************** 01540000
         PRINT NOGEN                                                    01560000
VTAMX    PCCU  ....                                                    X01570000
*              ....                                                     02380000
*********************************************************************** 02390000
*       BUILD MACRO SPECIFICATIONS                                    * 02400000
*********************************************************************** 02410000
NCPBUILD BUILD BFRS=(120),          NCP BUFFER SIZE,EP FREE BUFFER      X02420000
               BRANCH=500,          BRANCH TRACE ENTRIES              V3X02430000
               CA=(TYPE5-TPS,TYPE5),         CHANNEL ADAPTER TYPE     V3X02440000
               CATRACE=(YES,100),            CHANNEL ADAPTER TRACE    V3X02450000
               CSMHDR=27F5C711C3F0405C40C8C4D9405C, 3270 CRITSIT HEADERX02460000
               CSMHDRC=40E3C5E7E3405C5C, 3270 CRITST HEADER EXTRA TEXT X02470000
               CSMSG=C3D9C9E3E2C9E35A40E385819440F040, CRITSIT MESG    X02480000
               CSMSGC=6040C1D5E240828587A4954B, CRITST MESG EXTRA TEXT X02490000
               CWALL=26,            MIN. BUFFERS BEFORE SLOWDOWN        X02500000
               DELAY=(0.2,0.2),     CA ATT.-DELAY FOR V3 BUILD          X02510000
               DSABLTO=6.5,                                            X02520000
               ENABLTO=6.5,         IBM 386X REQUIRE 6.5 AS MINIMUM     X02530000
               LOADLIB=NCPLOAD,     LIB FOR LOAD MODULE                 X02540000
               LOCALTO=1.5,       * NTRI ACK.TIMER FOR LOCAL TOKEN-RING X02541006
               QUALIFY=NCP43725,    QUALIFIER FOR STAGE 2 DECK NAMES    X02550000
               LTRACE=4,            SIT FOR 4 LINES                     X02560000
               MAXSSCP=8,           8 SSCP'S CAN ACTIVATE THIS NCP      X02570000
               MAXSUBA=63,          ALLOW FOR UP TO  63 SUBAREAS        X02580000
               MEMSIZE=768,         3725 STORAGE SIZE IS 768K BYTES     X02590000
               MODEL=3725,                                             X02600000
               MXRLINE=2,         * NTRI PHYSICAL CONNECRIONS           X02600106
               MXVLINE=25,        * NTRI LOGICAL CONNECTIONS            X02600206
               TRANSFR=25,                                             X02601000
               NCPCA=(ACTIVE,ACTIVE),                                V3X02610000
               NEWNAME=N139F4M,     NAME OF THIS LOAD MODULE          V3X02620013
               NPA=YES,                                             NPAX02630000
               NUMHSAS=6,           6 HOSTS MAY COMMUNICATE CONCURRENTLYX02640000
               REMOTTO=1.5,       * NTRI ACK.TIMER FOR REMOTE TOKEN-RINGX02641006
               RESOEXT=64,          ALLOW 64 NAU'S TO BE REUSED         X02650000
               SUBAREA=13,          SUBAREA ADDRESS = 13                X02660000
               TIMEOUT=(120,120),   ANS BEGINS AFTER, FOR V3 IN BUILD   X02670000
               TRACE=(YES,64),      64 ADDRESS-TRACE ENTRIES            X02680000
               TYPGEN=NCP,        NCP ONLY                             X02690000
               TYPSYS=OS,         OS GENERATION                        X02700000
               VERSION=V4R2       NDF VERSION INDICATOR                 02710011
*              VNCA=('19F...........')                                  02720000
*********************************************************************** 02730008
*       SYSCNTRL MACRO SPECIFICATIONS                                 * 02740008
*********************************************************************** 02750008
NCPSYSC SYSCNTRL OPTIONS=(BHSASSC,ENDCALL,MODE,RCNTRL,RCOND,RECMD,RIMM,X02760008
               NAKLIM,SESSION,SSPAUSE,XMTLMT,STORDSP,DLRID,RDEVQ)       02770008
*********************************************************************** 02780000
*       HOST MACRO SPECIFICATIONS                                     * 02790000
*********************************************************************** 02800000
M01      HOST  INBFRS=10,          NCP BUFFERS  ALLOCATION             X02810000
               MAXBFRU=34,         UP TO 34 VTAM BUFFERS SHIPPED       X02820000
               UNITSZ=152,         VTAM IO BUFFERS SIZE                X02830000
               BFRPAD=0,           BUFFER PAD (MANDATORY FOR ACF)      X02840000
               SUBAREA=(11)        CHANNEL ATTACHED HOSTSA   REL 3      02850000
M02      HOST  ....                                                     02860000
*********************************************************************** 03120000
*        DYNAMIC RECONFIGURATION POOL SPACE                           * 03130000
*********************************************************************** 03140000
*                                                                       03150000
DRPOOLPU PUDRPOOL NUMBER=8,        CAN ADD 8 PUS                       X03160000
               MAXLU=64            A MAX OF 64 LUS PER PU               03170000
*                                                                     * 03180000
DRPOOLLU LUDRPOOL NUMTYP1=10,      RESERVE 10 LUS ON PU.T1 PUS         X03190000
               NUMTYP2=90          RESERVE 90  LUS ON PU.T2 PUS         03200000
*********************************************************************** 03210000
*        PATH SPECIFICATIONS                                          * 03220000
*                                        MARCH 25  1985               * 03230000
*********************************************************************** 03240000
         PATH  ....                                                    X03260000
*********************************************************************** 03786000
*********************************************************************** 37351104
*       PHYSICAL GROUP FOR NTRI TIC 1                                 * 37351204
*********************************************************************** 37351304
EG13P00  GROUP ECLTYPE=PHYSICAL                                         37351406
*                                                                       37351505
EL13080  LINE  ADDRESS=(80,FULL),PORTADD=0,LOCADD=400011301000,        X37351617
               RCVBUFC=4095,                                           X37351717
               MAXTSL=1108                                              37351817
*                                                                       37351905
EP13080  PU                                                             37352004
*                                                                       37352105
EU13080  LU    ISTATUS=INACTIVE                                         37352206
```

```
*********************************************************************** 37352304
*       PHYSICAL GROUP FOR NTRI TIC 2                                 * 37352404
*********************************************************************** 37352504
EG13P01  GROUP ECLTYPE=PHYSICAL                                         37352606
*                                                                       37352705
EL13081  LINE  ADDRESS=(81,FULL),PORTADD=1,LOCADD=400011301001,        X37352917
               RCVBUFC=4095,                                           X37353017
               MAXTSL=1108                                              37353117
*                                                                       37353205
EP13081  PU                                                             37353307
*                                                                       37353405
EU13081  LU    ISTATUS=INACTIVE                                         37353507
*********************************************************************** 37353604
*       LOGICAL  GROUP FOR NTRI TIC 1                                 * 37353704
*********************************************************************** 37353804
EG13L00  GROUP ECLTYPE=LOGICAL,                                        X37353906
               AUTOGEN=20,                                             X37354006
               CALL=INOUT,                                             X37354104
               MAXLU=5,                                                X37354215
               PHYPORT=0                                                37354306
*********************************************************************** 37354404
*       LOGICAL  GROUP FOR NTRI TIC 2                                 * 37354514
*********************************************************************** 37354605
EG13L01  GROUP ECLTYPE=LOGICAL,                                        X37354706
               AUTOGEN=5,                                              X37354806
               CALL=INOUT,                                             X37354904
               MAXLU=5,                                                X37355015
               PHYPORT=1                                                37355106
*********************************************************************** 37355204
         GENEND                                                         37358012
         END                                                            40280000
*********************************************************************** 37355204
```

11.2 NCP SOURCE - OUTPUT FROM NDF

```
NCPOPT   OPTIONS NEWDEFN=YES
************************************************************************ 01521008
*       PCCU'S    MACRO SPECIFICATIONS                                 * 01530000
************************************************************************ 01540000
VTAMX    PCCU ....
************************************************************************ 02390000
*       BUILD MACRO SPECIFICATIONS                                     * 02400000
************************************************************************ 02410000
NCPBUILD BUILD BFRS=(120),BRANCH=500,CA=(TYPE5-TPS,TYPE5),CATRACE=(YES,*
               100),CSMHDR=27F5C711C3F0405C40C8C4D9405C,CSMHDRC=40E3C5E*
               7E3405C5C,CSMSG=C3D9C9E3E2C9E35A40E385819440F040,CSMSGC=*
               6040C1D5E240828587A4954B,CWALL=26,DELAY=(0.2,0.2),DSABLT*
               O=6.5,ENABLTO=6.5,LOADLIB=NCPLOAD,LOCALTO=1.5,QUALIFY=NC*
               P43725,LTRACE=4,MAXSSCP=8,MAXSUBA=63,MEMSIZE=768,MODEL=3*
               725,MXRLINE=2,MXVLINE=25,TRANSFR=25,NCPCA=(ACTIVE,ACTIVE*
               ),NEWNAME=N139F4M,NPA=YES,NUMHSAS=6,REMOTTO=1.5,RESOEXT=*
               64,SUBAREA=13,TIMEOUT=(120,120),TRACE=(YES,64),TYPGEN=NC*
               P,TYPSYS=OS,VERSION=V4R2
*              VNCA=('19F...........')                                   02720000
************************************************************************ 02730008
*       SYSCNTRL MACRO SPECIFICATIONS                                  * 02740008
************************************************************************ 02750008
NCPSYSC  SYSCNTRL OPTIONS=(BHSASSC,ENDCALL,MODE,RCNTRL,RCOND,RECMD,RIMM*
               ,NAKLIM,SESSION,SSPAUSE,XMTLMT,STORDSP,DLRID,RDEVQ)
************************************************************************ 02780000
*       HOST MACRO SPECIFICATIONS                                      * 02790000
************************************************************************ 02800000
M01      HOST INBFRS=10,MAXBFRU=34,UNITSZ=152,BFRPAD=0,SUBAREA=(11)
************************************************************************ 03120000
*        DYNAMIC RECONFIGURATION POOL SPACE                            * 03130000
************************************************************************ 03140000
*                                                                      * 03180000
DRPOOLPU PUDRPOOL NUMBER=8,MAXLU=64
DRPOOLLU LUDRPOOL NUMTYP1=10,NUMTYP2=90
************************************************************************ 03210000
*        PATH SPECIFICATIONS                                           * 03220000
************************************************************************ 03240000
         PATH ...
************************************************************************ 03786000
************************************************************************ 37351104
*       PHYSICAL GROUP FOR NTRI TIC 1                                  * 37351204
************************************************************************ 37351304
EG13P00  GROUP ECLTYPE=PHYSICAL,TYPE=NCP,DIAL=NO,LNCTL=SDLC,LEVEL2=ECLN*
               ARL2,LEVEL3=ECLNARL3,LEVEL5=NCP,TIMER=(ECLNART1,,ECLNART*
               2,ECLNART3),XIO=(ECLNARXL,ECLNARXS,ECLNARXI,ECLNARXK),US*
               ERID=(5668854,ECLRBDT,NORECMS),MAXPU=1,SPEED=9600,NPACOL*
               L=NO,PUTYPE=1,PUDR=NO,MAXLU=1,TI=NO
*                                                                        37351505
EL13080  LINE ADDRESS=(80,FULL),PORTADD=0,LOCADD=400011301000,RCVBUFC=4*
               095,MAXTSL=1108,UACB=(X$P1AX,X$P1AR)
* GENERATED BY ECL
J000D01S SERVICE
*                                                                        37351905
EP13080  PU ADDR=01
*                                                                        37352105
EU13080  LU ISTATUS=INACTIVE,LOCADDR=0
```

```
*********************************************************************** 37352304
*       PHYSICAL GROUP FOR NTRI TIC 2                                 * 37352404
*********************************************************************** 37352504
EG13P01  GROUP ECLTYPE=PHYSICAL,TYPE=NCP,DIAL=NO,LNCTL=SDLC,LEVEL2=ECLN*
               ARL2,LEVEL3=ECLNARL3,LEVEL5=NCP,TIMER=(ECLNART1,,ECLNART*
               2,ECLNART3),XIO=(ECLNARXL,ECLNARXS,ECLNARXI,ECLNARXK),US*
               ERID=(5668854,ECLRBDT,NORECMS),MAXPU=1,SPEED=9600,NPACOL*
               L=NO,PUTYPE=1,PUDR=NO,MAXLU=1,TI=NO
*                                                                       37352705
EL13081  LINE ADDRESS=(81,FULL),PORTADD=1,LOCADD=400011301001,RCVBUFC=4*
               095,MAXTSL=1108,UACB=(X$P2AX,X$P2AR)
* GENERATED BY ECL
J000D02S SERVICE
*                                                                       37353205
EP13081  PU ADDR=01
*                                                                       37353405
EU13081  LU ISTATUS=INACTIVE,LOCADDR=0
*********************************************************************** 37353604
*       LOGICAL  GROUP FOR NTRI TIC 1                                 * 37353704
*********************************************************************** 37353804
EG13L00  GROUP ECLTYPE=LOGICAL,AUTOGEN=20,CALL=INOUT,MAXLU=5,PHYPORT=0,*
               TYPE=NCP,DIAL=YES,LNCTL=SDLC,LEVEL2=ECLNAVL2,LEVEL3=ECLN*
               AVL3,LEVEL5=NCP,TIMER=(ECLNAVT1,,ECLNAVT2,ECLNAVT3),XIO=*
               (ECLNAVXL,ECLNAVXS,ECLNAVXI,ECLNAVXK),USERID=(5668854,EC*
               LVBDT,NORECMS),LINEADD=NONE,LINEAUT=YES,MAXPU=1,NPACOLL=*
               NO,PUTYPE=2,TI=NO
* GENERATED BY ECL
J000D001 LINE UACB=X$L1A
* GENERATED BY ECL
J000D002 PU
* GENERATED BY ECL
J000D003 LINE UACB=X$L2A
* GENERATED BY ECL
J000D004 PU
* GENERATED BY ECL
J000D005 LINE UACB=X$L3A
* --------> UP TO X$L20A GENERATED BY ECL
*********************************************************************** 37354404
*       LOGICAL  GROUP FOR NTRI TIC 2                                 * 37354514
*********************************************************************** 37354605
EG13L01  GROUP ECLTYPE=LOGICAL,AUTOGEN=5,CALL=INOUT,MAXLU=5,PHYPORT=1,T*
               YPE=NCP,DIAL=YES,LNCTL=SDLC,LEVEL2=ECLNAVL2,LEVEL3=ECLNA*
               VL3,LEVEL5=NCP,TIMER=(ECLNAVT1,,ECLNAVT2,ECLNAVT3),XIO=(*
               ECLNAVXL,ECLNAVXS,ECLNAVXI,ECLNAVXK),USERID=(5668854,ECL*
               VBDT,NORECMS),LINEADD=NONE,LINEAUT=YES,MAXPU=1,NPACOLL=N*
               O,PUTYPE=2,TI=NO
* GENERATED BY ECL
J000D029 LINE UACB=X$L21A
* GENERATED BY ECL
J000D02A PU
* --------> UP TO X$L25A GENERATED BY ECL
*********************************************************************** 37355204
         GENEND INIT=ECLINIT,TMRTICK=ECLTICK
```

11.3 VTAM SWITCHED MAJOR NODE - DEFINITIONS FOR TRN DEVICES

```
**********************************************************************
*                                                                    *
*         VTAM SWITCHED MAJOR NODE FOR NTRI                          *
*                                                                    *
**********************************************************************
*                                                                    *
E13SW   VBUILD MAXGRP=5,               REQUIRED                        X00010480
               MAXNO=12,               REQUIRED                        X00010490
               TYPE=SWNET              REQUIRED                         00010500
*                                                                    *
*                                                                    *
**********************************************************************
*                  PU, LU and PATH for PC 1
**********************************************************************
*
*************************************
* PU, LU and PATH for PC 3270 EMUL. *
*************************************
**                                                                   *
E13PS01  PU    ADDR=04,                COULD BE ANYTHING (NOT USED)  * X00010540
               IDBLK=017,              PC 3270 EMULATION             * X00010550
               IDNUM=E0001,            PC 3270 EMULATION             * X00010560
               DISCNT=NO,                                            * X00010570
               MAXDATA=261,            NOT REQUIRED FOR SW LINES     * X00010580
               MAXOUT=1,                                             * X00010580
               MODETAB=MT327X,                                       * X00010580
               PASSLIM=7,                                            * X00010580
               MAXPATH=2,                                            * X00010590
               VPACING=0,                                            * X00010610
               PUTYPE=2,                                             * X00010620
               SSCPFM=USSSCS,                                        * X00010630
               USSTAB=US327X,                                        * X
               DLOGMOD=T3278M2
**                                                                      00010640
E13D0101 PATH  DIALNO=0004400031301001, TO 3270 EMUL.PC              * X00010830
               GRPNM=EG13L00,           FROM TIC 1                   * X00010840
               GID=1,                                                * X00010850
               PID=1                                                    00010860
**
E13D0102 PATH  DIALNO=0104400031301001, TO 3270 EMUL.PC              * X00010830
               GRPNM=EG13L01,           FROM TIC 2                   * X00010840
               GID=1,                                                * X00010850
               PID=2,                                                  X00010860
               USE=NO                   INITIALLY INACTIVE
E13L0102 LU    LOCADDR=2                                                00010650
**
**********************************************************************
```

```
*** VTAM  Cont. *******************************************************
************************************
* PU, LU and PATH for APPC/PC      *
************************************
**
E13PS02  PU    ADDR=04,                COULD BE ANYTHING (NOT USED)   * X00010540
               IDBLK=050,              FOR APPC/PC                    * X00010550
               IDNUM=A0001,            FOR APPC/PC                    * X00010560
               DISCNT=NO,                                             * X00010570
               MAXDATA=261,            NOT REQUIRED FOR SW LINES      * X00010580
               MAXOUT=1,                                              * X00010580
               MODETAB=MTAPPC,                                        * X00010580
               PASSLIM=7,                                             * X00010580
               MAXPATH=2,                                             * X00010590
               VPACING=0,                                             * X00010610
               PUTYPE=2,                                              * X00010620
               SSCPFM=USSSCS,                                         * X00010630
               USSTAB=US327X,                                         * X
               DLOGMOD=APPCNEGB
**                                                                       00010640
E13D0201 PATH  DIALNO=0004400031301001, TO APPC/PC PC                 * X00010830
               GRPNM=EG13L00,           FROM TIC 1                    * X00010840
               GID=1,                                                 * X00010850
               PID=1                                                     00010860
**
E13D0202 PATH  DIALNO=0104400031301001, TO APPC/PC PC                 * X00010830
               GRPNM=EG13L01,           FROM TIC 2                    * X00010840
               GID=1,                                                 * X00010850
               PID=2,                                                   X00010860
               USE=NO                   INITIALLY INACTIVE
**
E13L0203 LU    LOCADDR=3                                                 00010650
**
***********************************************************************
*                  PU, LU and PATH for PC 2
***********************************************************************
*
*************************************
* PU, LU and PATH for PC 3270 EMUL. *
*************************************
**
E13PS03  PU    ....
         PATH  ....
         LU    ....
**
*************************************
* PU, LU and PATH for APPC/PC       *
*************************************
**
E13PS04  PU    ....
         PATH  ....
         LU    ....
**** End of VTAM Definitions ******************************************
```

11.4 CICS SOURCE - TCT FOR PC 3270 EMUL. AND APPC/PC

```
*********************************************************************** 02100000
*                                                                     * 02110000
* THE FOLLOWING ENTRIES DEFINE SOME PC SYSTEMS ON A 3725 TRN          * 02120002
*                                                                     * 02130000
*        E201<---------->E13L0102  LUT2                               * 02140002
*        E603<---------->E13L0203  LUT6.2                             * 02160002
*        E202<---------->E13L0302  LUT2                               * 02150002
*        E604<---------->E13L0403  LUT6.2                             * 02170002
*                                             JOAN MCCORMACK          * 02240001
*                                                    JULY 1986        * 02250001
*********************************************************************** 02260000
*
         DFHTCT TYPE=TERMINAL,TRMIDNT=E201,PGESIZE=(24,80),            X02270002
               BUFFER=256,                                             X02280000
               TRMMODL=2,                                              X02290000
               LOGMODE=EM3270,                                         X02300000
               TRMTYPE=LUTYPE2,                                        X02310000
               ACCMETH=VTAM,FEATURE=(UCTRAN),                          X02320000
               GMMSG=YES,NETNAME=E13L0102,                             X02330002
               TCTUAL=64,TIOAL=(512,4096),RELREQ=(YES,YES),            X02340000
               TRMSTAT=TRANSCEIVE,CHNASSY=YES,RUSIZE=256                02350000
         DFHTCT TYPE=TERMINAL,TRMIDNT=E202,PGESIZE=(24,80),            X02360002
               BUFFER=256,                                             X02370000
               TRMMODL=2,                                              X02380000
               LOGMODE=EM3270,                                         X02390000
               TRMTYPE=LUTYPE2,                                        X02400000
               PGESTAT=PAGE,                                           X02410000
               ACCMETH=VTAM,FEATURE=(UCTRAN),                          X02420000
               GMMSG=YES,NETNAME=E13L0302,                             X02430002
               TCTUAL=64,TIOAL=(256,4096),RELREQ=(YES,YES),            X02440000
               TRMSTAT=TRANSACTION,CHNASSY=YES,RUSIZE=256               02450000
         DFHTCT TYPE=SYSTEM,       PC  NTRI LAN                        C02541000
               TRMTYPE=LUTYPE62,   (FOR DISSOS OR APPC)                C02542000
               ACCMETH=VTAM,                                           C02543000
               SYSIDNT=E603,                                           C02544002
               NETNAME=E13L0203,                                       C02545003
               FEATURE=SINGLE,                                         C02546000
               MODENAM=APPCNEGB,                                       C02547000
               RUSIZE=256,                                             C02548000
               BUFFER=256,                                             C02549000
               TRMSTAT=TRANSCEIVE                                       02549100
         DFHTCT TYPE=SYSTEM,       PC  NTRI LAN                        C02549202
               TRMTYPE=LUTYPE62,   (FOR DISSOS OR APPC)                C02549302
               ACCMETH=VTAM,                                           C02549402
               SYSIDNT=E604,                                           C02549502
               NETNAME=E13L0403,                                       C02549602
               FEATURE=SINGLE,                                         C02549702
               MODENAM=APPCNEGB,                                       C02549802
               RUSIZE=256,                                             C02549902
               BUFFER=256,                                             C02550002
               TRMSTAT=TRANSCEIVE                                       02560002
*
*********************************************************************** 02260000
```

12.1 TRACE EXAMPLE: PC 3270 EMULATION - HOST INITIATED CALL

```
                                 ADVANCED COMMUNICATIONS FUNCTION
                                      TRACE ANALYSIS PROGRAM
          DATE: 08:21:86                 LINE TRACE DETAIL
      TIMESTAMP:  11.45.14
ELEMNT             ENTRY TYPE/
NUMBER ID-COMMAND SCANNER STATE                 HEX
           ***************************************
           *  OPEN AND INITIALIZE THE TIC        *
           ***************************************
000007 NTRI LINE  OPEN         4007767C  0009548C 0007767E 70002343
000003 NTRI LINE  OPEN         4009440C 00 038000 00077856 78CC0303
000004 NTRI IOH                4A952000  4AC0A000
000005 NTRI LINE  SCB CLEAR    4009440C  00044000 000777C6 78CC03CB
000006 NTRI IOH                4AC09880  4A952000 4AC0A000
000007 NTRI LINE  OPEN         4007767C  0009548C 0007767E 70002343
000008 NTRI IOH                4AC09880  4A952000 4AC0A000
000009 NTRI LINE  OPEN         400776A0  00095594 000776A2 70001363
000010 NTRI IOH                4AC08680  4AC08880 4A952000 4AC0A000
000011 NTRI LINE  SCB CLEAR    4009440C  00044000 000777EA 78CC10CB
000012 NTRI IOH                4AA52000  4AC08880 4A952000 4AC0A000
000013 NTRI LINE  SCB CLEAR    4009440C  00044000 0007780E 78CC10CB
000014 NTRI IOH                4AA52000  4AC08180 4AC0A000
           ***************************************
           *  TRANSMIT AND RECEIVE TEST FRAME    *
           ***************************************
000015 NTRI LINE  TRANSMIT     4009AD3C  00044000 C107780E 78CC0404 00404000
                               00313000  40000113 00000004 F3000789 A8
000016 NTRI IOH                4AC0A400
000017 NTRI LINE  RECEIVE      40077778  00099710 C107777A 70001006 10404000
                               01130000  40000031 30000000 00000000 00000000
                               00000000  00000000 0401F300 0789A8
000018 NTRI IOH                4AC08280 4AC08180 4AC0A000
           ***************************************
           *  TRANSMIT AND RECEIVE XID           *
           ***************************************
000019 NTRI LINE  TRANSMIT     4009AD3C 00044000 00077832 78CC0404 00404000
                               00313000 40000113 00000404 BF
000020 NTRI IOH                4AC0A400
000021 NTRI LINE  RECEIVE      4007779C 0009989C 0007779E 70001006 10404000
                               01130000 40000031 30000000 00000000 00000000
                               00000000 00000000 0405BF02 00017E00 01
000022 NTRI IOH                4AC08280 4AC08180 4AC0A000
           ***************************************
           *  TRANSMIT SABME                     *
           ***************************************
000023 NTRI LINE  TRANSMIT     4009AD3C 00044000 00077856 78CC0404 00404000
                               00313000 40000113 00000404 7F
000024 NTRI IOH                4AC0A400
           ***************************************
           *  RECEIVE UA                         *
           ***************************************
000025 NTRI LINE  RECEIVE      400775C8 00099A28 000775CA 70001006 10404000
                               01130000 40000031 30000000 00000000 00000000
                               00000000 00000000 040573
000026 NTRI IOH                4AC08280 4AC08180 4AC0A000
           ***************************************
           *  TRANSMIT AND RECEIVE RR            *
           ***************************************
000027 NTRI LINE  TRANSMIT     4009AD3C 00044000 000777C6 78CC0404 00404000
                               00313000 40000113 00000404 0101
000028 NTRI IOH                4AC0A400
000029 NTRI LINE  RECEIVE      400775EC 00099BB4 000775EE 70001006 10404000
                               01130000 40000031 30000000 00000000 00000000
                               00000000 00000000 04050101
000030 NTRI IOH                4AC08280 4AC08180 4AC0A000
           ***************************************
           *  TRANSMIT ACT PU                    *
           ****************************************
000031 NTRI LINE  TRANSMIT     4009AD3C 00044000 000777EA 78CC0404 00404000
                               00313000 40000113 00000404 00002D00 00000D07
                               6B800011 02010500 000000                 0B
000032 NTRI IOH                4AC0A400
           ****************************************
           *  RECEIVE RR                          *
           ****************************************
000033 NTRI LINE     RECEIVE   40077610 00099D40 00077612 70001006 10404000
                               01130000 40000031 30000000 00000000 00000000
                               00000000 00000000 04050102
000034 NTRI IOH                4AC08280 4AC0A400
           ****************************************
           *  RECEIVE ACT PU RESPONSE             *
           ****************************************
000035 NTRI LINE     RECEIVE   C0077634 00099ECC 00077636 70001006 10404000
                               01130000 40000031 30000000 00000000 00000000
                               00000000 00000000 04040002 2D000000 0D07EB80
                               00111140 40404040 4040
000036 NTRI IOH                4AC08280 4AC08180 4AC0A000
           ****************************************
           *  TRANSMIT RR                         *
           ****************************************
000037 NTRI LINE     TRANSMIT  4009AD3C 00044000 0007780E 78CC0404 00404000
                               00313000 40000113 00000405 0102
000038 NTRI IOH                4AC08180 4AC0A000
           ****************************************
           *  TRANSMIT ACT LU                     *
           ****************************************
000039 NTRI LINE     TRANSMIT  4009AD3C 00044000 00077832 78CC0404 00404000
                               00313000 40000113 00000404 02022D00 02000D08
                               6B80000D 0201
000040 NTRI IOH                4AC0A400
           ****************************************
           *  RECEIVE RR                          *
           ****************************************
000041 NTRI LINE     RECEIVE   40077658 0009A058 0007765A 70001006 10404000
                               01130000 40000031 30000000 00000000 00000000
                               00000000 00000000 04050104
000042 NTRI IOH                4AC08280 4AC0A400
           ****************************************
           *  RECEIVE ACT LU RESPONSE             *
           ****************************************
000043 NTRI LINE     RECEIVE   4007767C 0009A1E4 0007767E 70001006 10404000
                               01130000 40000031 30000000 00000000 00000000
                               00000000 00000000 04040204 2D000002 0D08EB80
                               000D0101 00850000 000C
000044 NTRI IOH                4AC08280 4AC08180 4AC0A000
           ****************************************
           *  TRANSMIT RR                         *
           ****************************************
000045 NTRI LINE     TRANSMIT  4009AD3C 00044000 00077856 78CC0404 00404000
                               00313000 40000113 00000405 0104
000046 NTRI IOH                4AC0A400
           ****************************************
           *  RECEIVE NOTIFY                      *
           ****************************************
000047 NTRI LINE     RECEIVE   400776A0 0009A370 000776A2 70001006 10404000
                               01130000 40000031 30000000 00000000 00000000
                               00000000 00000000 04040404 2C000002 00000B80
                               00810620 0C060300 0100
000048 NTRI IOH                4AC08280 4AC08180 4AC0A000
           ****************************************
           *  TRANSMIT RR                         *
           ****************************************
000049 NTRI LINE     TRANSMIT  4009AD3C 00044000 000777C6 78CC0404 00404000
                               00313000 40000113 00000405 0106
000050 NTRI IOH                4AC08180 4AC0A000
           ****************************************
           *  TRANSMIT MSG 10                     *
           ****************************************
000051 NTRI LINE     TRANSMIT  4009AD3C 00044000 000777EA 78CC0404 00404000
                               00313000 40000113 00000404 04062800 02000001
                               03800040 15D4E2C7 F1F040D4
000052 NTRI IOH                4AC0A400
000053 NTRI LINE     RECEIVE   400776C4 0009A4FC 000776C6 70001006 10404000
                               01130000 40000031 30000000 00000000 00000000
                               00000000 00000000 04050106
```

13.0 APPENDIX E: SAMPLE SOURCES FOR APPC/PC

13.1 APPC/PC - APPLICATION SUBSYSTEM EXAMPLE

```
**********************************************************************
;*********************************************************************
;*                                                                   *
;*          APPLICATION SUBSYSTEM -                                  *
;*                                                                   *
;* Purpose  This is the application subsystem for the CICS TEST      *
;*                                                                   *
;*********************************************************************
;* Stack Segment - STACKSEG                                          *
;*********************************************************************
;* Note  The size of the stack is used in step 5.
stackseg   segment para stack 'stack'
           db      64 dup ("Stack...")
stackseg   ends
;*********************************************************************
;* Data Structure Definitions                                        *
;*********************************************************************
;*
;* Include the necessary APPC/PC verb structure definitions.
;*
include    attachpu.str                ; ATTACH_PU    Data Structure
include    attachlu.str                ; ATTACH_LU    Data Structure
include    part_lu.str                 ; PARTNER_LU   Data Structure
include    mode.str                    ; MODE         Data Structure
include    act_dlc.str                 ; ACTIVATE_DLC Data Structure
include    cnos.str                    ; CNOS         Data Structure
include    convert.str                 ; CONVERT      Data Structure
include    getalloc.str                ; GET_ALLOCATE Data Structure
include    tp_valid.str                ; TP_VALID     Data Structure
include    createtp.str                ; CREATE_TP    Data Structure
include    tp_init.str                 ; TP_INITIATE  Data Structure
include    detachlu.str                ; DETACH_LU    Data Structure
include    detachpu.str                ; DETACH_PU    Data Structure
include    tp_start.str                ; TP_STARTED   Data Structure

;*
;* Parameter Block structure used to Load and Execute the Transaction
;* Program
;*
parm_blk   struc                       ; Parameter Block (see step 6)
env_addr   dw      0                   ; Segment address of environment
parm_ptr   dd      0                   ; Parameter string address
fcb5c_ptr  dd      0                   ; FCB address for offset 5Ch
fcb_6c_ptr dd      0                   ; FCB address for offset 6Ch
parm_blk   ends
;*
;* Structure for information pertaining to Transaction Programs known
;*   to the Application Subsystem.
;*   The structure is 108 bytes in length.
known_tp   struc                       ; Structure for Known TP's
tpname_len db      0                   ; Length(TP_NAME)
tpname     db      '
'
luname     db      '        '          ; LU_NAME
luid       dq      0                   ; LU_ID
psid       dq      0                   ; PS_ID
resource   dd      0                   ; Resource
fileid     db      '              '    ; Fileid of program to be executed
           db      0                   ; Term. char for ASCIIZ string
known_tp   ends

;*
;* Structure for TP name entered by user from keyboard (DOS Function 0Ah)
;*
kbd_input  struc
maxlen     db      0                   ; Maximum input length
```

```
actlen     db      0                   ; Actual input length
tpn        db      65 dup (' ')        ; TP Name
           db      ' '                 ; Delimiter
kbd_input  ends
;*
;* Macro used to invoke APPC/PC
;* Note  AX & DX are modified.
;*
appc_pc    macro   request,ctrl_blk
           ifnb    <ctrl_blk>
           mov     dx,offset ctrl_blk  ; DS:DS points to APPC/PC control
                                       ; block
           endif
           mov     ah,request          ; APPC/PC Function Request
           int     appc_int            ; APPC/PC Interrupt Request
           endm

;*
;* Macro to display a message on the screen.
;* Notes  AX & DX are modified.
;*        The end-of-string (referenced by DS:DS) is terminated by '$'
;*
display    macro   msg
           mov     dx,offset msg       ; DS:DS points to message
           mov     ah,9                ; DOS function request number
           int     dos_int             ; Have DOS display the message
           endm
;*
kbinp      macro
           mov     dx,offset kbuf      ; DS:DS points to message
           mov     ah,0ah              ; DOS function request number
           int     dos_int             ; Have DOS display the message
           endm
;* Macro to display a single character on the screen.
;* Note  AX & DX are modified.
;*
putchar    macro   char
           mov     dl,char             ; DL = character to be displayed
           mov     ah,2                ; DOS function request number
           int     dos_int             ; Have DOS display the message
           endm
;*
;* Macro to set up registers for a CALL to procedure PRT_RC
;* Note  CX & SI are modified.
;*
printrc    macro   rc,type
           lea     si,rc               ; DS: SI -> Return Code
           mov     cl,type             ; CL indicates kind of return code
           call    prt_rc              ; Display the Return code
           endm
;*
;* Macro to copy a sequence of characters from source to target.
;* Note  CX, DI & SI are modified.
;*
copystr    macro   source,target,length
           cld                         ; Clear the direction flag
           lea     si,source           ; DS: SI = Address of source
           lea     di,target           ; ES: DI = Address of target
           mov     cx,length           ; CX = Length (bytes)
       rep movsb                       ; Copy CX bytes from source to
                                       ; target
           endm
copystr1   macro   source,target,length
           cld                         ; Clear the direction flag
           lea     si,source           ; DS:SI = Address of source
           push    bx
           mov     bx,es: tpn_offset
           lea     di,target           ; ES:DI = Address of target
           pop     bx
           mov     cx,length           ; CX = Length (bytes)
       rep movsb                       ; Copy CX bytes from source to
                                       ; target
           endm
copystr2   macro   source,target,length
```

```
           cld                       ; Clear the direction flag
           push    bx
           mov     bx,tpn_offset
           lea     si,source           ; DS:SI = Address of source
           pop     bx
           lea     di,target           ; ES:DI = Address of target
           mov     cx,length           ; CX = Length (bytes)
       rep movsb                       ; Copy CX bytes from source to
                                       ; target
           endm
;***********************************************************************
;* BEGIN OF DATA SEGMENT - DATASEG                                     *
;***********************************************************************
           .list
dataseg    segment para public 'data'
;************************************************************************
;* This data segment supports the ITRN adapter.                         *
;*                                                                      *
;* The Local LU E13L0203 has been defined as having a partner LU        *
;* located at the Ring Station address 400001130000.  This definition  *
;* is contained in trn_plu_aa.                                          *
;* The other definition that can be of interest is the indication of   *
;* Adapter Number (Primary or Secondary), set to Primary (0) as         *
;* contained in adapt_id.                                               *
;*                                                                      *
;* Allocate and initialize the control block data structures.          *
;***********************************************************************
;*
att_pu_cb  attachpu <,,,,,,,'NETWORK','E13PS01'>          ; ATTACH_PU
att_lu_cb  attachlu <,,,,,'E13L0203',,3,2,0,,,,1,1,,,ltot>  ; ATTACH_LU
;                    23456          78 9 0 1234 5 678
b1         equ      $
pnl_lu_cb  part_lu  <ltot,,'CICS11',,1,,'ITRN',adapt_id,6,,lmod>
;                        23          45 67      8        9 01
b2         equ      $
mode_cb    mode     <,'APPCNEGB',256,256,1,2>
;                    2        3   4   5 6
ltot       equ      $-b1
lmod       equ      $-b2
actdlc_cb  act_dlc  <,,,,'ITRN',adapt_id>            ; ACTIVATE_DLC
cnos_cb    cnos     <,,,,,,,,'CICS11','APPCNEGB',,1,1,0,1,0> ; CNOS
;                    2345678          9         01 2 3 4 5
conv_cb    convert  <>                               ; CONVERT
det_lu_cb  detachlu <,,,,,1>                         ; DETACH_LU (soft)
det_pu_cb  detachpu <>                               ; DETACH_PU
getall_cb  getalloc <>                               ; GET_ALLOCATE
tp_vald_cb tp_valid <>                               ; TP_VALID
tp_strt_cb tp_start <>                               ; TP_STARTED

;*
;* Equates used by the application subsystem
;*

RC_4       equ     0                   ; 4 Byte Return Code Indicator
RC_6       equ     1                   ; 6 Byte Return Code Indicator
adapt_id   equ     0                   ; Adapter ID number
CR         equ     13                  ; Carriage Return
LF         equ     10                  ; Line-Feed
dos_int    equ     21h                 ; DOS     Interrupt Request
appc_int   equ     68h                 ; APPC/PC Interrupt Request
q          equ     39                  ; For imbedded quotes
remote_tp  equ     00000001b           ; Remote TP mask
bad_tp     equ     00000010b           ; Bad TP mask

;*
;* Messages displayed by application subsystem.
;*

SNAmissing db      'APPC/PC is not loaded.$'
rc_equals  db      'Return Code = $'
xlate_msg  db      'Translating names from ASCII to EBCDIC.',CR,LF,'$'
att_pu_msg db      'Attaching the PU.',CR,LF,'$'
det_pu_msg db      'Detaching the PU.',CR,LF,'$'
att_lu_msg db      'Attaching the LU.',CR,LF,'$'
det_lu_msg db      'Detaching the LU.',CR,LF,'$'
actdlc_msg db      'Activating the DLC.',CR,LF,'$'
cnos_msg   db      'CNOS - Setting Session Limit to '
cnos_limit db      '1.',CR,LF,'$'
trmsg      db      'Enable  Traces ?   ','$'
setblk_err db      'DOS SETBLOCK failed.$'
await_msg  db      'Awaiting incoming ALLOCATE....',CR,LF
           db      'Ent.name of local TP (',q,'z',q,' to end) ',CR,LF,'$'
alloc_msg  db      'Incoming ALLOCATE received.',CR,LF,'$'
bad_alloc  db      'GET_ALLOCATE error.',CR,LF,'$'
acceptmsg  db      'Incoming ALLOCATE accepted.',CR,LF,'$'
rejectmsg  db      'Incoming ALLOCATE rejected.',CR,LF,'$'
exe_tp_msg db      'Load and execute the Transaction Program.',CR,LF,'$'
bad_tp_msg db      'Unrecognized TP name',CR,LF,'$'
fnd_tp_msg db      'TP name found in table',CR,LF,'$'

;*
;* Allocate memory for subsystem data structures, and values.
;*

tp_info    known_tp <7,'LOCALTP','E13L0203',,,,'DMPC.EXE',>
           known_tp <7,'DEBUG01','E13L0203',,,,'debug.com',>
tp_end     db      0                   ; Table delimiter
in_area    kbd_input <65,,,'$'>        ; Max length 65 allows 64 + CR
offset_env equ     es: ¢2Ch|            ; Offset of Env. Address in PSP
```

```
parmstr    db      0                   ; Null Parameter string-see stp 6
fcb        db      16 dup(0)           ; Null FCB entry for TP-see stp 6
tp_parmblk parm_blk <,parmstr,fcb,fcb> ; TP parameter block
prefix     dw      0                   ; Program Segm. Prefix (PSP) Addr.
signature  db      'APPC/PC'           ; APPC/PC signature
trn_plu_aa db      40h                 ; Partner LU TRN adapter address
           db      00h
           db      01h
           db      13h
           db      00h
           db      00h
tpn_offset dw      0                   ; Pointer to tp_info entry
as_ind     db      0                   ; Indicator bits (see equates)
kbuf       db      04h
           db      0
tflg       db      '    '              ;keyboard input
trinfo     dd      0
           db      20h
           db      20h
           dw      0000h
           dd      0
           dd      0
dataseg    ends
;************************************************************************
;* END OF DATA SEGMENT - DATASEG                                        *
;************************************************************************
;*
;************************************************************************
;* Code Segment - CODESEG                                               *
;************************************************************************
          .xlist
codeseg    segment para public 'code'
           assume  cs: codeseg,ss: stackseg,ds: dataseg
;*
;* Procedure to convert the binary value in AL into 2 hexadecimal
;*   characters, and display each of them on the screen.
;* Note  AX, CX & DX are modified.
;*

b2x        proc    near                ; Local to the Application Subsys.
           xor     ah,ah               ; Clear out AH
           push    ax                  ; Save the initial value of AL
           mov     cl,4                ; CL = shift count (bit pos.)
           shr     ax,cl               ; Shift low order bits out of AX
           call    b2x_digit           ; Convert the digit & display it
           pop     ax                  ; Restore initial value of AL
           and     al,0Fh              ; Mask off low order bits
b2x_digit: cmp     al,9                ; Is it a normal decimal digit?
           jle     b2x_dec             ;   Yes
           add     al,'A'-'9'-1        ;   No  - conv. 10..15 -> 'A'..'F'
b2x_dec:   add     al,'0'              ; Change binary value into char.
           putchar al                  ; Display the character
           ret                         ; Return
b2x        endp

;*
;* Procedure to display the return code on the screen.
;*
;* Input  DS:SI -> Return Code value to be displayed
;*        CL =  RC_4 implies 4 Byte Return Code
;*        CL <> RC_4 implies 6 Byte (primary/secondary) Return Code
;*
;*  Note  AX, CX, DX & SI are modified.
;*

prt_rc     proc    near
           push    cx                  ; Save Return Code Type indicator
           display rc_equals           ; Display Return Code message
           cld                         ; Insure Auto-increment on Loads
           lodsb                       ; AL = First byte of RC
           call    b2x                 ; Convert and Display it
           lodsb                       ; AL = 2nd   byte of RC
           call    b2x                 ; Convert and Display it
           pop     cx                  ; Restore Return Code Type ind.
           cmp     cl,RC_4             ; Is this a 4 byte Return Code?
           je      last_2_rc           ; -Yes - Displ. only 2 more bytes
           putchar ' '                 ; Display sep. (blank) character
           lodsb                       ; AL = 3rd   byte of RC
           call    b2x                 ; Convert and Display it
           lodsb                       ; AL = 4th   byte of RC
           call    b2x                 ; Convert and Display it
last_2_rc:                             ; Display the final 2 bytes
           lodsb                       ; AL = penultimate byte of RC
           call    b2x                 ; Convert and Display it
           lodsb                       ; AL = last  byte of RC
           call    b2x                 ; Convert and Display it
           putchar CR                  ; Display CR & LF
           putchar LF                  ;
           ret                         ;
prt_rc     endp

;***************************************************************************
;*
;* Procedure used to process the PASSTHROUGH requests made of the
;*   Application Subsystem.  The only verb handled by this sample
;*   is TP_INITIATE, which is a verb defined for this sample
;*   subsystem.
;*
;* Inputs  AH = 7 (required by APPC/PC)
;*         DS:DS points to the verb (TP_INITIATE) control block
;*         LU_ID of known TP (returned by APPC/PC on ATTACH_LU)
;*
;* Notes  1. The stack defined by the Transaction Program must be
;*           large enough to accommodate this exit, and its
;*           invocations of APPC/PC.
;*        2. Names are expected to be in EBCDIC (i.e., TP_NAME & LU_NAME)
;*
;* Warning  The passthrough routine must return to the invoking TP via
;*          the IRET (interrupt return) instruction, since APPC/PC jumps
;*          to (not calls) this code.
;*
;* Process  1. Verify Verb Opcode
;*          2. Verify Local/Remote Indicator
;*          3. Verify TP_NAME field
;*   If local init
;*      4. Put LU_ID of known TP into TP_STARTED Control Block
;*      5. Issue TP_STARTED
;*      6. Copy TP_ID from TP_STARTED CB into tp_info table
;*      7. Copy TP_ID from TP_STARTED CB into TP_INITIATE control block
;*          8. Retrieve TP_STARTED return code
;*   If remote init
;*      9. Copy TP_ID from tp_info table into TP_INITIATE control block
;*      10. Copy CONV_ID from tp_info table into TP_INITIATE cntrl.block
;*   In any case
;*      11. Store Return Code in TP_INITIATE Control Block
;*      12. Exit to invoking TP
;*
;* Return Codes 00000000 = OK
;*              00000003 = BAD_LU_ID                   (from TP_STARTED)
;*              00000243 = TOO_MANY_TPS                (from TP_STARTED)
;*              F0010000 = APPC_DISABLED               (from TP_STARTED)
;*              F0020000 = APPC_BUSY                   (from TP_STARTED)
;*              F0030000 = APPC_ABENDED                (from TP_STARTED)
;*              FFFFFFFD = INVALID_TP_NAME             (unique to subsys)
;*              FFFFFFFE = INVALID_LOCAL/REMOTE_INDICATOR (also unique)
;*              FFFFFFFF = INVALID_TP_VERB_VARIANT
;******************************************************************************
;*
passthru   proc    far                 ; Invoked by the TP (via APPC/PC)
           push    ax                  ; Save entry value of mod. regs.
           push    bx                  ;
           push    cx                  ;
           push    dx                  ;
           push    di                  ;
           push    si                  ;
           push    es                  ;
                                       ;
           mov     ax,dataseg          ; Point ES to Subsys Data Segment
           mov     es,ax                   ;
           mov     bx,dx                   ; have DS: BX point to Cntr Block
                                           ;
           xor     ax,ax                   ; Note  All 4 bytes are swapped
           not     ax                      ; DX: DX = 4 byte Return code
           mov     dx,ax                   ;      = INVALID_TP_VERB_VARIANT
                                           ;
           cmp     word ptr ¢bx|.tp_init_opcode,0    ; Ver. verb opcode
           je      a1                      ;
           jmp     set_rc                  ;
a1:        dec     ah                      ; Return Code = INVALID_
                                           ;  LOCAL/REMOTE_INDICATOR
;*
;* See if TP agrees with us on source of its init (local or remote)
;*
           xor     cl,cl                                ; Zero CL
           or      cl,es: as_ind                         ; CL=as_ind
           and     cl,remote_tp                         ; Mask off 0-6
           xor     cl,byte ptr ¢bx|.tp_init_indicator ; Agreement?
           jz      check_tpn                            ;   Yes
           jmp     set_rc                               ;   No

;*
;* See if TP Name supplied on TP_INITIATE matches that suppl. initially
;* (i.e. via keyboard or incoming ALLOCATE)
;*
check_tpn: dec     ah                      ; Return Code = INVALID_TP_NAME
           xor     ch,ch                             ; CL = length of
           mov     cl,¢bx|.tp_init_tpname_len    ;  TP name
           lea     si,offset ¢bx|.tp_init_tpname ; Addr of TP Name field
           mov     di,es: tpn_offset
           inc     di
;*
;* At this point
;*   ES:DI -> Table entry of TP Name known to subsystem   (EBCDIC)
;*   DS:SI -> TP name specified on TP_INITIATE            (EBCDIC)
;*   CX    =  Length(TP Name)  = 1..64     (Trail EBCDIC blanks allowed)
;*
           cld                             ;
      repe cmpsb                           ; Are TP Names equal?
           je      comp_loc                ;  Yes  - TP_INITIATE is valid
           jmp     set_rc                  ;  No   - Invalid TPN specified

;* Is it a local initiate?
comp_loc:  test    byte ptr ¢bx|.tp_init_indicator,remote_tp
           jz      loc_init                ; Yes
;*
;* It's a remote init  return TP_ID and CONV_ID to TP
;*
           push    ds                      ;
           push    es                      ; Exchange DS & ES segm. pointers
           pop     ds                      ; so DS points to subsys data segm.
           pop     es                      ; & ES points to the TP's data seg

          copystr2 ¢bx|¢81|,¢bx|.tp_init_tpid,8
          copystr2 ¢bx|¢89|,¢bx|.tp_init_conv_id,4

           push    es                      ; Restore DS, so it points back
           pop     ds                      ;  to the TP's data segment

           xor     ax,ax                   ; Set 4 byte return code = OK
           xor     dx,dx                   ;
           jmp     set_rc                  ;
;*
;* If local, we need to issue TP_STARTED, which needs LU_ID parameter,
;*
loc_init:  push    ds                      ;
           push    es                      ; Have both ES & DS point to subsys
           pop     ds                      ;   data segment
          copystr2 ¢bx|¢73|,tp_strt_cb.tp_start_luid,8
                                           ;
           appc_pc 3,tp_strt_cb            ; APPC/PC (TP_STARTED request)
                                           ;
           push    bx                      ; Save offset for TP_INITIATE CB
           mov     bx,dx                   ; Have DS: BX>TP_STARTED Ctrl.Blk

          copystrl ¢bx|.tp_start_tpid,¢bx|¢81|,8 ; Put TP_ID into table

           pop     di                      ; DI = offset for TP_INITIATE CB
                                           ;
           pop     es                      ;Setup ES for cpy from TP_STARTED
           push    es                      ;   control block to TP_INTITIATE

           copystr ¢bx|.tp_start_tpid,¢di|.tp_init_tpid,8 ; Copy TP_ID

           les     dx,¢bx|.tp_start_rc     ; Retrieve TP_STARTED return code
           mov     ax,es                   ; Make DX: AX the 4 byte ret.codede
           pop     ds                      ; Rest. DS (Inv. TP's Data Segm)
                                           ;
set_rc:    mov     word ptr ¢bx|.tp_init_rc+0,dx ; Store 4 byte Ret.code
           mov     word ptr ¢bx|.tp_init_rc+2,ax ;
           pop     es                      ; Restore entry value of mod.regs
           pop     si                      ;
           pop     di                      ;
           pop     dx                      ;
           pop     cx                      ;
           pop     bx                      ;
           pop     ax                      ;
           iret                            ; Return to invoking TP
passthru   endp
;--Stop and wait for operator to hit Enter....
;stop       proc    near
;          display  tstop
;          mov     ah,0ah
;          mov     dx,offset dumbuf
;          int     21h
;          ret
;stop       endp
;--Stop and wait for operator to hit Enter....
;*
;* Data area for saving stack pointer across Load & Exec (DOS) Requ.
;* Note  The reason for placing it here (rather than in the data seg),
;*        is so that it is addressable off of the CS register.
;*

saved_ss   dw      0                       ; Saved SS: SP (see step 8)
saved_sp   dw      0                       ;
;
;***********************************************************************
;* DOS entry point.                                                    *
;***********************************************************************
        .list
gen_as     proc    far

;*
;* Initialize the stack for a return to DOS.
;* On entry, DS & ES point to the Program Segment Prefix (PSP).
;*
           push    ds                      ; Use the Seg value of PSP
           xor     ax,ax                   ; Offset of "Return to DOS"
           push    ax                      ;   request (INT 20) in the PSP
;*
;* Point DS and ES to the data segment.
;* N.B. These pointers remain in effect until Step 7, except for a
;*       temporary DS change in Step 1.
;*
           mov     ax,dataseg              ; AX = Segment value of dataseg
           mov     ds,ax                   ;   Put it in DS
           mov     bx,offset_env           ;  (save environment address in
           mov     tp_parmblk.env_addr,bx ; parameter block, see step 8)
           push    es                      ;
           mov     es,ax                   ;   Put it in ES too
           pop     prefix                  ; Save PSP address (see step 5)
           assume  es: dataseg              ; Inform assembler about ES
;*
;* 1. Verify that APPC/PC is installed.
;*
           push    ds                      ; Save DS (i.e., dataseg)
           xor     ax,ax               ; Clear AX
           mov     ds,ax               ; DS points to low memory
           mov     si,4 * appc_int     ; SI > to APPC/PC interr. vector
           lds     si,¢si|             ; DS:SI > to APPC/PC entry point
           sub     si,9                ; > DS:SI to APPC/PC signature
           mov     di,offset signature ; DI points to expected value
           cld                         ; Clear the direction flag
           mov     cx,7                ; Length of signature
      rep  cmpsb                       ; Is the APPC/PC sign. present?
           pop     ds                  ; Restore DS
           je      cont1               ; Is APPC/PC loaded?
           display SNAmissing          ;   No - display message
           jmp     exit                ;      & exit to DOS
           .list                       ;
cont1:     nop
;*
;* 2. Translate ASCII names to EBCDIC (e.g., TP names)
;*
       display  xlate_msg
                          ; First translate all the 8-char fields
       mov  conv_cb.convert_length,8
       mov  word ptr conv_cb.convert_source+2,ds
   mov  word ptr conv_cb.convert_source,offset att_pu_cb.attachpu_netid
       mov  word ptr conv_cb.convert_target+2,ds
   mov  word ptr conv_cb.convert_target,offset att_pu_cb.attachpu_netid
       appc_pc 251,conv_cb          ; Translate PU Net ID to EBCDIC
       printrc conv_cb.convert_rc,RC_4
;
  mov  word ptr conv_cb.convert_source,offset att_pu_cb.attachpu_puname
  mov  word ptr conv_cb.convert_target,offset att_pu_cb.attachpu_puname
       appc_pc 251,conv_cb          ; Translate PU Name to EBCDIC
       printrc conv_cb.convert_rc,RC_4
;
    mov     word ptr conv_cb.convert_source,offset cnos_cb.cnos_pluname
    mov     word ptr conv_cb.convert_target,offset cnos_cb.cnos_pluname
    appc_pc 251,conv_cb          ; Translate Partner LU Name to EBCDIC
    printrc conv_cb.convert_rc,RC_4
                                ;
   mov     word ptr conv_cb.convert_source,offset cnos_cb.cnos_modename
   mov     word ptr conv_cb.convert_target,offset cnos_cb.cnos_modename
    appc_pc 251,conv_cb          ; Translate Mode Name to EBCDIC
    printrc conv_cb.convert_rc,RC_4
;  call    stop                  ;
;* Following CONVERTs loop through tp_info table

           lea     si,tp_info
conv_tbl:  cmp     byte ptr ¢si|,0     ; End of table?
           je      set_p_t             ; If so, go to Step 3
           mov     conv_cb.convert_length,8
           mov     word ptr conv_cb.convert_source,si
           add     word ptr conv_cb.convert_source,65
           mov     word ptr conv_cb.convert_source+2,ds
           mov     word ptr conv_cb.convert_target,si
           add     word ptr conv_cb.convert_target,65
           mov     word ptr conv_cb.convert_target+2,ds
           appc_pc 251,conv_cb         ; Translate LU Name to EBCDIC
;          call    stop
           push    si
           printrc conv_cb.convert_rc,RC_4
           pop     si
           mov     conv_cb.convert_length,64
           mov     word ptr conv_cb.convert_source,si
           inc     word ptr conv_cb.convert_source
           mov     word ptr conv_cb.convert_target,si
           inc     word ptr conv_cb.convert_target
           appc_pc 251,conv_cb         ; Translate TP Name to EBCDIC
           push    si
           printrc conv_cb.convert_rc,RC_4
           pop     si
           add     si,108              ; SI -> next entry in table
           jmp     conv_tbl            ; Do it again
;*
;* 3. Set up Passthrough mechanism.
```

```
;*
set_p_t:   nop
           push    ds
           push    cs                  ;
           pop     ds                  ;
           mov     dx,offset passthru  ; DS:DS -> Passthru routine
           appc_pc 255                 ; SET_PASSTHROUGH request
           pop     ds                  ; Restore DS
;=====================TRACE===============
           DISPLAY   TRMSG
           KBINP
           CMP   TFLG,'Y'
           JE    DOT
           CMP   TFLG,'y'
           JE    DOT
           jmp   trcon
dot:       nop
           mov   ah,252                ;trace msgs
           mov   al,1
           mov   dx,80
           int   68h
           mov   ah,253                ;trace api
           mov   al,1
           mov   dx,80
           int   68h
           mov   ah,254                ;output dest
           mov   al,4                  ;data set
           mov   dx,offset trinfo
           int   68h
trcon:     nop
;=====================TRACE===============

;*
;*  ATTACH PU
;*
           display   att_pu_msg
           appc_pc 1,att_pu_cb         ; 4a. ATTACH_PU
           printrc att_pu_cb.attachpu_rc,RC_4
;          call    stop                ;temp stop for testing
;*
;* ATTACH LU
;*
       copystr trn_plu_aa,pnl_lu_cb.part_lu_adapt_addr,6
;      mov     pnl_lu_cb.part_lu_adp_adrlen,06h    ;len-TR-adp-addr
       mov     word ptr att_lu_cb.attachlu_createtp,0000h   ;queue
       mov     word ptr att_lu_cb.attachlu_createtp+2,0000h ;Allocates
       mov     conv_cb.convert_length,8
  mov  word ptr conv_cb.convert_source,offset att_lu_cb.attachlu_luname
  mov  word ptr conv_cb.convert_target,offset att_lu_cb.attachlu_luname
       appc_pc 251,conv_cb         ; Translate LU Name to EBCDIC
       printrc conv_cb.convert_rc,RC_4
  mov  word ptr conv_cb.convert_source,offset pnl_lu_cb.part_lu_pluname
  mov  word ptr conv_cb.convert_target,offset pnl_lu_cb.part_lu_pluname
       appc_pc 251,conv_cb         ; Translate Partner LU Name to EBCDIC
       printrc conv_cb.convert_rc,RC_4
  mov     word ptr conv_cb.convert_source,offset mode_cb.mode_modename
  mov     word ptr conv_cb.convert_target,offset mode_cb.mode_modename
           appc_pc 251,conv_cb         ; Translate Mode Name to EBCDIC
           printrc conv_cb.convert_rc,RC_4
;****
           display   att_lu_msg
           appc_pc 1,att_lu_cb         ;     ATTACH_LU
;****
           printrc att_lu_cb.attachlu_rc,RC_4
           copystr  att_lu_cb.attachlu_luid,tp_info.luid,8
;          call    stop                ;temp stop for testing
           display actdlc_msg          ;
;****
           appc_pc 1,actdlc_cb         ;     ACTIVATE the TR
;****
           printrc actdlc_cb.act_dlc_rc,RC_4
;          call    stop                ;temp stop for testing
           display cnos_msg            ;
           mov     cnos_cb.cnos_modesesslim,01h    ;test
           copystr att_lu_cb.attachlu_luid,cnos_cb.cnos_luid,8
;****
           appc_pc 6,cnos_cb          ; 4d. CNOS - Session Limit = 1
;****
           printrc cnos_cb.cnos_pri_rc,RC_6
;          call    stop               ;temp stop for testing
;*
;* 5. Invoke DOS to release (i.e., shrink) memory thus making room for
;*    the transaction program (TP).
;*    The size of the application subsystem must be computed to perform
;*    this function.  This value is computed using the following:
;*
;*    size (in paragraphs) = Addr(Stack) + Length(Stack) - Addr(PSP)
;*
;*    The reason this works is that the stack area is placed last by
;*    the linker, since its segment name (stackseg) is higher than that
;*    of both the code segment (codeseg) and the data segment (dataseg).
;*
;*    Note: ES = Segment of block to be shrunk
;*          BX = Requested size (paragraphs)
;*
shrink:    push    es                 ; Save current ES value
           mov     es,prefix          ; ES = PSP   Segment address
           mov     bx,ss              ; BX = Stack Segment address
           add     bx,20h             ;  Add stack size (paragraphs)
           mov     ax,es              ;  Subtrack starting segment
           sub     bx,ax              ;  from ending to compute length
           mov     ah,4Ah             ; Request - SETBLOCK
           int     dos_int            ; DOS Function Request
           pop     es                 ; Restore initial ES value
           jc      shrink_err         ; SETBLOCK request unsuccessful
           jmp     init_wt1           ; SETBLOCK request   successful
shrink_err:display setblk_err         ;   No - display message and
           jmp     takedown           ;        take down the session
;*
;* 6. Loop to wait for an init request, local (via keyboard) or remote.
;*
init_wt1:  copystr att_lu_cb.attachlu_luid,getall_cb.getalloc_luid,8
init_wt2:  lea     bx,getall_cb       ; DS:BX >GET_ALLOCATE contr.block
;* CHANGE_LU here
           and     as_ind,not bad_tp  ; Reset 'Bad TP' indicator
           and     as_ind,not remote_tp; Reset 'Remote TP' indicator
           display await_msg          ; Inf. user of what we are doing
rd_kbd:    mov     ah,0Bh             ; Check keyboard input status
           int     dos_int            ; DOS function call
           cmp     al,0FFh            ; Character available?
           je      a5                 ;   Yes: go and see what
           jmp     get_alloc          ;   No:  check incoming ALLOCATE
a5:        mov     ah,0Ah             ; Buffered keyboard input
           lea     dx,in_area         ; DS:DS -> input area
           int     dos_int            ; DOS function call
           cmp     in_area.actlen,0   ; Anything actually entered?
           jne     a4                 ; If so, convert it
           jmp     get_alloc          ; If not, go to GET_ALLOCATE
a4:        mov     conv_cb.convert_length,64
           mov     word ptr conv_cb.convert_source,offset in_area.tpn
           mov     word ptr conv_cb.convert_target,offset in_area.tpn
           appc_pc 251,conv_cb        ; Translate entered TPN to EBCDIC
           printrc conv_cb.convert_rc,RC_4
;          call    stop
           cmp     in_area.actlen,1   ; Single char entered?
           jne     findtp_1           ; If not, find TP name in table
           cmp     in_area.tpn,0A9h   ; Did he enter 'z' (converted)?
           jne     findtp_1           ;
           jmp     takedown           ; If so, end the subsystem
;*
;* Search for TP name in table
;*
findtp_1:
           cld                        ; Compare forwards
           lea     si,tp_info         ; DS:SI -> tp_info table
comp_1:    lea     di,in_area.tpn     ; ES:DI -> entered TP name
           xor     ch,ch              ; CX = Length of entered TP name
           mov     cl,in_area.actlen  ;
           cmp     byte ptr ¢si|,0     ; End of table?
           je      not_fnd_l           ;   Yes
           cmp     cl,¢si|             ; Do the lengths agree?
           jne     next_l              ;   No  - move to next entry
           push    si                  ;   Yes - save SI for later
           inc     si                  ; DS:SI -> tpname field
      repe cmpsb                       ; Have we found it?
           je      found_l             ;   Yes
           pop     si                  ;   No  - first restore SI
next_l:    add     si,108              ; DS:SI -> next entry in table
           jmp     comp_l              ; Do it again
not_fnd_l: display bad_tp_msg          ; Tell the user his TPN no good
           or      as_ind,bad_tp       ; Set the 'Bad TP' indicator
           jmp     clr_input           ;
found_l:   pop     tpn_offset          ; Rest. offset of matching entry
           display fnd_tp_msg          ; Tell user TP name found

;* Copy LUID into table entry
           mov     ax,tpn_offset
          copystrl att_lu_cb.attachlu_luid,¢bx|¢73|,8

;* Clear the input area

clr_input: xor     ch,ch
           mov     cl,in_area.maxlen   ; Length of input area
           mov     si,0                ; Initial offset
clr_char:  mov     in_area.tpn¢si|,20h ; Blank a byte
           inc     si                  ; Point to next char
           loop    clr_char            ; Do it again

           test    as_ind,bad_tp       ; Do we have a bad TP?
           jnz     get_allocl          ;  Yes - try a GET_ALLOCATE
           jmp     execute             ;  It's good so load it

get_allocl:
           and     as_ind,not bad_tp   ; Reset 'Bad TP' indicator
get_alloc: appc_pc 3,getall_cb         ; Try a GET_ALLOCATE request
           push    es                  ; Save ES for later
           les     dx,¢bx|.getalloc_rc ; Retrieve the Return Code
           mov     ax,es               ; DX:AX = 4 byte Return Code
           pop     es                  ; Restore saved ES
           or      dx,dx               ; Are first 2 bytes zero?
           jnz     alloc_err           ; No  - something is really wrong.
           cmp     ax,8202h            ; Yes - is RC = UNSUCCESSFUL?
           jne     a3
           jmp     rd_kbd              ;    Yes - Try keyboard
a3:        or      ax,ax               ;    No  - is RC = OK?
           jz      good_alloc          ;        Yes - ALLOCATE received
alloc_err:                             ;        No  - takedown
           display bad_alloc           ; Dsply "GET_ALLOCATE error" msg
           printrc getall_cb.getalloc_rc,RC_4 ;
           jmp     takedown            ; Unexpected GET_ALLOCATE error
good_alloc:                            ;
           display alloc_msg           ; Dsply "ALLOCATE received" msg
           printrc getall_cb.getalloc_rc,RC_4
                                       ;
;*
;* 7. Verify incoming ALLOCATE request against known TP list.
;*
;*    N.B. DS & ES chop and change in this step to cater for references
;*         to CREATE_TP record within APPC/PC.
;*
           lds     bx,¢bx|.getalloc_createtp ; DS:BX >CREATE_TP ctrl.blk
                                       ; Note: ES -> Subsys data segment
                                       ;
           push    ds                  ; Exchange DS & ES, so
           push    es                  ; DS       > Subsys data segment
           pop     ds                  ; & ES:BX > CREATE_TP ctrl record
           pop     es                  ;
           mov     ax,0810h            ; AX:DX = SENSE_CODE
           mov     dx,2160h            ; AX:DX (10086021) TPN not Recogn.
;* Loop to find TP name in table
           cld                         ; Compare forward
           lea     si,tp_info          ;        DS:SI -> tp_info table
comp_r:    lea     di,¢bx|.createtp_tpname ;    ES:DI -> rcvd TP name
           xor     ch,ch               ; CX = Length of recvt TP name
           mov     cl,es:¢bx|.createtp_tpnamelen
           cmp     byte ptr ¢si|,0     ; End of table?
           je      not_fnd_r           ;   Yes
           cmp     cl,¢si|             ;    comparing name lengths
           jne     next_r              ; Not the same, so move on
           push    si                  ; Otherwise save SI
           inc     si                  ; DS:SI -> tpname field
     repe  cmpsb                       ;   the names
           je      found_r             ; We have a match
           pop     si                  ; No match - first restore SI
next_r:    add     si,108              ; DS:SI -> next entry in table
           jmp     comp_r              ; Do it again
found_r:   pop     tpn_offset          ; Rest. offset of matching entry
           display fnd_tp_msg          ; Tell user TP name found
;          call    stop
;**** test  next instr
           jmp     setx                ;
           mov     dx,3460h            ; AX:DX (10086034) CONVERSATION
                                       ;  Type Mismatch
           cmp     es:¢bx|.createtp_convtype,0
           jne     set_sense           ;
           mov     dx,4160h            ; AX:DX (10086041) Sync Level
                                       ;  not Supported
           cmp     es:¢bx|.createtp_synclevel,0
           jne     set_sense           ;
           mov     ax,0F08h            ;
           mov     dx,5160h            ; AX:DX (080F6051) Sec. Not Valid
           cmp     es:¢bx|.createtp_pwlen,0
           jne     set_sense           ;
           cmp     es:¢bx|.createtp_useridlen,0
           jne     set_sense           ;
setx:      xor     ax,ax               ;
           xor     dx,dx               ; AX:DX (00000000) OK
           jmp     set_sense           ;
not_fnd_r: or      as_ind,bad_tp       ; Set the 'Bad TP' indicator
           push    dx
           push    ax
           display bad_tp_msg          ; Tell user bad TPN received
           pop     ax
           pop     dx
set_sense: mov     word ptr es:¢bx|.createtp_sensecode+0,ax ; Store the
           mov     word ptr es:¢bx|.createtp_sensecode+2,dx ; SENSE_CODE
           push    ds                  ;
           push    es                  ;
           pop     ds                  ; DS -> CREATE_TP
           pop     es                  ; ES -> Subsystem data segment
           test    es:as_ind,bad_tp    ; Do we have a good TP?
           jz      upd_table           ;   Yes - update table
           jmp     tpvalid             ;   No  - issue TP_VALID
;* Put APPC/PC info into our table entry:
upd_table:
          copystr1 ¢bx|.createtp_tpid,¢bx|¢81|,8
          copystr1 ¢bx|.createtp_conv_id,¢bx|¢89|,4
           push    ds                  ;
           push    es                  ;
           pop     ds                  ; DS, ES -> Subsystem dataseg
          copystr1 att_lu_cb.attachlu_luid,¢bx|¢73|,8
           pop     ds                  ; DS > CREATE_TP, ES > dataseg
tpvalid:   copystr ¢bx|.createtp_tpid,tp_vald_cb.tp_valid_tpid,8

           push    es                  ;
           pop     ds                  ; DS, ES -> Subsystem dataseg

 copystr getall_cb.getalloc_createtp,tp_vald_cb.tp_valid_createtp,4
           push    ax
           push    dx
           appc_pc 4,tp_vald_cb        ; Issue TP_VALID verb with
                                       ;   appropriate block
           pop     dx                  ; Restore the Sense Code
           pop     ax                  ;
                                       ;
           or      ax,ax               ; Non-zero SENSE_CODE?
           jnz     rejected            ;   Yes - ALLOCATE rejected
           or      dx,dx               ; Maybe - check next 2 bytes
           jnz     rejected            ;   Yes - ALLOCATE rejected
           display acceptmsg           ;   No  - ALLOCATE accepted
;          call    stop
;* CHANGE_LU here with message
           or      as_ind,remote_tp    ; Set 'Remote TP' indicator
           jmp     short execute       ; Execute the TP
                                       ;
rejected:  display rejectmsg           ; Tell local user of rejection
           jmp     init_wt2            ; Resume waiting for an init
                                       ;
;*
;* 8. Invoke DOS to Load and Execute the TP
;*
;* Warning: The only registers which are "preserved" across the
;*          LOAD_AND_EXECUTE request are CS & IP, so care must be
;*          taken to save any registers important to your program
;*          on the stack, then save SS & SP in the code segment or
;*          some other addressable area.
;*
execute:
           display exe_tp_msg          ; Inf. user of what we are doing
;          call    stop
           push    ds                  ; Preserve DS
           push    es                  ;  & ES across execution request
           mov     cs:saved_ss,ss      ; Save the stack pointer (SS:SP)
           mov     cs:saved_sp,sp      ;   in the code segment
           mov     dx,tpn_offset
           add     dx,93               ;
           mov     bx,offset tp_parmblk; ES:BX point to DOS Param. Block
           mov     ax,4B00h            ; Load & Exec DOS Funct. Request
           int     dos_int             ; Execute the Trans. Program
           cli                         ; Dis. interr. while chg SS:SP
           mov     sp,cs:saved_sp      ; Rest. the stack pointer (SS:SP)
           mov     ss,cs:saved_ss      ;   from within the code segment
           sti                         ; Re-enable interrupts
           pop     es                  ;
           pop     ds                  ; Restore DS & ES
;*
;*  The TP has ended (and returned). Resume waiting for an init request.
;*
           jmp     init_wt2
;*
;* 9.  Deactivate resources as prelude to subsystem termination.
;*
takedown:                              ;
           mov     cnos_limit,'0'      ; Correct CNOS message
           display cnos_msg            ;   and display it
           copystr att_lu_cb.attachlu_luid,cnos_cb.cnos_luid,8
           mov     cnos_cb.cnos_modesesslim,0
           mov     cnos_cb.cnos_winnersesslim,0
           mov     cnos_cb.cnos_autoactivate,0
           appc_pc 6,cnos_cb           ; CNOS - Session Limit = 0
           printrc cnos_cb.cnos_pri_rc,RC_6

           display det_lu_msg          ;
           copystr att_lu_cb.attachlu_luid,det_lu_cb.detachlu_luid,8
           appc_pc 1,det_lu_cb         ; DETACH_LU
           printrc det_lu_cb.detachlu_rc,RC_4

           display det_pu_msg          ;
           appc_pc 1,det_pu_cb         ; DETACH_PU
           printrc det_pu_cb.detachpu_rc,RC_4
;*
;*10. Reset the Pass-through mechanism
;*
           push    ds                  ; Save DS
           xor     dx,dx               ;
           not     dx                  ; DS:DS = FFFF:FFFF
           mov     ds,dx               ;
           appc_pc 255                 ; SET_PASSTHROUGH (reset) request
           pop     ds                  ; Restore DS
```

```
;*
;*11. Exit to DOS
;*
exit:      ret                         ; See notes at progr. entry point
gen_as     endp
codeseg    ends
           end     gen_as
;************************************************************************
;* END of APPLICATION SUBSYSTEM                                         *
;************************************************************************
;************************************************************************
```

13.2 APPC/PC - PC TRANSACTION PROGRAM EXAMPLE

```
;***********************************************************************
;*                                                                     *
;*          TEST TRANSACTION PROGRAM                                   *
;*                                                                     *
;* Purpose: To provide a sample transaction program which invokes the  *
;*          remote CICS transaction ,AIBR, over an SDLC link.          *
;*  Notes: This program uses a verb implemented in the appl. subsys.   *
;*          (via the PASSTHROUGH mechanism) called TP_INITIATE.        *
;*         It also uses the EZ-VU product for handling panels.         *
;*                                                                     *
;*  APPC/PC & Application Subsystem verbs issued:                      *
;*          1. Translate ASCII names to EBCDIC                         *
;*          2. TP_INITIATE(local)  (Application Subsystem unique verb) *
;*          3. ALLOCATE            (start the remote TP)               *
;*          4. SEND_DATA           (send some data to the other TP)    *
;*          5. RECEIVE             (Receive CICS response)             *
;*          6. DEALLOCATE          (terminate the conversation)        *
;*          7. TP_ENDED            (bring down TP)                     *
;*          7. Exit to DOS         (and Application Subsystem)         *
;***********************************************************************
;***********************************************************************
;* Data Structure Definitions                                          *
;***********************************************************************

;*
;* Include the necessary APPC/PC verb structure definitions.
;*
include    allocate.str                 ; ALLOCATE     Data Structure
include    convert.str                  ; CONVERT      Data Structure
include    dealloca.str                 ; DEALLOCATE   Data Structure
include    senddata.str                 ; SEND_DATA    Data Structure
include    rcv_wait.str                 ; RECEIVE & WAIT Data Structure
include    tp_ended.str                 ; TP_ENDED     Data Structure
include    tp_init.str                  ; TP_INITIATE  Data Structure

;*
;* Macro used to invoke APPC/PC
;* Note: AX & DX are modified.
;*
appc_pc    macro   request,ctrl_blk
           ifnb    <ctrl_blk>
           mov     dx,offset ctrl_blk   ; DS:DS >to APPC/PC control block
           endif
           mov     ah,request           ; APPC/PC Function Request
           int     appc_int             ; APPC/PC Interrupt Request
           endm

;*
;* Macro to display a message on the screen.
;* Notes: AX & DX are modified.
;*        The end-of-string (referenced by DS:DS) is terminated by '$'
;*
display    macro   msg
           mov     dx,offset msg        ; DS:DS points to message
           mov     ah,9                 ; DOS function request number
           int     dos_int              ; Have DOS display the message
           endm

;*
;* Macro to display a single character on the screen.
;* Note: AX & DX are modified.
;*
putchar    macro   char
           mov     dl,char              ; DL = character to be displayed
           mov     ah,2                 ; DOS function request number
           int     dos_int              ; Have DOS display the message
           endm
;*
;* Macro to call EZ-VU dialog manager
;* Note: AX  is  modified.
```

```
;*
DMPC      MACRO TYPE,PARMS
          IRP  X,<PARMS>
          MOV  AX,OFFSET X
          PUSH AX
          ENDM
          CALL TYPE
         ENDM

;*
;* Macro to set up registers for a CALL to procedure PRT_RC
;* Note: CX & SI are modified.
;*
printrc     macro    rc,type
            lea      si,rc                 ; DS:SI -> Return Code
            mov      cl,type               ; CL ind. kind of return code
            call     prt_rc                ; Display the Return code
            endm

;*
;* Macro to copy a sequence of characters from source to target.
;* Note: CX, DI & SI are modified.
;*
copystr     macro    source,target,length
            cld                            ; Clear the direction flag
            lea      si,source             ; DS:SI = Address of source
            lea      di,target             ; ES:DI = Address of target
            mov      cx,length             ; CX = Length (bytes)
        rep movsb                          ; Copy CX from source to target
            endm
            page
;*********************************************************************
;* Data Segment - DATA                                               *
;*********************************************************************
data        segment para public 'data'

;*
;* llocate and initialize the control block data structures.
;*
;===>     MC_ALLOCATE block for CICS - AIBR transaction
alloc_buf allocate <,,1,,,,,,1,,,0,,'CICS11','APPCNEGB',4,'AIBR',,,,,,,,
;                    23 456789 012 34        5             6 7      89012345
conv_buf  convert  <>                                   ; CONVERT
conv_asc  convert  <,,,,1,,,,>                          ; CONVERT
deall_buf dealloca <,,1,,,,,,,,01,,>                    ; MC_DEALL. (flush)
send_buf  senddata <,,1,,,,,,,,data_len,data_msg>       ; MC_SEND_DATA
rcv_buf   rcv_wait <,,1,,,,,,,,1,,326,,data_buf>        ; MC_RCV_WAIT
tpend_buf tp_ended <>                                   ; TP_ENDED
;===>    TP_INIT control block for TSK1_TP.EXE
tpinit_buf tp_init  <,,,,,local_tp,'E13L0203',,,7,'LOCALTP'>

;*
;* Equates used by the transaction program
;*
izero       equ      0
E           EQU      'E'
F           EQU      'F'
BK          EQU      'B'
;------ error message numbers --------
initer      equ      1                     ; TP_INIT error message
vdefer      equ      2                     ; VDEFINE error
;------ error message numbers --------
RC_4        equ      0                     ; 4 Byte Return Code Indicator
RC_6        equ      1                     ; 6 Byte Return Code Indicator
local_tp    equ      0                     ; Local/Remote Indicator values
remote_tp   equ      1                     ;
CR          equ      13                    ; Carriage Return
LF          equ      10                    ; Line-Feed
dos_int     equ      21h                   ; DOS     Interrupt Request
appc_int    equ      68h                   ; APPC/PC Interrupt Request

;*
;* Messages displayed by transaction program
;*
```

```
rc_equals  db      'Return Code = $'
xlate_msg  db      'Translating names from ASCII to EBCDIC.',CR,LF,'$'
inittp_msg db      'Initiating the Transaction Program.',CR,LF,'$'
initerr    db      'TP_INIT  error                    ',CR,LF,'$'
alloc_msg  db      'ALLOCATING the conversation.',CR,LF,'$'
ezvuer     db      'EZ-VU ERROR   RETURN CODE =        ',CR,LF,'$'
vgetmsg    db      'VGET  ERROR   RETURN CODE =        ',CR,LF,'$'
send_msg   db      'Sending the following data:'
alloc_cmp  db      'ALLOCATE COMPLETE          ',CR,LF,'$'
alloc_bad  db      'ALLOCATE FAILED            ',CR,LF,'$'
sendbad    db      'SEND     FAILED            ',CR,LF,'$'
sendok     db      'SEND   COMPLETE            ',CR,LF,'$'
pf3msg     db      '*                         * ',CR,LF,'$'
pf4msg     db      '***   P F 4    ****        ',CR,LF,'$'
rcvbad     db      'RECEIVE  FAILED            ',CR,LF,'$'
badata     db      'BAD DATA RECEIVED          ',CR,LF,'$'
crlf_msg   db       CR,LF,LF,'$'
deallocmsg db      'DEALLOCATING the conversation.',CR,LF,'$'
tp_end_msg db      'Ending the Transaction Program.',CR,LF,'$'

;*
;* Allocate memory for TP data structures, and values.
;*
           db      CR,LF,LF,'$'        ; Term. char. for DOS display r
equest
;
;  parameters for the dialog manager
;
LEN      DW   13
ONE      DW   1
PARM     DB   'DISPLAY tsk11'
PARM2    DB   'DISPLAY TSK12'
RC       DW    0000H
LENC     DW   13
PARMC    DB   'CONTROL CLEAR'
VDP1     DB   'BRKEY C'                ;FOR DEFINING THE USER
BRKEY    DB   '       '                ;ENTERED BROWSE KEY DATA
B        DB   '       '                ;FOR
C        DB   '       '                ;   PAGING
D        DB   '       '                ;         LOGIC
VDP2     DB   'SCRL C'
SCRL     DB   ' '
VDL2     DW   7
VDL21    DW   4
SYSID    DB   '     '
VDL1     DW   7
VDL11    DW   6
BLNK6    DB   '       '                ;
QFLG     DB   0
TFLG     DB   0
ZLEN     DW   11
VDPL1    DB   'LINE1 C'
LINE1    DB   39 DUP('  ')            ;78 CHARACTERS
VD1SIZ   DW   78
VDPL2    DB   'LINE2 C'
LINE2    DB   39 DUP('  ')            ;78 CHARACTERS
VDPL3    DB   'LINE3 C'
LINE3    DB   39 DUP('  ')            ;78 CHARACTERS
VDPL4    DB   'LINE4 C'
LINE4    DB   39 DUP('  ')            ;78 CHARACTERS
TSTBUF   DB   8,0,'         '
TRINFO   DD   0
         DW   100
         DW   0000h
         DD   0
LCNT     DW   0
RESET    DW   0
;
;INPUT TO CICS
;
data_msg    equ     $
TRAN     DB 0C1H,0C9H,0C2H,0D9H,40H   ;AIBR (EBCIDIC)
KEY      DB '000000'                  ;USER ENTERED BROWSE KEY
```

```
data_len   equ     $-data_msg        ;
;
;OUTPUT FROM CICS
;
data_buf db  200 dup('  ')          ;receive buffer
;
; PUT THE STACK AREA WITHIN THE DATA SEGMENT
;
SAK      DB    64 DUP("STACK...")
ESAK     DB    0FFH
data       ends
;*********************************************************************
;* Code Segment - CSEG                                               *
;*********************************************************************

cseg    segment para public 'code'
           assume  cs:cseg,ss:data,ds:data,es:data

;*
;* Procedure to convert the binary value in AL into 2 hexidecimal
;*   characters, and display each of them on the screen.
;* Note: AX, CX & DX are modified.
;*

b2x        proc    near                ; Local to the Appl. Subsystem
           xor     ah,ah               ; Clear out AH
           push    ax                  ; Save the initial value of AL
           mov     cl,4                ; CL = shift count (bit pos.)
           shr     ax,cl               ; Shift low order bits out of AX
           call    b2x_digit           ; Convert the digit & display it
           pop     ax                  ; Restore initial value of AL
           and     al,0Fh              ; Mask off low order bits
b2x_digit: cmp     al,9                ; Is it a normal decimal digit?
           jle     b2x_dec             ;  Yes
           add     al,'A'-'9'-1        ;  No - convert 10..15 -> 'A'..'F'
b2x_dec:   add     al,'0'              ; Change binary value into char.
           putchar al                  ; Display the character
           ret                         ; Return
b2x        endp

;*
;* Procedure to display the return code on the screen.
;*
;* Input: DS:SI -> Return Code value to be displayed
;*        CL =  RC_4 implies 4 Byte Return Code
;*        CL <> RC_4 implies 6 Byte (primary/secondary) Return Code
;*
;*  Note: AX, CX, DX & SI are modified.
;*

prt_rc     proc    near
           push    cx                  ; Save Return Code Type ind.
           display rc_equals           ; Display Return Code message
           cld                         ; Insure Auto-increment on Loads
           lodsb                       ; AL = First byte of RC
           call    b2x                 ; Convert and Display it
           lodsb                       ; AL = 2nd   byte of RC
           call    b2x                 ; Convert and Display it
           pop     cx                  ; Restore Return Code Type ind.
           cmp     cl,RC_4             ; Is this a 4 byte Return Code?
           je      last_2_rc           ;   Yes - Dsply only 2 more bytes
           putchar ' '                 ; Display separator (blank) char.
           lodsb                       ; AL = 3rd   byte of RC
           call    b2x                 ; Convert and Display it
           lodsb                       ; AL = 4th   byte of RC
           call    b2x                 ; Convert and Display it
last_2_rc:                             ; Display the final 2 bytes
           lodsb                       ; AL = penultimate byte of RC
           call    b2x                 ; Convert and Display it
           lodsb                       ; AL = last  byte of RC
           call    b2x                 ; Convert and Display it
           putchar CR                  ; Display CR & LF
           putchar LF                  ;
           ret                         ;
prt_rc     endp
dbugd      proc   near
           MOV    AH,0AH                      ;*
           MOV    DX,OFFSET TSTBUF            ; TEMP
           INT    21H                         ;     *
           ret
dbugd      endp
;
;**********************************************************************
;* DOS entry point.                                                   *
;**********************************************************************
MAIN    PROC   FAR                     ;ENTRY POINT FROM DOS
        EXTRN  ISPASM:FAR              ;NAME OF INTERFACE MODULE
        EXTRN  ISPASMV:FAR
BEG1:   MOV    AX,DATA
        MOV    SS,AX                   ;SS==>DATA SEG
        MOV    ES,AX                   ; SET AUX  SEGMENT REG
        MOV    BX,OFFSET ESAK          ; STACK IS WITHIN DATA SEG
        DEC    BX
        MOV    SP,BX                   ;SP==>END OF STACK AREA
        PUSH   DS                      ;SAVE PSP ID
        MOV    DS,AX                   ; SET DATA SEGMENT REG
        XOR    BX,BX                   ; THE DOUBLE WORD VECTOR SO THE
        PUSH   BX                      ; FAR RETURN WILL GO BACK TO DOS
;
; START  USER CODE HERE
;

        DMPC   ISPASM,<LENC,PARMC,RC>         ;CLEAR SCREEN
;
; DEFINE EZ-VU  VARIABLES
;
        DMPC  ISPASMV,<VDL1,VDP1,RC,BRKEY,VDL11>
        CMP   RC,IZERO                        ;RETURN CODE OF
        JE    ELSE03                          ; 0 OR 8
        CMP   RC,8                            ; IS OK.
        JE    ELSE03
        display  ezvuer
        CALL  dbugd
        JMP   ENDIF06
ELSE03: NOP
        DMPC  ISPASMV,<VDL11,VDP2,RC,SCRL,ONE>
        CMP   RC,IZERO                        ;RETURN CODE OF
        JE    ELSE06                          ; 0 OR 8
        CMP   RC,8                            ; IS OK.
        JE    ELSE06
        DISPLAY EZVUER
        CALL  DBUGD
        JMP   ENDIF06
ELSE06: NOP
        DMPC  ISPASMV,<VDL1,VDPL1,RC,LINE1,VD1SIZ>     ; VDEFINES
        DMPC  ISPASMV,<VDL1,VDPL2,RC,LINE2,VD1SIZ>     ;    FOR
        DMPC  ISPASMV,<VDL1,VDPL3,RC,LINE3,VD1SIZ>     ; SCREEN
        DMPC  ISPASMV,<VDL1,VDPL4,RC,LINE4,VD1SIZ>     ; VARIABLES
;*
;* 1. Translate ASCII names to EBCDIC (e.g., TP_NAMES)
;*
translate: nop
  mov conv_buf.convert_length,8 ; Translate all the 8 character names
  mov word ptr conv_buf.convert_source+2,ds
  mov word ptr conv_buf.convert_source,offset alloc_buf.allocate_pluname
  mov word ptr conv_buf.convert_target+2,ds
  mov word ptr conv_buf.convert_target,offset alloc_buf.allocate_pluname
  appc_pc 251,conv_buf         ; Translate Partner LU name to EBCDIC
                               ;
  mov word ptr conv_buf.convert_source,offset alloc_buf.allocate_modenam
  mov word ptr conv_buf.convert_target,offset alloc_buf.allocate_modenam
  appc_pc 251,conv_buf         ; Translate Mode Name to EBCDIC
                                        ;
  mov word ptr conv_buf.convert_source,offset tpinit_buf.tp_init_luname
  mov word ptr conv_buf.convert_target,offset tpinit_buf.tp_init_luname
  appc_pc 251,conv_buf         ; Translate LU Name to EBCDIC
                               ;
  mov conv_buf.convert_length,64 ; Now translate the 64 character names
  mov word ptr conv_buf.convert_source,offset alloc_buf.allocate_tp name
  mov word ptr conv_buf.convert_target,offset alloc_buf.allocate_tp name
  appc_pc 251,conv_buf        ; Translate TP Name to EBCDIC
                                       ;
  mov word ptr conv_buf.convert_source,offset tpinit_buf.tp_init_tp name
  mov word ptr conv_buf.convert_target,offset tpinit_buf.tp_init_tp name
  appc_pc 251,conv_buf        ; Translate TP Name to EBCDIC
                                       ;
;*
;* 2. TP_INITIATE(local)      (Application Subsystem unique verb)
;*
  appc_pc 7,tpinit_buf        ; TP_INITIATE(local) (Appl.Subs. verb)
           cmp    word ptr tpinit_buf.tp_init_rc+2,0000h     ;ok ?
           je     gcon
           display  initerr
           printrc tpinit_buf.tp_init_rc,RC_4
           jmp     endif07
gcon:      nop
           CLD                                  ;CLEAR  BRKEY'
           MOV    SI,OFFSET BLNK6
           MOV    DI,OFFSET BRKEY
           MOV    CX,3
REP        MOVSW
;
; DISPLAY A CICS-LIKE PANEL AND GET USER ENTRY
;
        DMPC   ISPASM,<LEN,PARM,RC>           ;DISPLAY THE PANEL
        CMP   RC,20                           ;20 = SEVERE ERROR
        JNE   ELSE09                          ; 0 OR 8
        display  ezvuer
        CALL  dbugd
        JMP   ENDIF07
ELSE09: NOP
        CMP    WORD PTR BRKEY,'  '             ;IF NULL ENTRY
        JNE    ELSE04
        JMP    ENDIF07                        ;QUIT
ELSE04: NOP
        DMPC   ISPASM,<LENC,PARMC,RC>         ;CLEAR SCREEN
;  Convert the key value from ASCII to EBCIDIC
;
   mov     conv_buf.convert_length,6           ; Translate 6 char key
   mov     word ptr conv_buf.convert_source+2,ds
   mov     word ptr conv_buf.convert_source,offset brkey
   mov     word ptr conv_buf.convert_target+2,ds
   mov     word ptr conv_buf.convert_target,offset key
   appc_pc 251,conv_buf                       ;TRANSLATE TO EBCIDIC
;
; INITIAL PG_FD RIPPLE
           copystr  key,brkey,6              ;make brkey ebcidic
           CLD
           MOV    SI,OFFSET BRKEY
           MOV    DI,OFFSET B
           MOV    CX,9
REP        MOVSW
        MOV   QFLG,00h                       ;RESET QUIT_FLG
;=============  TRACES =================
;       cmp   tflg,00h
;       je    conl
;       mov   ah,253                         ;** enable trace_api
;       mov   al,1
;       int   68h
;       mov   ah,252                         ;** enable trace_msg
;       mov   al,1
;       mov   dx,80
;       int   68h
;       mov   ah,254                         ;** PUT DATA ON
;       mov   al,4                           ;   OUTPUT.PC
;       mov   dx,offset trinfo
;       int   68h
conl:   nop
;=============  TRACES  =================
; DO UNTIL (QFLG IS SET)
;
```

```
DONTL1: NOP
;*
;* 3. ALLOCATE                 (allocate a session and a conversation
;*                              with  AIBR')
           copystr tpinit_buf.tp_init_tpid,alloc_buf.allocate_tpid,8
           display alloc_msg           ; Inf. user of what we are doing
;          CALL   DBUGD
allo_loop: appc_pc 2,alloc_buf         ; ALLOCATE the remote TP
           cmp     alloc_buf.allocate_pri_rc,0000h ; ok ?
           je      alloc_ok                         ;yes
           cmp     alloc_buf.allocate_pri_rc,1400h ; Prim. RC = 0014?
           jne     alloc_el            ; Error
           printrc alloc_buf.allocate_pri_rc,RC_6
           jmp     allo_loop          ; Keep trying
;          push    es                  ; Save ES
;          les     dx,alloc_buf.allocate_sec_rc
;          mov     ax,es               ; DX:AX = Sec. Return Code
;          pop     es                  ; Restore ES
;          or      dx,dx               ; Is secondary RC = 00000005 ?
;          jne     alloc_el            ;   No
;          cmp     ax,0500h            ;   Maybe
;          je      allo_loop           ;   Yes - Retry the ALLOCATE
alloc_el:  display alloc_bad           ;allocate bad
           printrc alloc_buf.allocate_pri_rc,RC_6
           call    dbugd
           mov     qflg,01h            ; stop the dountil
           jmp     enddol
alloc_ok:  nop
           display alloc_cmp           ;allocate complete
           printrc alloc_buf.allocate_pri_rc,RC_6
           call    dbugd
;
; SEND A BUFFER TO CICS
;
           copystr tpinit_buf.tp_init_tpid,send_buf.senddata_tpid,8
           copystr alloc_buf.allocate_conv_id,send_buf.senddata_conv_id,4
           appc_pc 2,send_buf          ; Send the message
           cmp     send_buf.senddata_pri_rc,0000h  ; ok ?
           je      else0a                           ;yes
           display sendbad
           printrc send_buf.senddata_pri_rc,RC_6
           call    dbugd               ;
           mov     qflg,01h            ; stop the dountil
           jmp     enddol
else0a:    nop
;          display  sendok
;          call  dbugd
;WAIT FOR CICS TO RESPOND
           copystr reset,rcv_buf.rcv_wait_datalen,2           ;clear
           copystr tpinit_buf.tp_init_tpid,rcv_buf.rcv_wait_tpid,8
           copystr alloc_buf.allocate_conv_id,rcv_buf.rcv_wait_conv_id,4
           appc_pc 2,rcv_buf          ; Wait for data from CICS
           mov     ax,rcv_buf.rcv_wait_pri_rc ; AX = Primary Return Code
           cmp     ax,0                ; zero return code?
           je      else0b              ; Yes
           display rcvbad              ; No
           printrc rcv_buf.rcv_wait_pri_rc,RC_6
           call    dbugd
           mov     qflg,01h            ;  And display the received message
           jmp     enddol
else0b:    nop
;*
;* 5. DEALLOCATE               (terminate the conversation)
;*
        copystr tpinit_buf.tp_init_tpid,deall_buf.dealloca_tpid,8
        copystr alloc_buf.allocate_conv_id,deall_buf.dealloca_conv_id,4
;       display deallocmsg          ; Inform user of what we are doing
        appc_pc 2,deall_buf         ; Deallocate the conversation
;       printrc deall_buf.dealloca_pri_rc,RC_6
;
; display the data received from CICS
           call    ddom
enddol:    nop                         ;check dountil condition
           cmp  qflg,00h
           jne  pass                   ;stop
           copystr  brkey,key,6        ;set next start browse key
           jmp  dont11
pass:     nop
ENDIF07:  NOP
;*
;* 6. TP_ENDED                (bring down TP)
;*
           copystr tpinit_buf.tp_init_tpid,tpend_buf.tp_ended_tpid,8
;          display tp_end_msg          ; Inf. user of what we are doing
           appc_pc 4,tpend_buf         ; TP_ENDED verb
;          printrc tpend_buf.tp_ended_rc,RC_4
;*
;* 7. Exit to DOS             (and Application Subsystem)
;*
endif06:   nop                         ;
exit:      ret                         ; See notes at prog. entry point
main       endp
;*
;* Procedure to display data on the EZ-VU panel
;*

DDOM       PROC    NEAR
           mov     ax,rcv_buf.rcv_wait_datalen ;get length of data
           cmp     ax,86                       ;min size is 86
           ja      ddok
           display badata
           call    dbugd
           mov     qflg,01h
           jmp     ddeX
ddok:      nop
           copystr data_buf,brkey,6            ;save last key
           mov     bx,offset data_buf          ;start of data
           add     bx,6                        ;step past last-key
; init for display loop
           mov     ax,offset line4
           add     ax,1
           push    ax
           mov     ax,offset line3
           add     ax,1
           push    ax
           mov     ax,offset line2
           add     ax,1
           push    ax
           mov     ax,offset line1
           add     ax,1
           push    ax
           mov     lcnt,4                  ;initialize loop count
dolp1:     nop
           pop     dx                       ;get ptr to next line
           mov     ax,0¢bx|                 ;get 1st char
           cmp     ah,0D5H                  ;is this the -end file-
           je      pas3
           add     bx,1                     ;1st byte is blank
pas3:      nop
           mov     conv_asc.convert_length,6 ; convert key field
           mov     word ptr conv_asc.convert_source+2,ds
           mov     word ptr conv_asc.convert_source,bx
           mov     word ptr conv_asc.convert_target+2,ds
           mov     word ptr conv_asc.convert_target,dx
           push    dx
           push    bx
           appc_pc 251,conv_asc            ;convert KEY field
;
           pop     bx
           pop     dx
           add     dx,11                 ;5+Key field size
           add     bx,6                  ;step to Name field
           mov     conv_asc.convert_length,20; convert name field
           mov     word ptr conv_asc.convert_source+2,ds
           mov     word ptr conv_asc.convert_source,bx
           mov     word ptr conv_asc.convert_target+2,ds
           mov     word ptr conv_asc.convert_target,dx
           push    dx
           push    bx
           appc_pc 251,conv_asc         ;convert NAME field
;
           pop     bx
           pop     dx
           add     dx,20                ;step past Name field
           add     bx,20
           mov     conv_asc.convert_length,20; convert ADDRESS field
           mov     word ptr conv_asc.convert_source+2,ds
           mov     word ptr conv_asc.convert_source,bx
           mov     word ptr conv_asc.convert_target+2,ds
           mov     word ptr conv_asc.convert_target,dx
           push    dx
           push    bx
           appc_pc 251,conv_asc       ;conver ADDRESS field
;
           pop     bx
           pop     dx
           add     dx,20                ;step past Address field
           add     bx,20
           mov     conv_asc.convert_length,10; convert SERIAL field
           mov     word ptr conv_asc.convert_source+2,ds
           mov     word ptr conv_asc.convert_source,bx
           mov     word ptr conv_asc.convert_target+2,ds
           mov     word ptr conv_asc.convert_target,dx
           push    dx
           push    bx
           appc_pc 251,conv_asc         ;SERIAL field
;
           pop     bx
           pop     dx
           add     dx,14
           add     bx,16                         ;step by SERIAL
           mov     conv_asc.convert_length,8     ;AMOUNT field
           mov     word ptr conv_asc.convert_source+2,ds
           mov     word ptr conv_asc.convert_source,bx
           mov     word ptr conv_asc.convert_target+2,ds
           mov     word ptr conv_asc.convert_target,dx
           push    dx
           push    bx
           appc_pc 251,conv_asc          ;AMOUNT field
           pop     bx
           pop     dx
           add     bx,17                 ;step to next data rcd
           sub     lcnt,1
           cmp     lcnt,0
           je      pass2
           jmp     dolpl
pass2:     nop
;
; DISPLAY THE DATA PANEL
;
        DMPC   ISPASM,<LEN,PARM2,RC>           ;DISPLAY THE PANEL
        CMP   RC,20                          ;20 = SEVERE ERROR
        JNE   ELSEDO                          ; 0 OR 8
        display  ezvuer
        CALL  dbugd
        JMP   DDEX
ELSEDO: NOP
        DMPC   ISPASM,<LENC,PARMC,RC>         ;CLEAR SCREEN
        CMP   SCRL,E
        JE    DDEQ
        CMP   SCRL,F
        JNE   CHKB
        COPYSTR C,D,6                         ; PAGE
        COPYSTR B,C,6                         ;   FORWARD
        COPYSTR BRKEY,B                       ;
        JMP   DDEX
CHKB:   NOP
        CMP   SCRL,BK                        ;IS IT PG_BK
        JNE   DDEQ
        COPYSTR C,BRKEY,6                    ;PAGE
        COPYSTR C,B,6                        ;BACKWARD
        COPYSTR D,C,6
        JMP    DDEX
DDEQ:   NOP
        MOV    QFLG,1
DDEX:   NOP
           RET
ddom       endp
cseg       ends
           END     BEG1
*******************************************************************
*******************************************************************
```

14.0 APPENDIX F: PUBLICATION REFERENCE

14.1 RELATED ITSC PUBLICATIONS

Token-Ring Network Bridges and Management
GG24-3062
Introduction to Programing for APPC/PC
GG24-3034
Guide to IBM PC 3270 Emulation Program V2
GG24-3038

14.2 ARCHITECTURE PUBLICATIONS

SNA Architecture Reference Summary
GA27-3136/3236?
Token-Ring Network Architecture Reference (6165877) SH19-6558

14.3 TOKEN-RING NETWORK PUBLICATIONS

IBM Cabling System Planning and Installation Guide
GA27-3361
Using the IBM Cabling System with Comm. Products
GA27-3620
IBM Token-Ring Network Introd. and Planning Guide
GA27-3677
IBM Token-Ring Network Problem Determination Guide
SY27-0280-1
IBM Token-Ring Network Telephone Twisted-Pair Media Guide
GA27-3714
IBM Token-Ring Network Installation Guide
GA27-3678
A Building Planning Guide for Communication Wiring
G320-8059
IBM Token-Ring Network Optical Fiber Cable Options
GA27-3747
IBM Token-Ring Network PC Adapter Guide to Operations
(comes
with adapter and diskette)
IBM Token-Ring Network PC Adapter Hardware Maintenance and Service
(comes with diskette)
IBM Token-Ring Network NETBIOS User's Guide
(with diskette)
IBM Token-Ring Network Manager User's Guide
(with diskette)
IBM Token-Ring Network Bridge Program U.G.
(with diskette)

IBM 3725 CONTROL PROGRAM
TOKEN-RING INTERFACE
PLANNING AND IMPLEMENTATION
GG24-3110-01

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, are deemed appropriate.

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:

____

____

____

____

What is your occupation? ______________________________

Number of latest Newsletter associated with this publication: ______________________________

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

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

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 3725 CONTROL PROGRAM
TOKEN-RING INTERFACE
PLANNING AND IMPLEMENTATION
GG24-3110-01

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, are deemed appropriate.

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:

____

____

____

____

What is your occupation? ______________________________

Number of latest Newsletter associated with this publication: ____________________

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

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

Reader's Comment Form

Fold and tape

Please Do Not Staple

Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM International Technical Support Center
Department 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®

Cut or Fold Along Line

GG24-3110-01

IBM 3725 NETWORK CONTROL PROGRAM
TOKEN-RING INTERFACE
PLANNING AND IMPLEMENTATION

GG24-3110-01

PRINTED IN THE U.S.A.

GG24-3110-01