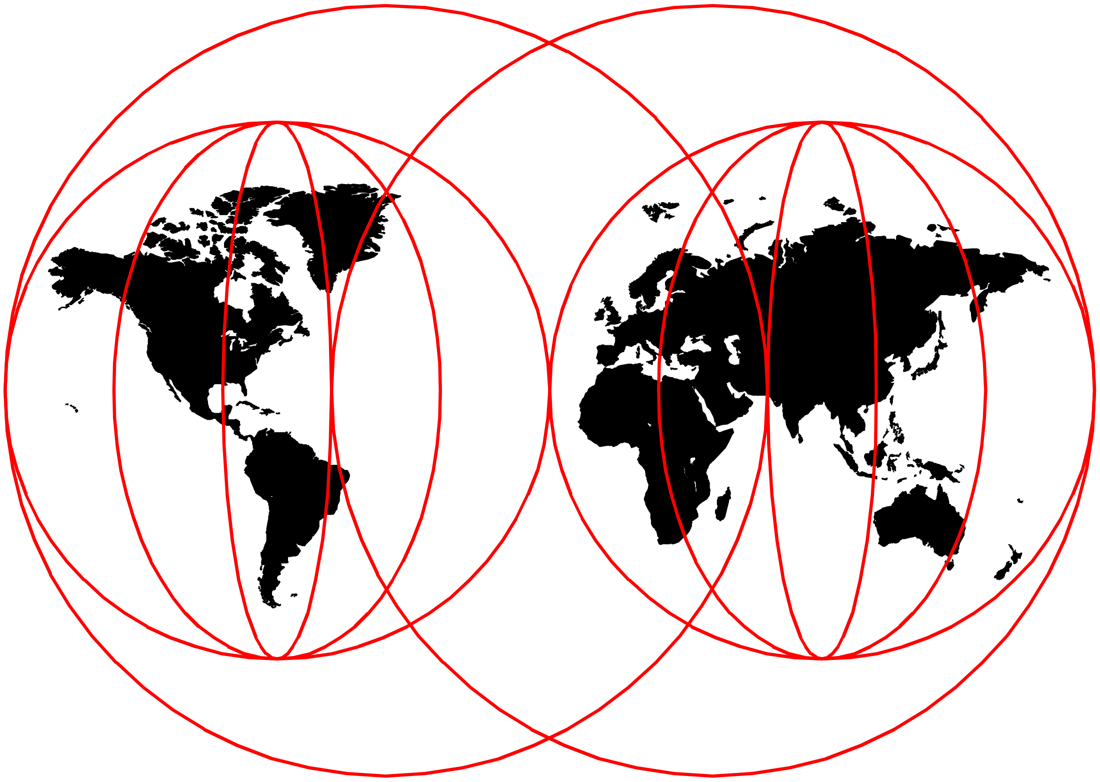
SNA and TCP/IP Integration

Jerzy Buczak, Karl Wozabal, Antonio Luca Castrichella, Heikki Lehikoinen, Maria Cristina Madureira, Tsutomu Masaoka



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-5291-00

International Technical Support Organization

SG24-5291-00

SNA and TCP/IP Integration

April 1999

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, “Special Notices” on page 235.

First Edition (April 1999)

This edition applies to the following products:

- OS/390 eNetwork Communications Server, Version 2 Release 7
- eNetwork Communications Server for OS/2, Version 5
- eNetwork Communications Server for Windows NT, Version 6
- 2216 Multiprotocol Access Services, Version 3 Release 2, together with equivalent releases of software on other members of the 221X router family

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 678
P.O. Box 12195
Research Triangle Park, NC 27709-2195

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

Contents

Figures

Preface

This redbook addresses the major networking issue facing many large enterprises today: how to integrate their present SNA network with tomorrow's TCP/IP and Web-based communication requirements.

There are many techniques available for carrying multiple protocols in the same backbone network, but they fall into two major categories:

- Multiplexing or switching, in which all protocols are carried independently over a common transport facility. Such a facility could be asynchronous transfer mode (ATM), frame relay, X.25 or a bridged LAN. Multiprotocol routing also falls into this category.
- Routing, which uses the network/routing layer of one protocol to carry all the others. Generally (but not always), the backbone routing protocol chosen is IP.

In this book we concentrate on the routing techniques. In the first chapter we offer an overview of each of them, and contrast their benefits and drawbacks. In later chapters we illustrate each routing technique with practical examples conducted at our laboratory in the Raleigh ITSO Center.

Many of the SNA/IP integration techniques have already been documented in other redbooks, but never have they been brought together in one volume. This book will help you to decide which method (or methods!) of SNA/IP integration is most relevant to your needs, and to render assistance in implementing that method.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

Jerzy Buczak is an IT Consultant at the International Technical Support Organization, Raleigh Center. He writes extensively and teaches IBM classes worldwide on VTAM and APPN. Before joining the ITSO in 1996, Jerzy worked for Networking Systems in the UK. He has 17 years' experience in SNA networking, network management, and a wide variety of product implementations. Jerzy holds an M.A. degree in mathematics from Cambridge University, England.

Karl Wozabal is a Senior Networking Specialist at the International Technical Support Organization, Raleigh Center. He writes extensively and teaches IBM classes worldwide in all areas of TCP/IP. Before joining the ITSO, Karl worked in IBM Austria as a Networking Support Specialist.

Antonio Luca Castrichella is an IT specialist in IBM Italy. He has six years' experience in professional networking services with a variety of large customers. Luca is skilled in the installation and customization of TCP/IP in the MVS, UNIX Systems Services, AIX and Windows NT environments.

Heikki Lehikoinen is a networking specialist in IBM Finland. He has 26 years' experience in a wide variety of networking environments. His areas of particular expertise include SNA/APPN architecture and products, network design, tuning, analysis and problem determination.

Maria Cristina Madureira is a Networking Support Specialist with IBM Brazil. Her particular area of expertise lies in implementing and maintaining TCP/IP on both IBM and non-IBM routers. She also has experience in implementing both SNA and TCP/IP on the CS for OS/390 platform.

Tsutomu Masaoka is an IT specialist in IBM Japan, and the leader of the networking team on a large banking account. His extensive knowledge includes designing large SNA networks, implementing a wide variety of networking products, and organizing network management procedures. Masaoka-san has 16 years' experience in both SNA and TCP/IP with large customers.

Thanks to the following people for their invaluable contributions to this project. Some have reviewed the material and made corrections and additions. Others have kindly allowed their redbook material to be used as a basis for parts of this book:

Roy Brabson
CS for OS/390 Development, Research Triangle Park

Nancy Gates
Advanced Technical Support, Washington Systems Center

John Parker
IBM United Kingdom

Wolfgang Singer
IBM Austria

Jonathan Follows
Bob Haimowitz
Tim Kearby
Julian Over
Juan Rodriguez
Carla Sadtler
International Technical Support Organization, Raleigh Center

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 249 to the fax number shown on the form.
- Use the online evaluation form found at `http://www.redbooks.ibm.com`.
- Send your comments in an Internet note to `redbook@us.ibm.com`.

Chapter 1. Integrating SNA and TCP/IP

There was a time long ago when a typical enterprise installation used only one networking protocol to access the applications running on its computers. Exactly which protocol was used depended almost entirely on which manufacturer's *hardware* and *operating system* were installed. Most IBM installations used SNA, most DEC installations used DECNet, most UNIX installations used TCP/IP and so on.

In recent years the direction has changed; the networking protocols have, more and more, been determined by the *applications* that the installation has chosen to install. This has led inevitably to a proliferation of networking protocols, followed by a drive to rationalize them because of the costs involved in running many protocols within a single network. This rationalization process will probably never end, but great progress has been made and most large enterprises now have, at most, two or three networking protocols in serious use. For installations using IBM mainframes or mid-range systems these will include SNA and TCP/IP, with perhaps NetBIOS or IPX running on the site LANs. Since LAN protocols (NetBIOS in particular) do not take kindly to being run over the wide area, customers will often use either SNA or TCP/IP to carry these (where needed) between sites. Thus the major issue facing large installations today is how to carry both SNA and TCP/IP traffic across the enterprise network *most efficiently*.

It is important to realize that both these protocols will remain in widespread use for a very long time. To convert completely to one or the other would involve a total rewrite or replacement of all the applications that use the less desirable one. There is no protocol conversion technique available today (and none visible in the future) that will provide a complete translation between all the features used by SNA applications and all the features used by TCP/IP applications. At the application programming interface, therefore, both SNA and TCP/IP *must* be provided.

Fortunately this is not true at the lower layers of the networking (OSI) reference model. SNA and TCP/IP are almost completely independent of the physical and data link layers of the architecture, and their differences at the next (network) layer are manageable. Only in the upper layers are SNA and TCP/IP sufficiently different to make it virtually impossible to translate fully between them. Therefore, it is possible to combine SNA and TCP/IP at a *transport* level without affecting the upper layers. There are three practical choices:

1. To integrate SNA and TCP/IP at the physical layer. We will refer to this technique as *multiplexing*. This is the technique used originally by channelized modems, then by time division multiplexers, then by the integrated-services digital network (ISDN). Each protocol sees its own physical circuit and uses its own data link control (DLC) over it, while the network combines these DLCs on to a single real physical circuit. The nodes on the network then route each protocol independently using the protocol's native techniques.

   Multiplexing is regarded with little favor these days because it requires a separate physical connection for each protocol at each routing node. A much more efficient way is to run both protocols over the same physical connection, which brings us to the second and third options.

2. To integrate SNA and TCP/IP at the link layer. The nodes on the network combine both protocols on the same physical connection, utilizing the fact that both support much the same range of link-level protocols. Again, the two protocols must be routed independently using the native technique of each.

   The DLC and the transport medium used must be able to distinguish between the higher-layer protocols being carried, so that they can be passed to the appropriate routing function. Such media include:

   - Local area networks. A LAN station can support multiple logical connections in addition to connectionless transport. Each may carry a different protocol. LANs may be bridged together to provide a single transport medium across the wide area network.
   - Frame relay. A frame relay device can support multiple virtual circuits (logical connections), each of which can carry a different protocol. In addition, protocols may be multiplexed over the same virtual circuit using RFC 2427 (an update of RFC 1490 and RFC 1294).
   - Asynchronous transfer mode (ATM), which is similar to frame relay in concept but designed for higher speeds. Each ATM virtual circuit can carry one protocol, or a number of protocols multiplexed together as defined in RFC 1483.
   - X.25. Each X.25 device can maintain multiple virtual circuits and therefore multiple independent protocols.
   - Point-to-Point Protocol (PPP). Routers are able to carry multiple protocols independently over PPP connections between them.

   Use of these shared transport media confers an additional advantage, in that you can now build an end-to-end switched network that does not require intermediate routing function for any of the protocols carried. However, it still has the disadvantage that you need a complete stack of each protocol implemented on each endpoint node in the network.

3. To integrate SNA and TCP/IP at the network/routing layer. This is more complex because the SNA and IP routing techniques are rather different. In essence it means building a backbone routing network for one protocol, then somehow encapsulating the essential information from the other protocol's packets and carrying it over the backbone. Thus we may see SNA carried over an IP network, or IP carried over an SNA network.

   With this method you need install only the upper layers of each stack on the application nodes in the network, and the lower layers of the chosen single stack on all nodes. On the other hand the management and operation of the network tend to be more complex and less efficient.

Note

This book concerns itself solely with network-layer integration (the third option). For an up-to-date view of what you can do with link-layer integration, please refer to *MSS Release 2.1 Including the MSS Client and Domain Client,* SG24-5231.

Both DLC-layer and network-layer integration have their advantages and disadvantages:

- With DLC-layer integration, the endpoints and routing nodes of the connections require multiple complete protocol stacks. While this is not usually a problem on large servers or routers, and less of a problem in an end-to-end switched environment, it can make a significant difference to the configuration and the management of user workstations and branch routers. Network-layer integration normally requires only one complete stack at each node, plus multiple *partial* stacks at the endpoints.
- Network-layer integration techniques tend to be more complex, and it is more difficult to ensure that all the functions of the *carried* protocol are available when the underlying *carrying* protocol is different in philosophy.
- The optimum level of service could be provided by either technique, depending on the mix of protocols needed and the service levels required for each one.

One solution might be to use network-layer integration at the periphery of the network but link-layer integration in the backbone. Thus user workstations would contain a single protocol stack, but their gateways into the backbone routing network would have both. Typically this would mean an IP stack on the workstations (because it is easier and cheaper to provide legacy 3270 application access from an IP workstation than to provide Web access from an SNA workstation). The gateway routers would then provide TN3270 conversion for SNA applications but leave the IP traffic alone. This solution, however, is not so easy if the SNA applications are more recent types such as advanced program-to-program communication (APPC); dual stacks may still be needed throughout.

It is important to understand that the choice between link-layer and network-layer integration is one of finding the most cost-effective way of running a multiprotocol network, and not one of interoperability. An application written to (for example) an SNA programming interface will always require a partner SNA application, regardless of what kind of network is used to transport the data; it will never talk to a (TCP/IP) Sockets application without some form of protocol conversion. The integration techniques we describe here provide protocol coexistence, not conversion.

In an ideal world the decision between link-layer and network-layer integration would be taken after extensive design studies, based on future projections for application requirements and technology trends. In practice, however, it often depends on the existing infrastructure at the time of the decision, as well as (often enough) prevailing fashion. Our book, therefore, does not attempt to give advice on this particular choice beyond some general observations.

1.1 Summary of Conclusions

As stated above, it is beyond the scope of this book to give detailed advice on the choice between link-layer integration (running SNA and TCP/IP in parallel over a shared transport medium) and network-layer integration (running both protocols on top of the routing functions of one). In principle:

- With link-layer integration the more complex functions tend to be at the edges of the network, whereas with network-layer integration they tend to be distributed throughout the network. This is especially true in the case of a switched multiprotocol network, where there is no routing at all in the backbone, merely switching.

- If your need is for a high-speed network with a high level of service, you should consider link-layer integration, and in particular a switched backbone. The combination of simple switching in the backbone, plus native routing where it is needed, gives (all other things being equal) better performance than routing throughout, especially considering possible inefficiencies due to non-native routing for one protocol.

  Switched backbone technologies for this case start with ATM as the clear winner, because it was designed to carry very high traffic rates and give the correct service to such diverse traffic types as video, voice and data. Second choice must be frame relay or X.25; X.25 is old technology, but frame relay does not have mature switched virtual circuit standards. If your backbone costs are based on connection time you need switched virtual circuits. Bridged LANs are not recommended across a wide area network because of the difficulty of limiting broadcast traffic (but ATM LAN emulation may be a solution).

- In a link-layer integration solution the costs of installing (or upgrading) the network are biased towards the end stations and the edges of the network, whereas with network-layer integration they are more biased towards the backbone routers.

- If one protocol predominates in your network, then you should consider network-layer integration. You have the choice of whether to route or to switch on the backbone, and the dominant protocol can be carried with maximum efficiency while minimizing the costs of carrying the other(s).

- On the other hand, if the two protocols are both of major importance, you may not wish to introduce inefficiencies in one of them by making it use a non-native routing protocol. However, there is an inherent bias towards TCP/IP in this because the technologies for carrying IP over SNA are less numerous and less advanced than those for carrying SNA over IP.

If you decide, or have decided, to implement network-layer integration then we feel we can offer some more specific advice:

1. The backbone should be IP unless the overwhelming majority of your traffic is, and will remain, SNA. Methods of running IP over SNA are not as mature as methods of running SNA over IP, so a network whose traffic is fairly evenly balanced between SNA and IP is likely to benefit from the choice of IP.

2. If you have decided to use IP over SNA, then AnyNet is the only practicable choice. Sockets APIs are supported on all current IBM platforms.

3. If you have decided to use SNA over IP, the SNA boundary should be placed as far out in the network as possible. With Enterprise Extender this is perfectly possible even if the backbone is totally IP. While SNA over IP is currently unable to provide the same level of service as native SNA (because of IP’s unpredictable connectionless transport), the use of end-to-end SNA protocols should maximize the level of availability and performance to their users.

   This is where Telnet/3270 comes into its own, if your SNA traffic is almost all 3270-type traffic. With TN3270 you can minimize the costs of upgrading your workstations (since an IP stack tends to be cheaper to implement than an SNA stack), while at the same time positioning your TN3270 Server (and therefore your SNA boundary) close to the workstation.

4. With SNA over IP, there are three main choices of technique: data link switching, AnyNet and Enterprise Extender. We strongly recommend that you choose Enterprise Extender because:

   - It provides end-to-end SNA priority and the high availability of high-performance routing (HPR), even if there is not a single native SNA link in your entire network.
   - It is not susceptible to the failure of an SNA-to-IP boundary node in the way that data link switching and AnyNet are susceptible.
   - It provides the full dynamics of Advanced Peer-to-Peer Networking (APPN), unlike AnyNet.
   - Its use of HPR routing minimizes the overhead on intermediate routers that provide Enterprise Extender-to-native SNA boundary functions.
   - It is the only technique that gives you end-to-end nondisruptive session recovery, again through its use of HPR.

1.2 Options for Integration

There are several different ways of running mixed protocol communication over single protocol transport networks. IBM provides the following solutions to intermix SNA and TCP/IP:

- Data link switching (DLSw). SNA traffic is encapsulated in TCP packets.
- AnyNet Sockets over SNA. TCP/IP traffic is carried across the SNA network using LU 6.2 sessions.
- AnyNet SNA over IP. SNA LU 6.2 and dependent LU traffic is carried over TCP connections.
- Telnet/3270 (TN3270). 3270 data streams are carried over TCP connections to a server which replaces the TCP transport with SNA transport.
- Host On-Demand (HOD). This is a TN3270 client implemented as a Java applet downloaded from a Web server.
- Enterprise Extender. SNA (HPR) packets are carried as User Datagram Protocol (UDP) packets over an IP network.

Most of these provide the ability to run SNA traffic over an IP network. Only one, Sockets over SNA, provides the ability to run TCP/IP Sockets applications over an SNA network.

Two of the methods, TN3270 and HOD, provide IBM 3270 terminal access to SNA applications from workstations having only a TCP/IP or Internet connection.

Each of these multiprotocol solutions is briefly described below. More detailed descriptions with practical examples are in the subsequent chapters.

1.3 Data Link Switching

Data link switching (DLSw) is an IBM-architected extension to source route bridging that allows SNA and NetBIOS frames to be routed through an intervening IP network. It is implemented as an extra feature in IP router products, such as the IBM 2210, 2212, 2216 and 3746 MAE.

There are two types of DLSw: local and remote. Here we concentrate on remote DLSw for SNA, as it uses the IP transport network while local DLSw does not. Local data link switching enables communication between a LAN-attached SNA device and an SDLC station that is link-attached to the same DLSw router. The SDLC station is assigned a MAC address so that it appears to other network devices to be on a LAN, and SDLC frames are converted to IEEE 802.2 logical link control (LLC) type 2 frames.

Figure 1 illustrates the principles of data link switching.

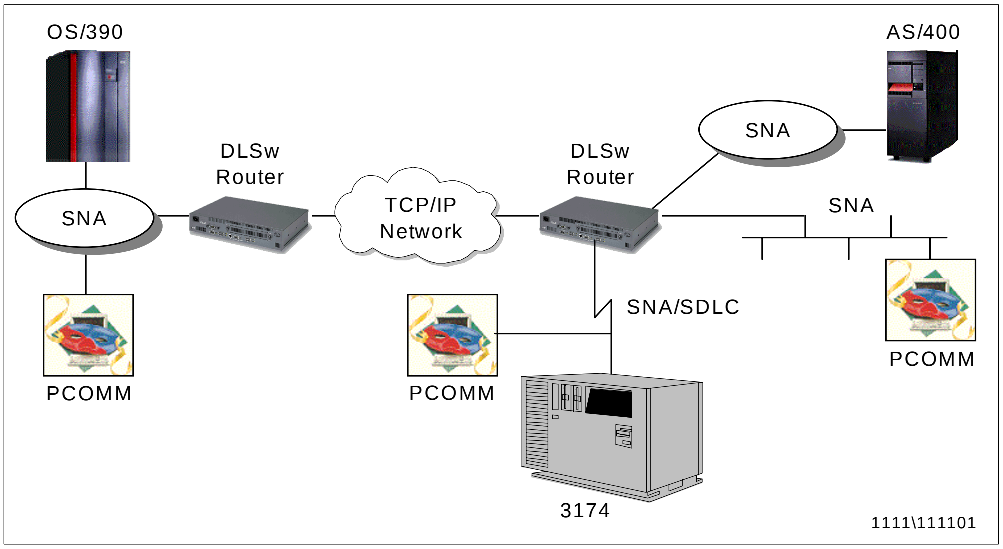


Figure 1. SNA over IP Using Data Link Switching

To SNA users, a DLSw connection looks like a bridged LAN connection. In fact, DLSw is an extended form of the LAN bridge function that interconnects local area networks across wide area networks (WANs). DLSw utilizes a TCP connection between the DLSw routers to carry the SNA traffic.

Remote DLSw implementation requires at least two routers (with the DLSw feature), connected to each other via an IP network. Routers within the IP network need not know anything about DLSw. The DLSw routers are only needed at the edge of the IP network.

SNA stations, in turn, may be LAN- or link-attached to the DLSw routers.

Remote data link switching supports:

- SDLC-to-LAN communication over WAN. DLSw performs SDLC-to-IEEE 802.2 conversion. This allows a link-attached SDLC station to communicate with a LAN-attached SNA station on another DLSw router.
- LAN-to-LAN communication over WAN. SNA stations can be on token-ring or Ethernet LANs. DLSw can convert token-ring MAC sublayer frames to Ethernet MAC sublayer frames, and vice versa. LAN-to-LAN can be done between SNA devices connected to the same router, but this is bridging and not DLSw.

LAN-attached SNA devices use LLC type 2, which is the connection-oriented version of Logical Link Control and requires timely responses between stations. With DLSw, the LLC 2 connections are terminated at the DLSw routers, which acknowledge the frames locally instead of transmitting the acknowledgements across the WAN. This practice is called spoofing, and it can reduce the WAN traffic drastically. Also, it eliminates possible LLC timeouts across the WAN.

WAN traffic is reduced in similar fashion for SDLC-attached devices. Polling is done locally by the DLSw station, and only productive polling results in data transmission over the IP network.

DLSw-attached stations, even if they are on a leased SDLC link, are always defined as LAN-attached, and are thus considered by an SNA host as switched resources. To VTAM they are defined in switched major nodes.

DLSw supports subarea (intermediate network node) as well as peripheral (boundary network node) communications. However, performance tests indicate that, for INN traffic, DLSw can increase NCP CCU utilization as compared with plain bridging; spoofing can actually be counter-productive in this situation. If you plan to use DLSw for INN traffic you might want to evaluate plain bridging as an alternative.

Data link switching is fairly commonly used, as it was the first SNA over IP solution that became available. It has some attractive features such as spoofing, and it is easy to understand and configure. DLSw, like any other TCP connection, utilizes the dynamism and rerouting capabilities of the IP network.

DLSw has, however, some disadvantages:

- It may require additional routers on each side of the TCP/IP network if performance and availability requirements are to be met.
- Heavy workload demands may be placed on central (concentration point) routers implementing DLSw.
- While DLSw is an industry standard, many vendors have proprietary extensions that work only with their products.
- DLSw as implemented in IBM products does not support HPR. DLSw is based on the use of LLC2 connections using MAC and SAP addresses exchanged on TEST and XID frames. HPR does not require LLC2 connections, and normally uses a SAP other than the one seen in the XID frames. However, some vendors’ extensions to DLSw support HPR connections in certain circumstances.
- When a DLSw router fails, session outages occur, even if backup routers are in place.
- There are no SNA-like class of service or management functions within the IP part of the network, although it is often possible to distinguish between SNA and other types of IP traffic.

You can find more information on data link switching in the following publications:

- *3746 Nways Controller Models 900 and 950: Multiaccess Enclosure Primer*, SG24-5238
- *MSS Release 2.1*, *Including the MSS Client and Domain Client*, SG24-5231

- *IBM 2210 Nways Multiprotocol Router and IBM 2216 Nways Multiaccess Connector Description and Configuration Scenarios - Volume II*, SG24-4956
- *3746, 2210, 2216, and 2220 Interconnectivity: Frame Relay and Related Functions*, SG24-2146
- *6611 Network Processor Introduction and Planning Guide Version 1 Release 4,* GK2T-0334 (contains an excellent description of DLSw functions)
- RFCs 1795 and 2166, which can be found on the IETF Web site `www.ietf.org/rfc/`.
- The APPN Implementers' Workshop home page at `www.networking.ibm.com/app/aiwhome.htm`.

1.4 AnyNet and MPTN

AnyNet products implement the IBM Multiprotocol Transport Networking (MPTN) architecture, which supports mixed protocol networking. With AnyNet, application programs that were designed to operate over one transport networking protocol, such as SNA, OSI or TCP/IP, can run over a different transport network.

MPTN architecture also supports network interconnection, which means connecting transport networks of different types in a serial fashion to provide a multinetwork path for communicating partners.

In the context of MPTN, application programs or other end users are defined as *transport users* and each network using a protocol of its own is defined as a *transport provider*.

To build this type of multiprotocol network, some of the functions needed are:

- Address mapping, by which network addresses are changed from one network protocol format to another. In SNA networks, addresses are expressed in textual format, consisting of a NETWORK_NAME.LU_NAME pair. In IP networks addresses are expressed as four-byte binary numbers, usually presented in dotted decimal format, such as 155.130.96.29. (This four-byte addressing of TCP/IP will be replaced by a larger address space, in the so-called IPNG or IPv6).
- Function compensations to provide required functions to the transport user but which are not available natively in the transport provider network. MPTN provides a standard set of compensations for missing functions, which can be used by any protocols. An example of compensation is connection establishment and termination. This is needed when running SNA (connection oriented) over IP (which is connectionless in nature).
- Network interconnection. MPTN transport gateways are used to interconnect different transport providers when the end users are connected to different transport providers (one to its native transport and the other to a non-native one).
- Network management. Management of an MPTN network requires correlation between different native management protocols in the transport networks.

1.4.1 AnyNet/MPTN Node Types

An *access node* contains MPTN functions that allow applications to run over non-native transport providers. The access node function is used in systems that only need access to a transport network in order to reach their communications partners. The node is typically an endpoint of a connection. An access node may contain one or multiple transport providers, and multiple transport users.

Figure 2 on page 9 shows the structure of an MPTN access node. In this example the access node is a Sockets over SNA node. It contains two transport users (an SNA CPI-C application and a Sockets application) and a single transport provider (the SNA network). Two such nodes connected by an SNA network can allow Sockets applications to communicate with each other without the presence of an IP network.

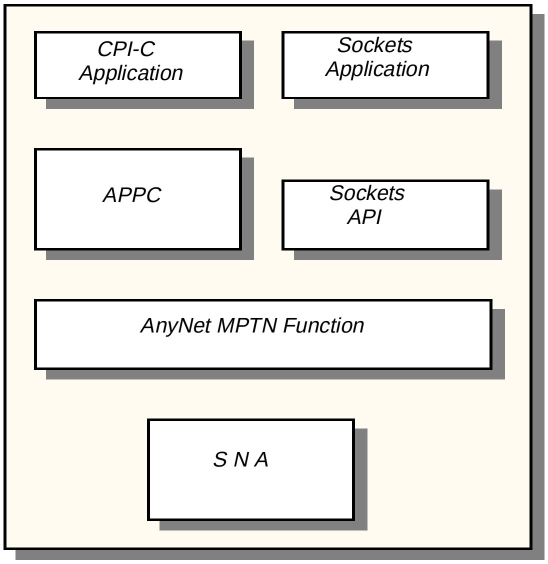


Figure 2. AnyNet Access Node Structure

The end-to-end connection is always between matching transport users, for example SNA transport user to SNA transport user, but not SNA transport user to TCP/IP transport user.

Figure 3 on page 10 shows how protocol users of the same type (shown as X) can communicate with each other when using AnyNet. On the left there are two systems with MPTN access node capabilities. This allows their type X transport users to communicate over a type Y transport provider. The type X transport users in the two nodes on the right can communicate natively over the type X transport providers.

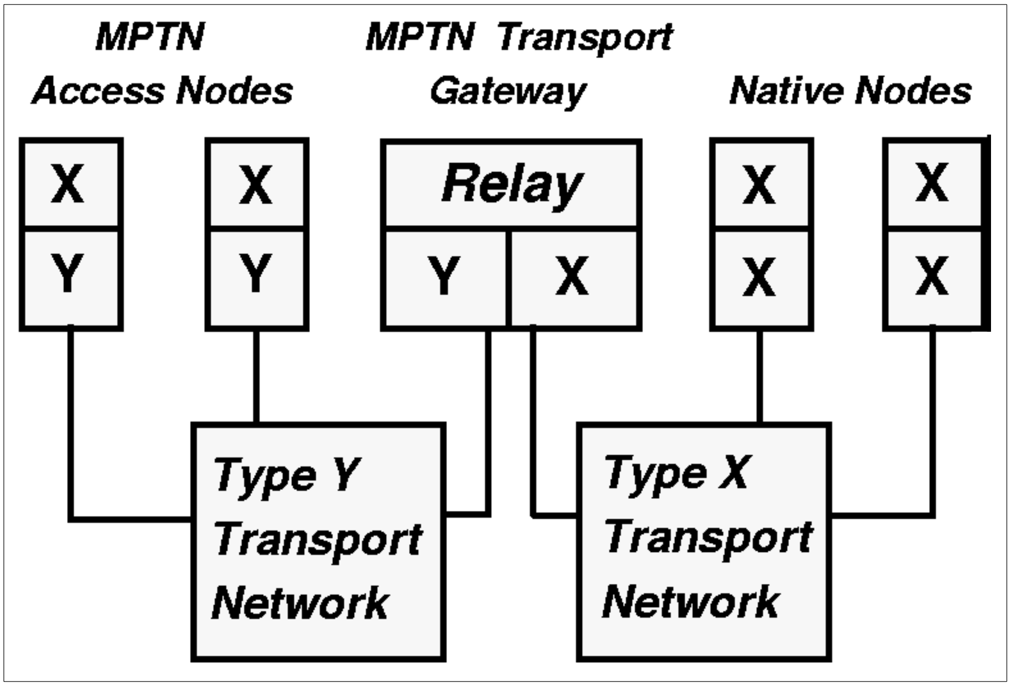


Figure 3. AnyNet/MPTN Node Types

For communication across the two networks, an *MPTN transport gateway* is needed. It connects the two different transport networks, relaying data from one transport network to the other and doing the necessary conversion from one transport protocol to another.

The MPTN architecture and AnyNet products support communications through multiple transport gateways connecting several transport networks of differing protocols.

A *native node* is defined in the architecture as a node that does not implement MPTN functions. An example could be a non-IBM UNIX platform that implements native Sockets applications on a standard TCP/IP stack. To use AnyNet functions, such a host has to route IP packets via an AnyNet gateway node.

In IBM products, MPTN is implemented as part of the Common Transport Semantics (CTS) layer of the Networking Blueprint, as shown in Figure 4 on page 11.

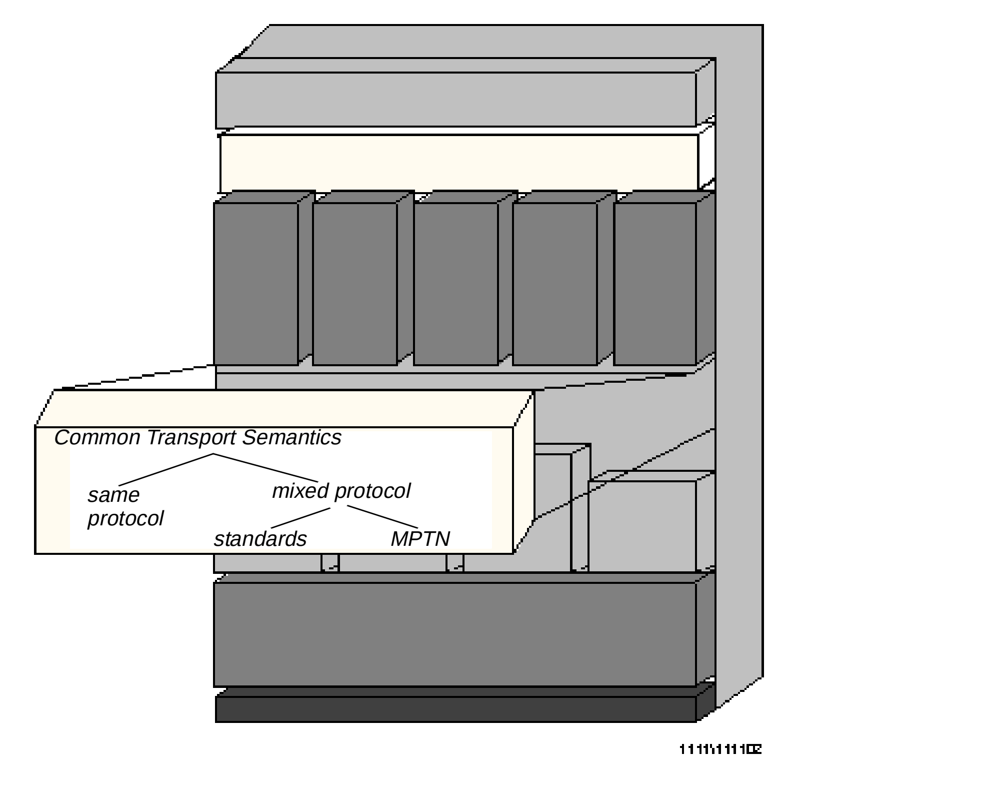


Figure 4. IBM Networking Blueprint with CTS and MPTN

Common Transport Semantics (CTS) provides a common boundary between the application support layer and the transport network layer, separating the transport user and the transport network from each other.

CTS includes all of the functions in the underlying transport providers. If required functions are missing from any of the transport providers, CTS itself provides those functions via compensations. CTS function is delivered in different ways depending on the situation:

1. Same protocol: CTS includes the case where the transport user protocol matches the transport provider protocol. The connection or datagram is native, and no changes are made to the protocol flows.
2. Mixed protocol includes two situations:
   - Standards: CTS function can be delivered in specific situations by accepted industry standards, such as RFC 1006 for OSI over TCP/IP or RFCs 1001 and 1002 for NetBIOS over TCP/IP.
   - MPTN: This part of CTS was introduced by IBM. MPTN is typically used when SNA is one of protocols being used.

The architecture is described in detail in the following manuals:

- *Multiprotocol Transport Networking (MPTN) Architecture: Technical Overview*, GC31-7073
- *Multiprotocol Transport Networking (MPTN) Architecture: Formats*, GC31-7074

AnyNet implementations of the MPTN architecture in different platforms include:

- SNA over IPX
- SNA over TCP/IP
- Sockets over IPX
- Sockets over NetBIOS
- Sockets over SNA
- IPX over TCP/IP
- IPX over SNA
- NetBIOS over TCP/IP
- NetBIOS over SNA

1.4.2 AnyNet Products for TCP/IP and SNA/APPN

For the purposes of this redbook, the AnyNet protocols that concern us are Sockets over SNA and SNA over TCP/IP.

Sockets over SNA utilizes independent LU 6.2 sessions to carry TCP/IP traffic across an SNA (subarea or APPN) network. The following current IBM products provide various options of Sockets over SNA:

1. Access node and gateway functions:
   - SecureWay Communications Server for Windows/NT, Version 5 and above
   - SecureWay Communications Server for OS/2 Warp, Version 4.1 and above
   - SecureWay Communications Server for AIX, Version 4 Release 2 and above
2. Sockets over SNA access node only:
   - SecureWay Communications Server for OS/390 SNA (formerly known as VTAM), from VTAM Version 3 Release 4.2
   - Operating System/400 (OS/400) Version 3.7 and above
   - SecureWay Personal Communications for Windows NT, Version 4 and above
   - SecureWay Personal Communications for OS/2, Version 4 and above

AnyNet SNA over TCP/IP utilizes TCP connections to carry dependent and independent LU sessions across an IP network. All implementations support independent LUs, but CS for OS/390 requires that dependent LUs utilize the dependent LU requester/server (DLUR/S) function. This is the only case where the DLUR/S control sessions do not require an APPN path. The following current IBM products provide various options of SNA over TCP/IP:

1. Access node and gateway functions:
   - SecureWay Communications Server for OS/390, from VTAM Version 3 Release 4.2 (dependent LU sessions require VTAM V4R2 or above)
   - SecureWay Communications Server for Windows NT, Version 5 and above
   - SecureWay Communications Server for OS/2 Warp, Version 4.1 and above
   - SecureWay Communications Server for AIX, Version 4 Release 2 and above
2. Access node only:
   - Operating System/400 (OS/400) Version 3.7 and above
   - Personal Communications for Windows NT, Version 4 and above
   - Personal Communications for OS/2, Version 4 and above

In addition to these current AnyNet products, there are earlier implementations, such as AnyNet/2 for OS/2. They are compatible with, and can still be used together with, the newer products.

Note: The SecureWay Communications Server family of products was previously known as the eNetwork Communications Server family.

1.4.3 AnyNet in Summary

The AnyNet product family provides a convenient way to run SNA over TCP/IP and Sockets programs over SNA, as well as some additional protocol combinations.

AnyNet Sockets over SNA is the only solution that we have for Sockets applications to utilize SNA transport.

AnyNet has the following benefits:

- It is a software only solution. Generally no additional hardware is needed.
- It is based on a solid architecture, MPTN, covering most protocol requirements.
- AnyNet has been available for several years; the products are mature and stable.
- AnyNet capability is available as a built-in feature in all Communications Server products for most IBM platforms, and in two Personal Communications products (OS/2 and Windows).

1.5 Enterprise Extender

The Enterprise Extender technology is a simple set of extensions to the existing, open high-performance routing (HPR) technology. Its purpose is to carry SNA traffic over an IP network while maintaining the SNA class of service to the maximum possible extent. Enterprise Extender is an up-to-date replacement for data link switching.

To the HPR network, the IP backbone is a logical link; to the IP network, the SNA traffic comprises UDP datagrams that are routed without hardware or software changes to the IP backbone. There is no protocol transformation, as UDP/IP is seen as just another type of SNA DLC. Nor is there the overhead of additional transport functions, since TCP is not used.

Since Enterprise Extender is HPR, it provides the benefits of HPR all the way from the mainframe to the furthest SNA-capable node in the network, such as a remote TN3270 Server. It does this regardless of whether the underlying routing network is SNA, IP or a combination. Thus, for example, sessions using an SNA backbone can be rerouted nondisruptively to a path over an IP backbone (or even the Internet) should a critical SNA component fail. No other technology can provide this; data link switching is susceptible to failure of the DLSw routers at the edges of the IP network.

The other major benefit of Enterprise Extender is the ability to maintain the SNA class of service across and within the IP network. Routers typically use either the precedence bits in the IP header, or the UDP port number, to prioritize traffic. Enterprise Extender can use one or both of these to indicate the APPN transmission priority to the network. Once again, this feature is unique to Enterprise Extender.

Figure 5 on page 14 illustrates one example of an Enterprise Extender configuration.

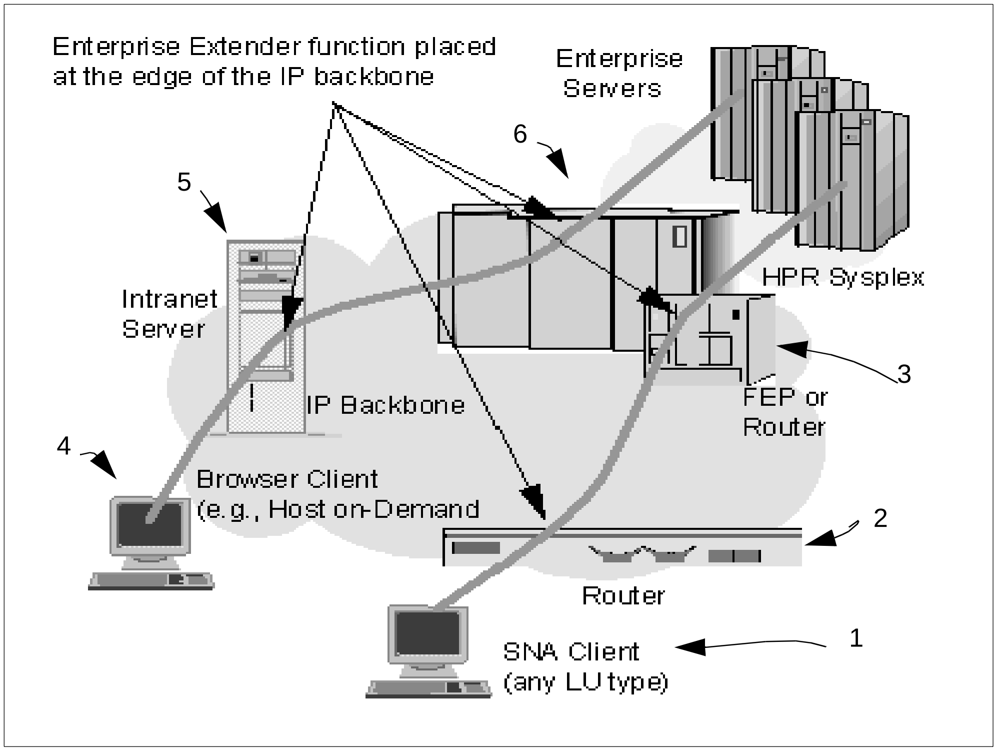


Figure 5. Enterprise Extender Operation

In this example the Enterprise Extender function is placed in the channel-attached routers at the mainframe site (for example, 2216 or 3746 MAE) and the remote gateways or routers closest to the user workstations. The workstations themselves could be any of the following:

- An IP workstation, with a Web browser or a TN3270 client. The nearest gateway (for example, CS/2 or CS/NT) is then a TN3270 and HOD server, and HPR is used all the way between the gateway and the host application.
- An SNA workstation with 3270 emulation alone. The nearest router (for example, 2210 or 2212) is then a dependent LU requester node, and HPR is used all the way between the router and the host.
- An SNA workstation with a full SNA/APPN stack. Now the nearest router is simply an ANR node and HPR is available all the way from end to end, still with a pure IP backbone.

This example, with a Parallel Sysplex at the host site, provides all the VTAM sysplex benefits, particularly generic resources and multinode persistent sessions, to all the users whether SNA-attached or IP-attached. In the example:

- The native SNA workstation (1) is using HPR all the way to the application in the sysplex. The first hop of the HPR path is native SNA to the router (2); the second hop is Enterprise Extender across the IP network to the channel-attached router (3); and the third hop is native SNA again to the host.

- The Web browser workstation (4) is using TN3270 over native IP to the Intranet server (5). This server runs the HOD server as well as the TN3270E Server. The connection from there on is a two-hop HPR path carrying the SNA LU 2 session: one hop using Enterprise Extender to the channel-attached router (6) and the second hop native SNA to the host.

An even better solution might be to run Enterprise Extender on the OS/390 host itself. In that case, the channel-attached routers run only IP; eliminating the DLSw function can provide a very large increase in throughput of such routers. Again all the benefits of HPR are realized. A TN3270E Server at each remote location, coupled with Enterprise Extender on CS for OS/390, provides all the SNA/HPR capabilities even though the *whole* network is IP.

1.5.1 Enterprise Extender Implementations

Enterprise Extender is available on the following products from IBM:

- SecureWay Communications Server for OS/390, Release 6 (with APAR OW36113) and above
- 2216 Multiprotocol Access Services, Version 2 Release 2 and above
- 2212 Access Integration Services
- 2210 Multiprotocol Routing Services, Version 2 Release 2 and above
- 3746 Multiaccess Enclosure with Feature Code 5805
- Network Utility
- SecureWay Communications Server for Windows/NT, Version 6 and above

1.6 Telnet/3270

TN3270 is a protocol by which a workstation user can get access to IBM 3270 applications running on an SNA host.

TN3270 requires a client function on the workstation and a TN3270 Server somewhere in the network (sometimes even at the host). TCP/IP must be running in both the client and the server system. The TN3270 Server translates the TCP/IP protocol to SNA 3270 (LU type 2), and maintains SNA sessions with the desired host application(s).

The TN3270 client program runs typically in a PC workstation, using the installed TCP/IP stack for transport. No other protocols need to be installed in the workstation. TN3270 is a popular method for integrating SNA and IP for workstations that use SNA only for 3270 sessions, because the TCP/IP stack is typically installed in most workstations anyway.

The old TN3270 had certain limitations such as lack of printer support. For some time, an updated version of the protocol called TN3270E (E stands for Extended) has been available. The old TN3270 clients can be used to access an extended server, since the two protocols are compatible. Note that TN3270E is *not* required for the use of the 3270 extended data stream; basic TN3270 supports this.

A TN3270 Server in today's IBM routers will normally use APPN and DLUR to access the host applications, but subarea protocols can also be used if the router is adjacent to a VTAM or NCP node.

By moving the TN3270 Server out from the mainframe, one can achieve the following benefits:

- Reduction in processor load, as the protocol conversion is done elsewhere.
- Better service level to users, in the form of constant response times and better availability. The further out in the network the TN3270 Server is placed, the greater the level of service. This is due to the fact that SNA and APPN are more deterministic by nature than IP, and use the class of service for routing and prioritization.
- Use of HPR with its dynamic route switching capability, giving even greater availability. In addition, the presence of at least one HPR-capable SNA hop in the network allows you to exploit multinode persistent sessions in a Parallel Sysplex.
- A TCP/IP stack is not required in the OS/390 host unless needed for other purposes.

However, if all your 3270 applications are concentrated in the same place then placing the TN3270 Server on the host can be the most economical solution. As a general principle, we believe that the TN3270 Server function should be as close as possible to either the workstation or the host. Having it somewhere in between can introduce a potential disruptive failure point at the SNA/IP boundary.

1.6.1 TN3270E Server Implementations

The TN3270E Server function is available on the following current platforms:

- SecureWay Communications Server for OS/390
- OS/400
- SecureWay Communications Server for OS/2
- SecureWay Communications Server for AIX
- SecureWay Communications Server for Windows/NT
- 2216, 2212, 2210, 3746 MAE and Network Utility

1.7 Host On-Demand

Host On-Demand (HOD) is a means by which 3270 applications can be accessed from an Internet browser. It is based on Java, and thus requires a Java-capable browser. Nothing else is needed in the client workstation, except for a connection to the Internet or other TCP/IP network.

HOD is actually a TN3270 (or other Telnet) client implemented in Java. To use HOD, a Web server is needed somewhere in the network for users to connect to. The TN3270 (or TN5250, or other Telnet) Java applet is downloaded from this server to the workstation. This Web server may also be running the TN3270 Server function, through which the applications in the SNA hosts are accessed.

The HOD Web server and/or the TN3270 Server can be in the application host itself. Alternatively, they could be on completely different platform(s), somewhere in the SNA network.

The Host On-Demand server function is available for the following environments:

- OS/390
- OS/400
- OS/2
- AIX, Sun Solaris and HP-UX
- Novell Netware
- Windows/NT

The latest release, Host On-Demand Version 3, supports TN3270, TN5250, VT52, VT100, VT220 and CICS Java Gateway displays.

HOD is an excellent solution for temporary access to 3270 applications from a Web browser. It can be used, for example, to provide safe external access to your enterprise data to the public or to selected customers. Your normal IP firewall, together with your existing SNA security measures, should prevent unauthorized access.

Host On-Demand supports end-to-end Secure Sockets Layer (SSL) encryption between the client and the HOD server. In addition, HOD server profile access is protected by user ID and password, and authentication is performed by the server sending an X.509 certificate.

Host On-Demand presents the Web user with the traditional green screen as an interface. By using a product such as Host Publisher, you can instead present a Web-like GUI interface to users not familiar with the 3270 green screen (such as the public).

1.7.1 Host On-Demand Summary

Host On-Demand allows Web users to access host programs without installing a terminal emulator on their computer. A Java-capable Web browser is all they need. In our opinion, however, it cannot be recommended for constant professional use because of the additional overhead of running Java and a browser as well as the 3270 protocol. Native SNA or TN3270E are better solutions for heavy use.

With HOD you have these benefits:

- Multiple concurrent sessions with one or more hosts.
- No end-user installation or configuration required to establish a session.
- User's browser remains available for concurrent Web use.
- Cost-effective distribution of emulator software updates with Java.
- Print screen, file transfer, cut and paste, and keyboard remapping.
- Host On-Demand is Java-based, so users in different operating environments (whether they are using network computers, traditional personal computers, or advanced workstations) get the same look and feel.
- A Java-based application programming interface (API) for application development is available to customize desktops.

- National language support is available, including support for double-byte character sets.

Additional information on HOD can be found on the Host On-Demand home page at URL:

`http://www.software.ibm.com/network/hostondemand/`

Chapter 2. Enterprise Extender

The Enterprise Extender technology was implemented on most of IBM's networking platforms during 1998. Its main objective is to provide SNA-over-IP integration that is significantly superior to its predecessors such as data link switching and AnyNet.

2.1 Benefits of Enterprise Extender

Enterprise Extender combines features of SNA and IP to offer the best of both worlds when running SNA traffic over an IP backbone. Its advantages are in the three major areas of availability, performance and usability.

2.1.1 Failure Protection

TCP/IP has always had the ability to reroute packets around failing components, without disrupting the connection, by means of its connectionless IP transport. More recently SNA has implemented the same function, albeit in a rather different fashion. The high-performance routing (HPR) extension to SNA is connection-oriented as SNA has always been, but when it detects a failure it will move an existing connection around a failing component. The use of HPR transport over an IP backbone provides four benefits, only one of which is available to traffic using native TCP/IP:

1. Non-disruptive rerouting upon a failure could be accomplished using either IP or HPR methods, depending on the location of the failure.
2. HPR retains the connection-oriented nature of SNA, making it more predictable and easier to manage.
3. The use of HPR allows both the above benefits to be realized in a combined network which uses both SNA and IP routing in different portions. Enterprise Extender provides *seamless* integration, in that the IP network is seen as an additional hop in an SNA connection path.
4. Adaptive rate-based flow control provides a fair allocation of bandwidth among SNA sessions and IP traffic while maintaining the SNA class of service.

By contrast, data link switching provides non-disruptive rerouting across the IP network alone, and is susceptible to failures of the DLSw routers at the edge of the network. Since it does not support HPR, it cannot provide full end-to-end non-disruptive rerouting in a combined SNA/IP network.

Enterprise Extender is particularly suited for customers wishing to exploit the multinode persistent sessions (MNPS) function in a Parallel Sysplex. MNPS allows an application to move between MVS images in a sysplex after a failure, without disrupting existing sessions. Since MNPS utilizes HPR to achieve this, Enterprise Extender is a perfect complement to MNPS even in the situation where there is no SNA backbone network whatsoever. The only requirement is a remote partner node capable of Enterprise Extender (and therefore HPR).

2.1.2 Class of Service

One of the biggest issues facing those who wish to transport SNA over an IP backbone is the question of maintaining SNA's class of service. In native SNA the class of service specified for a particular session is used to determine both the route taken by the session and the transmission priority allotted to it.

With an IP backbone the route is essentially unpredictable because of IP's connectionless transport. However, IP provides for a transmission priority using the precedence bits in the IP header. Many routers now support the use of these bits, but in the past they have tended to use the TCP or UDP port number as a means of assigning priorities to packets.

Enterprise Extender supports the use of both the precedence bits and the port numbers to inform the IP network of the transmission priority. Use of the precedence bits is recommended because the UDP or TCP port numbers are carried inside the IP datagram, whereas the precedence bits are in the IP header. Thus encrypted packets have unreadable port numbers and fragmented packets have no port numbers after the first fragment; therefore, intermediate routers cannot determine the priority in these cases.

2.1.3 Flow and Congestion Control

TCP/IP and HPR both provide their own unique, network-specific mechanisms for flow and congestion control. TCP uses a windowed technique, whereas HPR uses a technique based on data rate. Enterprise Extender introduced a new variant of the HPR flow control method known as responsive mode adaptive rate-based (ARB) flow control. Responsive mode ARB, like basic mode ARB, is designed to prevent network congestion; however, it also ensures a fair division of network capacity between the four SNA priorities and the native IP traffic.

2.1.4 Usability and Cost Effectiveness

Enterprise Extender technology can reduce the demands on the data center routing platforms such as routers and front-end processors and thus provide a more cost-effective solution than other integration technologies. The boundary nodes between the SNA and IP backbones have less work to do than, for example, DLSw routers which must maintain TCP connections with their attendant flow control and error recovery requirements. Enterprise Extender does all that at the HPR endpoints wherever they may be located.

Enterprise Extender has been designed to run over existing IP networks without requiring any change to applications or to IP routers. SNA applications see the same network interface as before, whereas IP routers see the same IP (UDP) packets as before. Only the HPR nodes at the edges of the IP network need to be aware of Enterprise Extender.

In conjunction with the branch extender technology, which was introduced in 1997, Enterprise Extender permits the implementation of extremely large networks that provide SNA application access using any combination of SNA and IP networks and clients.

Together, these technologies can dramatically reduce the overhead in building the real-world networks that integrate the SNA and IP network and application assets.

2.2 HPR Overview

Enterprise Extender is a new DLC type used by the SNA high-performance routing architecture, so before describing Enterprise Extender itself we offer a brief overview of HPR. Further details on HPR can be found in *Subarea to APPN Migration: HPR and DLUR Implementation,* SG24-5204*,* and in *Inside APPN - The Essential Guide to the Next-Generation SNA,* SG24-3669*.*

HPR is a set of enhancements for APPN whose main objectives are:

- **To improve APPN data routing**

  With faster, more reliable communication lines it is neither necessary nor desirable to perform error recovery, flow control and complicated routing functions at intermediate nodes in a network. HPR takes full advantage of modern technology to eliminate these functions, by ensuring that they are performed only at the endpoints of a session path. With HPR, each intermediate node has only a minimal switching function to perform.

- **To improve APPN reliability**

  Customers have always wanted the network to recover from errors, and to find an alternate path without requiring the end user to take action. HPR switches session routes to bypass link and node failures if an acceptable alternate path is available. This occurs transparently to the sessions; in other words, the session is not disrupted.

- **Compatibility and easy migration**

  HPR implements many functions in exactly the same way as base APPN. The same topology, the same directory, the same search methods, and the same route calculations are used. Priority queueing, connection networks, DLUR/S, VR-TG and cross-network sessions are all supported by HPR. The same management data is carried on the same sessions in the same ways. HPR functions are invoked only at session initiation time, when the first HPR-capable node on the session path discovers the new session request.

In HPR, sessions are grouped together on logical connections called rapid transport protocol (RTP) connections. Sessions traversing the same route and using the same class of service will normally share an RTP connection. Data flowing on RTP connections flows as network layer packets (NLPs). Figure 6 on page 23 illustrates an RTP connection.

The intermediate nodes in an HPR network have only one function, to route data quickly and efficiently from one link to another. This technique is known as automatic network routing (ANR). A node performing ANR is aware of neither the sessions nor the RTP connections passing through it. If there is no awareness there is no need for cleanup or restoration after a failure, so the ANR technique allows easy switching of session paths. Aside from the routing information, ANR nodes are aware of the transmission priority so that this can be maintained throughout the session path. Please refer to 2.2.1, “Automatic Network Routing” on page 22 for more details.

All the rest of the HPR function is implemented in the endpoints of a connection. The major part of this function is called rapid transport protocol (RTP). A node capable of RTP must be able to:

- Establish and maintain an RTP connection with a partner node

- Recover from the loss or corruption of packets
- Resequence packets arriving out of order
- Perform adaptive rate-based (ARB) flow control for the RTP connection
- Act as the boundary between base APPN and HPR portions of a network
- Perform a nondisruptive path switch if a permanent failure occurs and an alternative path is available

Please see 2.2.2, “Rapid Transport Protocol” on page 23 for more details on RTP.

The ability to understand ANR is part of the HPR base function, which is a prerequisite for RTP. The HPR base function includes:

- The ability to exchange HPR capabilities via XID exchange
- The ability to exchange HPR-related topology information on CP-CP sessions
- For a network node, the ability to calculate HPR-only routes, which are required for a successful path switch

There is one further HPR option that is particularly relevant to Enterprise Extender, and that is called control flows over RTP. If a node implements the HPR base and RTP options alone, then the CP-CP sessions and the route setup messages (which are used to establish the RTP connection, much as a BIND establishes a session) flow as base APPN (FID-2) packets. If the connection does not permit base APPN flows (in other words, the connection is HPR-only) then the CP-CP and route setup traffic must itself flow on RTP connections. The control flows over RTP option permits exactly this. RTP support is a prerequisite to the control flows option. Since Enterprise Extender is an HPR-only connection, the nodes at either end of such a connection must support the control flows option.

2.2.1 Automatic Network Routing

Automatic network routing (ANR) is a low-level routing mechanism that minimizes processing cycles and storage requirements for routing packets through intermediate nodes.

ANR is a source-routing protocol; the routing information (ANR label) for every node on the path is contained in the packet header. Furthermore, the ANR label represents the onward link for each node, *not* the session as with base APPN routing. There is *no* session awareness in a node performing ANR routing. All it has to do is inspect the first ANR label in the packet header, strip it off, and forward the packet to the correct outbound link.

This label stripping technique is much more efficient than the label swapping technique (intermediate session routing) used by base APPN nodes. ISR requires that the node inspects the session identifier in the incoming packet, uses it to look up a session table, swaps it to the outbound session identifier, and forwards the packet.

Figure 6 on page 23 illustrates the way ANR labels are used.

1. Node A sends an NLP to node B with ANR labels 21 / 33 / 65 / FF in the NLP header as shown.
2. Node B looks in the header for the first ANR label. This is 21, so node B removes it from the header and transfers the truncated NLP to the link it knows as 21.

3. Node C receives the NLP, removes the next ANR label (33), and sends the NLP on the link it knows as 33.
4. Node D receives the NLP and recognizes that the next ANR label (65) represents not a link but the endpoint of the RTP connection. Therefore, it passes the data in the NLP to the higher protocol layers for processing.
5. The response to this message takes a similar course through the network in the opposite direction.

In the example, the ANR labels are one byte in length but this is not necessarily so; different products implement different lengths of label. Since each node on an HPR path assigns the ANR labels which it is to interpret, there is no need for any other node to be aware of their length or meaning. Each node will find its own label at the start of any NLP it receives.

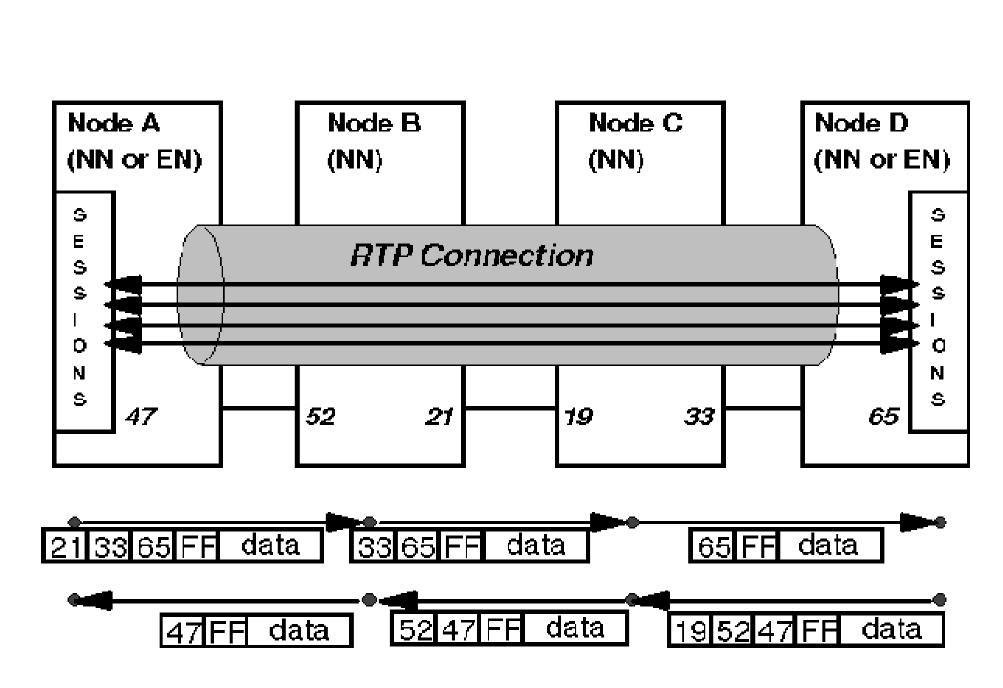


Figure 6. RTP Connection and Use of ANR Labels

2.2.2 Rapid Transport Protocol

An HPR network consists of a minimum of two APPN nodes which have implemented the HPR base functions and the RTP functions. These RTP nodes must be directly connected to each other, or they must be connected by a path of consecutive network nodes that support the ANR functions.

During activation of a link between two HPR nodes, these nodes exchange their HPR capabilities. When the APPN topology database is updated to reflect the new connection, the HPR capability of each node and link is included in the update. Thereafter, the APPN searching and route calculation algorithms are the same for HPR as for base APPN; the major difference comes when a session is to be started.

At session initiation time, the node that contains the primary LU receives a route selection control vector (RSCV) from its network node server, which represents the route that the session is to take. Base APPN processing requires that the RSCV is appended to the BIND, which then flows through the network establishing the session path. With HPR, however, the first RTP-capable node on the path must establish an RTP connection, if possible, over which the BIND may flow as an NLP. This RTP-capable node, therefore, examines the RSCV (which contains the HPR information from the topology database) to see if such a connection exists or can be set up.

The RTP-capable node determines the furthest RTP-capable partner on the session path that is linked to it by a contiguous chain of ANR-capable nodes. If an RTP connection already exists over the same route and for the same class of service, the BIND (and therefore the new session) flows over that connection. If such a connection does not exist, one is established by means of the route setup message and its response. The route setup carries that portion of the RSCV that represents the HPR part of the session route, and finds its own way through the network exactly as a BIND would. The route setup exchange allows each intermediate node to assign ANR labels to the links on the path, and to make those labels known to the RTP endpoints that will use them to route NLPs. The route setup exchange also determines (among other things) the performance characteristics of the RTP connection, so that the ARB flow control algorithm can be initialized with reasonable values.

2.2.2.1 End-to-End Error Recovery

On an RTP connection it is the responsibility of the endpoint nodes to recover from errors such as lost packets or packets arriving out of order. NLPs themselves are subject to CRC checking as they pass through the network, but any error detected at an intermediate (ANR) node may result simply in the packet being discarded.

RTP nodes count the data bytes sent on each RTP connection, and include a byte sequence number in the header of each NLP. This is similar to the way TCP detects lost data. Any discrepancy between the number of bytes received and the sequence number received results in a request for retransmission. RTP nodes have the ability to request selective retransmission, so that only the missing packets need to be resent. This is clearly more efficient than resending *all* the data since the first erroneous packet.

2.2.2.2 Non-Disruptive Path Switch

A path switch can be triggered by a timeout on an RTP connection, or by detection of an error on a link adjacent to one of the RTP endpoints. It can also be initiated deliberately to move a session path in a planned fashion. When this happens, one or both of the RTP endpoints request a new route (RSCV), which must be calculated by the appropriate network node(s). If an HPR-only path is available that satisfies the class of service required, a route setup is sent to restore the RTP connection and to establish the new set of ANR labels. As soon as this is done the NLPs continue to flow without disruption to the sessions using the RTP connection. If no HPR-only path is available, the RTP connection is terminated, together with all the sessions using it.

2.2.2.3 Adaptive Rate-Based Flow/Congestion Control

Adaptive rate-based (ARB) flow control is performed by the RTP endpoints of a connection. Intermediate (ANR) nodes take no part in ARB flow control, nor in adaptive session pacing (which is still performed between RTP endpoints, which treat the connection as a single hop). The objective of ARB flow control is to anticipate congestion before it occurs, providing a fair share of the network capacity to each RTP connection while maintaining optimum throughput. ARB flow control works by measuring the round-trip delay across the RTP connection, and monitoring changes to this delay. An increase in the delay over a period of time results in a message being sent from the receiver to the sender requesting that the sending rate be cut back. A full description of the algorithm is in *Inside APPN - The Essential Guide to the Next-Generation SNA,* SG24-3669.

2.3 Enterprise Extender Description

The Enterprise Extender architecture must carry SNA (HPR) traffic over an IP backbone without requiring changes to that backbone. Therefore, it must treat the IP network as a particular type of SNA logical connection, in much the same way as an ATM or frame relay network is treated.

2.3.1 TCP/IP Protocol Summary

For a comprehensive guide to TCP/IP, please refer to *TCP/IP Tutorial and Technical Overview,* GG24-3376. Here we summarize the basic principles necessary to an understanding of Enterprise Extender.

Each node capable of communicating using TCP/IP is called a *host*. This host may have more than one connection to the network, and each interface is assigned an IP address comprising four decimal digits separated by dots. The IP address is sufficient to identify a particular interface, but not sufficient to identify the process on the host to which packets are directed. For this purpose an additional address called a *port* is defined. A process may use one or more ports to communicate with another process in the network.

It is sometimes necessary for a TCP/IP host to present a single IP address to the outside world even though it has many physical connections. On OS/390, this is accomplished by using virtual IP addressing (VIPA). The host using VIPA pretends to be a router that can redirect packets to the VIPA address, but in reality keeps those packets for its own applications. On outbound datagrams, the VIPA address is identified as the source for these datagrams if the source VIPA option is enabled.

The unit of transfer in an IP network is the IP datagram. IP is an unreliable, best-effort, connectionless packet delivery protocol. An IP datagram carries the IP address of the destination host but no port number. It also carries a *service type* field which contains:

- Three *precedence* bits, which indicate the nature and priority of this datagram
- Four *type of service* bits, which specify the level of service requested in terms of delay, throughput, reliability and cost

IP itself is limited in usefulness because it can identify no process within the host beyond the interface address. Therefore, useful data is carried over IP networks in one of two forms:

1. User datagram protocol (UDP) adds multiplexing to IP by carrying a port number in its header. It adds nothing by way of reliability, flow control or error recovery. It can detect corrupted packets but its only choice is to discard such packets.
2. Transmission control protocol (TCP) provides a reliable, multiplexed (again via port numbers), connection-oriented transport whose basic principles are akin to those of SNA.

2.3.2 Enterprise Extender TCP/IP Protocol Usage

The designers of Enterprise Extender, therefore, had three choices as to the method of transporting NLPs:

1. Raw IP datagrams. Datagrams are completely compatible with the HPR principles, as they flow through the network with minimal overhead and provide no error recovery of any sort. However, raw IP provides no means of multiplexing; it would be impossible to distinguish between HPR over IP and other IP traffic to the same node, let alone between various types of HPR over IP packets. Although raw IP allows priority and type of service to be specified, in practice not all networks or routers are (or can be) configured to support this.
2. UDP packets provide the multiplexing required because they contain port numbers. This allows Enterprise Extender packets to be distinguished from other IP packets. It also permits a priority scheme to be implemented independent of the type of service bits, since many routers are able to prioritize traffic based on the received port number. UDP, in addition, has low overhead since it does not concern itself with error recovery or flow control.
3. A TCP connection also provides multiplexing by means of port numbers, but it incurs a significantly higher overhead than raw IP or UDP. A TCP connection handles error recovery, retransmission and flow control; none of these is required for an HPR connection because the RTP endpoints are responsible for all of them. Moreover, if the Enterprise Extender connection is only part of the HPR path then one end or the other will be an ANR node; the additional burden of a TCP connection is unacceptable for an ANR node which should do as little SNA processing as possible.

UDP was therefore the method chosen for Enterprise Extender, as Figure 7 on page 27 illustrates.

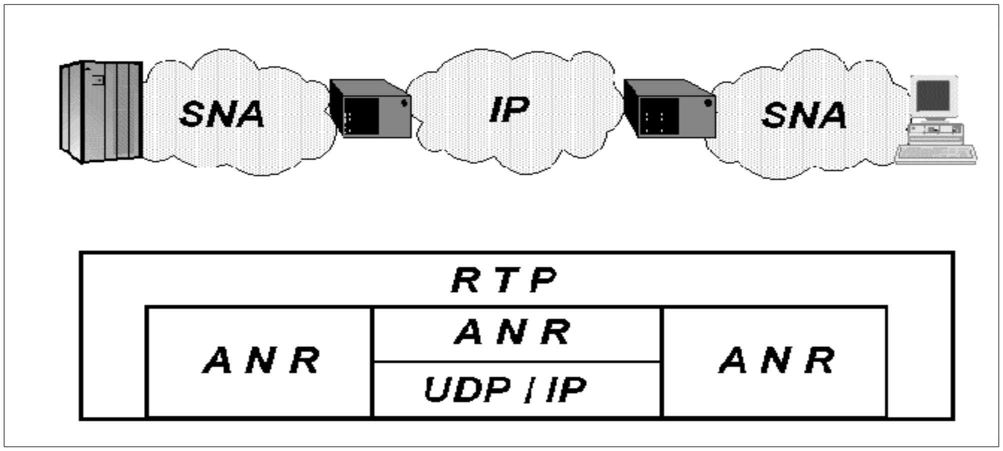


Figure 7. Enterprise Extender Operation

2.3.3 Enterprise Extender Implementation

Enterprise Extender is very similar in concept to the way native SNA over ATM is implemented:

- The underlying transport network appears as an APPN TG but uses logical data link control (LDLC) to exchange XIDs and NLPs. LDLC is a subset of LLC2 that eliminates much of the error handling and acknowledging that RTP makes unnecessary at link level. It is similar in concept to the qualified logical link control (QLLC) used to transport SNA traffic over an X.25 network, but there are some major differences that LDLC has to allow for:

  - LLC2 requires the use of several fixed timers, which are by their nature incompatible with a variable-route variable-delay IP network.
  - LLC2 performs error recovery, which is not necessary in HPR.
  - LLC2 requires in-order delivery, which cannot be guaranteed on an IP network.

  LDLC, used also for native SNA over ATM, includes only the XID, TEST, DISC, DM and UI frame types. These are sufficient to establish the connection (XID), send data (UI), terminate the connection (DISC) and respond in the negative to a previous frame (DM). The TEST frames are used to check whether a connection is still active, a function required by HPR over IP.

- The UDP port number identifies the destination of the datagram as being the partner IP host’s ANR routing function. Several UDP ports (12000-12004) have been registered with the Internet Assigned Number Authority (IANA) for this purpose. Each of these default ports is mapped to one of the APPN transmission priority values, with the fifth (12000) being used for XID exchange. An Enterprise Extender implementation may choose to alter these port numbers, but by using the registered defaults you can be reasonably sure that no other application will conflict with Enterprise Extender. ANR labels are mapped to the partner’s IP address.

- Because there is no link-level error recovery and no guarantee that packets will arrive in order on an Enterprise Extender connection, only HPR NLPs can be transported once the XID flows are completed. Therefore, both partner nodes must support control flows over RTP.

- A connection network can be defined on the IP network, which uses logical Enterprise Extender links. Defining all such logical links between each pair of a large number of IP addresses would be an unpleasant task, just as it would be on a LAN.
- The SNA transmission priority is mapped to the UDP port number, which is why five UDP ports have been registered for Enterprise Extender use. The main reason for this is that many IP routers can be configured to prioritize traffic based on the port number. However, the Enterprise Extender architecture permits the use of the precedence bits in the IP header for the same purpose. These bits are reserved in the TCP/IP architecture for exactly this usage, but not all routers take account of them. Among those that do are the 2216, 2210 and MAE which set both the precedence bits and the UDP port number. LLC commands (XID, TEST, DISC, DM) use the same precedence bit setting (the highest) as network priority NLPs. Table 1 shows the correspondence between the APPN priorities, the IP precedence bits and the UDP port numbers.

Table 1. Use of UDP/IP for APPN Priorities

APPN Priority	IP Precedence	UDP Port
N/A (LLC commands)	B’110’	12000
Network	B’110’	12001
High	B’100’	12002
Medium	B’010’	12003
Low	B’001’	12004

Because of its design, Enterprise Extender is extremely flexible. It can be used in all networks from the smallest to the largest, and provides the customer with a wide choice of where the SNA/IP boundary is placed.

Enterprise Extender enables remote branches or workstations to be connected to the SNA backbone using the Internet, with no application changes required, while maintaining SNA connectivity from end to end. Dependent LU sessions can be carried on an Enterprise Extender connection as easily as any others, and by utilizing the dependent LU requester function (available on all current IBM workstation and router platforms) they can take advantage of the Enterprise Extender technology all the way into the most remote locations.

One issue of which you should be aware if using Enterprise Extender between distinct enterprises is that many firewalls are configured by default to block UDP datagrams.

2.3.4 A Restriction to Be Aware Of

Because an Enterprise Extender connection is HPR-only, there is an important restriction that must be understood relating to subarea SNA. Such a connection cannot be adjacent to an interchange transmission group (IC-TG) unless the other end of the connection is an end node. The most common scenario in which this might occur is where a CS for OS/390 host performing the Enterprise Extender role is also an SNI gateway to another network, as illustrated in Figure 8 on page 29.

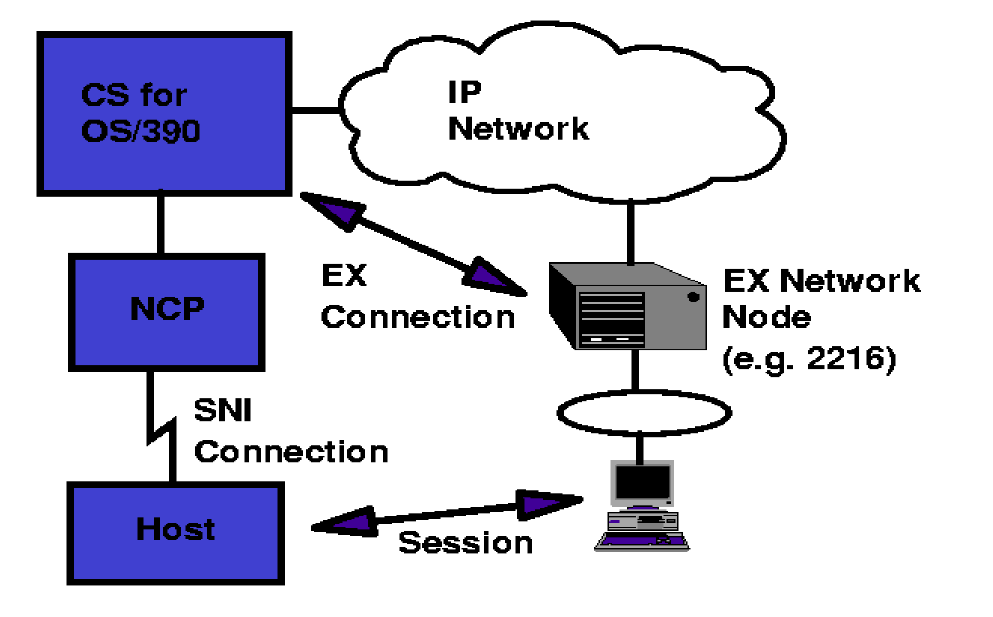


Figure 8. HPR-to-Subarea Restriction

In this example, the session between the user terminal and the cross-network host would fail. This is because:

- There is an Enterprise Extender connection on the session path, between the 2216 network node and the CS for OS/390 host. This connection can support only HPR NLPs, and no other form of SNA traffic.
- The adjacent hop on the session path is an SNI connection, which can support only subarea traffic, and no other form of SNA. In APPN terms this connection is called an interchange transmission group since the CS for OS/390 host, which acts as an interchange node, represents it as a TG in the APPN topology.

It is this juxtaposition between subarea-only and HPR-only links that prevents the session from working. All other sessions that use either the SNI connection or the Enterprise Extender connection will work. For the subarea-to-HPR route to work the 2216 would have to be an APPN end node, which it cannot be.

There are various ways around this restriction:

- If there are two OS/390 hosts running Enterprise Extender in your site, define a FID-2 connection between the gateway NCP and the other host. This will introduce an additional hop, capable of base APPN, into the session path. The subarea-only hop is no longer adjacent to the HPR-only hop.
- If there is an alternative Enterprise Extender node (such as a 2216) in your site, define a native APPN connection between the gateway NCP and the 2216. This also introduces a base APPN hop, and still allows the use of Enterprise Extender across the IP backbone.

If the subarea connections are same-network, the solution is much more straightforward: define VR-TGs on the VTAM-to-VTAM connections. VR-TG is APPN and therefore does not break the subarea-to-HPR restriction. Unfortunately VR-TG cannot be defined across an SNI gateway.

2.3.5 Responsive Mode Adaptive Rate-Based Flow Control

The original ARB algorithm introduced with HPR works very well with SNA traffic alone, but is less efficient in the Enterprise Extender environment when SNA and IP traffic must coexist. Upon detection of a lost packet (a common occurrence in IP networks), ARB would immediately reduce its sending rate by a significant amount, thus impacting performance.

An enhanced algorithm known as Responsive Mode ARB, or ARB-2, was introduced with Enterprise Extender, and is now an option for RTP nodes whether or not their HPR connection includes an Enterprise Extender link. Nodes that support Responsive Mode ARB can negotiate their level of ARB support during route setup exchange, and fall back to the original ARB if their partner does not support Responsive Mode. Thus each individual RTP connection can choose which option to use. Responsive Mode ARB provides the following features:

- It competes fairly with TCP congestion control. Therefore, reservation of bandwidth for SNA traffic is no longer necessary (as it often is with DLSw).
- It can be tuned to tolerate a certain level of lost data.
- It gives priority to short transmissions.
- It allocates a fair bandwidth to sustained transmissions, independent of propagation delays.
- It can ramp up its transmission rate faster at startup.

2.4 Enterprise Extender Implementation

At the time this redbook is being written, there are eight products which implement, or plan to implement, Enterprise Extender:

- The 221X router family, which comprises the 2216, 2212, 2210, Network Utility and 3746 Multiaccess Enclosure. The minimum software level required is Version 2, Release 2 of Multiprotocol Access Services or its equivalent on the appropriate platform.
- SecureWay Communications Server for OS/390, Release 6 (with APAR OW36113) or later.
- Communications Server for Windows NT, Version 6 or later.
- Communications Server for OS/2, the version planned for availability in Spring 1999.

2.4.1 SecureWay Communications Server for OS/390

Enterprise Extender on the OS/390 platform requires CS for OS/390 Release 6 or above. The function was not present in the initial shipment of Release 6, but was enabled by a subsequent PTF for APAR OW36113. It requires no additional networking hardware or software beyond this release of OS/390.

CS for OS/390 allows multiple TCP/IP stacks to run concurrently with a single VTAM, but the Enterprise Extender function can utilize only one of these at a time. VTAM can change its allegiance from one TCP/IP stack to another, but only when all Enterprise Extender connections have been terminated. A VTAM start option, TCPNAME, allows you to specify which of several stacks VTAM is to use.

If this option is not specified VTAM will choose a suitable stack that supports Enterprise Extender.

If VTAM is to act as an Enterprise Extender node, it must have an IP address associated with it. This IP address is specified using the IPADDR start option, but the default IP address of the chosen stack will be used if the start option is not coded. The IP address needs to be predictable (because partner Enterprise Extender nodes may need to connect to it), but it is not desirable that a single IP port always be used for such connections. Therefore, this address must be a virtual (VIPA) IP address. Moreover, it must be defined as a *source* VIPA address so that datagrams sent out by VTAM have this address on them as the source.

The Enterprise Extender connections themselves are defined to VTAM as switched connections, in a fashion similar to that used for 3172 and open systems adapter (OSA) connections. The practical examples shown later in this chapter illustrate how these definitions are coded. The TCP/IP stack also needs a definition for the port represented by the VTAM application; this is done using a same-host (IUTSAMEH) statement as shown in the examples. This definition must be active before VTAM can establish any Enterprise Extender connections.

2.4.1.1 VTAM Definitions

Figure 9 shows what VTAM statements are required for a predefined Enterprise Extender connection.

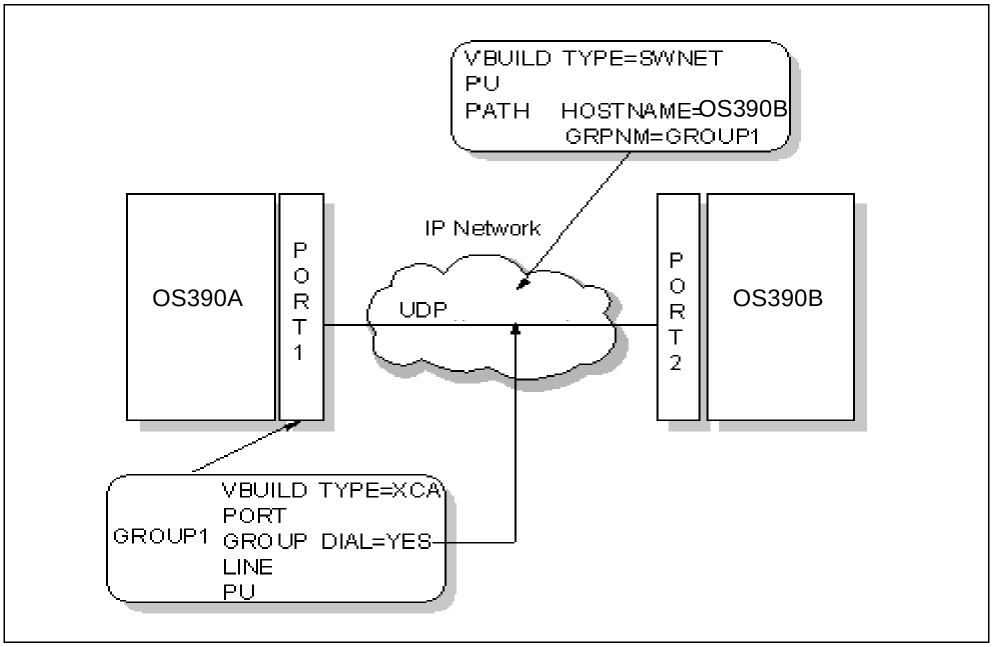


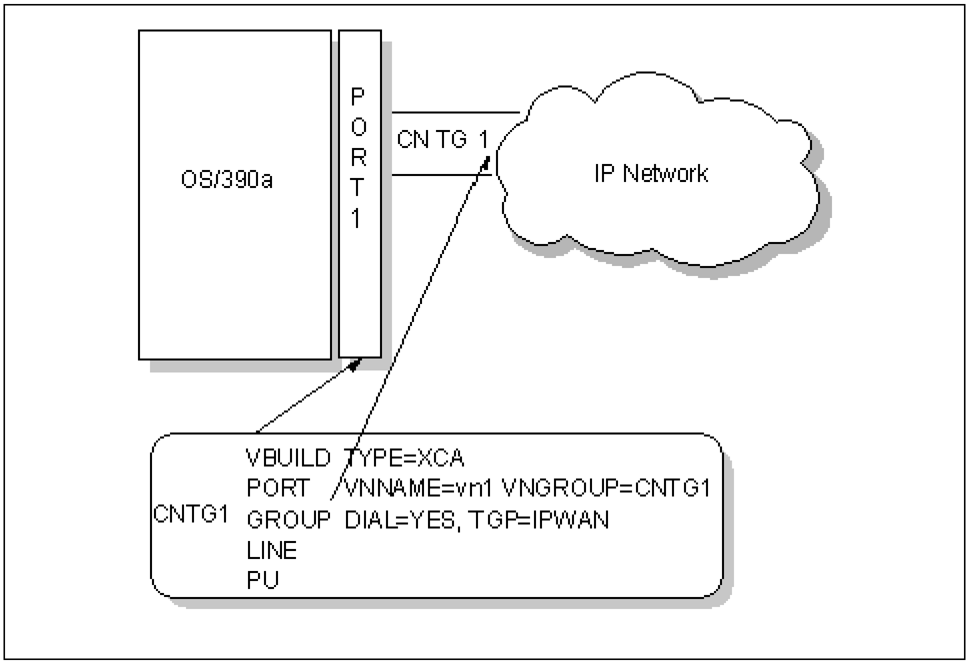
Figure 9. Switched Connection Definition

The external communications adapter (XCA) major node defines the IP port (the connection to the adjoining MVS TCP/IP stack) that VTAM will use for Enterprise Extender connections. Just as with OSA LAN connections, the individual links are defined in switched major nodes, or they can be dynamically defined using the VTAM exits available for this purpose. If VTAM is to initiate the connection then the IP host name (or address, if host name resolution is not available) of the partner Enterprise Extender node must be specified on the PATH statement.

VTAM is able to use the domain name service of TCP/IP just as any other TCP/IP application.

The PU statement in the switched major node may be used to define the APPN TG characteristics of the Enterprise Extender connection, as is true for any APPN link. To cater for Enterprise Extender, two new TG profiles have been provided in CS for OS/390. They are called IPWAN (for a wide-area IP network) and IPCAMPUS (for an IP connection across a LAN). Coding these on the PU statement gives you reasonable TG characteristics for route calculation and ARB flow control purposes without having to determine and code the actual characteristics. By default VTAM assumes IPCAMPUS, so if your Enterprise Extender connection crosses, for example, the Internet then you should code IPWAN which has lower capacity and security, and higher delay, characteristics.

Figure 10 shows the corresponding VTAM definitions required for a connection network. A connection network saves having to define links between every single pair of nodes that can communicate directly across a shared transport facility such as an IP network or a LAN. Within each node, you define just two connections instead of one per potential partner node. The first connection is required for CP-CP sessions (since these cannot flow across a connection network) and the second is to a virtual node that represents the shared transport facility. When the session path is calculated and the RSCV is presented to the primary end of the session, that node recognizes the virtual node in the RSCV and replaces it with a direct connection to its desired partner. For this to work, the address (IP address in the case of Enterprise Extender) of the partner must be present in the RSCV, and therefore in the topology database. Nodes that support connection networks include these addresses in their topology update reporting.



```
      VBUILD TYPE=XCA
      PORT   VNNAME=vn1 VNGROUP=CNTG1
CNTG1 GROUP  DIAL=YES, TGP=IPWAN
      LINE
      PU
```

Figure 10. Connection Network Definitions

Again, the familiar XCA major node is used. This time the VNNAME keyword specifies the name of the virtual node representing the connection network, and VNGROUP points to the group of switched line definitions that will be used for the dynamically defined connections (PUs).

2.4.2 Communications Server for Windows NT

Enterprise Extender was implemented in Version 6 of Communications Server for Windows NT. There is only one TCP/IP stack in Windows NT, and only one Enterprise Extender port can be defined to CS/NT, so the process of defining an Enterprise Extender connection is somewhat simpler than that used in CS for OS/390.

The process consists essentially of defining an SNA connection on a new device type of IBM-EEDLC. This creates the link between CS/NT and the TCP/IP stack. After that you can create a link station for each predefined connection to a remote device, supplying the IP address of the remote partner where a LAN connection would require a MAC address. As with a LAN port, you can allow incoming calls to create link stations dynamically.

CS/NT does not support a connection network over IP in this release. This is planned for late 1999.

2.4.3 The 221X Router Family

Enterprise Extender has been implemented in the following releases of the 221X router family products:

- 3746 Multiaccess Enclosure Extended Functions Part 2, feature code 5805
- 2216 Multiprotocol Access Services, Version 2 Release 2 and above
- 2210 Multiprotocol Routing Services, Version 2 Release 2 and above
- 2212 Access Integration Services
- Network Utility, models TN1 and TX1

In each of these products, Enterprise Extender capability is defined by means of a new APPN port type, IP. Only one such port (logical interface) can be defined, and all Enterprise Extender connections are associated with this “virtual” port. As with all shareable APPN ports, link stations can be defined dynamically for incoming calls, or explicitly for all calls. For explicit definitions the configuration process asks for the remote IP address. The assignment of UDP port numbers can be changed from the default, and the 221X family allows you to specify whether the IP precedence bits will be used for APPN priority *in addition to* the UDP port numbers.

2.5 Enterprise Extender between CS for OS/390 and CS/NT via 3172

Figure 11 on page 35 shows the network configuration that we used in our first testing scenario. We established an Enterprise Extender connection between an OS/390 host and a PC running Communications Server for Windows NT. We then used dependent LU requester to establish sessions between our terminal emulator (PComm/NT) and TSO on the host.

The essential features of the configuration are:

- There are no native SNA connections in the network; the Enterprise Extender link is the only TG in the APPN topology.
- The CS/NT machine is an APPN end node and the OS/390 host is a network node. Aside from the necessity to have at least one NN in an APPN network, the use of DLUR requires that the DLU server is a network node.
- The IP network connecting the two SNA nodes consists of a 2216 router connected to each node by means of a LAN. The 3172 runs ICP and is transparent to the IP network as well as the SNA network.
- The relevant IP addresses are:
  - 192.169.236 is the network connecting the CS/NT machine to the 2216. The PC's interface address is 192.169.236.1.
  - 192.168.221 is the network connecting the 2216 to the host. The host's real interface address is 192.168.221.39.
  - There is a single TCP/IP stack running as part of CS for OS/390 on the host. Its VIPA address has been defined as 192.168.232.39. This is the address that the CS/NT machine will contact; the routing function within MVS TCP/IP will ensure that packets are routed correctly for this address.
  - The real address of the interface between VTAM and MVS TCP/IP is 192.168.232.40. VTAM is not aware of this address; it only knows the VIPA address. Although this address is in the same subnetwork as the VIPA address, this does not cause a problem because both are internal to this CS for OS/390. A VIPA address must never be in the same subnetwork as a real physical interface address.
- The APPN CP names are USIBMRA.RA39M for the OS/390 host and USIBMRA.WTR05212 for the CS/NT PC.

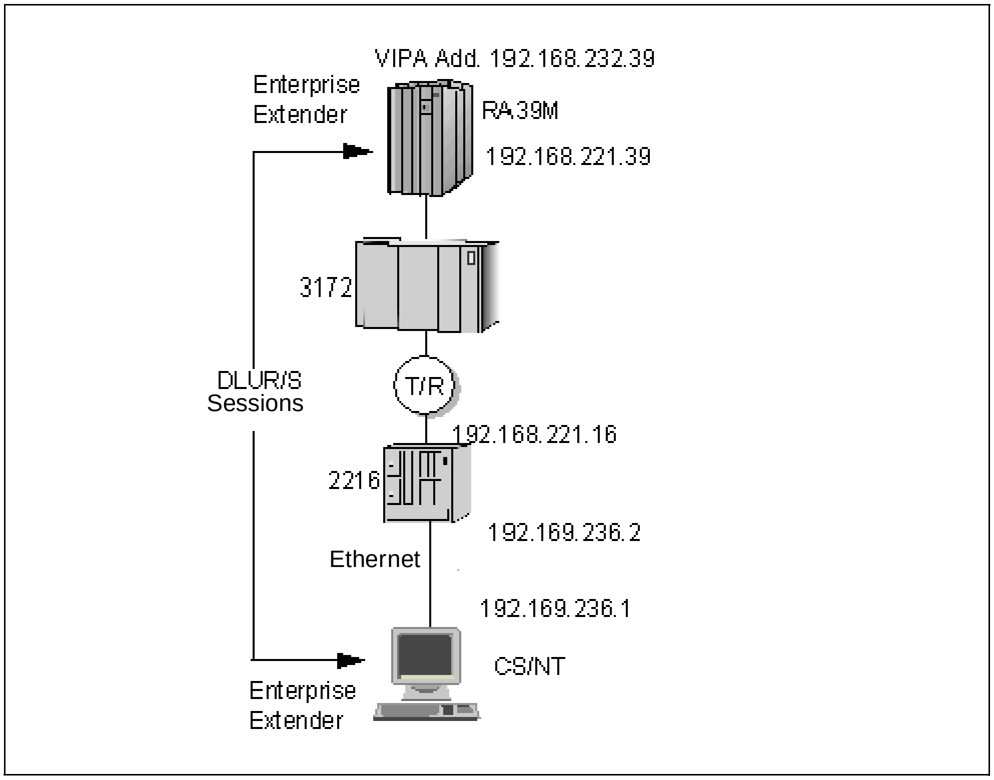
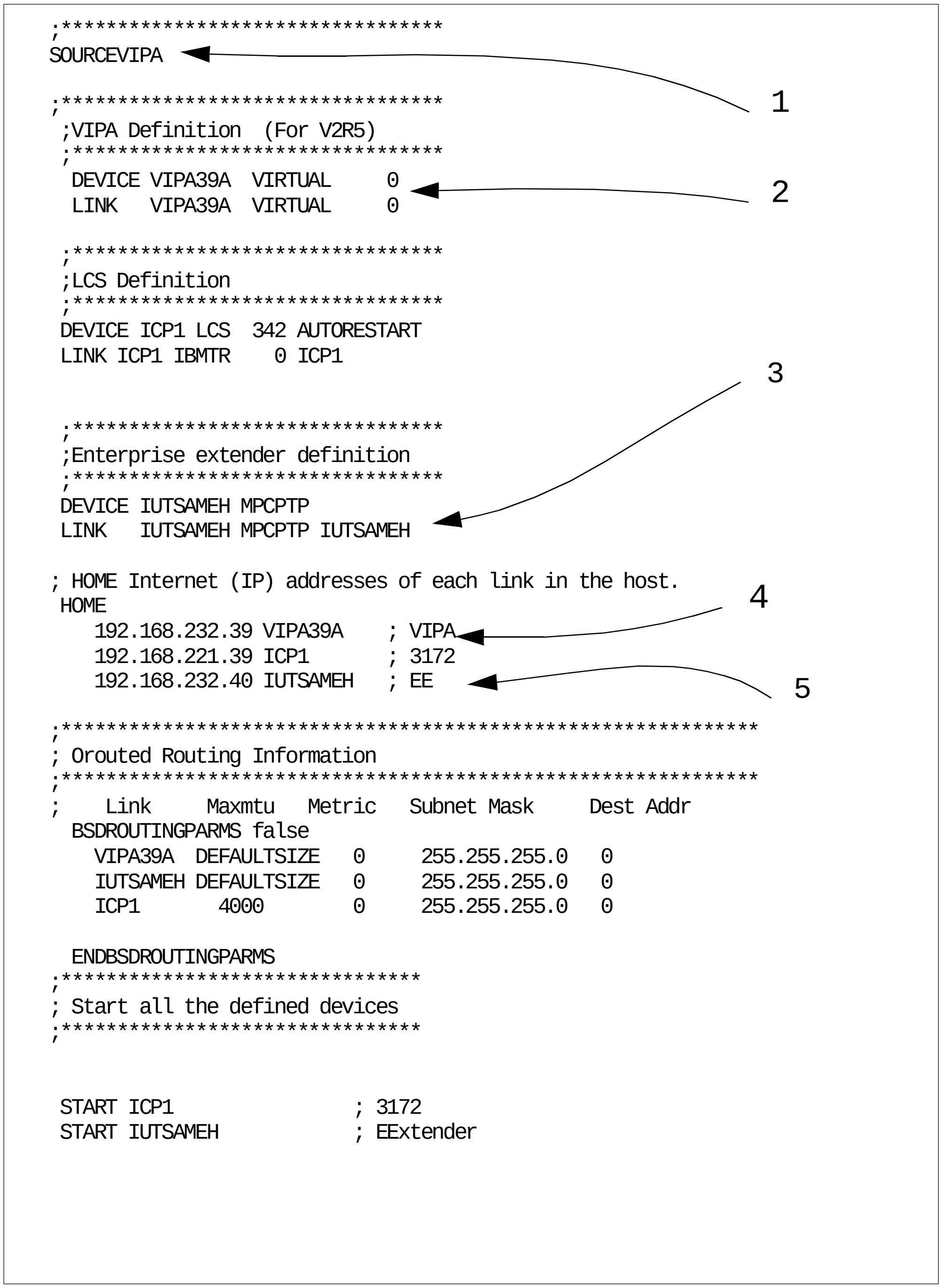


Figure 11. Enterprise Extender between OS/390 and CS/NT

In this section we do not show the configurations of the 2216 and 3172, as they are not relevant to the use of Enterprise Extender. We had to configure both parts of CS for OS/390 (VTAM and TCP/IP) and CS/NT to get this configuration working.

2.5.1 MVS TCP/IP Definitions

Because TCP/IP underlies SNA in an Enterprise Extender configuration, we established the TCP/IP connections before defining the VTAM ones. Figure 12 on page 36 shows the TCP/IP profile that we used on the OS/390 host.



```
;*********************************
SOURCEVIPA

;*********************************
 ;VIPA Definition  (For V2R5)
 ;*********************************
  DEVICE VIPA39A  VIRTUAL     0
  LINK   VIPA39A  VIRTUAL     0

 ;*********************************
 ;LCS Definition
 ;*********************************
 DEVICE ICP1 LCS  342 AUTORESTART
 LINK ICP1 IBMTR    0 ICP1


 ;*********************************
 ;Enterprise extender definition
 ;*********************************
 DEVICE IUTSAMEH MPCPTP
 LINK   IUTSAMEH MPCPTP IUTSAMEH

; HOME Internet (IP) addresses of each link in the host.
 HOME
    192.168.232.39 VIPA39A    ; VIPA
    192.168.221.39 ICP1       ; 3172
    192.168.232.40 IUTSAMEH   ; EE

;**************************************************************
; Orouted Routing Information
;**************************************************************
;    Link     Maxmtu   Metric   Subnet Mask     Dest Addr
  BSDROUTINGPARMS false
    VIPA39A  DEFAULTSIZE   0     255.255.255.0   0
    IUTSAMEH DEFAULTSIZE   0     255.255.255.0   0
    ICP1       4000        0     255.255.255.0   0

  ENDBSDROUTINGPARMS
;*******************************
; Start all the defined devices
;*******************************


 START ICP1               ; 3172
 START IUTSAMEH           ; EExtender
```

Figure 12. MVS TCP/IP Definitions

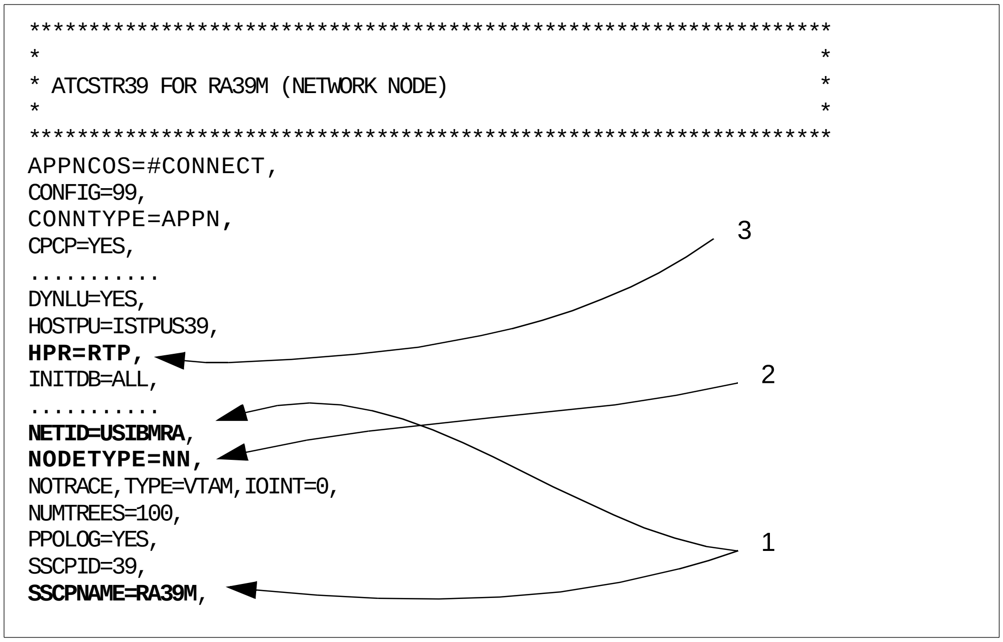
In the TCP/IP profile we coded the following parameters:

1. SOURCEVIPA tells TCP/IP to return the VIPA address as the source address on all outbound datagrams. It also means that a client program on this MVS image (in our case, VTAM is that client) will use the VIPA address as its local address. This provides for maximum resilience against the failure of any one physical interface. SOURCEVIPA is mandatory for Enterprise Extender connections.

2. The VIPA address needs a virtual device and link defined; we have named it VIPA39A. The VIPA address looks like an interface to a virtual network that is concealed behind this MVS TCP/IP acting as a router.
3. The Enterprise Extender connection to VTAM also requires a device and link definition. It uses the same interface as to another TCP/IP stack on the same MVS image, known as Samehost. TCP/IP recognizes the name IUTSAMEH as being a Samehost interface, and treats it in a similar fashion to an MPC connection (hence the device type MPCPTP). TCP/IP uses VTAM’s common DLC connection manager for communication across MPC links; in the case of the Samehost interface VTAM will define a TRLE dynamically for use by TCP/IP.
4. We assign the address 192.168.232.39 to the VIPA interface.
5. Similarly, we give the address 192.168.232.40 to the Samehost Enterprise Extender interface.

2.5.2 VTAM Definitions

Once MVS TCP/IP was working to our satisfaction, we defined the VTAM APPN node RA39M. Figure 13 on page 37 shows an extract from our VTAM start options.



```
*******************************************************************
*                                                                 *
* ATCSTR39 FOR RA39M (NETWORK NODE)                               *
*                                                                 *
*******************************************************************
APPNCOS=#CONNECT,
CONFIG=99,
CONNTYPE=APPN,
CPCP=YES,
...........
DYNLU=YES,
HOSTPU=ISTPUS39,
HPR=RTP,
INITDB=ALL,
...........
NETID=USIBMRA,
NODETYPE=NN,
NOTRACE,TYPE=VTAM,IOINT=0,
NUMTREES=100,
PPOLOG=YES,
SSCPID=39,
SSCPNAME=RA39M,
```

Figure 13. VTAM Start Options

The important definitions here are:

1. The NETID and SSCPNAME options define the APPN control point name of this node.
2. The NODETYPE option must be coded as NN if this VTAM is to act as a network node server or a dependent LU server for other nodes. An Enterprise Extender connection with dependent LU sessions will work just as well to a VTAM end node that does not have to act as one of these.
3. HPR=RTP must be specified (it is in fact the default) in order for Enterprise Extender to work. The value RTP also implies support for the control flows option.

Next, we defined the Enterprise Extender port itself. This is done using an XCA major node as shown in Figure 14.

```
EEXCA    VBUILD TYPE=XCA
EEXCAP   PORT  MEDIUM=HPRIP,SAPADDR=4
EEXCAG   GROUP DIAL=YES
EEXCAL   LINE  CALL=INOUT
EEXCAP   PU
```

Figure 14. XCA Major Node for Enterprise Extender

The keyword MEDIUM=HPRIP tells VTAM that this is an Enterprise Extender connection rather than a LAN or ATM connection. Since Enterprise Extender connections use switched protocols between APPN nodes, the LINE definitions specify DIAL=YES. In our example we had only one connection active, but multiple LINE statements may be defined if VTAM is to maintain multiple connections.

The SAP address (SAPADDR) differs in usage from that on a real local area network. If you require parallel TGs between the same two MAC addresses on a real LAN, you must distinguish between them by the use of SAPs, which are effectively sub-addresses. Each node assigns its own SAP address(es) to its interface, and the partner node must be aware of them. Each TG is distinguished by its local/remote SAP pair.

On an Enterprise Extender connection, SAPs are also used to distinguish between parallel TGs, between the same *IP* addresses rather than the same MAC addresses. However, the SAP addresses for *both* ends of a connection are defined by the node which establishes the connection, and the partner node merely accepts them. The SAP defined at the partner node is used only for connections initiated at that node. Therefore, if you plan to use parallel TGs between Enterprise Extender nodes you must ensure that the local/remote SAP pairs are not duplicated. A good scheme is to allow all the local SAP definitions to default to 4 and to code (on the switched major node PATH statements) values other than 4 for the dial-out connections. In this way the SAP pair (4,4) never occurs; all outbound connections have SAP pair (4,x) and all inbound connections have SAP pair (x,4). Note that with Enterprise Extender the SAP values can be multiples of two rather than four.

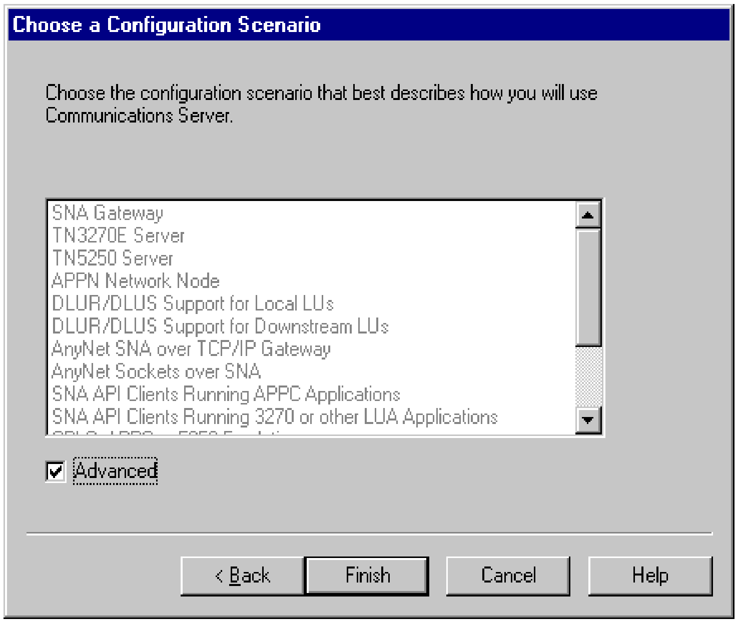
We established the connection from the CS/NT node, so we did not need a switched major node for this exercise. It was, in fact, defined dynamically by the configuration services exit. If VTAM is to dial the connection itself, however, a switched major node is required. This is similar to that used for a LAN connection except that:

- The address of the remote node in the PATH statement must be specified in IP terms, as either IPADDR (the IP address) or HOSTNAME (the host name to be resolved by the name server).
- The remote SAP address should be specified in SAPADDR on the PATH statement, to identify the connection uniquely. If parallel connections are not to be used then the SAPADDR can be left to default to 4.

2.5.3 CS/NT Configuration

In CS/NT we had to configure Enterprise Extender, and also dependent LU requester because we wished to establish emulator sessions to TSO. On an HPR-only connection DLUR is the only way to achieve this.

When you invoke the SNA node configuration in CS/NT, the first thing it asks you (Figure 15) is to choose the scenario that most closely resembles your setup. Your response determines which configuration options will be presented on the main panel. If you choose **Advanced**, you get all the options which is what we did.



Choose a Configuration Scenario

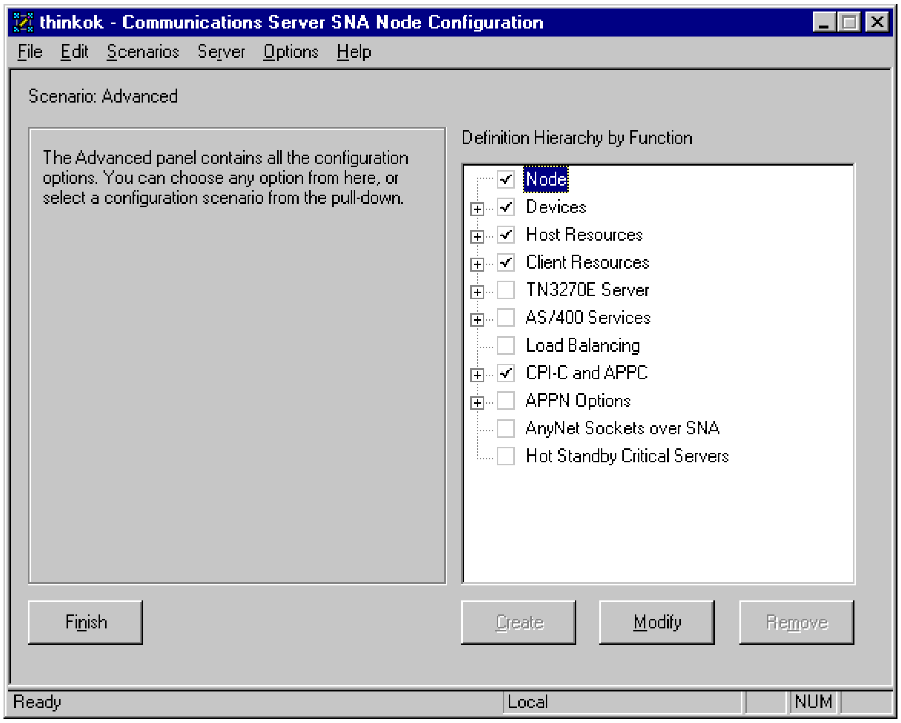
Choose the configuration scenario that best describes how you will use Communications Server.

SNA Gateway
TN3270E Server
TN5250 Server
APPN Network Node
DLUR/DLUS Support for Local LUs
DLUR/DLUS Support for Downstream LUs
AnyNet SNA over TCP/IP Gateway
AnyNet Sockets over SNA
SNA API Clients Running APPC Applications
SNA API Clients Running 3270 or other LUA Applications

☑ Advanced

Figure 15. Configuration Scenario Choice

Now the configuration main panel is displayed, as seen in Figure 16 on page 40.



thinkok - Communications Server SNA Node Configuration
File
Edit
Scenarios
Server
Options
Help
Definition Hierarchy by Function
Finish
Create
Modify
Remove
Ready
Local
NUM

Scenario: Advanced

The Advanced panel contains all the configuration options. You can choose any option from here, or select a configuration scenario from the pull-down.

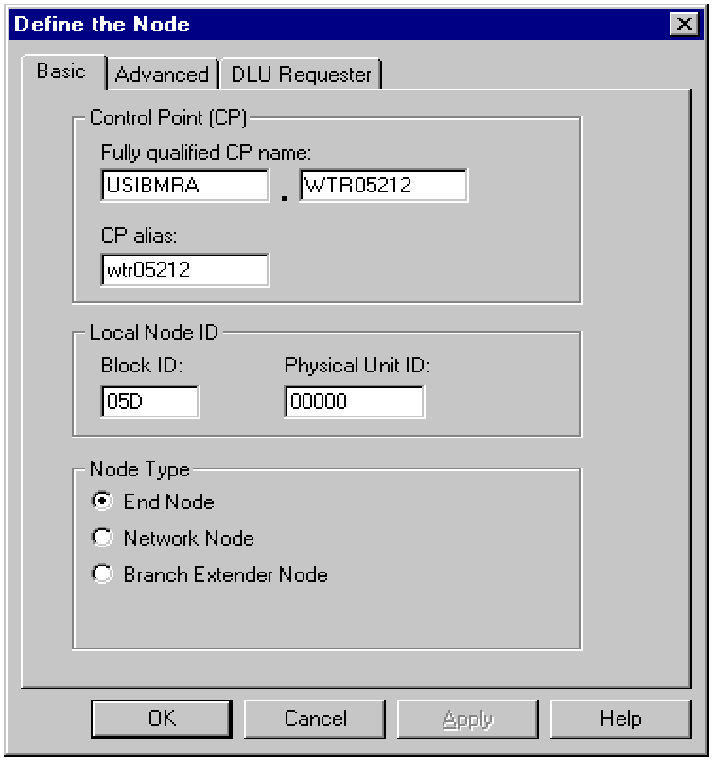
- ☑ Node
- ☑ Devices
- ☑ Host Resources
- ☑ Client Resources
- □ TN3270E Server
- □ AS/400 Services
- □ Load Balancing
- ☑ CPI-C and APPC
- □ APPN Options
- □ AnyNet Sockets over SNA
- □ Hot Standby Critical Servers

Figure 16. CS/NT Configuration Main Panel

In this simple configuration, we had to define the following:

- The APPN node
- The Enterprise Extender port, reached from the **Devices** option
- The Enterprise Extender connection to VTAM, reached from the **CPI-C and APPC** option
- The dependent LUs and PU for the emulator sessions, reached from the **Host Resources** option

Selecting **Node** and then **Create** (or **Modify**, if you are changing an existing configuration) gives the panel shown in Figure 17.


Define the Node

Basic | Advanced | DLU Requester

Control Point (CP)

Fully qualified CP name: USIBMRA . WTR05212

CP alias: wtr05212

Local Node ID

Block ID: 05D

Physical Unit ID: 00000

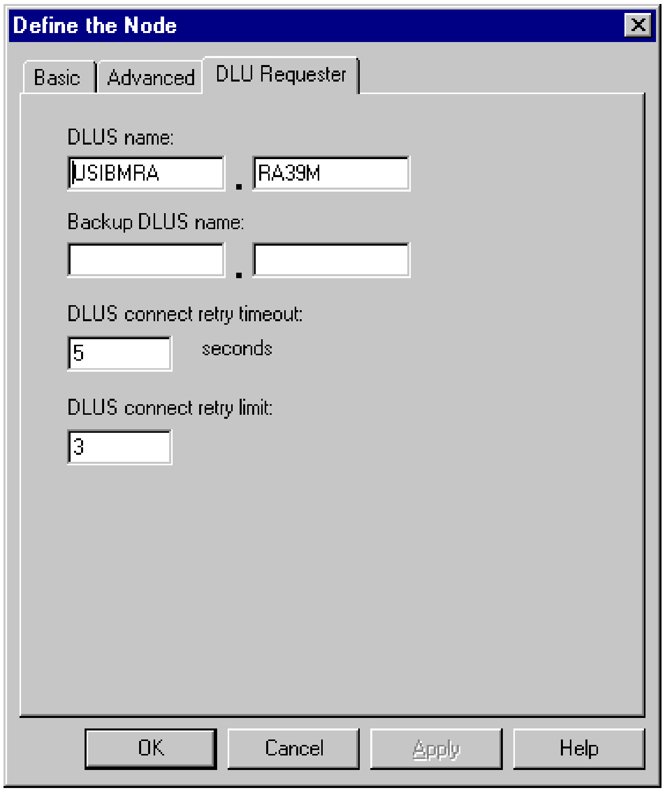
Node Type

● End Node
○ Network Node
○ Branch Extender Node

Figure 17. Basic Node Definition

We filled in the name of this node (USIBMRA.WTR05212) and the node ID (05D-00000). We also selected **End Node** as shown. The node ID would be used by our configuration services exit to define a link station (PU) to VTAM, since we did not have a suitable switched major node coded.

Next, we selected the **Advanced** tab on the configuration panel to define the default DLUR characteristics. These definitions will be used for all dependent LUs unless specifically overridden on the LU/PU configuration panels. Figure 18 on page 42 illustrates this.



Define the Node

DLUS name: USIBMRA . RA39M

Backup DLUS name: .

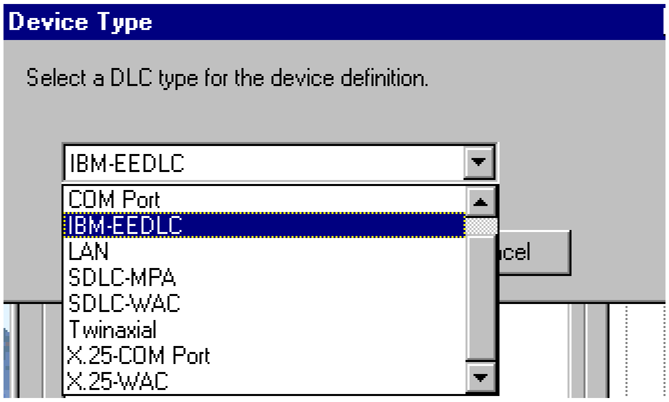
DLUS connect retry timeout: 5 seconds

DLUS connect retry limit: 3

Figure 18. DLU Requester Default Parameters

The DLUR parameters include the primary DLU server name (our VTAM NN, USIBMRA.RA39M) and the backup DLU server (not specified).

Next, we defined the Enterprise Extender port by selecting **Devices** and **Create**. The correct selection is IBM-EEDLC as shown in Figure 19.



Device Type

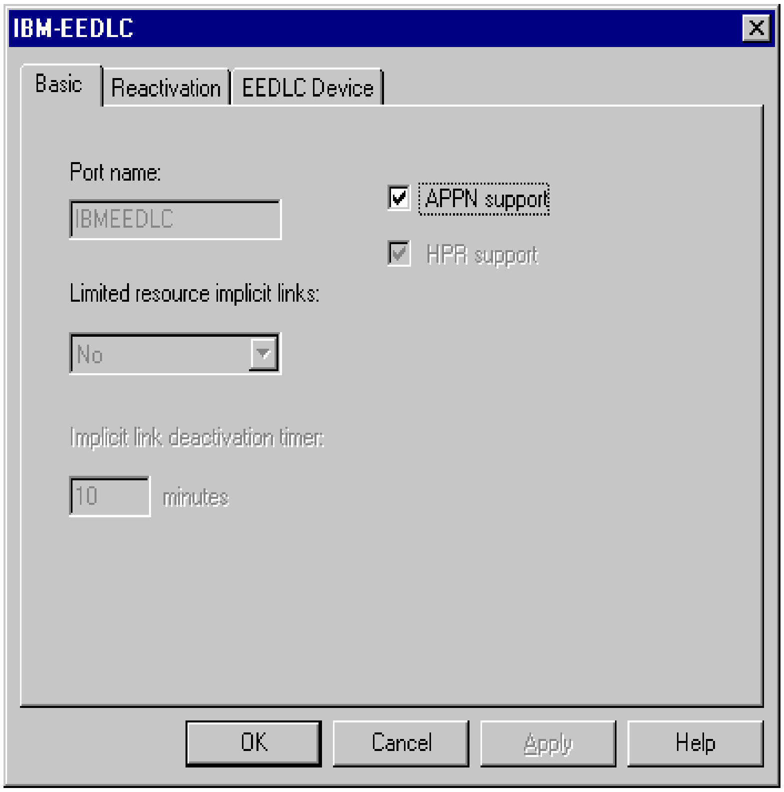
Select a DLC type for the device definition.

IBM-EEDLC

Figure 19. Device (Port) Choices

Selecting the IBM-EEDLC device type gives us Figure 20 on page 43. We allowed all the defaults to remain in force on this set of three panels. The option **APPN support** is misnamed; it actually means CP-CP session support, which we required in this case because the connection over this port was to our network node server.

Note that the name given to the port is always IBMEEDLC; CS/NT does not allow multiple Enterprise Extender ports to be defined.



IBM-EEDLC

Port name: IBMEEDLC

☑ APPN support

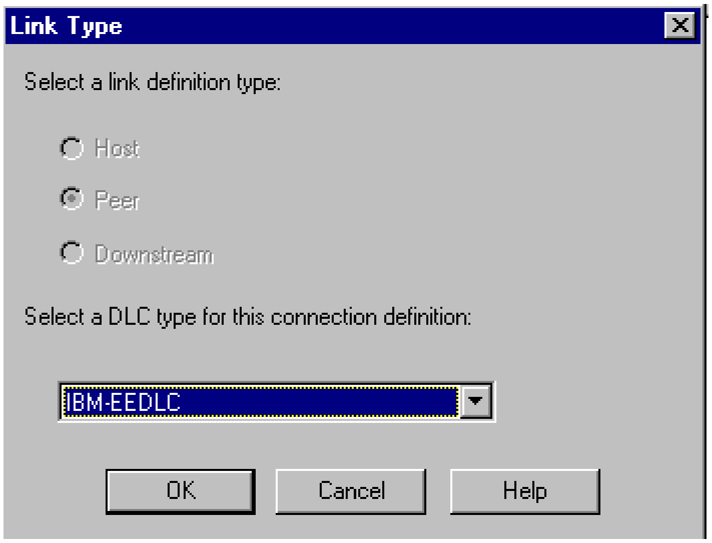
☑ HPR support

Limited resource implicit links: No

Implicit link deactivation timer: 10 minutes

Figure 20. Enterprise Extender DLC Configuration

Our next task was to define the Enterprise Extender connection (link station) itself. We selected **CPI-C and APPC** then **Peer Connections** and **Create** to see the panel shown in Figure 21 on page 44. The **CPI-C and APPC** box is already checked on a new configuration, since the modes and the APINGD transaction program are predefined.



Link Type

Select a link definition type:

○ Host

● Peer

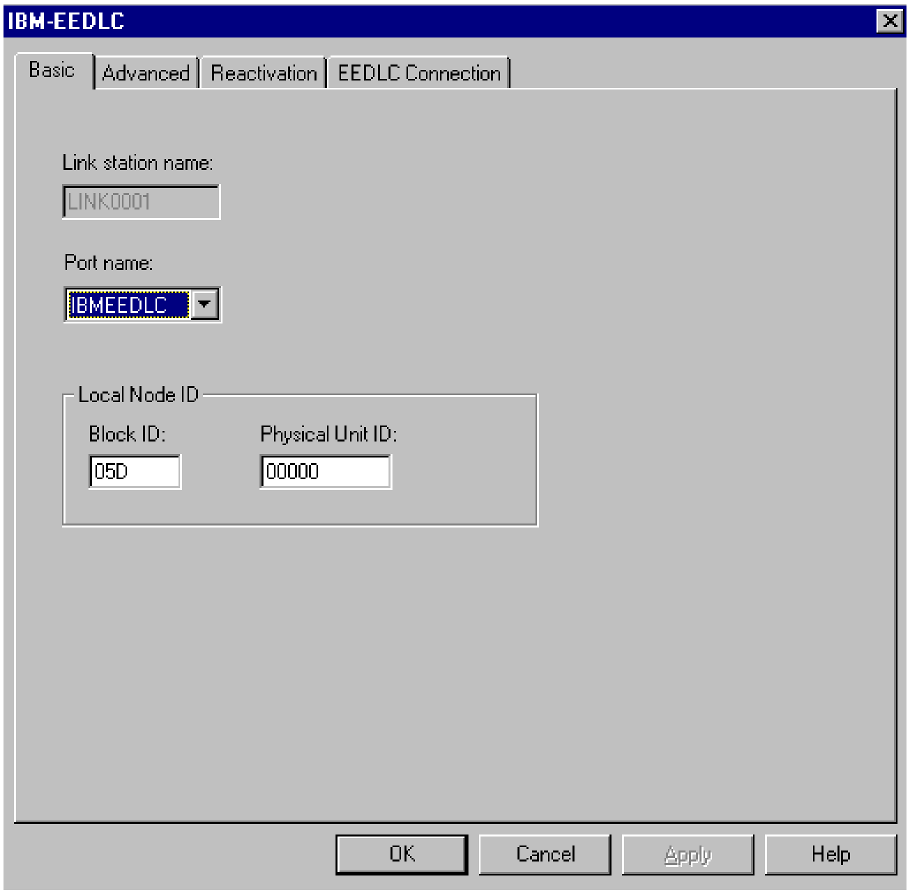
○ Downstream

Select a DLC type for this connection definition:

IBM-EEDLC

Figure 21. Link Station Type for Enterprise Extender

In this panel **Peer** was preselected because this is purely an APPN connection. **Host** means that dependent LUs are supported natively on this link (impossible with Enterprise Extender) and **Downstream** means a link to nodes supported by this node as a branch extender (impossible on an end node). Clicking **OK** brought us at last to the link station definition panel shown in Figure 22 on page 45.



Link station name: LINK0001

Port name: IBMEEDLC

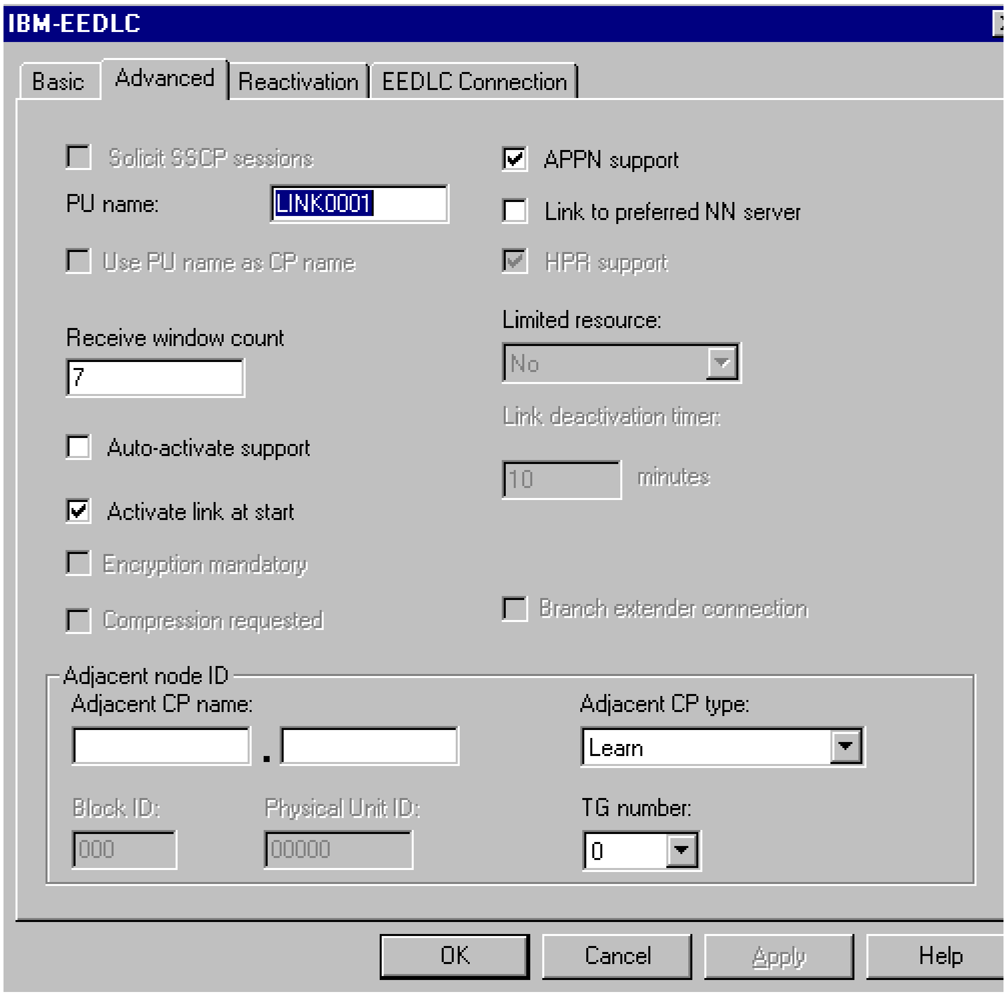
Local Node ID

Block ID: 05D

Physical Unit ID: 00000

Figure 22. Link Station Definition (Basic Tab)

The **Basic**, **Advanced** (Figure 23 on page 46) and **Reactivation** tabs were left alone to default. The Local Node ID on the **Basic** tab is used by VTAM to identify the link station; in this case the configuration services exit would dynamically define a PU name of W00000 from this information. The **Advanced** tab can be used to override some of the port definitions (such as CP-CP session support) as well as to control what kind of partner node you will allow on this connection. We were happy for CS/NT to learn about its partner dynamically.



□ Solicit SSCP sessions

PU name: LINK0001

□ Use PU name as CP name

Receive window count: 7

□ Auto-activate support

☑ Activate link at start

□ Encryption mandatory

□ Compression requested

☑ APPN support

□ Link to preferred NN server

☑ HPR support

Limited resource: No

Link deactivation timer: 10 minutes

□ Branch extender connection

Adjacent node ID

Adjacent CP name: .

Adjacent CP type: Learn

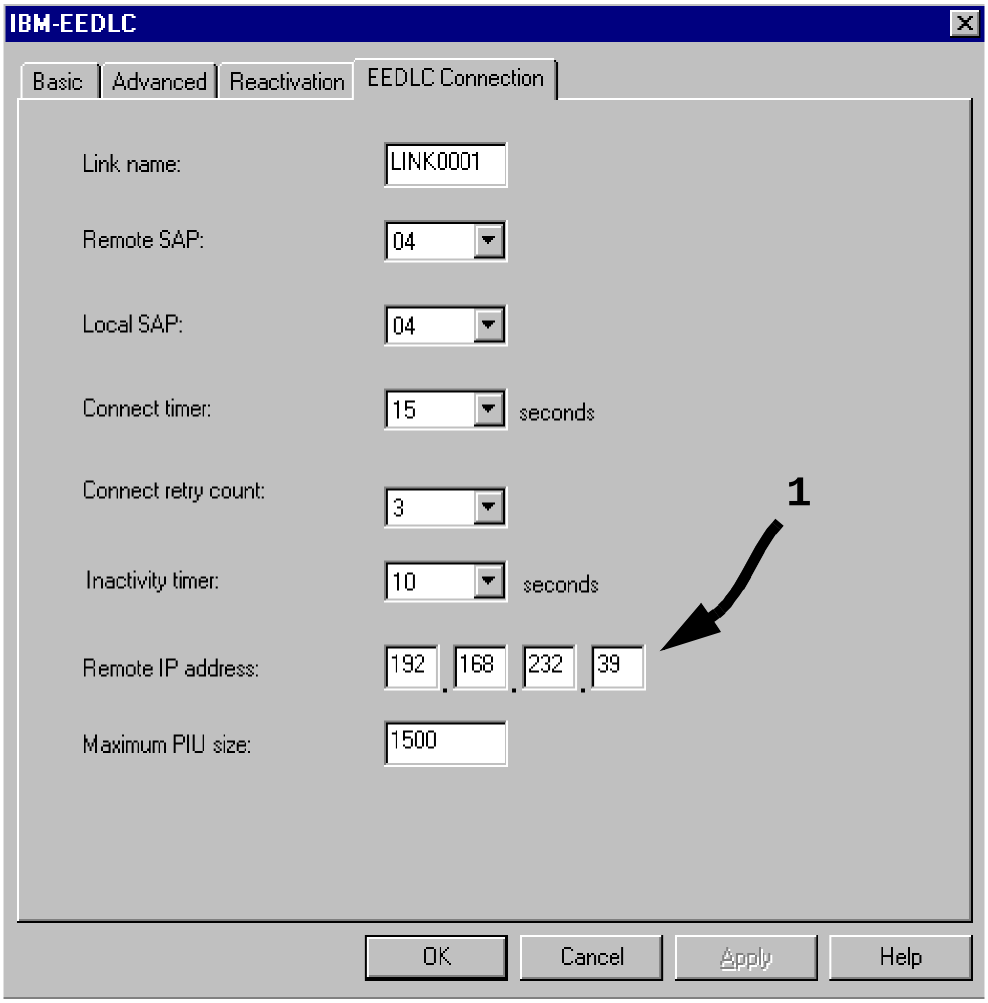
Block ID: 000

Physical Unit ID: 00000

TG number: 0

Figure 23. Link Station Configuration (Advanced Tab)

Finally, we moved to the **EEDLC Connection** tab (Figure 24 on page 47) where we were, at last, able to define the partner's IP address at (1). This is the VIPA address of the TCP/IP stack on the RA39M host.



Link name: LINK0001

Remote SAP: 04

Local SAP: 04

Connect timer: 15 seconds

Connect retry count: 3

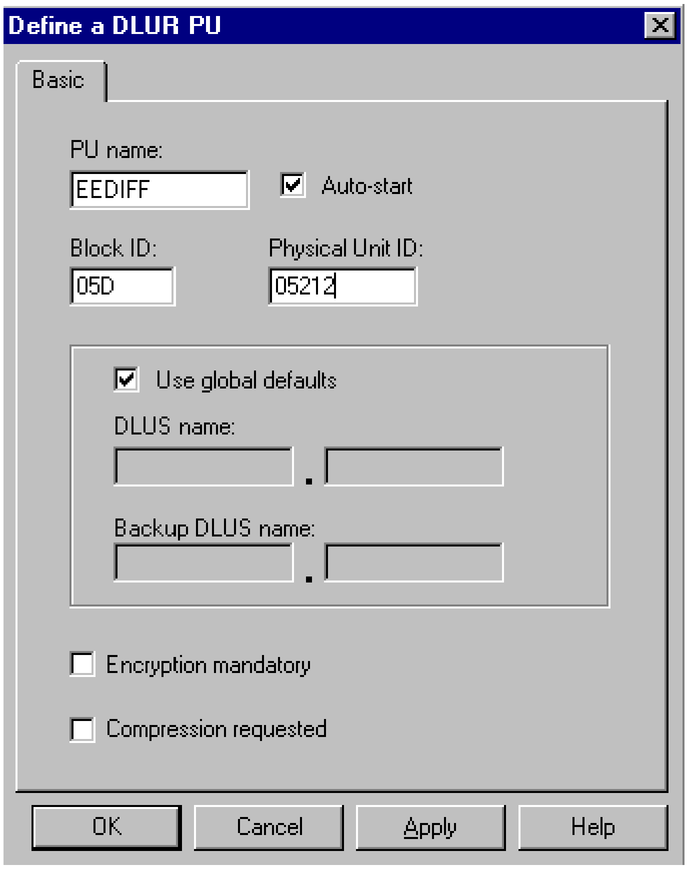
Inactivity timer: 10 seconds

Remote IP address: 192 . 168 . 232 . 39

Maximum PIU size: 1500

Figure 24. Link Station Definition (EEDLC Connection Tab)

Now our connection definition is complete, but we can only set up independent LU sessions such as that for APING. To allow dependent LU logon to TSO we must define some dependent resources to be used by DLUR. This we did by selecting **Host Resources**, **DLUR PUs** and **Create** to see Figure 25 on page 48.



PU name: EEDIFF

☑ Auto-start

Block ID: 05D

Physical Unit ID: 05212

☑ Use global defaults

DLUS name:

Backup DLUS name:

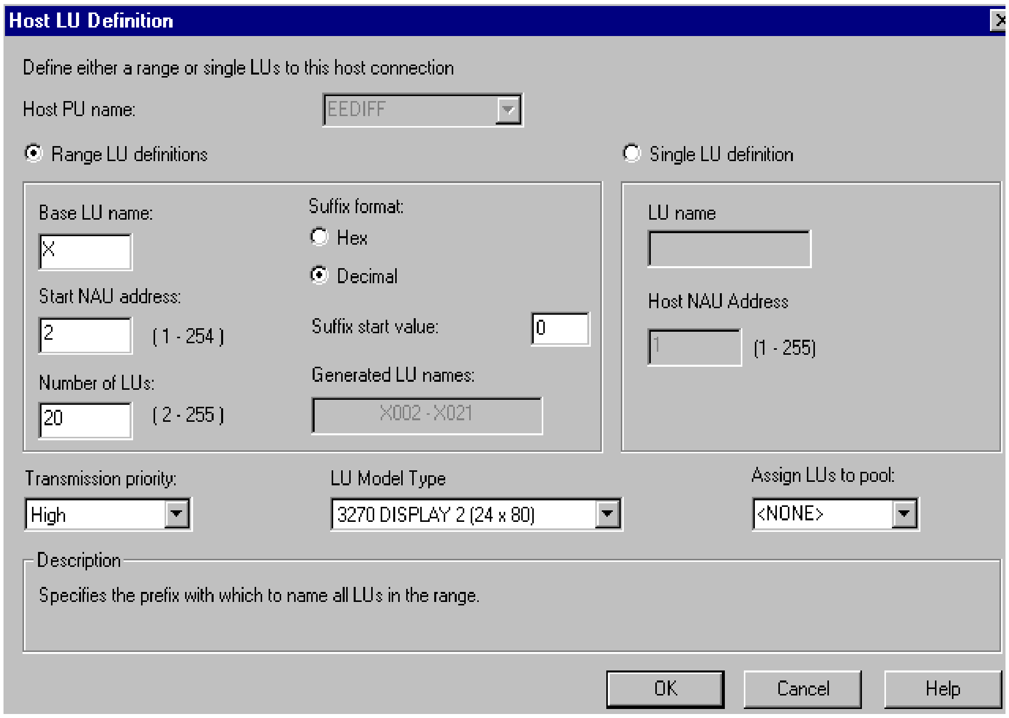
□ Encryption mandatory

□ Compression requested

Figure 25. DLUR PU Definition

Here we allowed the DLU server definition to default from the node-level definition that we entered in Figure 18 on page 42. However, we had to give VTAM a new node ID from which to identify the dependent resources. This particular node is adjacent to VTAM (the Enterprise Extender connection is a single APPN hop), so VTAM must be able to distinguish between the real link station (which will be W00000) and the dependent PU (which will now be W05212) with its dependent LUs. We entered 05D/05212 in the node ID field as shown.

When you create a new PU on CS/NT, it asks, “Do you wish to create and assign new LUs to this connection?”. We replied Yes and were then able to define the dependent emulation LUs as shown in Figure 26 on page 49. We defined 20 24x80 emulator sessions with local addresses 2-21.


Host LU Definition

Define either a range or single LUs to this host connection

Host PU name: EEDIFF

● Range LU definitions

○ Single LU definition

Base LU name: X

Start NAU address: 2 (1 - 254)

Number of LUs: 20 (2 - 255)

Suffix format:

○ Hex

● Decimal

Suffix start value: 0

Generated LU names: X002 - X021

LU name

Host NAU Address 1 (1 - 255)

Transmission priority: High

LU Model Type 3270 DISPLAY 2 (24 x 80)

Assign LUs to pool: <NONE>

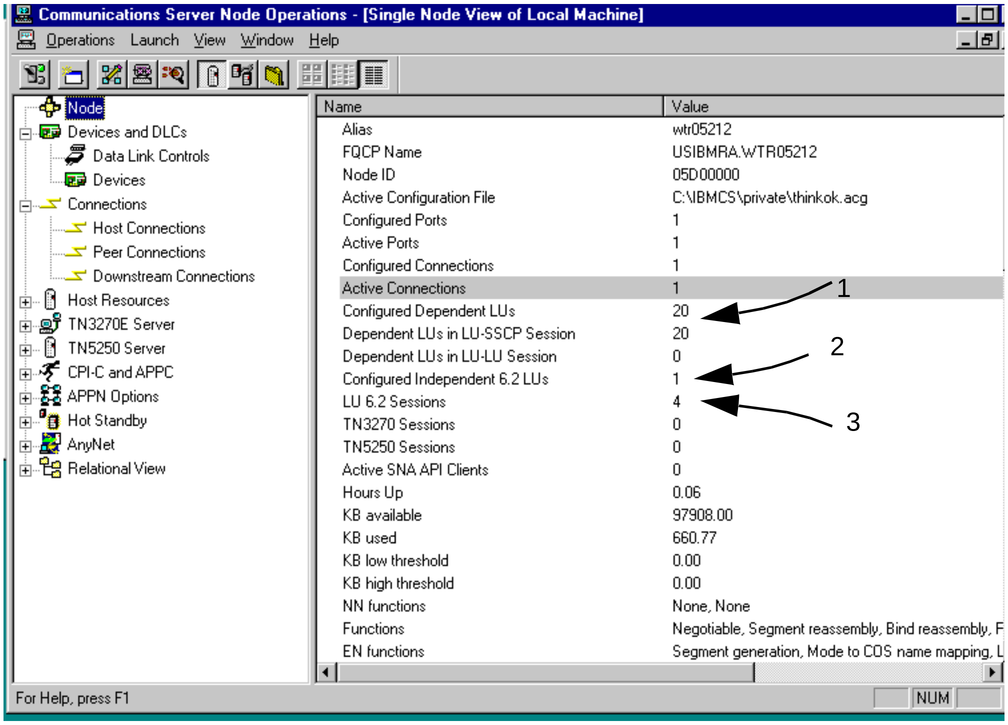
Description

Specifies the prefix with which to name all LUs in the range.

Figure 26. Host LU Definition Panel

2.5.4 Displays

Now the definitions are complete. We started the CS/NT node and activated the XCA major node on RA39M. Figure 27 on page 50 shows the resulting display on the CS/NT node when we invoked SNA Node Operations.



Name	Value
Alias	wtr05212
FQCP Name	USIBMRA.WTR05212
Node ID	05D00000
Active Configuration File	C:\IBMCS\private\thinkok.acg
Configured Ports	1
Active Ports	1
Configured Connections	1
Active Connections	1
Configured Dependent LUs	20
Dependent LUs in LU-SSCP Session	20
Dependent LUs in LU-LU Session	0
Configured Independent 6.2 LUs	1
LU 6.2 Sessions	4
TN3270 Sessions	0
TN5250 Sessions	0
Active SNA API Clients	0
Hours Up	0.06
KB available	97908.00
KB used	660.77
KB low threshold	0.00
KB high threshold	0.00
NN functions	None, None
Functions	Negotiable, Segment reassembly, Bind reassembly, F
EN functions	Segment generation, Mode to COS name mapping, L

Figure 27. Node Display on CS/NT

Note that:

1. There are 20 dependent LUs with SSCP-LU sessions, in other words they have been activated by VTAM.
2. The single independent LU is the control point USIBMRA.WTR05212.
3. The four LU 6.2 sessions are the two CP-CP sessions and the two DLUR/S sessions, all between WTR05212 and RA39M.

Now we were able to log on to TSO (or, indeed, any application in the SNA network) and do some useful work.

We displayed some of the new Enterprise Extender-related resources from VTAM RA39M. Figure 28 shows VTAM's view of the XCA major node.

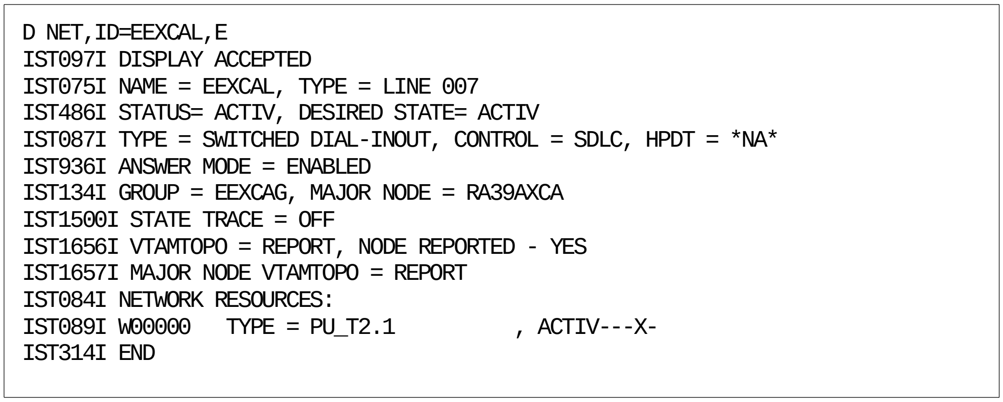
```
D NET,E,ID=RA39AXCA
IST097I DISPLAY ACCEPTED
IST075I NAME = RA39AXCA, TYPE = XCA MAJOR NODE 024              1
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST1679I MEDIUM = HPRIP
IST1685I TCP/IP JOB NAME = T39ATCP                                2
IST1680I LOCAL IP ADDRESS 192.168.232.39
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF                       3
IST1656I VTAMTOPO = REPORT, NODE REPORTED - YES
IST170I LINES:
IST232I EEXCAL    ACTIV
IST314I END
```

Figure 28. XCA Major Node Display

Information unique to Enterprise Extender here comprises:

1. MEDIUM=HPRIP as opposed to RING, CSMACD or ATM
2. The TCP/IP job name
3. The local (VIPA) IP address

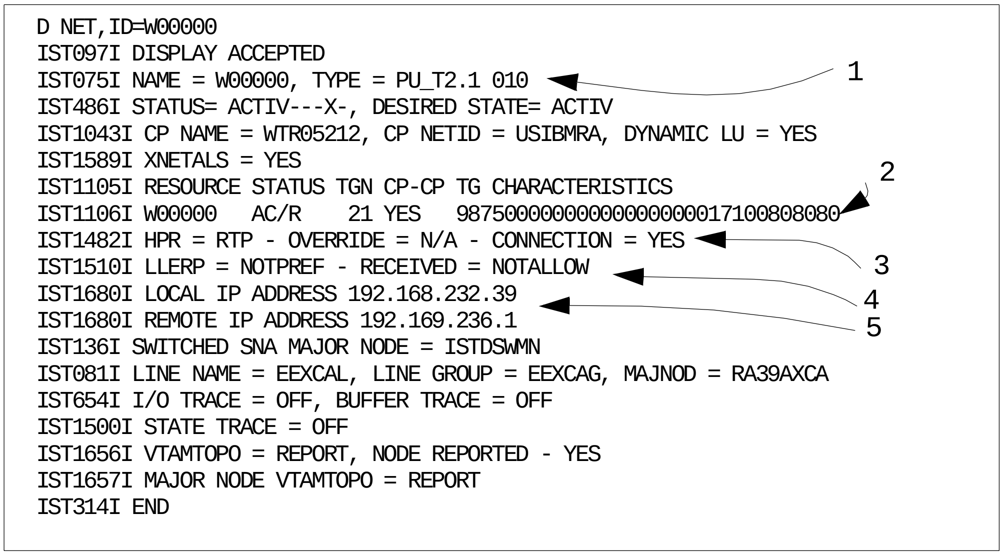
A display of the line EEXCAL shows nothing new (Figure 29). EEXCAL is a logical line (like a LAN line) that serves merely as an anchor point for the link station.



```
D NET,ID=EEXCAL,E
IST097I DISPLAY ACCEPTED
IST075I NAME = EEXCAL, TYPE = LINE 007
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST087I TYPE = SWITCHED DIAL-INOUT, CONTROL = SDLC, HPDT = *NA*
IST936I ANSWER MODE = ENABLED
IST134I GROUP = EEXCAG, MAJOR NODE = RA39AXCA
IST1500I STATE TRACE = OFF
IST1656I VTAMTOPO = REPORT, NODE REPORTED - YES
IST1657I MAJOR NODE VTAMTOPO = REPORT
IST084I NETWORK RESOURCES:
IST089I W00000   TYPE = PU_T2.1          , ACTIV---X-
IST314I END
```

Figure 29. Enterprise Extender Logical Line Display

The link station itself is shown in Figure 30.



```
D NET,ID=W00000
IST097I DISPLAY ACCEPTED
IST075I NAME = W00000, TYPE = PU_T2.1 010
IST486I STATUS= ACTIV---X-, DESIRED STATE= ACTIV
IST1043I CP NAME = WTR05212, CP NETID = USIBMRA, DYNAMIC LU = YES
IST1589I XNETALS = YES
IST1105I RESOURCE STATUS TGN CP-CP TG CHARACTERISTICS
IST1106I W00000   AC/R    21 YES   98750000000000000000017100808080
IST1482I HPR = RTP - OVERRIDE = N/A - CONNECTION = YES
IST1510I LLERP = NOTPREF - RECEIVED = NOTALLOW
IST1680I LOCAL IP ADDRESS 192.168.232.39
IST1680I REMOTE IP ADDRESS 192.169.236.1
IST136I SWITCHED SNA MAJOR NODE = ISTDSWMN
IST081I LINE NAME = EEXCAL, LINE GROUP = EEXCAG, MAJNOD = RA39AXCA
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST1656I VTAMTOPO = REPORT, NODE REPORTED - YES
IST1657I MAJOR NODE VTAMTOPO = REPORT
IST314I END
```

Figure 30. Enterprise Extender Link Station Display

This display shows that:

1. The link station is a PU type 2.1 (APPN) connection.
2. The link station is in the APPN topology database as TG 21.
3. The level of HPR supported is RTP, as it must be.
4. Link-level error recovery is not permitted. With UDP as the logical link this is impossible in any case.

5. The local IP address is the VIPA address of the TCP/IP stack being used by this VTAM, and the remote IP address is that of the workstation.

2.6 Enterprise Extender between CS for OS/390 and CS/NT via 2216

Figure 31 shows the second scenario that we tested. This time the 3172 is missing and the 2216 is directly connected to the host via an ESCON MPC channel.

The observant reader will notice that, in terms of Enterprise Extender, this configuration is absolutely identical to the previous one, and the definitions for that connection will be identical. We include it here because of its relevance to another aspect of SNA and IP integration: the sharing of MPC links between SNA and TCP/IP.

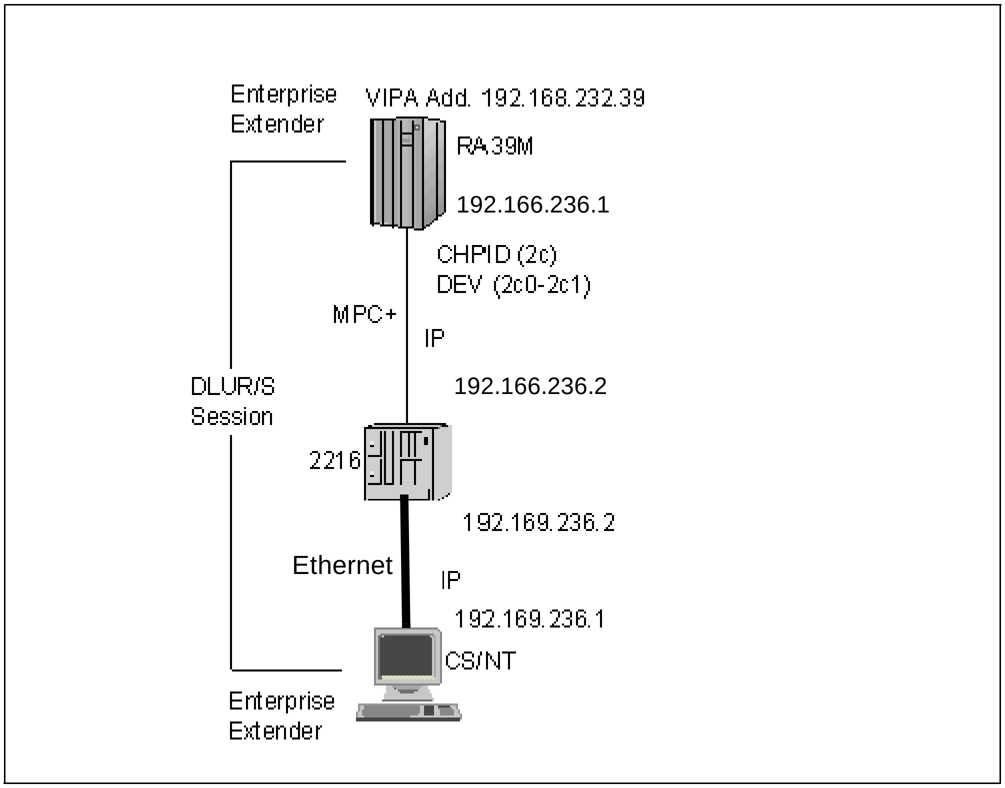


Figure 31. Enterprise Extender Connection via 2216

The main differences between this configuration and the previous one are:

- The MPC channel has addresses of 2C0 (read) and 2C1 (write) on the RA39M image.
- The IP subnetwork comprising the MPC connection is 192.166.236. The interfaces on this subnetwork are 192.166.236.1 (the host) and 192.166.236.2 (the 2216).

2.6.1 VTAM and IOCP Definitions

Our VTAM and CS/NT definitions were the same as before, with the one major exception that we had to define the 2216 MPC connection to VTAM. This is because TCP/IP has no DLCs of its own in CS for OS/390 Release 5 and above. All are controlled by the common DLC connection manager which knows them in terms of VTAM transport resource list elements (TRLEs). Most of the connection types are for the exclusive use of TCP/IP, and have their TRLEs dynamically defined when TCP/IP requests a connection. Two of them (MPC and ATM) can be used by both VTAM and TCP/IP, and these must be manually defined by means of VTAM major nodes. Figure 32 shows how we coded the definitions for the MPC connection.

```
*********************************************************************
*                                                                   *
*         VTAM TRL NODE FOR 2216 MPC                                *
*********************************************************************
M2216   VBUILD TYPE=TRL
M3A2216A TRLE   LNCTL=MPC,                                            X
               MAXBFRU=9,                                             X
               READ=2C0,                                              X
               WRITE=2C1,                                             X
               MPCLEVEL=HPDT,                                         X
               REPLYTO=3.0
```

Figure 32. TRLE Definitions for MPC Connection

The TRL major node shown specifies the read and write channel addresses for the connection. Since VTAM will not be using the connection itself, we do not need to code an additional local SNA major node.

The channel addresses must somehow be logically connected to the 2216 definitions and the ESCON configuration, and this is done through the MVS IOCP definitions as shown in Figure 33.

```
  RESOURCE PARTITION=((A1,1),(A2,2),(A3,3),(A4,4),(A5,5),(C1,6),*
        (C2,7))                                                    3
  CHPID PATH=(2C),SHARED,                                        *
        PARTITION=((A1,A2,A3,A4),(A1,A2,A3,A4,A5)),SWITCH=E1,    *  4
        TYPE=CNC
  CNTLUNIT CUNUMBR=0280,PATH=(2C),UNITADD=((00,032)),LINK=(C9),  *
        CUADD=1,UNIT=3172
  CNTLUNIT CUNUMBR=02A0,PATH=(2C),UNITADD=((00,032)),LINK=(C9),  *
        CUADD=2,UNIT=3172
  CNTLUNIT CUNUMBR=02C0,PATH=(2C),UNITADD=((00,032)),LINK=(C9),  *  2
        CUADD=4,UNIT=3172
  IODEVICE ADDRESS=(280,32),UNITADD=00,CUNUMBR=(0280),STADET=Y,  *
        PARTITION=(A1),UNIT=3172
  IODEVICE ADDRESS=(2A0,32),UNITADD=00,CUNUMBR=(02A0),STADET=Y,  *
        PARTITION=(A2),UNIT=3172                                    1
  IODEVICE ADDRESS=(2C0,32),UNITADD=00,CUNUMBR=(02C0),STADET=Y,  *
        PARTITION=(A4),UNIT=3172
```

Figure 33. IOCP Definitions for MPC Connection

The definitions are for a seven-LPAR sysplex in which our RA39M image is LPAR number 4. There is an ESCON Director and the ESCON multiple image facility (EMIF) is in use. The logic is as follows:

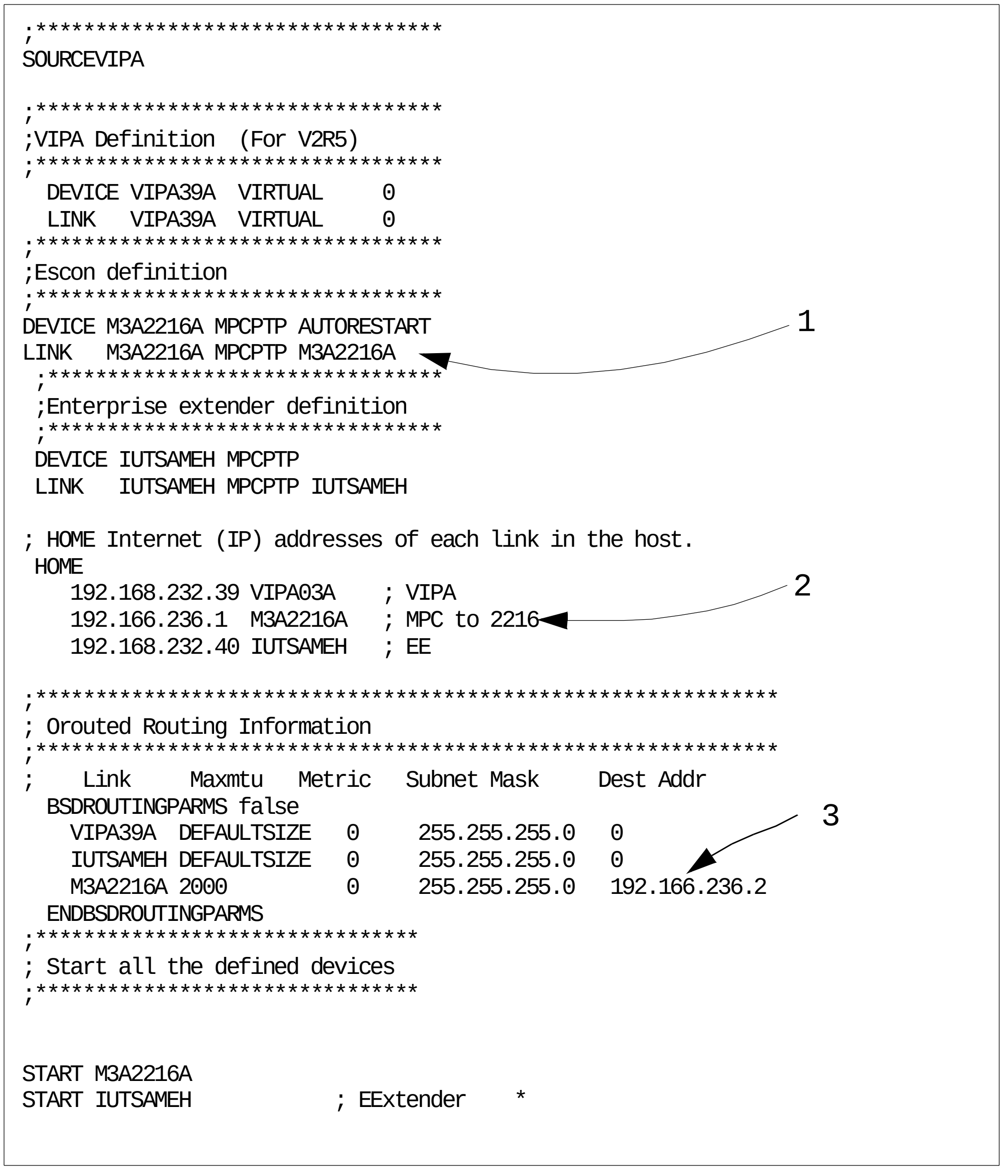
1. The channel addresses are defined in the ADDRESS keyword of the IODEVICE statement. The IODEVICE statement also contains the partition identifier (A4) and the low-order unit address (UNITADD=00) which corresponds to the device addresses configured in the 2216.
2. The CUNUMBR keyword on the IODEVICE statement connects it to the CNTLUNIT statement which defines the logical control unit (CUADD=4, which allows the 2216 to distinguish between the three EMIF connections), the downstream ESCON Director port (LINK=C9) and the channel path ID (CHPID=2C).

   Note that, from V3R2 of the 221X software (with PTF01), it is no longer necessary to have a separate control unit number for each LPAR; the LPAR number is sufficient to identify the source of the data. This means that you can define just one set of CNTLUNIT and IODEVICE statements for all LPARs, and have the same device address for each.
3. The RESOURCE statement states that the partition identified as A4 is in fact number 4.
4. The CHPID keyword in the CNTLUNIT statement identifies the CHPID statement which contains the ESCON Director identifier (SWITCH=E1).

Figure 36 on page 56 shows how the 2216 definitions are related to these.

2.6.2 TCP/IP Definitions

Our TCP/IP profile differed from the previous one, as shown in Figure 34 on page 55.



```
;*********************************
SOURCEVIPA

;*********************************
;VIPA Definition  (For V2R5)
;*********************************
  DEVICE VIPA39A  VIRTUAL     0
  LINK   VIPA39A  VIRTUAL     0
;*********************************
;Escon definition
;*********************************
DEVICE M3A2216A MPCPTP AUTORESTART
LINK   M3A2216A MPCPTP M3A2216A
 ;********************************
 ;Enterprise extender definition
 ;********************************
 DEVICE IUTSAMEH MPCPTP
 LINK   IUTSAMEH MPCPTP IUTSAMEH

; HOME Internet (IP) addresses of each link in the host.
 HOME
    192.168.232.39 VIPA03A    ; VIPA
    192.166.236.1  M3A2216A   ; MPC to 2216
    192.168.232.40 IUTSAMEH   ; EE

;**************************************************************
; Orouted Routing Information
;**************************************************************
;    Link     Maxmtu   Metric   Subnet Mask     Dest Addr
  BSDROUTINGPARMS false
    VIPA39A  DEFAULTSIZE   0     255.255.255.0   0
    IUTSAMEH DEFAULTSIZE   0     255.255.255.0   0
    M3A2216A 2000          0     255.255.255.0   192.166.236.2
  ENDBSDROUTINGPARMS
;********************************
; Start all the defined devices
;********************************


START M3A2216A
START IUTSAMEH            ; EExtender    *
```

Figure 34. TCP/IP Profile with 2216 MPC

The main differences are as follows:

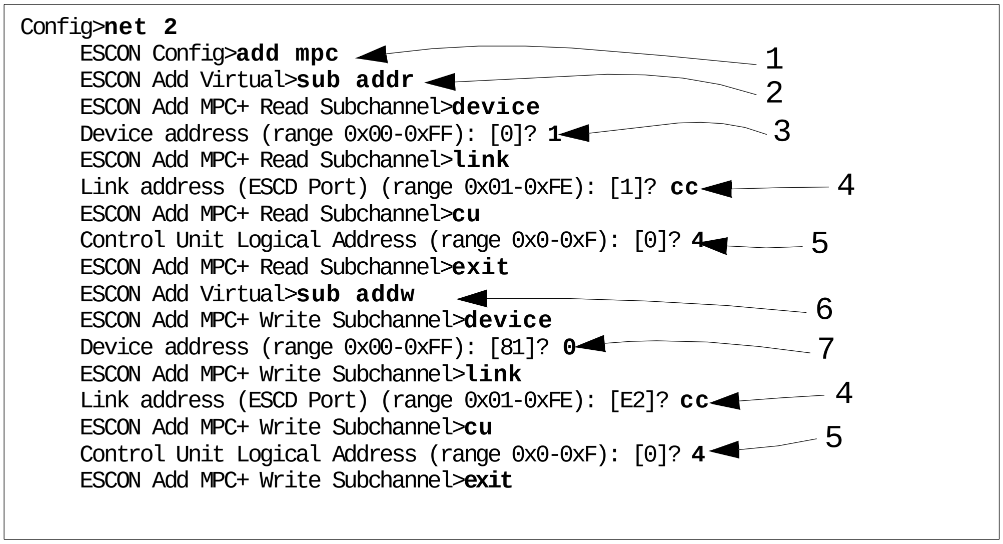
1. The 3172 interface is replaced by the 2216 MPC device and link statements. Note the interface type is MPCPTP, the same as the same-host interface to VTAM. The device name must match the TRLE name (M3A2216A).
2. The address 192.166.236.1 is assigned to this interface.
3. The routing characteristics of this link are defined in the BSDROUTINGPARMS statement. The destination IP address must be coded because this is a point-to-point link incapable of broadcast or multicast. BSDROUTINGPARMS is required only if RIP is used on this connection.

2.6.3 2216 Definitions

In the 2216 router, we first added the ESCON device (Figure 35 on page 56) and then configured the physical connection for MPC (Figure 36 on page 56).

```
Config>add device escon
     Device Slot #(1-8) [1]? 3
     Adding ESCON Channel device in slot 3  port 1 as interface #2
     Use net 2 to configure ESCON Channel parameters
```

Figure 35. Adding ESCON Device to 2216



```
Config>net 2
     ESCON Config>add mpc                                             1
     ESCON Add Virtual>sub addr                                       2
     ESCON Add MPC+ Read Subchannel>device
     Device address (range 0x00-0xFF): [0]? 1                         3
     ESCON Add MPC+ Read Subchannel>link
     Link address (ESCD Port) (range 0x01-0xFE): [1]? cc                4
     ESCON Add MPC+ Read Subchannel>cu
     Control Unit Logical Address (range 0x0-0xF): [0]? 4              5
     ESCON Add MPC+ Read Subchannel>exit
     ESCON Add Virtual>sub addw                                         6
     ESCON Add MPC+ Write Subchannel>device
     Device address (range 0x00-0xFF): [81]? 0                          7
     ESCON Add MPC+ Write Subchannel>link
     Link address (ESCD Port) (range 0x01-0xFE): [E2]? cc               4
     ESCON Add MPC+ Write Subchannel>cu
     Control Unit Logical Address (range 0x0-0xF): [0]? 4               5
     ESCON Add MPC+ Write Subchannel>exit
```

Figure 36. Configuring ESCON MPC Interface

The relevant entries in this definition are:

1. The `add mpc` command tells the 2216 that this is an MPC connection.
2. The `sub addr` command adds a read subchannel.
3. The read subchannel address is 1, which corresponds to UNITADD=01 in IODEVICE in IOCP, which corresponds to ADDRESS=2C1 in IOCP, which is the write subchannel address defined in the TRL major node (Figure 32 on page 53).
4. The upstream (host side) port on the ESCON Director is CC.
5. The control unit logical address is 4, which corresponds to CUADD=4 on the CNTLUNIT statement in the IOCP. Note that these definitions are no longer necessary after V3R2 PTF01 of the 2216 software.
6. The `sub addw` command adds a write subchannel.
7. The write subchannel is 0, which corresponds to UNITADD=00 in IODEVICE in IOCP, which corresponds to ADDRESS=2C0 in IOCP, which is the read subchannel address in the TRL major node.

Once the MPC definitions are completed you can configure TCP/IP and/or APPN on those connections. We did not use APPN in this test; at least, not to the knowledge of the 2216. The TCP/IP configuration consists merely of defining the IP address of this interface, as in Figure 37 on page 57.

```
Config>protocol ip
Internet protocol user configuration
IP Config>add address 5 192.166.236.2 255.255.255.0
                                    1
```

Figure 37. IP Address Assignment

The address 5 (1) is the virtual interface address assigned to the MPC interface by the 2216. It can be displayed by entering the `list` command during ESCON MPC configuration.

2.6.4 Displays

When we started the nodes in the new configuration we received the message in Figure 38 from MVS TCP/IP, showing that an IP connection has been established with the 2216.

```
   EZZ4313I INITIALIZATION COMPLETE FOR DEVICE M3A2216A
   EZZ4324I CONNECTION TO 192.166.236.2 ACTIVE FOR DEVICE M3A2216A
```

Figure 38. TCP/IP MPC Initialization

A VTAM display (Figure 39) confirms that the TRLE is active and that something (not necessarily VTAM) is using it.

```
D NET,TRL,TRLE=M3A2216A
IST097I DISPLAY ACCEPTED
IST075I NAME = M3A2216A, TYPE = TRLE 856
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST087I TYPE = LEASED             , CONTROL = MPC , HPDT = YES
IST1577I HEADER SIZE = 4096 DATA SIZE = 32 STORAGE = ***NA***
IST1221I WRITE DEV = 02C1 STATUS = ACTIVE     STATE = ONLINE
IST1577I HEADER SIZE = 4092 DATA SIZE = 32 STORAGE = DATASPACE
IST1221I READ  DEV = 02C0 STATUS = ACTIVE     STATE = ONLINE
IST314I END
```

Figure 39. VTAM MPC TRLE Display

2.7 Enterprise Extender and Native APPN in Parallel

Our third network configuration is shown in Figure 40 on page 58. Here we have the 2216 acting as a gateway to the CS/NT machine, with both native SNA(HPR) and IP connectivity between the 2216 and the mainframe. The 2216 now acts as the Enterprise Extender boundary between IP and SNA. Thus we can use the native SNA and Enterprise Extender connections as backup for each other in case of failure, without disrupting existing sessions.

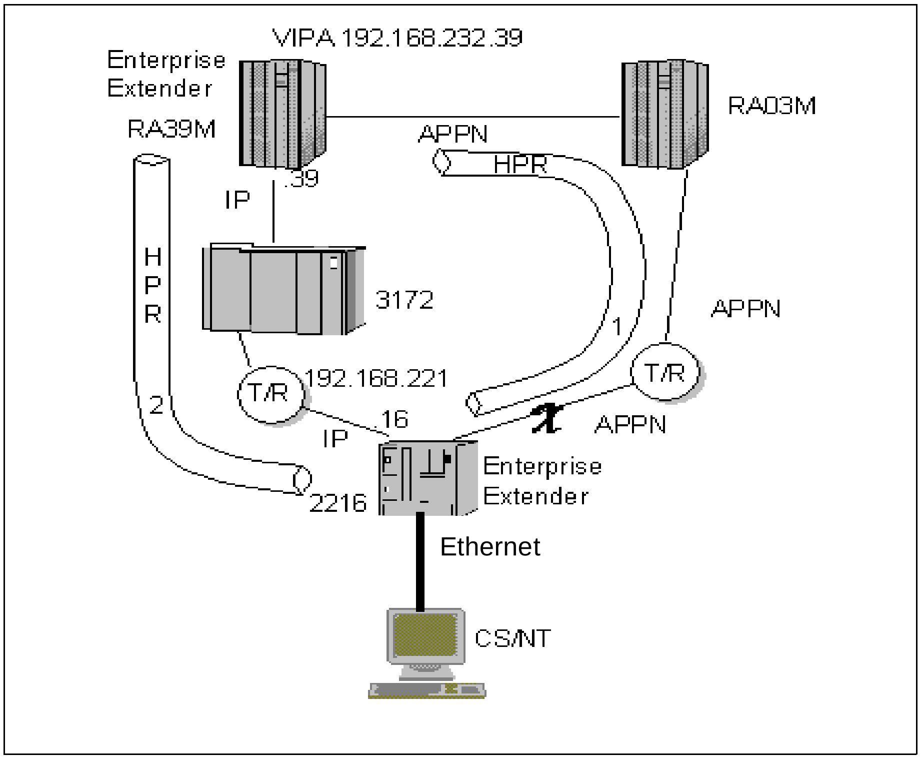


Figure 40. Native APPN and Enterprise Extender Backup

The features of this configuration are:

- The connection between RA39M and the 2216 is exactly the same as in "Enterprise Extender between CS for OS/390 and CS/NT via 3172" on page 33. In particular, the IP addresses are the same and the TCP/IP profile is exactly the same.
- The connection between the 2216 and CS/NT is now native APPN. The 2216, therefore, needs an APPN control point which we called USIBMRA.CP22162.
- There is an XCF connection between RA39M and another network node in the sysplex, USIBMRA.RA03M. XCF connections are always HPR.
- There is an APPN connection between RA03M and CP22162. This connection traverses a token-ring attached to the mainframe by an OSA.
- There are APPN CP-CP sessions between RA39M and RA03M (over XCF), between RA03M and CP2216A (over the token-ring), between RA39M and CP22162 (over the Enterprise Extender connection) and between CP22162 and WTR05212 (over the Ethernet).
- The configurations for MVS TCP/IP and the 3172 are identical to those in the first example.

2.7.1 VTAM Definitions

There are no new VTAM definitions for the XCF connection because such connections are dynamically defined. However, we wished in this test to make VTAM establish the Enterprise Extender connection to the 2216, so we had to code a switched major node as shown in Figure 41. The XCA major node was the same as in the previous examples.

```
EESWJ   VBUILD TYPE=SWNET
EEPU    PU     ISTATUS=ACTIVE,PUTYPE=2,CPNAME=CP22162
EESW    PATH   IPADDR=192.168.221.16,GRPNM=EEXCAG
```

Figure 41. Switched Major Node for Enterprise Extender

Note that we specified the IP address of the 2216, namely 192.168.221.16. Ideally, we should have used an IP host name (on the HOSTNAME keyword) if such a name had been registered to the domain name server.

2.7.2 2216 Configuration

The TCP/IP configuration for the 2216 was the same as in “Enterprise Extender between CS for OS/390 and CS/NT via 3172” on page 33, but we now had to add APPN and Enterprise Extender definitions. For APPN, we needed to define:

1. The node
2. The LAN port connecting the 2216 to the OSA and thus to RA03M
3. The LAN port connecting the 2216 to the CS/NT node (which was on a different LAN)
4. The link station from the 2216 to RA03M, as the 2216 was to initiate this connection
5. The Enterprise Extender port
6. The Enterprise Extender link station to connect with RA39M. Even though RA39M has this link station defined, we also defined it on the 2216 to show how it was done.

The APPN node definition is shown in Figure 42.

```
    Config>p appn
    APPN user configuration
    APPN config>set node                                          1
    Enable APPN (Y)es (N)o [Y]? y
    Network ID (Max 8 characters) [  ]? usibmra
    Control point name (Max 8 characters) [  ]? cp22162
    Route addition resistance(0-255) [128]? 128
    XID ID number for subarea connection (5 hex digits) [00000]? 00000
    Write this record? [Y]?Y
    The record has been written.
    APPN config>exit
```

Figure 42. 2216 APPN Node Configuration

The node definition is performed by entering `talk 6` from the base (*) prompt. The main thing that has to be defined here is the CP name (1).

Next, we defined the token-ring port as shown in Figure 43 on page 60.

```
2216  APPN config>add port
APPN Port
Link Type: (P)PP, (FR)AME RELAY, (E)THERNET, (T)OKEN RING,
(M)PC, (S)DLC, (X)25, (FD)DI, (D)LSw, (A)TM, (I)P  [ ]? t
Interface number(Default 0): [0]?
Port name (Max 8 characters) [T00000]? tkr001

Enable APPN on this port (Y)es (N)o [Y]?
Port Definition
        Support multiple PU (Y)es (N)o [N]?
        Service any node: (Y)es (N)o [Y]?
        High performance routing: (Y)es (N)o [Y]?
        Maximum BTU size (768-17745) [2048]?
        Maximum number of link stations (1-976) [512]?
        Percent of link stations reserved for incoming calls (0-100) [0]?
        Percent of link stations reserved for outgoing calls (0-100) [0]?
        Local SAP address (04-EC) [4]?
        Local HPR SAP address (04-EC) [C8]?
Edit TG Characteristics: (Y)es (N)o [N]?
Edit LLC Characteristics: (Y)es (N)o [N]? n
Edit HPR defaults: (Y)es (N)o [N]?
Write this record? [Y]? y
```

Figure 43. 2216 Token-Ring Port Configuration

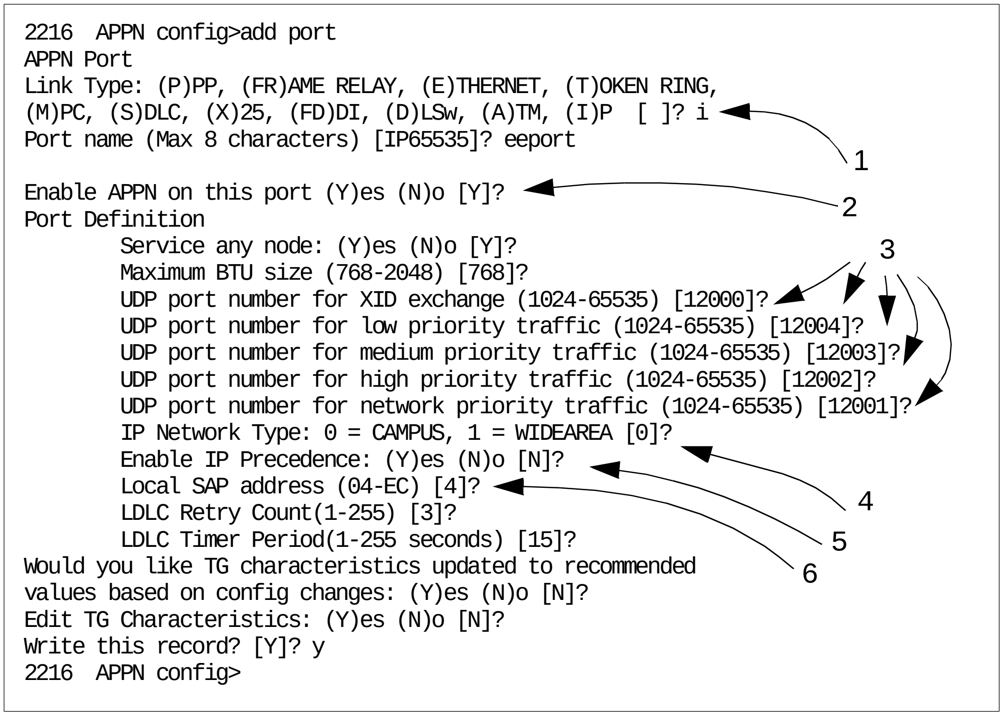
The corresponding link station definition is shown in Figure 44.

```
2216  APPN config> add link
APPN Station
Port name for the link station [ ]? tkr001                    1
Station name (Max 8 characters) [ ]? lnks001                        2

        Activate link automatically (Y)es (N)o [Y]?
        MAC address of adjacent node [400052005042]?
        Solicit SSCP Session: (Y)es (N)o [N]?
        Does link support APPN function: (Y)es (N)o [Y]?
        Adjacent node type: 0 = APPN network node,
        1 = APPN end node or Unknown node type,
        2 = LEN end node [0]?
        High performance routing: (Y)es (N)o [Y]?
        Allow CP-CP sessions on this link (Y)es (N)o [Y]?
        CP-CP session level security (Y)es (N)o [N]?
        Configure CP name of adjacent node: (Y)es (N)o [N]?
Edit TG Characteristics: (Y)es (N)o [N]?
Edit LLC Characteristics: (Y)es (N)o [N]? y                3
        Remote SAP(04-EC) [8]?
        Maximum number of outstanding I-format LPDUs (1-127) [26]?
        Receive window size (1-127) [26]?
        Inactivity timer(1-254 seconds) [30]?
        Reply timer (1-254 half seconds) [2]?
        Maximum number of retransmissions(1-254) [8]?
        Receive acknowledgement timer (1-254 half seconds) [1]?
        Acknowledgements needed to increment working window(0-127) [1]?
Edit HPR defaults: (Y)es (N)o [N]?
Write this record? [Y]? y
```

Figure 44. Token-Ring Link Station Definition

Note the reference to the port tkr001 (1) and the MAC/SAP address of the OSA which will connect us to RA03M (2,3).

The definitions for the Ethernet port to the CS/NT machine were similar. The Enterprise Extender port, however, is more interesting and its definition is shown in Figure 45.



```
2216  APPN config>add port
APPN Port
Link Type: (P)PP, (FR)AME RELAY, (E)THERNET, (T)OKEN RING,
(M)PC, (S)DLC, (X)25, (FD)DI, (D)LSw, (A)TM, (I)P  [ ]? i
Port name (Max 8 characters) [IP65535]? eeport

Enable APPN on this port (Y)es (N)o [Y]?
Port Definition
        Service any node: (Y)es (N)o [Y]?
        Maximum BTU size (768-2048) [768]?
        UDP port number for XID exchange (1024-65535) [12000]?
        UDP port number for low priority traffic (1024-65535) [12004]?
        UDP port number for medium priority traffic (1024-65535) [12003]?
        UDP port number for high priority traffic (1024-65535) [12002]?
        UDP port number for network priority traffic (1024-65535) [12001]?
        IP Network Type: 0 = CAMPUS, 1 = WIDEAREA [0]?
        Enable IP Precedence: (Y)es (N)o [N]?
        Local SAP address (04-EC) [4]?
        LDLC Retry Count(1-255) [3]?
        LDLC Timer Period(1-255 seconds) [15]?
Would you like TG characteristics updated to recommended
values based on config changes: (Y)es (N)o [N]?
Edit TG Characteristics: (Y)es (N)o [N]?
Write this record? [Y]? y
2216  APPN config>
```

Figure 45. 2216 Enterprise Extender Port Definition

Note the following:

1. This is an APPN port, but the port type is chosen to be I meaning IP.
2. As with CS/NT, enabling APPN really means enabling CP-CP sessions. Since these are two network nodes talking to each other, CP-CP session support is desirable.
3. The configuration offers you the option to change the default UDP port numbers to be used for the five types of HPR traffic (four APPN priorities plus LDLC commands).
4. The IP network type (default CAMPUS) determines the default TG characteristics of the connections that will be made on this port. The same TG profiles (LAN and WAN) are available as are supplied with VTAM.
5. You are also offered the option to use the IP precedence bits for specifying the priority.
6. The local SAP address, together with the remote SAP address defined on the link station, will uniquely identify this connection just as it does with VTAM.

Next, we defined the Enterprise Extender link station to RA39M. Figure 46 on page 62 illustrates.

```
2216  APPN config>add link
APPN Station
Port name for the link station [ ]? eeport
Station name (Max 8 characters) [ ]? eestat
                                                                  1
        Activate link automatically (Y)es (N)o [Y]?
        IP address of adjacent node  [192.168.232.39]?
        Adjacent node type: 0 = APPN network node,
        1 = APPN end node or Unknown node type  [0]?
        Allow CP-CP sessions on this link (Y)es (N)o [Y]?
        CP-CP session level security (Y)es (N)o [N]?
        Configure CP name of adjacent node: (Y)es (N)o [N]?
        Remote SAP(04-EC) [4]?
        IP Network Type: 0 = CAMPUS, 1 = WIDEAREA [0]?
        LDLC Retry Count(1-255) [3]?
        LDLC Timer Period(1-255 seconds) [15]?
Would you like TG characteristics updated to recommended
values based on config changes: (Y)es (N)o [N]?
Edit TG Characteristics: (Y)es (N)o [N]?
Write this record? [Y]? n
```

Figure 46. Enterprise Extender Link Station on 2216

Note the VIPA address of the RA39M TCP/IP stack (1). The link station definition also allows you to control what kind of APPN partner will be accepted on this connection, and to override the default TG characteristics. The remote SAP is also specified here.

Last but not least, we had to define the internal IP address of the 2216. This acts rather like a VIPA address on MVS, in that it represents the 2216 as a whole and not one particular interface. This is the address we defined in the call-out switched major node in VTAM (Figure 41 on page 59), namely 192.168.221.16. Figure 47 shows the definition.

```
2216  IP config>set internal-ip-add
Internal IP address [0.0.0.0]? 192.168.221.16
```

Figure 47. 2216 Internal IP Address

When we had completed the 2216 configuration we displayed it using the `list all` command as shown in Figure 48 on page 63.

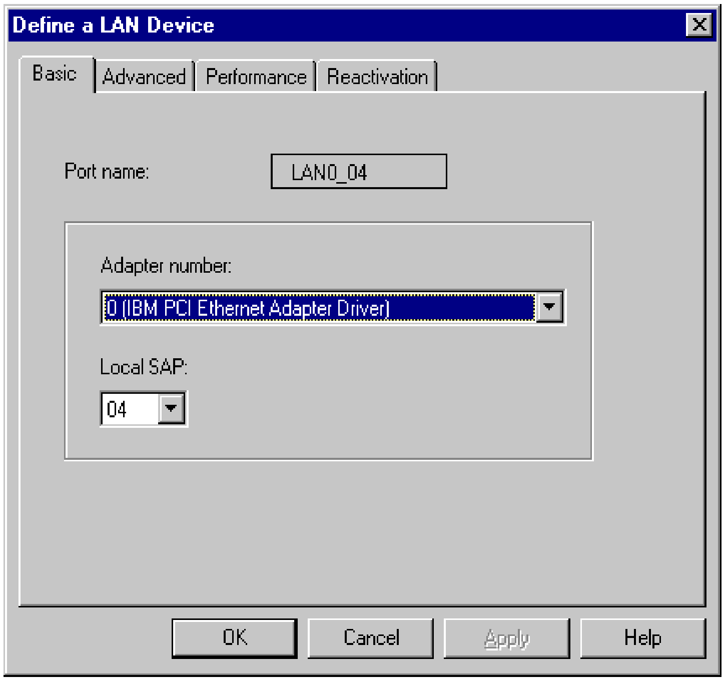
```
2216  APPN config>list all
NODE:
        NETWORK ID: USIBMRA
        CONTROL POINT NAME: CP22162
        XID: 00000
        APPN ENABLED: YES
        BREX OR BORDER NODE: NEITHER
        MAX SHARED MEMORY: 5108
        MAX CACHED: 4000
DLUR:
        DLUR ENABLED: NO
        PRIMARY DLUS NAME:
CONNECTION NETWORK:
              CN NAME       LINK TYPE  PORT INTERFACES
        -------------------------------------------------------------
COS:
        COS NAME
        --------
          #BATCH
        #BATCHSC
        #CONNECT
          #INTER
        #INTERSC
         CPSVCMG
        SNASVCMG
MODE:
         MODE NAME  COS NAME
        ---------------------
PORT:
          INTF     PORT      LINK      HPR    SERVICE   PORT
         NUMBER    NAME      TYPE    ENABLED    ANY    ENABLED
        ------------------------------------------------------
             0    TKR001   IBMTRNET    YES      YES      YES
             6    ETH001   ETHERAND    YES      YES      YES
         65535    EEPORT     HPR_IP    YES      YES      YES

STATION:
         STATION    PORT       DESTINATION      HPR    ALLOW  ADJ NODE
          NAME      NAME         ADDRESS      ENABLED  CP-CP    TYPE
        ------------------------------------------------------------
         LNKS002   ETH001      0020358337C6     YES     YES       0
          EESTAT   EEPORT    192.168.232.39     YES     YES       0
         LNKS001   TKR001      400052005042     YES     YES       0

LU NAME:
              LU NAME        STATION NAME          CP NAME
        ------------------------------------------------------------
2216  APPN config>
```

Figure 48. 2216 Configuration Display

2.7.3 CS/NT Configuration

On the CS/NT machine, the APPN node, DLU server, and dependent LU/PU definitions are the same as in the first example. The difference is now CS/NT will have a direct APPN connection to the 2216 over a real LAN, rather than a direct APPN connection to VTAM over an IP network. Therefore, we had to replace the Enterprise Extender port and link station with an Ethernet port and link station. There was no need to change the dependent LU/PU definitions because they were tied to the DLU server name rather than to any physical connection. We performed the following steps:

1. Select **Devices**, **LAN** and **Create** to define the port. This gives you a panel like that shown in Figure 49, where we chose IBM PCI Ethernet Adapter and clicked on OK.
2. Select **CPI-C and APPC**, **Peer Connections**, **LAN** and **Create** to define the link station. The resulting group of panels is similar to that for an Enterprise Extender link station except that the important thing to fill in is the MAC/SAP address of the 2216 Ethernet port rather than the IP address of the Enterprise Extender partner.



Port name: LAN0_04

Adapter number: 0 (IBM PCI Ethernet Adapter Driver)

Local SAP: 04

Figure 49. Ethernet Port Selection on CS/NT

2.7.4 Displays

When we activated the switched major node definition shown in Figure 41, the connection was made and VTAM was able to display the link station as in Figure 50 on page 65.

```
D NET,E,ID=EEPU
IST097I DISPLAY ACCEPTED
IST075I NAME = EEPU, TYPE = PU_T2.1 049
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST1043I CP NAME = CP22162, CP NETID = USIBMRA, DYNAMIC LU = YES
IST1589I XNETALS = YES
IST1105I RESOURCE STATUS TGN CP-CP TG CHARACTERISTICS
IST1106I EEPU     AC/R    21 YES   98750000000000000000017100808080
IST1482I HPR = RTP - OVERRIDE = N/A - CONNECTION = YES
IST1510I LLERP = NOTPREF - RECEIVED = NOTALLOW
IST1680I LOCAL IP ADDRESS 192.168.232.39                            1
IST1680I REMOTE IP ADDRESS 192.168.221.16
IST136I SWITCHED SNA MAJOR NODE = RA39ASWJ
IST081I LINE NAME = EEXCAL, LINE GROUP = EEXCAG, MAJNOD = RA39AXCA
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST1656I VTAMTOPO = REPORT, NODE REPORTED - YES
IST1657I MAJOR NODE VTAMTOPO = REPORT
IST172I NO LOGICAL UNITS EXIST
IST314I END
```

Figure 50. Link Station for 2216 Enterprise Extender

Note the local and remote IP addresses displayed at (1).

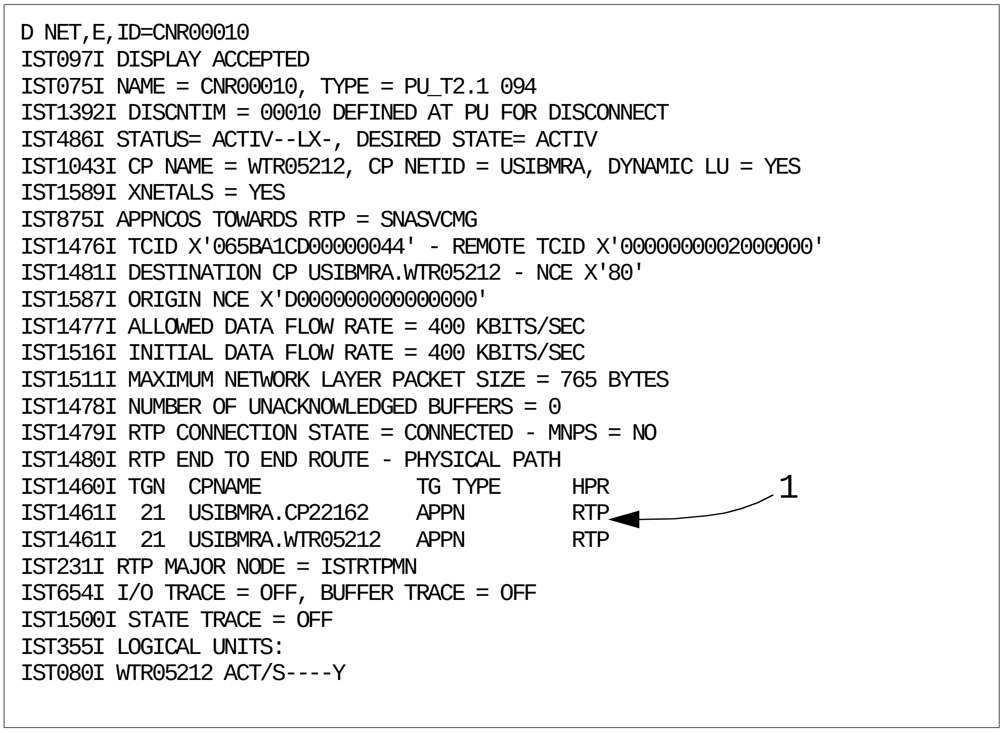
A display of the APPN topology as seen from RA39M (Figure 51) confirmed that the APPN connections are as expected.

.

```
D NET,TOPO,LIST=ALL,ID=RA39M
 IST097I DISPLAY ACCEPTED
 IST350I DISPLAY TYPE = TOPOLOGY 046
 IST1295I CP NAME           NODETYPE ROUTERES CONGESTION  CP-CP WEIGHT
 IST1296I USIBMRA.RA39M     NN       128      NONE        *NA*  *NA*
 IST1579I                   ----------------------------------------
 IST1297I                   ICN/MDH  CDSERVR  RSN         HPR
 IST1298I                   NO       NO       2           RTP
 IST1579I                   ----------------------------------------
 IST1223I                   BN       NATIVE   TIME LEFT
 IST1224I                   NO       YES      13
 IST1299I TRANSMISSION GROUPS ORIGINATING AT CP USIBMRA.RA39M
 IST1357I                                                 CPCP
 IST1300I DESTINATION CP    TGN      STATUS   TGTYPE      VALUE WEIGHT
 IST1301I USIBMRA.RA03M     21       OPER     INTERM      YES   *NA*
 IST1301I USIBMRA.RA28M     21       OPER     INTERM      YES   *NA*
 IST1301I USIBMRA.CP22162   21       OPER     INTERM      YES   *NA*
 IST314I END
```

Figure 51. APPN Topology from RA39M

When all the network connections were in place, we displayed one of the RTP connections from RA39M, as in Figure 52 on page 66.



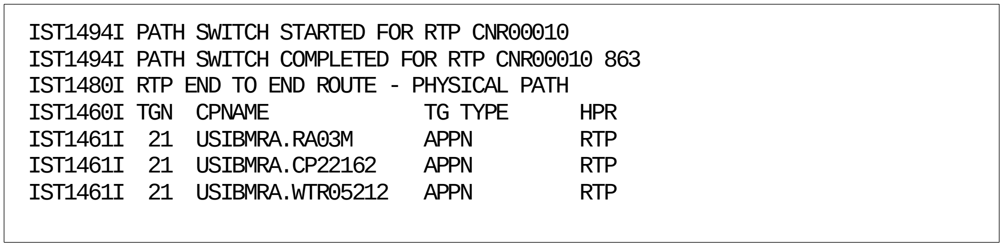
```
D NET,E,ID=CNR00010
IST097I DISPLAY ACCEPTED
IST075I NAME = CNR00010, TYPE = PU_T2.1 094
IST1392I DISCNTIM = 00010 DEFINED AT PU FOR DISCONNECT
IST486I STATUS= ACTIV--LX-, DESIRED STATE= ACTIV
IST1043I CP NAME = WTR05212, CP NETID = USIBMRA, DYNAMIC LU = YES
IST1589I XNETALS = YES
IST875I APPNCOS TOWARDS RTP = SNASVCMG
IST1476I TCID X'065BA1CD00000044' - REMOTE TCID X'0000000002000000'
IST1481I DESTINATION CP USIBMRA.WTR05212 - NCE X'80'
IST1587I ORIGIN NCE X'D000000000000000'
IST1477I ALLOWED DATA FLOW RATE = 400 KBITS/SEC
IST1516I INITIAL DATA FLOW RATE = 400 KBITS/SEC
IST1511I MAXIMUM NETWORK LAYER PACKET SIZE = 765 BYTES
IST1478I NUMBER OF UNACKNOWLEDGED BUFFERS = 0
IST1479I RTP CONNECTION STATE = CONNECTED - MNPS = NO
IST1480I RTP END TO END ROUTE - PHYSICAL PATH
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I  21  USIBMRA.CP22162    APPN         RTP
IST1461I  21  USIBMRA.WTR05212   APPN         RTP
IST231I RTP MAJOR NODE = ISTRTPMN
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST355I LOGICAL UNITS:
IST080I WTR05212 ACT/S----Y
```

Figure 52. RTP Connection for DLUR/S Sessions

This particular connection was the DLUR/S pipe carrying the LU 6.2 dependent LU requester sessions. The route taken (1) comprised the Enterprise Extender connection (TG 21) from RA39M to CP22162, followed by the Ethernet connection (TG 21) from CP22162 to WTR05212. This path obviously had a lower weight for the SNASVCMG class of service than the longer route via RA03M, even though the longer route was all native APPN.

After we had established some independent and dependent LU sessions between WTR05212 and the VTAM hosts, we unplugged the 3172 connection to the token-ring. This had the effect of breaking the Enterprise Extender link, because there was no other IP path between the 2216 and RA39M.

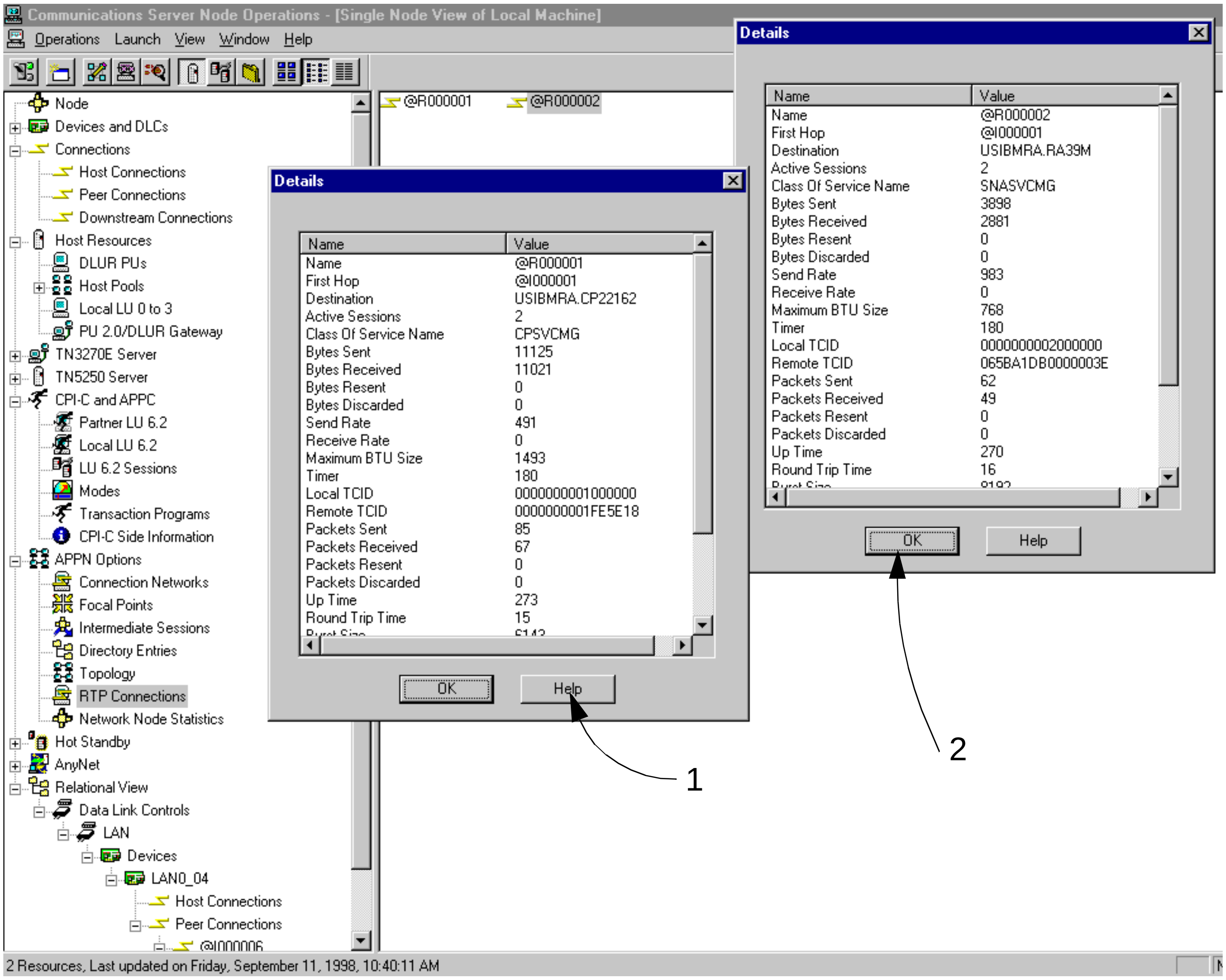
The sessions remained operational. To see why, we observed the console log on RA39M to see messages such as that in Figure 53. A non-disruptive path switch had occurred.



```
IST1494I PATH SWITCH STARTED FOR RTP CNR00010
IST1494I PATH SWITCH COMPLETED FOR RTP CNR00010 863
IST1480I RTP END TO END ROUTE - PHYSICAL PATH
IST1460I TGN  CPNAME            TG TYPE      HPR
IST1461I  21  USIBMRA.RA03M      APPN         RTP
IST1461I  21  USIBMRA.CP22162    APPN         RTP
IST1461I  21  USIBMRA.WTR05212   APPN         RTP
```

Figure 53. Path Switch for DLUR/S Pipe

We were able to confirm that sessions remained in place from the CS/NT node as well. As an example of the displays available on CS/NT, Figure 54 on page 67 shows two RTP connections which remained active across the failure.



Name	Value
Name	@R000001
First Hop	@I000001
Destination	USIBMRA.CP22162
Active Sessions	2
Class Of Service Name	CPSVCMG
Bytes Sent	11125
Bytes Received	11021
Bytes Resent	0
Bytes Discarded	0
Send Rate	491
Receive Rate	0
Maximum BTU Size	1493
Timer	180
Local TCID	0000000001000000
Remote TCID	000000001FE5E18
Packets Sent	85
Packets Received	67
Packets Resent	0
Packets Discarded	0
Up Time	273
Round Trip Time	15
Burst Size	6143

Name	Value
Name	@R000002
First Hop	@I000001
Destination	USIBMRA.RA39M
Active Sessions	2
Class Of Service Name	SNASVCMG
Bytes Sent	3898
Bytes Received	2881
Bytes Resent	0
Bytes Discarded	0
Send Rate	983
Receive Rate	0
Maximum BTU Size	768
Timer	180
Local TCID	0000000002000000
Remote TCID	065BA1DB0000003E
Packets Sent	62
Packets Received	49
Packets Resent	0
Packets Discarded	0
Up Time	270
Round Trip Time	16
Burst Size	8192

2 Resources, Last updated on Friday, September 11, 1998, 10:40:11 AM

Figure 54. RTP Connection Display from CS/NT

In the display, (1) shows the CP-CP session pipe between WTR05212 and CP22162, whereas (2) shows the DLUR/S session pipe between WTR05212 and RA39M. The latter is the one that survived the token-ring failure, and is the one known as CNR00010 on RA39M, as the transport connection identifiers (TCIDs) confirm.

Chapter 3. AnyNet Sockets over SNA

AnyNet Sockets over SNA is the only comprehensive technique available for running Sockets applications over an SNA transport network. It translates the Sockets calls directly to SNA requests, so it does not require a TCP/IP stack on the same node unless native IP transport is also needed.

AnyNet Sockets over SNA is currently implemented as part of the Communications Server product family, comprising the following products:

- SecureWay Communications Server for OS/390 (access node only)
- OS/400 (access node only)
- Communications Server for AIX (access node and gateway)
- Communications Server for OS/2 (access node and gateway)
- Communications Server for Windows NT (access node and gateway)

In addition, AnyNet access node function only is available in Personal Communications for OS/2, Windows NT and Windows 95/98.

AnyNet functions have been available earlier for various platforms in different packages, such as AnyNet/2 for OS/2. These older versions are still compatible with the more current implementations.

The Sockets over SNA part of AnyNet has changed in OS/390 OpenEdition Version 2 Release 5, as well as in Communications Server for Windows NT Version 6. Therefore, we concentrated on these platforms for the examples described in this chapter.

3.1 Sockets over SNA Overview

AnyNet Sockets over SNA provides SNA transport to applications written to the Sockets interface that would normally expect a TCP/IP network to satisfy their transport needs. You do not need to have native TCP/IP running in the same system, if all the transport needs are satisfied through the SNA network.

If you, however, run native TCP/IP and AnyNet Sockets over SNA at the same time on a single system, you must give them separate IP subnetworks and therefore different IP addresses. This is because the Sockets over SNA application looks like a new IP interface to the application; this new virtual IP interface is normally known as *sna0*. As with all IP interfaces, *sna0* must be given an IP address and a subnet mask to allow communication over that interface (in other words, over the SNA network).

Sockets over SNA and native TCP/IP operate independently of one another and have no awareness of each other's presence in an access node, where any communication between them must be done through an IP router. In a gateway node, they can route traffic to each other’s interfaces.

Because AnyNet Sockets over SNA must cause the SNA network to look like an IP network, it must map IP addresses to SNA logical unit names. Thus an IP packet addressed to a given IP address can be assigned to the appropriate SNA partner node, and an SNA session created if none exists. Address mapping can be done in one of two ways:

- Explicitly. In the explicit method, you have to define each IP address and its corresponding LU name manually. This has to be done in every AnyNet node for every IP address used as a destination address from that node. As the network grows, this can become a tedious task.
- Algorithmically. The recommended method for address mapping is the algorithmic one. Using this method, it is possible to code only one definition and allow the provided address mapping program to do the rest. Dynamic definition of LUs in subarea networks, and especially in APPN, will allow creation of new LUs automatically as they are needed, making network growth easy for the systems programmer. Also, as the mapping is based on an algorithm, no memory or disk space is needed for large mapping tables. The drawback is that you need a reasonably consistent naming and addressing convention in order to allow the algorithm to do its work.

The mapping algorithm is the same in all AnyNet products. This means that for a specific IP address the same LU name will always be generated, provided that the seed values given to the algorithm are the same. The algorithm works as follows:

- The IP address is logically divided into the two obvious portions: the network address (that part defined by the 1 bits in the subnet mask) and the host address (the rest). For example, the IP address 9.24.104.251 with the subnet mask 255.255.224.0 comprises a network address of 9.24.96.0 and a host address of 8.251.
- The network address is used to look up a user-defined table which maps it to an SNA network ID together with an LU name prefix (typically SX). So, in our example, 9.24.96.0 might be mapped to USIBMRA.SX.
- Starting at the right-hand end, the host address is divided into groups of five bits each. In our example above, the host address 8.251 (08FB in hexadecimal) becomes (00010,00111,11011) or (2,7,27).
- Each five-bit section now has 32 possible values, which are mapped to the 36 alphanumeric characters with EIOU missing, as follows:

  ```
  Value:       0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3
               0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
  Mapped to:   0 1 2 3 4 5 6 7 8 9 A B C D F G H J K L M N P Q R S T V W X Y Z
  ```

  In this way, our example of (2,7,27) becomes 27V.
- Finally, this result is padded with leading zeroes and appended to the network ID and the name prefix. In our case it produces USIBMRA.SX00027V.

  You must ensure that the prefix is short enough to accommodate the maximum length of suffix that follows it. The default value SX of two characters is sufficient to cover any eventuality since a host address of more than 30 bits is not possible in the IP architecture.

  You can convert IP addresses to LU names the easy way by using the SXMAP command with the convert option on CS/2 or CS/NT.

From an IP topology point of view, the AnyNet Sockets over SNA section of the network can be seen as a separate IP subnetwork. The Sockets over SNA gateway nodes act as normal IP routers that route IP packets between two interfaces: one standard TCP/IP interface and one AnyNet interface (sna0).

For data sent to the sna0 interface, Sockets over SNA maps the destination IP address to an SNA LU name and sends the data over LU 6.2 sessions established between Sockets over SNA nodes. The LU 6.2 sessions can run over any SNA routing protocol (APPN, HPR, subarea) supported by the AnyNet nodes on the session path.

For Sockets applications of the stream type (where the local and remote applications establish a full-duplex byte-stream connection with each other), Sockets over SNA also needs full-duplex SNA communication. If both SNA LUs support full-duplex LU 6.2, then only one session is established for the stream connection. If either LU does not support full-duplex LU 6.2, two half-duplex sessions are set up (one contention winner for each end).

For IP datagrams (no stream connection), Sockets over SNA establishes one LU 6.2 conversation for each destination, and thus one session whether full duplex or half duplex.

Although most of the nodes we used in our examples support full-duplex LU 6.2, the reader will notice that most of the displays show parallel LU 6.2 sessions in pairs. This is because we used PING extensively to test the connectivity. PING uses a raw Sockets interface rather than a stream connection. Thus one LU 6.2 session is set up by the node sending the PING (ICMP Echo datagram) and one is set up by the node sending the PING response (ICMP Echo Reply datagram).

3.2 Sockets over SNA in SecureWay Communications Server for OS/390

The Sockets over SNA part of AnyNet was redesigned in OS/390 Version 2 Release 5.

In previous versions of OS/390 and MVS, there were three Sockets application programming interfaces, all different:

1. The AnyNet Sockets API
2. The MVS TCP/IP Sockets API
3. The OpenEdition (now UNIX Systems Services) Sockets API.

Programs that required AnyNet had to be written to the AnyNet Sockets API, which meant in practice that most of the supplied TCP/IP application stacks would not work with AnyNet. From OS/390 Release 5 the only API supported by AnyNet Sockets over SNA is the UNIX Systems Services API. While this means that any user-written applications must be ported to the new API, it has the happy side effect that most of the supplied TCP/IP applications now *do* work. It also means that native TCP/IP and/or AnyNet Sockets over SNA connections can be assigned dynamically to an application depending on the routing requirements to the partner host. Figure 55 on page 72 shows how it fits together.

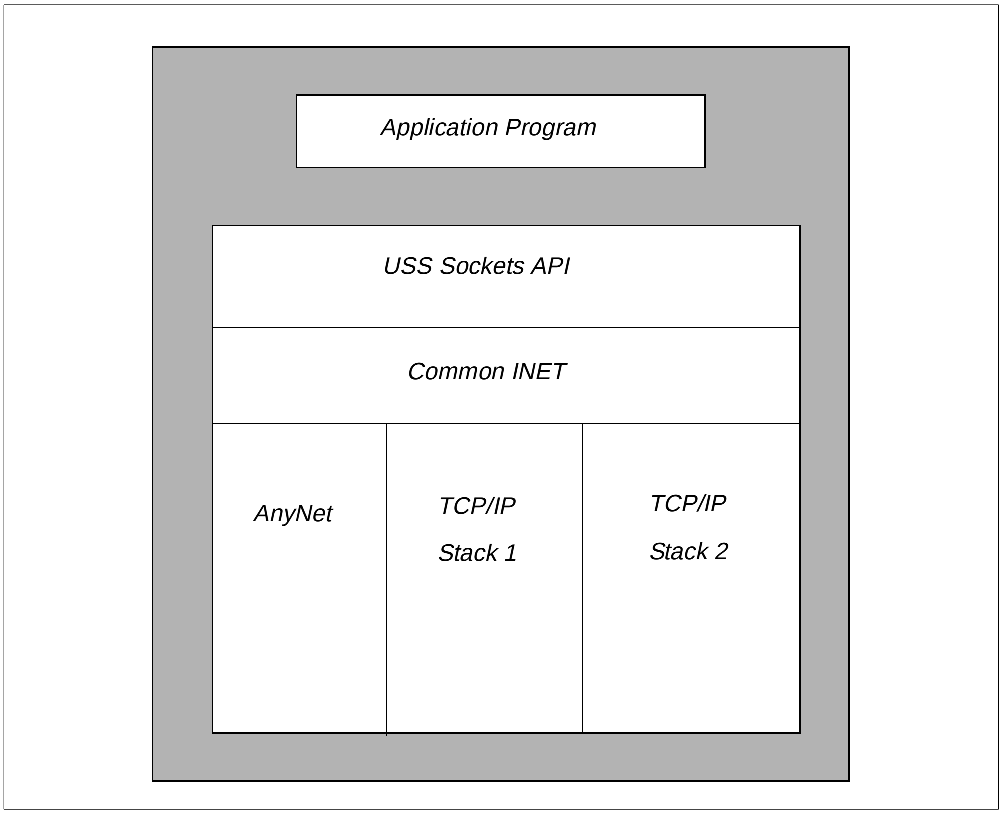


Figure 55. AnyNet Sockets over SNA in OS/390

To run Sockets over SNA in OS/390 you need to consider the following:

- The Sockets over SNA physical file system (PFS) must be defined to UNIX Systems Services. In UNIX terminology, a file is a stream of data which could be anywhere: on a disk or at the other end of a network. A physical file system represents the access method used to read and write the file. INET is a term used to describe the PFS that talks to the network, and CS for OS/390 provides a common INET interface that allows the application to use any one of several stacks (PFSs) without being aware of their existence.
- The Sockets over SNA environment variables must be defined in a data set referred to by DD name ENVVAR. Some of these have changed for OS/390 Release 5, some are new and some have been removed.
- You can have only one instance of Sockets over SNA running on a single OS/390 host. However, you may run any number of TCP/IP stacks at the same time for native TCP/IP. To use Sockets over SNA only, TCP/IP need not be started.
- Sockets over SNA utilities must now be authorized.
- Sockets over SNA now uses UNIX Systems Services HOSTS and NAMES files. It is no longer necessary, or indeed possible, to define the old HOSTS, NETWORKS, and SERVICES files for Sockets over SNA.
- For connectivity testing you can no longer use PING. Use OPING instead. It can be issued as a command from the UNIX Systems Services shell.

For more details on each of the items listed, please refer to *OS/390 eNetwork Communications Server, AnyNet: Guide to Sockets over SNA, Version 2 Release 5,* SC31-8577*.*

3.2.1 Definitions and Setup

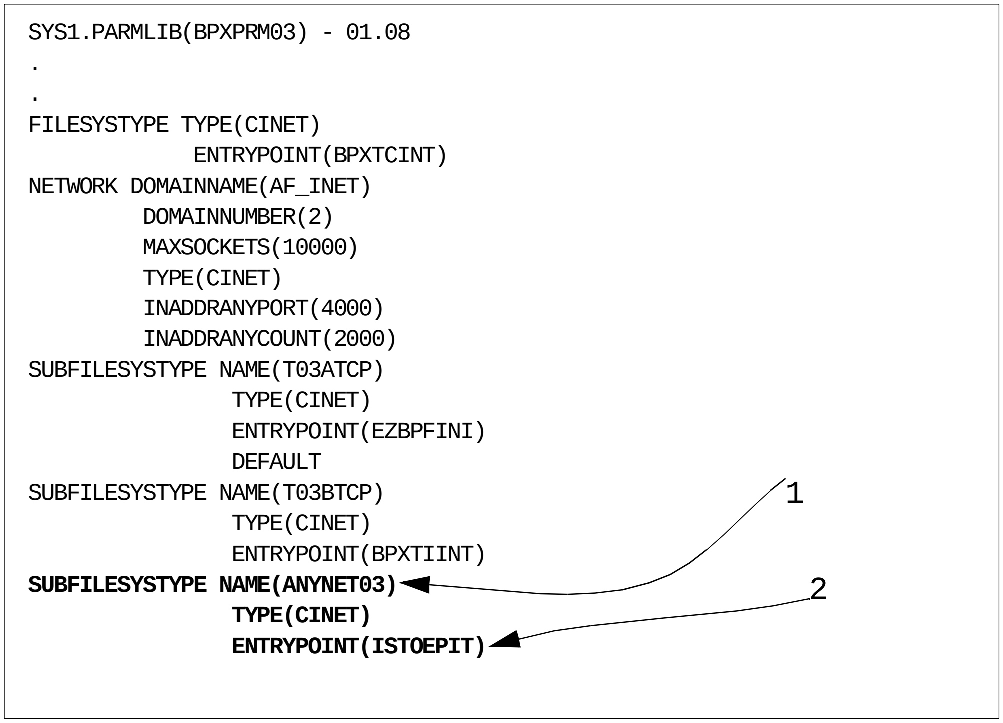
The setup required for Sockets over SNA comprises the following steps which must be executed *in the following order:*

1. Define the file system to UNIX Systems Services, the VTAM applications, the modes and the environment variables.
2. Start the AnyNet Sockets over SNA task (ISTSKDMN).
3. Run the address mapping program (ISTSKMAP).
4. Run the interface definition program (ISTSKIFC).

If your OS/390 LUs will be communicating with remote partners across a Sockets over SNA gateway, you will also need to define a routing table using the ISTSKRTE program. At the very least you will need to define the default router as being the gateway node, so that your Sockets over SNA subnetwork is able to access other IP subnetworks.

To be able to run AnyNet in the UNIX Systems Services environment, the BPXPRMxx member of SYS1.PARMLIB has to be changed to define the physical file system. Please note that changing these values requires an IPL of OS/390.

The example shown in Figure 56 is for one of our MVS images in our first testing scenario. A similar member must be defined for every OS/390 running AnyNet Sockets over SNA. The new, required definition lines are shown in bold.



```
SYS1.PARMLIB(BPXPRM03) - 01.08
.
.
FILESYSTYPE TYPE(CINET)
             ENTRYPOINT(BPXTCINT)
NETWORK DOMAINNAME(AF_INET)
         DOMAINNUMBER(2)
         MAXSOCKETS(10000)
         TYPE(CINET)
         INADDRANYPORT(4000)
         INADDRANYCOUNT(2000)
SUBFILESYSTYPE NAME(T03ATCP)
                TYPE(CINET)
                ENTRYPOINT(EZBPFINI)
                DEFAULT
SUBFILESYSTYPE NAME(T03BTCP)
                TYPE(CINET)
                ENTRYPOINT(BPXTIINT)
SUBFILESYSTYPE NAME(ANYNET03)
                TYPE(CINET)
                ENTRYPOINT(ISTOEPIT)
```

Figure 56. BPXPRM03 Member of PARMLIB

The NAME keyword in the SUBFILESYSTYPE statement (1) must be the name of the procedure that is used to start the AnyNet Sockets over SNA program, ISTSKDMN. The ENTRYPOINT keyword (2) must be coded as shown.

Having modifed the BPXPRMxx PARMLIB values, you must then define the SNA LUs to be used by Sockets over SNA. Sockets over SNA is a VTAM application which requires an application major node to be defined to VTAM. To save the effort involved in working out the LU names from the mapping algorithm, it is usual to define this major node using model applications such as that shown in Figure 57.

```
000100 SOS03A1 VBUILD  TYPE=APPL
000200 SX*     APPL    APPC=YES,
000300                 PARSESS=YES,
000400                 DSESLIM=10,
000500                 DMINWNL=5,
000600                 DMINWNR=5,
000700                 AUTOSES=0,
000800                 AUTH=(ACQ,PASS),
000900                 OPERCNOS=ALLOW,
001000                 ATNLOSS=ALL
```

Figure 57. Application Major Node for Sockets over SNA

This form of definition allows Sockets over SNA to define any LU whose name starts with SX, and therefore any IP host address within the Sockets over SNA subnetwork. The prefix (SX in this example) must match the LU name prefix defined to the address mapping algorithm.

In addition, as in any LU 6.2 environment, at least one mode entry must have been defined. By default, Sockets over SNA in OS/390 uses SNACKETS as the mode name. However, all the other current Sockets over SNA products use mode BLANK by default. To be compatible with the other platforms, you might want to change the default mode either in OS/390 or on the other platforms. The default mode can be defined in the SXMODE_DEFAULT statement in the ENVVAR data set. There you can also specify different modes for desired stream type Sockets applications. This is done with the statement SXMODEnn, where nn is the port number. Note that SNACKETS is in the IBM-supplied mode table in CS for OS/390 but BLANK is not.

If you have defined anything in the ENVVAR data set, you must make sure that the Sockets over SNA program ISTSKDMN is passed the name of the ENVVAR DD statement in the PARM value. During our testing we used the default setting wherever possible, so we did not code any environment variable definitions. Figure 97 on page 103 shows how an ENVVAR DD statement might be used.

When all the definitions above have been coded, you can start the AnyNet Sockets over SNA task. However, before you can actually set up any communications you need to run two more programs:

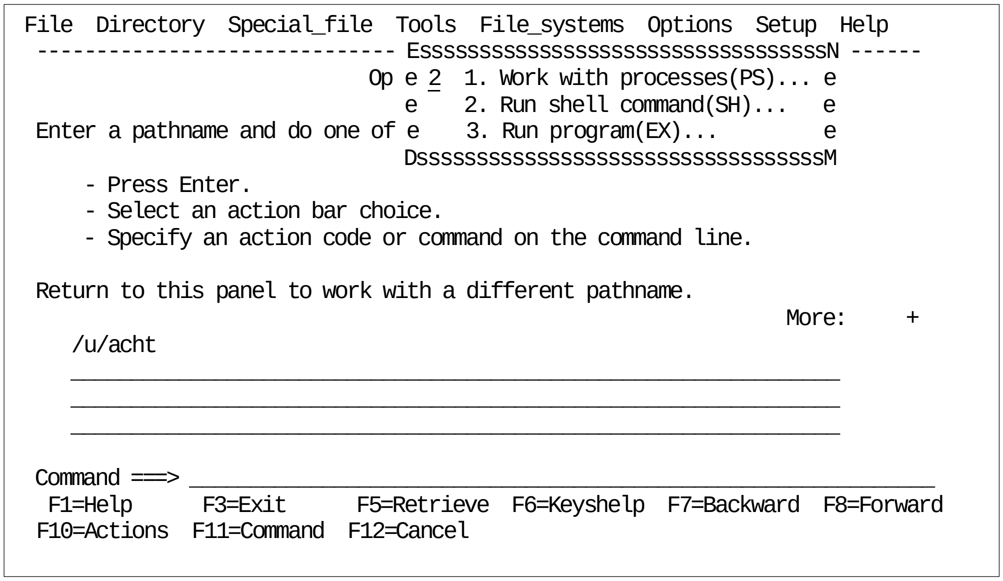
1. The address mapping program, ISTSKMAP. With this program you define either explicit or algorithmic LU name mappings for the IP addresses. We used, and recommend everybody to use, the algorithm. In this case the program needs to be run only once to provide the subnet address (with

mask) used for the virtual AnyNet IP network and the corresponding SNA network ID and the LU name seed.

2. The interface program, ISTSKIFC. This program initializes the sna0 interface to the TCP/IP stack and defines the IP address and network mask to be used by this interface. Only after running this can you start using Sockets over SNA functions.

3.2.2 Checking Connectivity

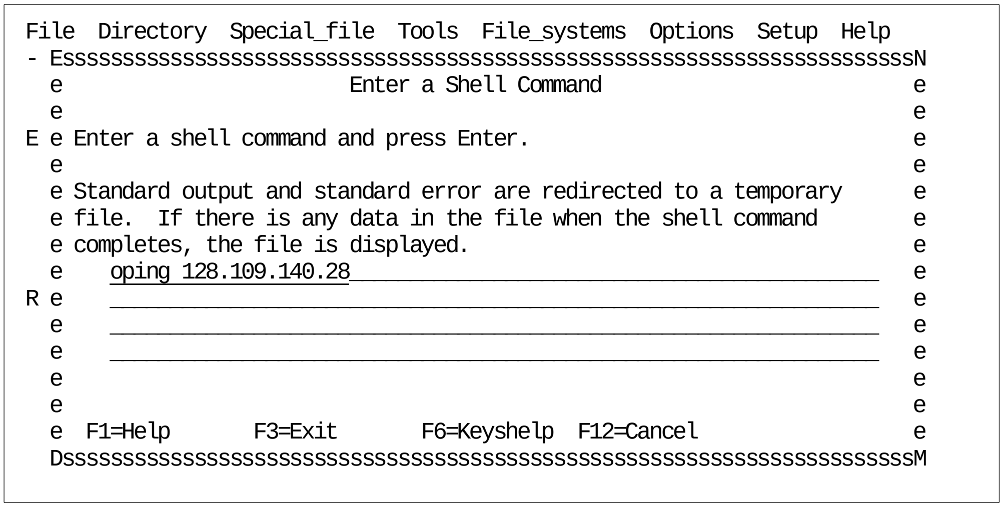
To test AnyNet connectivity, go to the UNIX Systems Services shell (enter `ish` from the TSO command prompt or use the ISPF menu), open Tools and select option 2 (Run shell command). Figure 58 shows the screen in question.



```
File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
 ------------------------------ EssssssssssssssssssssssssssssssssssN ------
                            Op e 2  1. Work with processes(PS)... e
                               e    2. Run shell command(SH)...   e
 Enter a pathname and do one of e    3. Run program(EX)...        e
                               DssssssssssssssssssssssssssssssssssM
     - Press Enter.
     - Select an action bar choice.
     - Specify an action code or command on the command line.

 Return to this panel to work with a different pathname.
                                                               More:     +
    /u/acht
    ________________________________________________________________
    ________________________________________________________________
    ________________________________________________________________

 Command ===> ______________________________________________________________
  F1=Help      F3=Exit      F5=Retrieve  F6=Keyshelp  F7=Backward  F8=Forward
 F10=Actions  F11=Command  F12=Cancel
```

Figure 58. UNIX Systems Services Menu Screen

Select 2 in the tools window, and you will get Figure 59 with a place to enter your command.

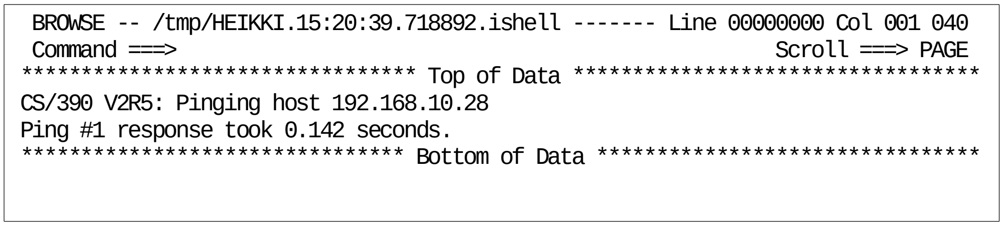


```
File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
- EssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
  e                        Enter a Shell Command                        e
  e                                                                     e
E e Enter a shell command and press Enter.                              e
  e                                                                     e
  e Standard output and standard error are redirected to a temporary    e
  e file.  If there is any data in the file when the shell command      e
  e completes, the file is displayed.                                   e
  e    oping 128.109.140.28____________________________________________  e
R e    ________________________________________________________________  e
  e    ________________________________________________________________  e
  e    ________________________________________________________________  e
  e                                                                     e
  e                                                                     e
  e  F1=Help       F3=Exit       F6=Keyshelp  F12=Cancel                e
  DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM
```

Figure 59. UNIX Systems Services Command Input

Because this version of AnyNet has changed the way it supports the Sockets APIs, you can no longer use the PING command. PING is still available, but only for native TCP/IP. With AnyNet, you have to use OPING, which runs under the UNIX Systems Services shell.

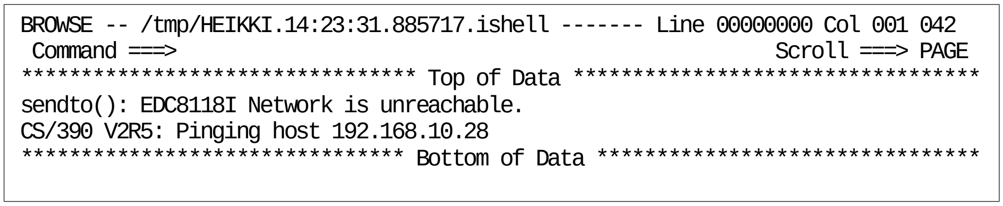
You enter the OPING command as shown, and after a short delay you will see the result as in Figure 60.



```
 BROWSE -- /tmp/HEIKKI.15:20:39.718892.ishell ------- Line 00000000 Col 001 040
 Command ===>                                                  Scroll ===> PAGE
********************************* Top of Data **********************************
CS/390 V2R5: Pinging host 192.168.10.28
Ping #1 response took 0.142 seconds.
******************************** Bottom of Data ********************************
```

Figure 60. OPING Command Output

If, on the other hand, you have not configured your network correctly, you are more likely to see the display shown in Figure 61.



```
BROWSE -- /tmp/HEIKKI.14:23:31.885717.ishell ------- Line 00000000 Col 001 042
 Command ===>                                                  Scroll ===> PAGE
********************************* Top of Data **********************************
sendto(): EDC8118I Network is unreachable.
CS/390 V2R5: Pinging host 192.168.10.28
******************************** Bottom of Data ********************************
```

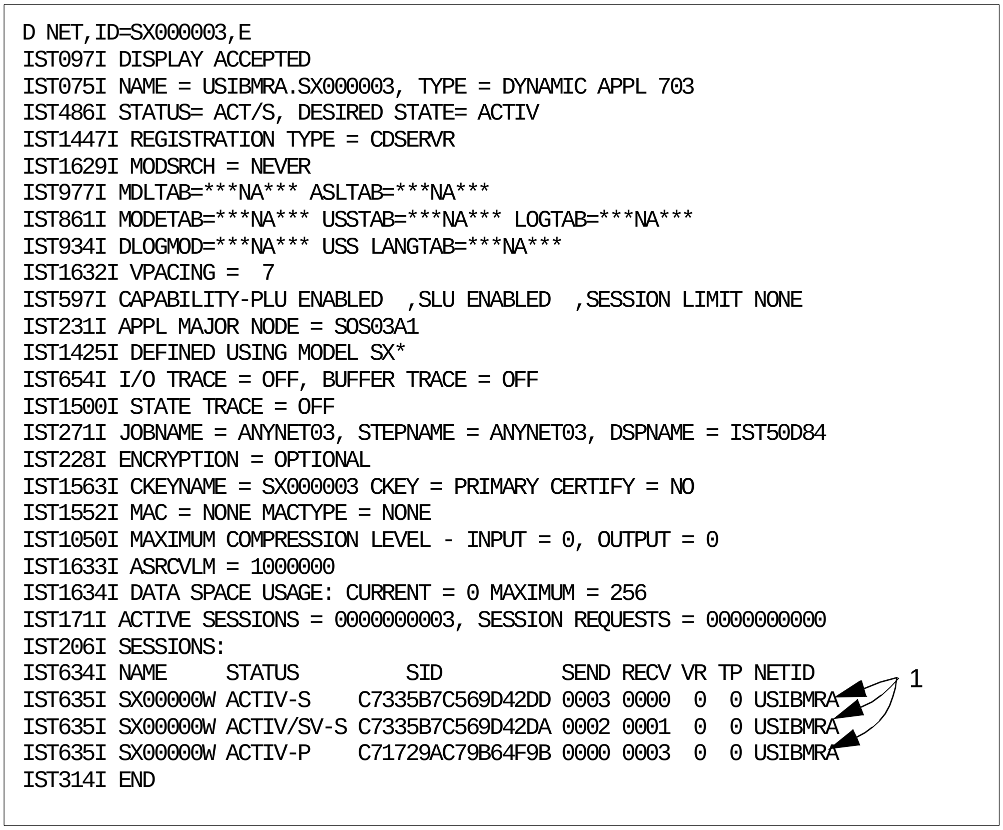
Figure 61. OPING Failed

When the OPING is issued (or, indeed, any remote IP host is contacted using Sockets over SNA) for the first time, it causes the following series of actions:

- Destination address mapping to an LU name.
- Session establishment to the remote LU. In fact there are three sessions: first an SNASVCMG session between the LUs for the change number of sessions (CNOS) processing, and then two productive sessions, both in half duplex mode. OPING (like all PINGs) is an ICMP application which uses raw Sockets and not stream Sockets.
- Actual transfer of the IP data to the remote LU, and over to AnyNet and the corresponding IP port.

For this reason, you can expect the first OPING to take a considerably longer time than subsequent OPINGs to the same destination.

The session status after the OPING is shown in the following VTAM display (Figure 62 on page 77).



```
D NET,ID=SX000003,E
IST097I DISPLAY ACCEPTED
IST075I NAME = USIBMRA.SX000003, TYPE = DYNAMIC APPL 703
IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
IST1447I REGISTRATION TYPE = CDSERVR
IST1629I MODSRCH = NEVER
IST977I MDLTAB=***NA*** ASLTAB=***NA***
IST861I MODETAB=***NA*** USSTAB=***NA*** LOGTAB=***NA***
IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
IST1632I VPACING =  7
IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT NONE
IST231I APPL MAJOR NODE = SOS03A1
IST1425I DEFINED USING MODEL SX*
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST271I JOBNAME = ANYNET03, STEPNAME = ANYNET03, DSPNAME = IST50D84
IST228I ENCRYPTION = OPTIONAL
IST1563I CKEYNAME = SX000003 CKEY = PRIMARY CERTIFY = NO
IST1552I MAC = NONE MACTYPE = NONE
IST1050I MAXIMUM COMPRESSION LEVEL - INPUT = 0, OUTPUT = 0
IST1633I ASRCVLM = 1000000
IST1634I DATA SPACE USAGE: CURRENT = 0 MAXIMUM = 256
IST171I ACTIVE SESSIONS = 0000000003, SESSION REQUESTS = 0000000000
IST206I SESSIONS:
IST634I NAME     STATUS         SID          SEND RECV VR TP NETID
IST635I SX00000W ACTIV-S    C7335B7C569D42DD 0003 0000  0  0 USIBMRA
IST635I SX00000W ACTIV/SV-S C7335B7C569D42DA 0002 0001  0  0 USIBMRA
IST635I SX00000W ACTIV-P    C71729AC79B64F9B 0000 0003  0  0 USIBMRA
IST314I END
```

Figure 62. VTAM Display of OPING Sessions

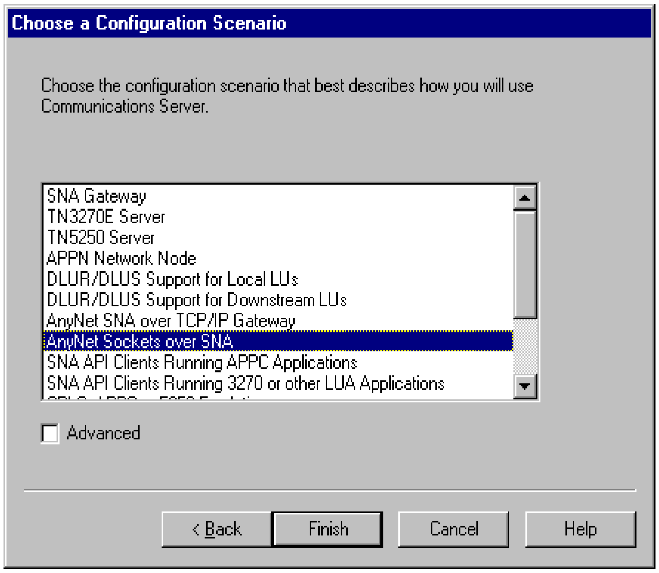
In the figure you can see the three sessions (1) between this application (SX000003) and the partner LU SX00000W. These are the algorithmically generated LU names for the IP addresses 192.168.10.3 and 192.168.10.28. SX000003 was dynamically created by VTAM from the application model major node (Figure 57 on page 74), while SX00000W is a dynamic cross-domain resource which VTAM can create without any predefinition.

3.3 AnyNet Sockets over SNA for CS/NT Access Node

This section describes how to define and operate the Sockets over SNA Access node environment on Communications Server for Windows NT, Version 6. Exactly the same description is applicable to the Access Feature which comprises part of Personal Communications for Windows NT. The Access Feature is a subset of CS/NT with certain functions (such as APPN network node, TN3270 server, SNA gateway and Sockets over SNA gateway) missing.

3.3.1 Configuration and Definition

Start the SNA Node Configuration function from the desktop, and select a configuration scenario that fits your requirements. With Enterprise Extender we selected **Advanced** to receive a menu (Figure 15 on page 39) that covers all the possibilities but provides no extra guidance. For our AnyNet scenarios we selected **AnyNet Sockets over SNA** (Figure 63 on page 78). This guides you through a series of steps that you must follow to ensure a successful configuration of Sockets over SNA. Please see Figure 64 on page 78.

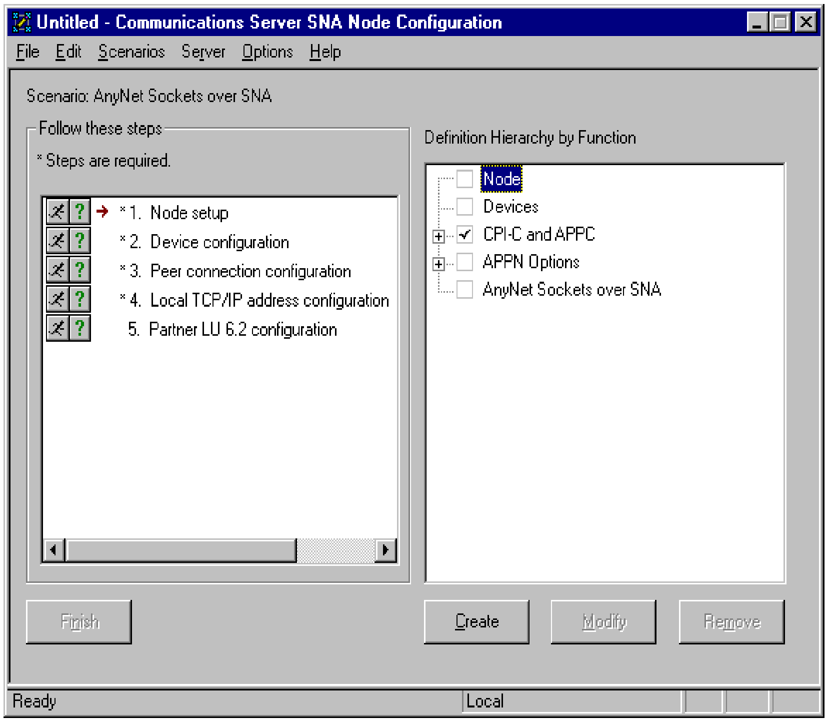

Choose a Configuration Scenario

Choose the configuration scenario that best describes how you will use Communications Server.

- SNA Gateway
- TN3270E Server
- TN5250 Server
- APPN Network Node
- DLUR/DLUS Support for Local LUs
- DLUR/DLUS Support for Downstream LUs
- AnyNet SNA over TCP/IP Gateway
- **AnyNet Sockets over SNA**
- SNA API Clients Running APPC Applications
- SNA API Clients Running 3270 or other LUA Applications

☐ Advanced

Figure 63. Configuration Selection: AnyNet Sockets over SNA on CS/NT


Untitled - Communications Server SNA Node Configuration

Scenario: AnyNet Sockets over SNA

Follow these steps

* Steps are required.

- * 1. Node setup
- * 2. Device configuration
- * 3. Peer connection configuration
- * 4. Local TCP/IP address configuration
- 5. Partner LU 6.2 configuration

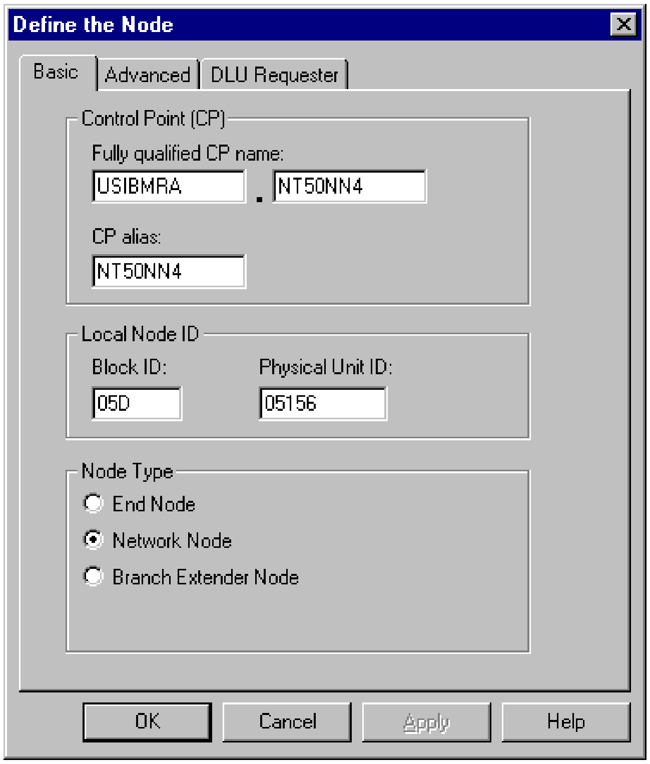
Definition Hierarchy by Function

- ☐ Node
- ☐ Devices
- ☑ CPI-C and APPC
- ☐ APPN Options
- ☐ AnyNet Sockets over SNA

Figure 64. Communications Server SNA Node Configuration

Just as with Enterprise Extender, you have to configure the node, the devices (ports) and the link stations (connections) as a minimum. To this you must add the AnyNet-specific definitions such as the mapping requirements. However, in the guided path (as opposed to the advanced path) you invoke in sequence the steps shown on the left-hand panel in Figure 64 rather than worry about which of the right-hand panel options you need to select.

First, you define the node itself (select **Node Setup**), as in Figure 65.



Control Point (CP)

Fully qualified CP name: USIBMRA . NT50NN4

CP alias: NT50NN4

Local Node ID

Block ID: 05D

Physical Unit ID: 05156

Node Type

○ End Node

● Network Node

○ Branch Extender Node

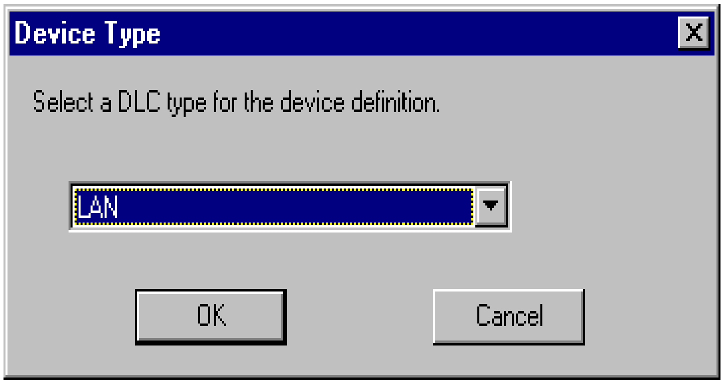
Figure 65. CS/NT Node Definition

With AnyNet Sockets over SNA the SNA part of the configuration is normal, since the transport network itself is pure SNA. The node definition shows the usual parameters:

- The fully qualified CP name, USIBMRA.NT50NN4
- The CP alias, NT50NN4
- The local node ID, 05D-05156
- The node type (APPN network node in our case, as we were to use the machine later in configurations where intermediate routing was required). The AnyNet access node function does not require network node capability.

The **DLU Requester** part of this panel is not relevant to Sockets over SNA, which uses only independent LU 6.2 sessions.

Next, you have to configure a port so invoke the **Device Configuration** option from the Node Configuration panel. This is real SNA transport so select a real device type (LAN in our example, as in Figure 66 on page 80) and click **OK**.



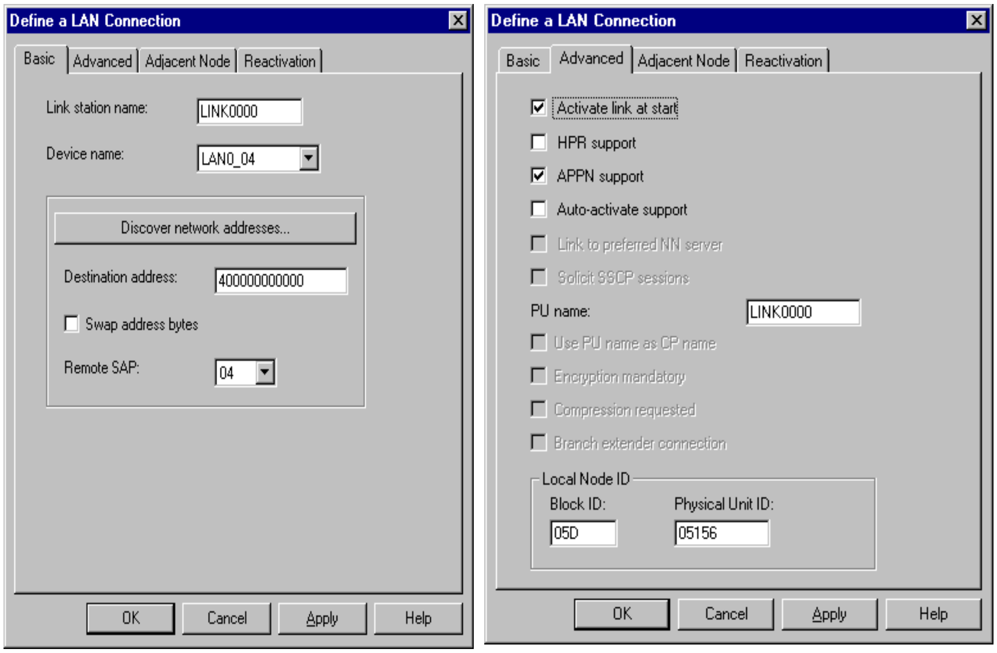
Select a DLC type for the device definition.

LAN

Figure 66. Device Type Selection

Next, select **Peer connection configuration** from the main panel (Figure 64 on page 78). This results in the panel entitled Define a LAN Connection which has four parts. The complete panel is shown in Figure 67 and Figure 68 on page 81, but there is very little to define if you are happy with the defaults as we were. The main things are to ensure that:

1. The **destination** (MAC/SAP) **address** is configured on the **Basic** tab, if this node is to establish the connection.
2. The **APPN Support** box is checked in the **Advanced** tab if you want CP-CP sessions to be supported on this link.
3. The **Device Name** on the **Basic** tab is the name you gave to your port in the previous configuration step. If you have only one port defined then the correct name is automatically presented and you have to do nothing.



Link station name: LINK0000

Device name: LAN0_04

Destination address: 400000000000

□ Swap address bytes

Remote SAP: 04

☑ Activate link at start

□ HPR support

☑ APPN support

□ Auto-activate support

□ Link to preferred NN server

□ Solicit SSCP sessions

PU name: LINK0000

□ Use PU name as CP name

□ Encryption mandatory

□ Compression requested

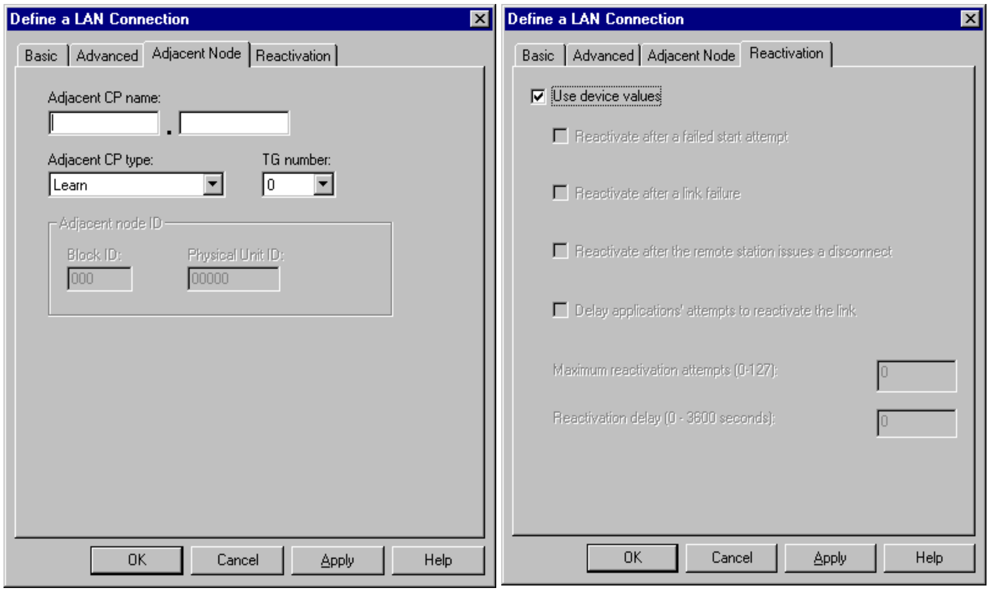
□ Branch extender connection

Local Node ID

Block ID: 05D

Physical Unit ID: 05156

Figure 67. Define a LAN Connection (Basic and Advanced)



Adjacent CP name: .

Adjacent CP type: Learn

TG number: 0

Adjacent node ID

Block ID: 000

Physical Unit ID: 00000

☑ Use device values

□ Reactivate after a failed start attempt

□ Reactivate after a link failure

□ Reactivate after the remote station issues a disconnect

□ Delay applications' attempts to reactivate the link

Maximum reactivation attempts (0-127): 0

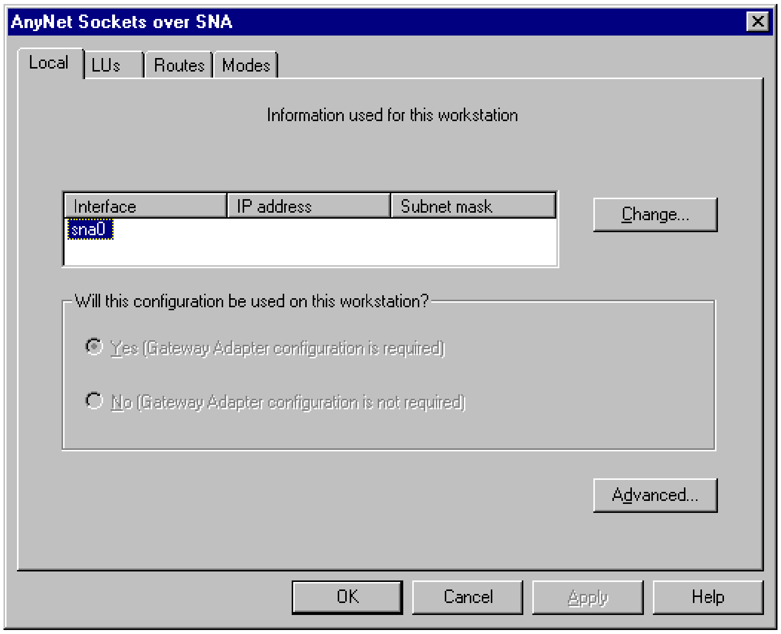
Reactivation delay (0 - 3600 seconds): 0

Figure 68. Define a LAN Connection (Adjacent Node and Reactivation)

To complete the definition and to store the configuration file, click **OK.**

The fourth step is to configure the AnyNet-specific values using the **TCP/IP Address Configuration** option on the main panel. This too has four parts: **Local**, **LUs**, **Routes** and **Modes**. So far everything has been straightforward but now some interesting things start to happen.

Figure 69 shows the **Local** part of the AnyNet configuration panel. By clicking **Change** you can configure the IP address and the subnet mask to be used for the sna0 interface.



Information used for this workstation

Interface	IP address	Subnet mask
sna0		

Will this configuration be used on this workstation?

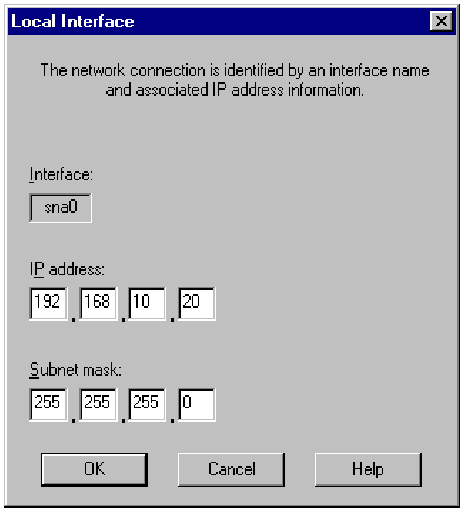
● Yes (Gateway Adapter configuration is required)

○ No (Gateway Adapter configuration is not required)

Figure 69. AnyNet Local Configuration Panel

The bottom half of the panel varies according to the state of your configuration, so do not expect yours to look exactly the same as this one. For it is at this stage that the CS/NT configuration process defines the sna0 interface. To do this it invokes the Windows NT adapter configuration process. Once the interface has been defined you can only delete it using the NT process, not the CS/NT configuration. To see what adapters have been defined (or to delete them), use **My Computer**, **Control Panel**, **Network** and **Adapters**.

Figure 70 shows the panel you see after clicking **Change** on the Local panel. Enter the IP address and subnet mask to be used for the sna0 interface.



Local Interface

The network connection is identified by an interface name and associated IP address information.

Interface: sna0

IP address: 192 . 168 . 10 . 20

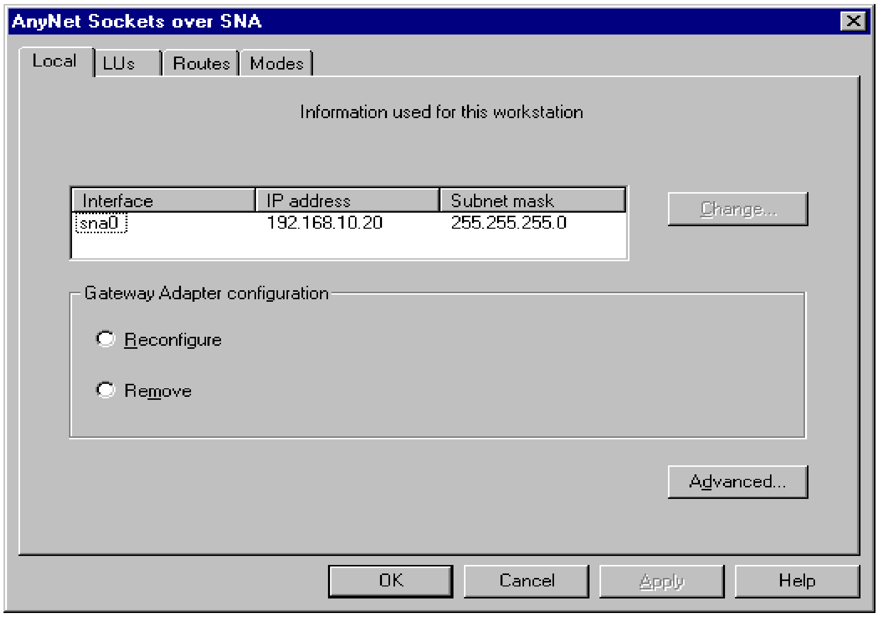
Subnet mask: 255 . 255 . 255 . 0

Figure 70. sna0 Interface Configuration

After you have entered your values and clicked **OK**, CS/NT invokes the Windows NT adapter configuration. CS/NT displays instructions on what you have to do, but the essential steps are:

1. Enter the same IP address and subnet mask for sna0 as you have just entered on the CS/NT configuration panel (Figure 70).
2. Click **Have Disk** when prompted to select an adapter driver.
3. Point to the directory in which the AnyNet driver is installed. This directory should contain the files ANYSOSGW.SYS and OEMSETUP.INF.
4. Do *not* restart NT, but exit and return to the CS/NT configuration.

Now the sna0 configuration is complete and you get the Local panel again with the new information filled in, as in Figure 71 on page 83.



Information used for this workstation

Interface	IP address	Subnet mask
sna0	192.168.10.20	255.255.255.0

Gateway Adapter configuration

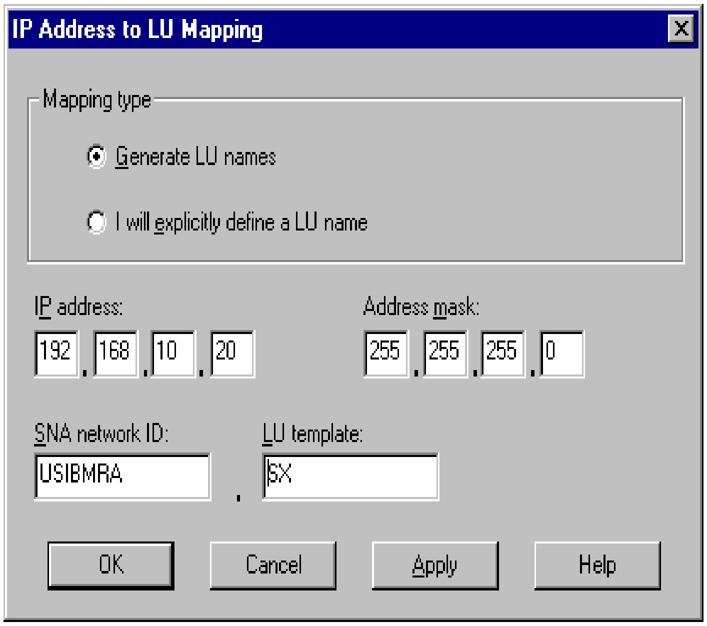
○ Reconfigure

○ Remove

Figure 71. AnyNet Local Configuration (Again)

The **Advanced** option on this panel prompts you to change the default Class A network address that AnyNet uses internally. It will never attempt to contact a host in this network, but if your installation uses addresses in that network you will need to change it. The default is 125.

Now it is time to define the LU-to-IP address mapping, so select the LUs panel and click **New** to see the panel in Figure 72.



Mapping type

● Generate LU names

○ I will explicitly define a LU name

IP address: 192 . 168 . 10 . 20

Address mask: 255 . 255 . 255 . 0

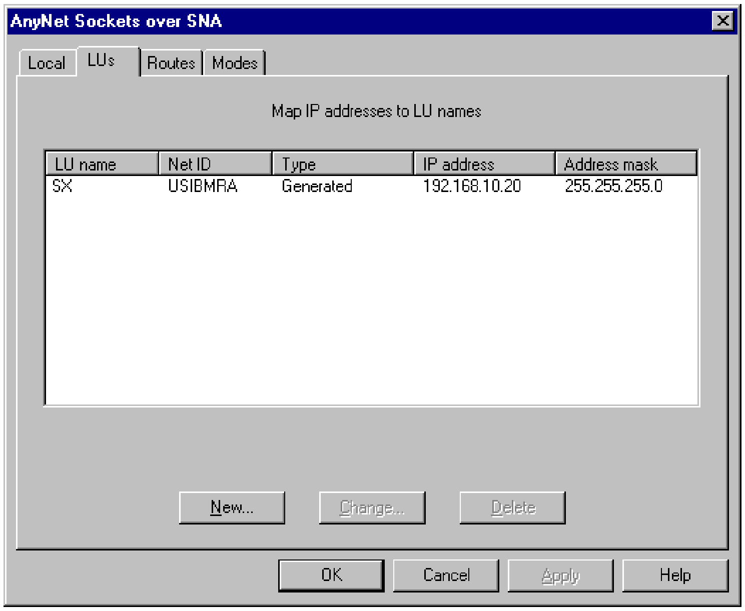
SNA network ID: USIBMRA .

LU template: $X

Figure 72. IP Address to LU Mapping Panel

CS/NT configuration allows you to define the mapping manually, or to let the algorithm do its work. If you choose the algorithm (**Generate LU names**) then the sna0 IP network address and subnet mask are already filled in, and you just have to enter the LU name seed. For other Sockets over SNA subnetworks that you wish to reach over an SNA connection, you must enter all the information in the appropriate LU mapping panels.

When you have entered the mapping definitions the LUs panel looks like Figure 73.



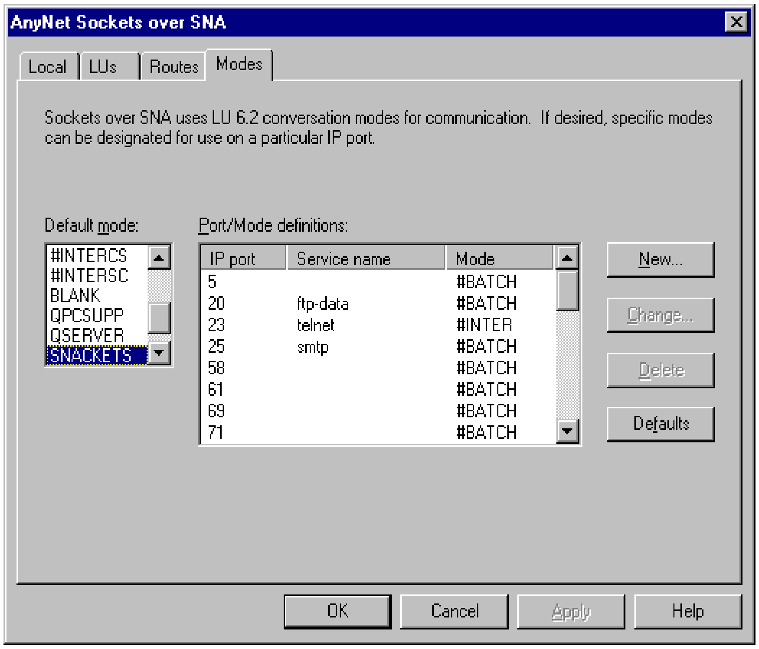
LU name	Net ID	Type	IP address	Address mask
SX	USIBMRA	Generated	192.168.10.20	255.255.255.0

Figure 73. AnyNet Sockets over SNA - LUs

Next comes the Routes panel. Just as with a normal IP interface, you need fill this in only if you are to access IP addresses in a remote subnetwork. In that case you will enter (at least) the address of the default router, which will actually be the Sockets over SNA gateway that takes you to the real IP world. An example of the Routes panel is displayed in the Gateway section in Figure 77 on page 88.

Finally comes the Modes panel. By default the mode used for Sockets over SNA is BLANK, but we wished to use SNACKETS to be compatible with CS for OS/390. Since SNACKETS is not one of the modes supplied with CS/NT, we had to create it using **CPI-C and APPC**, **Modes** and **Create**. The only change you might wish to make is to alter the default class of service from #CONNECT which gives you medium priority. Suitable alternatives are #INTER (high priority for interactive work) or #BATCH (low priority for data transfer).

Once the SNACKETS mode has been created it appears in the Modes panel as shown in Figure 74 on page 85.



Sockets over SNA uses LU 6.2 conversation modes for communication. If desired, specific modes can be designated for use on a particular IP port.

Default mode:

#INTERCS
#INTERSC
BLANK
QPCSUPP
QSERVER
SNACKETS

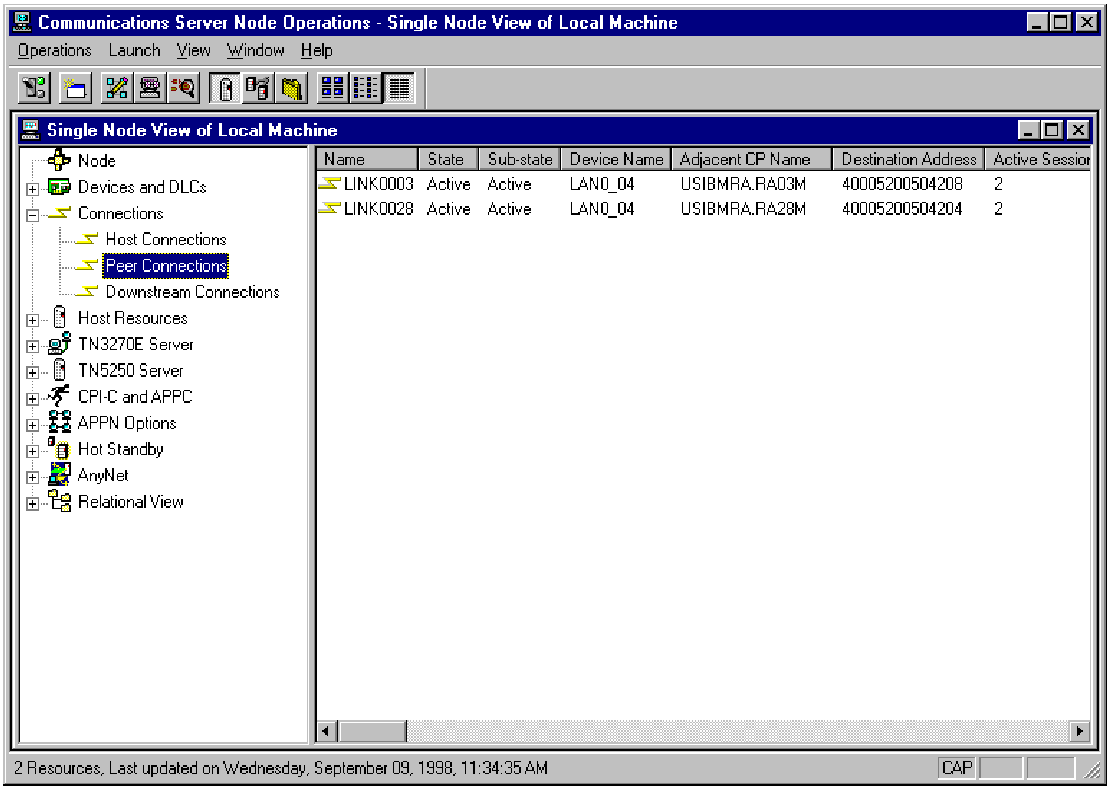
Port/Mode definitions:

IP port	Service name	Mode
5		#BATCH
20	ftp-data	#BATCH
23	telnet	#INTER
25	smtp	#BATCH
58		#BATCH
61		#BATCH
69		#BATCH
71		#BATCH

Figure 74. AnyNet Sockets over SNA - Modes

3.3.2 Node Operation

To start or stop the APPN node, CS/NT has the Node Operations function. When this is started it displays the Single Node View of Local Machine panel as shown in Figure 75 on page 86. It displays the current status of all the things you have defined in the Node Configuration, in much the same structure. This example shows two real LAN connections; more examples can be found in the working scenarios later in this chapter.



Name	State	Sub-state	Device Name	Adjacent CP Name	Destination Address	Active Sessior
LINK0003	Active	Active	LAN0_04	USIBMRA.RA03M	40005200504208	2
LINK0028	Active	Active	LAN0_04	USIBMRA.RA28M	40005200504204	2

2 Resources, Last updated on Wednesday, September 09, 1998, 11:34:35 AM

Figure 75. Node Operations Panel

3.4 AnyNet Sockets over SNA CS/NT Gateway

The difference between the access node and the gateway node in AnyNet Sockets over SNA is simply that the gateway node can act as a router between the sna0 interface and the other IP interfaces on the node, whereas the access node can only exchange IP datagrams across the Sockets interface. Indeed, an access node is likely to have no real IP interfaces at all.

The configuration of the gateway environment, therefore, is very similar to that of the access node environment. However, there is one very important thing to remember and that is that the gateway is an IP router. Therefore, you must enable **IP Forwarding** in the TCP/IP Routing Properties on Windows NT. This panel can be reached by invoking **Network Neighborhood Properties** then **Protocols** and **TCP/IP**.

3.4.1 Configuration and Definition

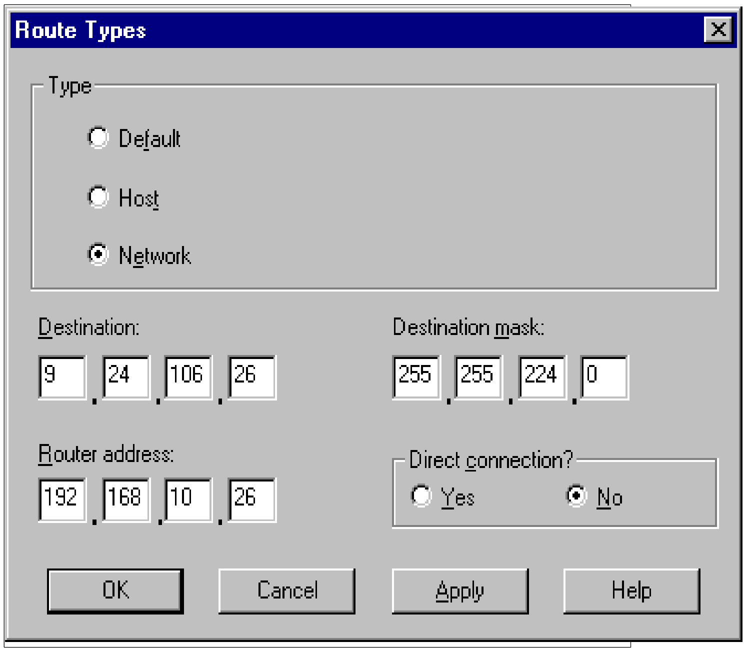
The first few steps of the gateway node configuration are the same as those described in “AnyNet Sockets over SNA for CS/NT Access Node” on page 77. The Node Setup, Device Configuration and Peer Connection procedures are therefore not repeated here.

In the TCP/IP address configuration procedure, the Local, LUs and Modes panels are treated similarly because the requirements are the same. The gateway configuration requires an sna0 adapter, some LU mapping instructions and some modes to use for its SNA sessions just as does the access node. The Routes panels is where the biggest difference is likely to occur.

Routes must be explicitly defined here if:

- This node will route data through another gateway or router. In particular, if this node is part of a back-to-back gateway configuration the route to the partner gateway must be defined.
- This node will route data directly to another Sockets over SNA node that is in a different IP subnetwork from that of the sna0 interface. Sockets over SNA does not know whether such a node should be reached by SNA or by native IP means.

If you select the Routes tab and click **New**, Figure 76 appears.



Route Types

Type

○ Default

○ Host

● Network

Destination: 9 . 24 . 106 . 26

Destination mask: 255 . 255 . 224 . 0

Router address: 192 . 168 . 10 . 26

Direct connection? ○ Yes ● No

Figure 76. AnyNet Sockets over SNA - Route Part

The choices of route type available here are:

- Default.

  When no host or network route matches a given destination, the default route (if specified) is used. Data is sent to the gateway or router whose IP address is defined here.

- Host.

  A route to a specific host (defined in the **Destination** field) is available through the gateway or router whose IP address is given in the **Router address** field.

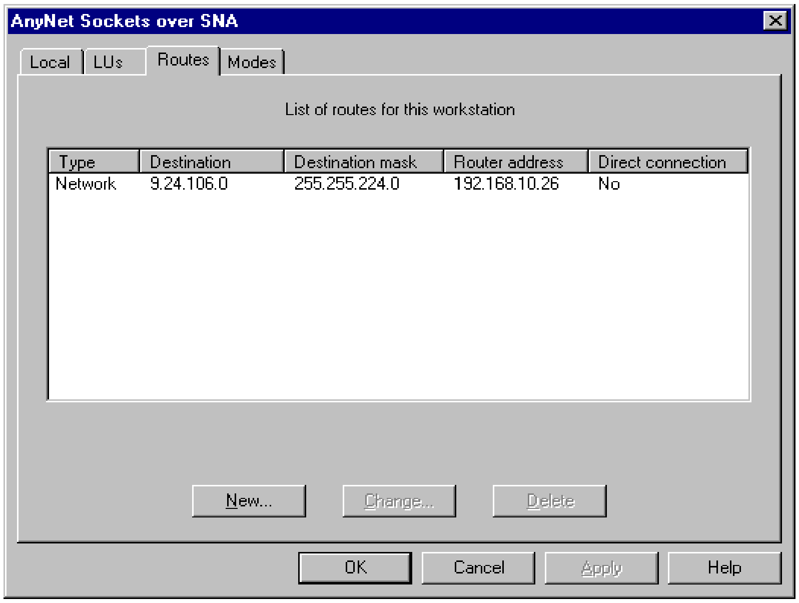
- Network.

  A route to a particular network (defined in the **Destination** and **Destination mask** fields) is available through the gateway or router whose IP address is in the **Router address** field.

This is, in fact, the same as any workstation TCP/IP route configuration. You just have to remember that Sockets over SNA has an extra virtual IP network on interface sna0, and it has to decide whether to send a particular packet over sna0 or to give it to a real IP interface on the same machine.

A **direct connection** is one to another Sockets over SNA node, reached via SNA but in a different IP subnetwork. We did not use any of these in our testing.

When you have defined the routes and clicked **OK** you will see the Routes panel as in Figure 77.



List of routes for this workstation

Type	Destination	Destination mask	Router address	Direct connection
Network	9.24.106.0	255.255.224.0	192.168.10.26	No

Figure 77. AnyNet Sockets over SNA - Routes

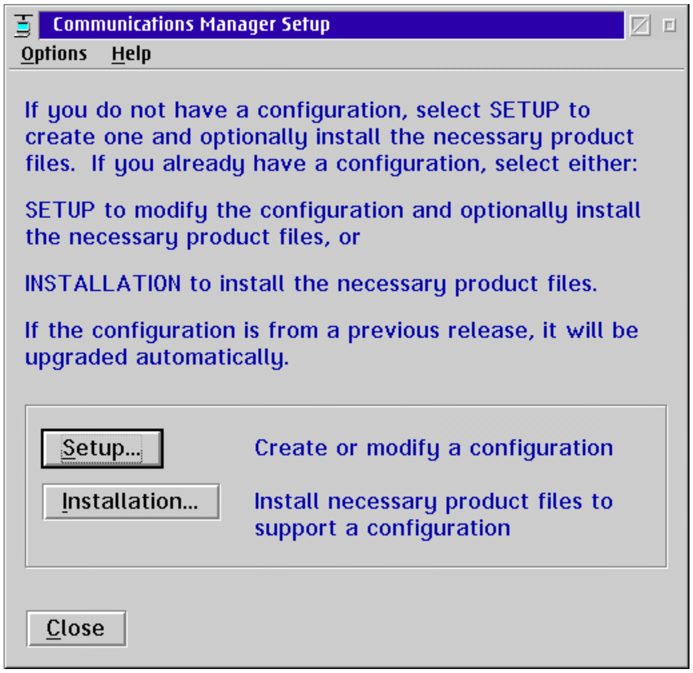
3.5 AnyNet Sockets over SNA CS/2 Access Node

This section describes how to define and operate the AnyNet Sockets over SNA access node on Communications Server for OS/2, Version 5. Although the principles of AnyNet are the same as for CS/NT, the definition and operation methods are very different on the two platforms.

The same considerations apply to Personal Communications for OS/2 as to PComm for Windows NT. PComm for OS/2 includes the Access Feature which is a subset of CS/2 without some of the functions, in particular the Sockets over SNA gateway.

3.5.1 Configuration and Definition

As with CS/NT, there are several ways of configuring CS/2, a detailed comparison of which is beyond the scope of this book. All begin by invocation of the Communications Manager Setup function, whose initial screen is shown in Figure 78 on page 89. Select **Setup** from this panel to see the Open Configuration panel (Figure 79 on page 89). Enter the name of your desired configuration file and click **OK**. Now you are faced with the main Configuration Definition panel as shown in Figure 80 on page 90.



Communications Manager Setup

If you do not have a configuration, select SETUP to create one and optionally install the necessary product files. If you already have a configuration, select either:

SETUP to modify the configuration and optionally install the necessary product files, or

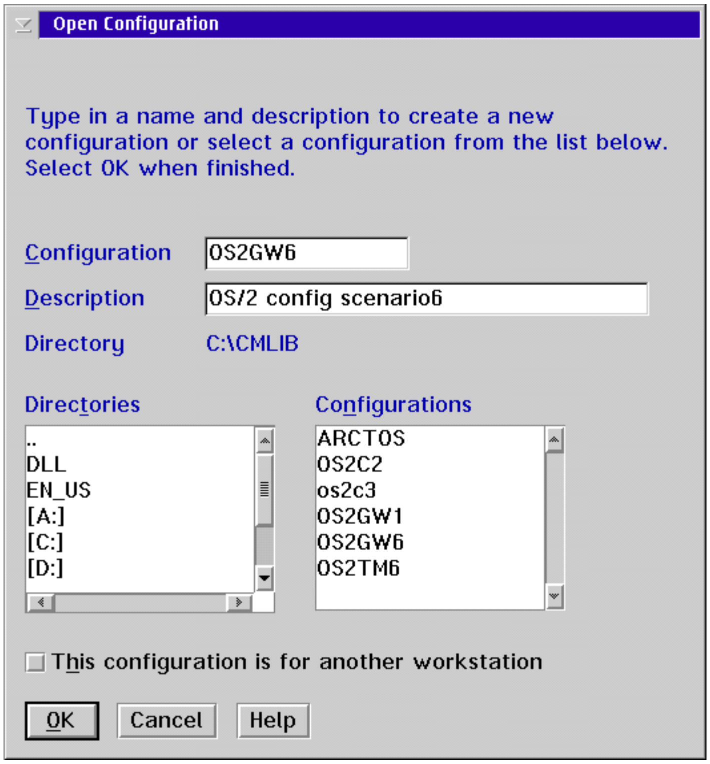
INSTALLATION to install the necessary product files.

If the configuration is from a previous release, it will be upgraded automatically.

Create or modify a configuration

Install necessary product files to support a configuration

Figure 78. Communications Manager Setup Initial Screen



Open Configuration

Type in a name and description to create a new configuration or select a configuration from the list below. Select OK when finished.

Configuration: OS2GW6

Description: OS/2 config scenario6

Directory: C:\CMLIB

Directories:

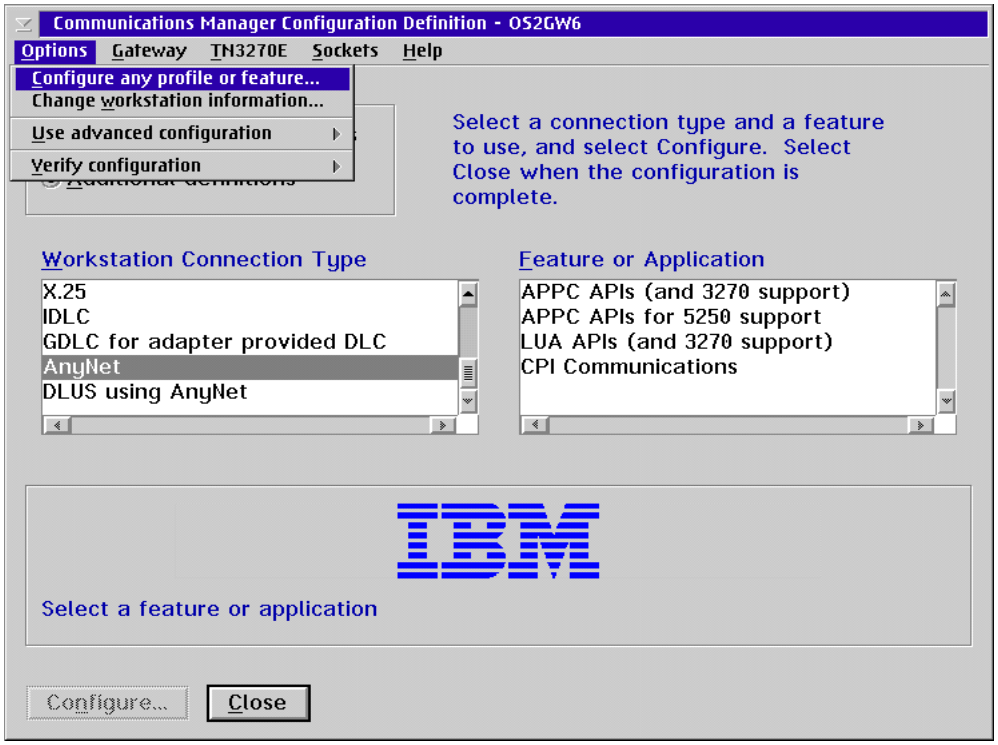
..
DLL
EN_US
[A:]
[C:]
[D:]

Configurations:

ARCTOS
OS2C2
os2c3
OS2GW1
OS2GW6
OS2TM6

□ This configuration is for another workstation

Figure 79. Open Configuration Panel



Communications Manager Configuration Definition - OS2GW6
Options
Gateway
TN3270E
Sockets
Help
Configure any profile or feature...
Change workstation information...
Use advanced configuration
Verify configuration
Workstation Connection Type
X.25
IDLC
GDLC for adapter provided DLC
AnyNet
DLUS using AnyNet
Feature or Application
APPC APIs (and 3270 support)
APPC APIs for 5250 support
LUA APIs (and 3270 support)
CPI Communications
IBM
Configure...
Close

Select a connection type and a feature to use, and select Configure. Select Close when the configuration is complete.

Select a feature or application

Figure 80. Communications Manager Configuration Definition

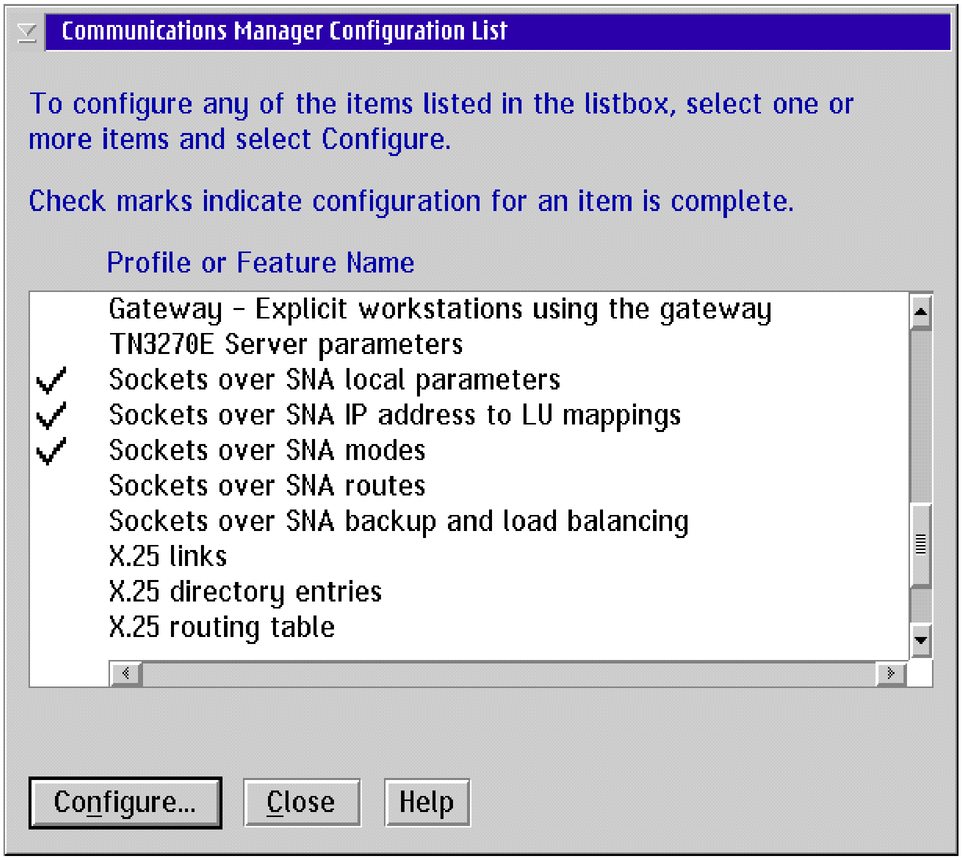
The AnyNet option (shown highlighted) actually refers to AnyNet SNA over IP, while CS/2 uses the term Sockets over SNA to mean the function in which we are currently interested. There are two ways to configure AnyNet Sockets over SNA from here:

1. Select **Options** then **Configure any profile or feature.** This presents you with a complete list of CS/2 functions from which you select the ones you want. This works in a similar fashion to the Advanced option on CS/NT.
2. Configure the real SNA environment in the normal way, then select **Sockets** and **Configure** from the menu bar. This presents you with a list of the Sockets over SNA features that you need to add to the basic CS/2 SNA configuration.

The features of particular interest to you if you are configuring a Sockets over SNA access node on a simple LAN-attached workstation (as in the example depicted in "AnyNet Sockets over SNA for CS/NT Access Node" on page 77) are:

- SNA local node characteristics
- DLC - whichever attachment you have on the workstation
- SNA connections - to define your real links to the SNA network
- SNA features - to define the SNACKETS mode if you wish to use it
- Sockets over SNA Local parameters
- Sockets over SNA IP address to LU name mappings
- Sockets over SNA modes
- Sockets over SNA routes
- Sockets over SNA backup and load balancing

As you can see, these correspond very closely with the steps taken for the CS/NT configuration; the only options not supported by CS/NT are backup and load balancing. Whichever method you choose to define the Sockets over SNA environment, the above options are presented in the Configuration List panel as shown in Figure 81.



Communications Manager Configuration List

To configure any of the items listed in the listbox, select one or more items and select Configure.

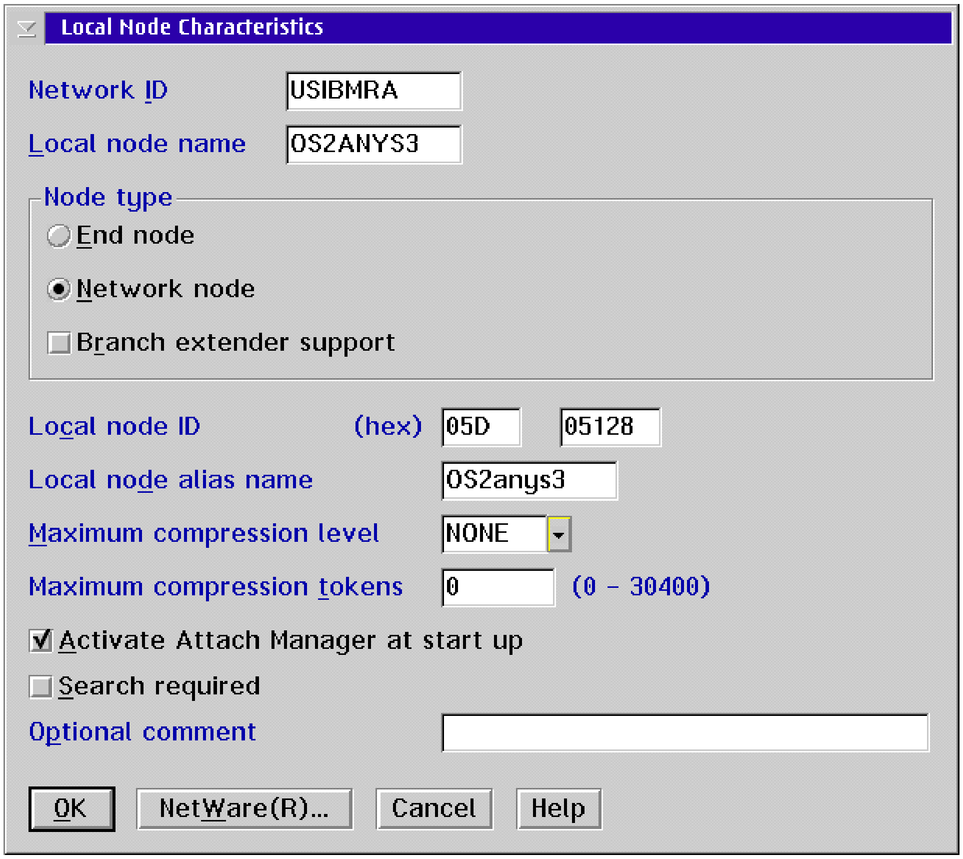
Check marks indicate configuration for an item is complete.

Profile or Feature Name

Gateway - Explicit workstations using the gateway
TN3270E Server parameters
✓ Sockets over SNA local parameters
✓ Sockets over SNA IP address to LU mappings
✓ Sockets over SNA modes
Sockets over SNA routes
Sockets over SNA backup and load balancing
X.25 links
X.25 directory entries
X.25 routing table

Figure 81. CS/2 Configuration List

Figure 82 on page 92 shows the configuration of the local SNA node, which you invoke by selecting **SNA local node characteristics** from the Configuration List.



Local Node Characteristics

Network ID: USIBMRA

Local node name: OS2ANYS3

Node type

○ End node

● Network node

□ Branch extender support

Local node ID (hex): 05D 05128

Local node alias name: OS2anys3

Maximum compression level: NONE

Maximum compression tokens: 0 (0 - 30400)

☑ Activate Attach Manager at start up

□ Search required

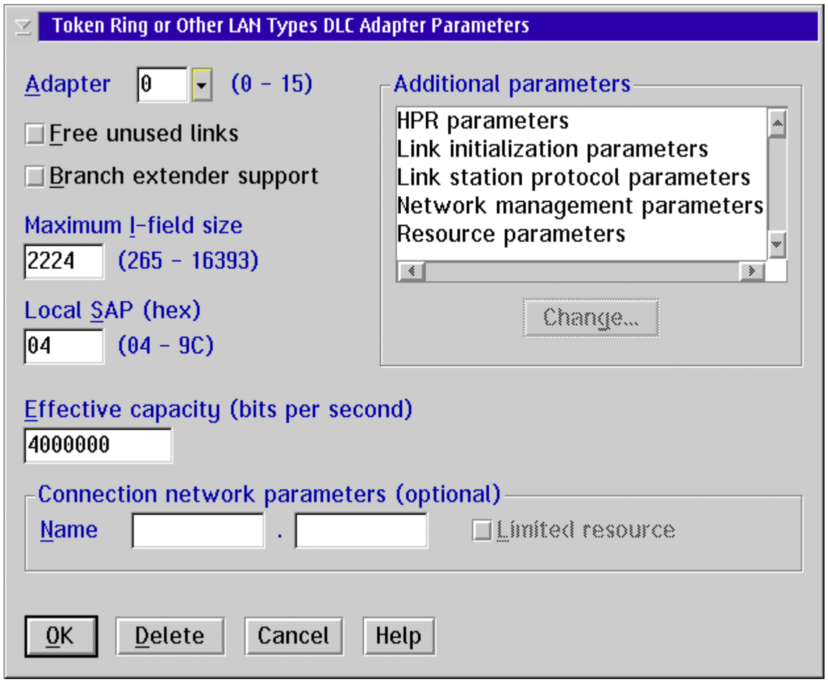
Optional comment:

Figure 82. Local SNA Node Definition

In this example we have defined:

- The fully qualified CP name, USIBMRA.OS2ANYS3
- The CP alias, OS2ANYS3
- The local node ID, 05D-05128
- The node type, an APPN network node

Next, you should configure the port so select the appropriate **DLC** option to define the adapter. Figure 83 on page 93 shows the token-ring port we defined.


Token Ring or Other LAN Types DLC Adapter Parameters

Adapter: 0 (0 - 15)

□ Free unused links

□ Branch extender support

Maximum I-field size: 2224 (265 - 16393)

Local SAP (hex): 04 (04 - 9C)

Effective capacity (bits per second): 4000000

Connection network parameters (optional)

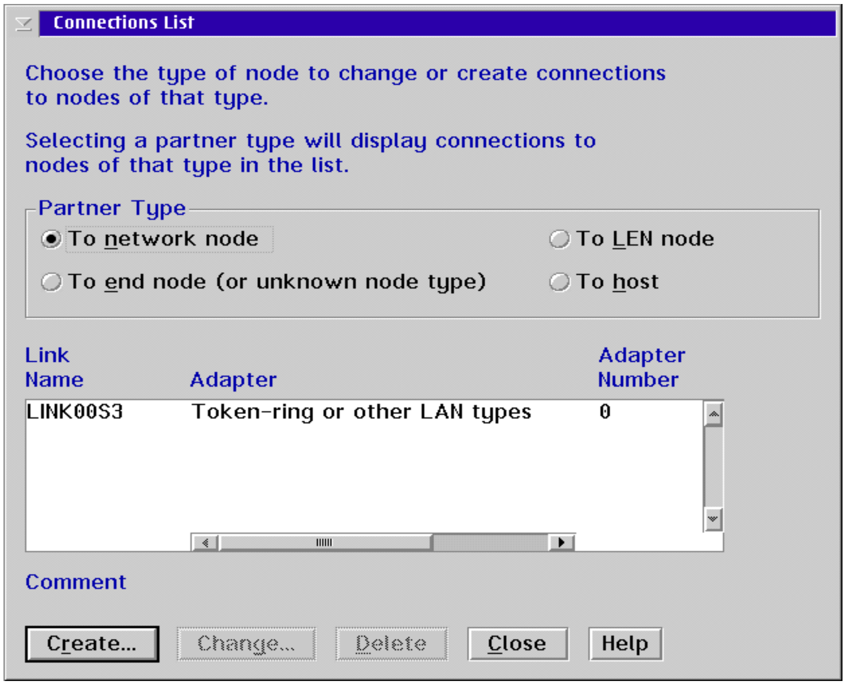
Name: .

□ Limited resource

Additional parameters
HPR parameters
Link initialization parameters
Link station protocol parameters
Network management parameters
Resource parameters
Change...
OK
Delete
Cancel
Help

Figure 83. Token-Ring DLC Definition

Here you define the port-level parameters such as branch extender and HPR support, many of which can be overridden at the link station (connection) level. The connections to other nodes can be defined dynamically (if the partner node initiates the connection) or manually by selecting **SNA Connections** as shown in Figure 84.


Connections List

Choose the type of node to change or create connections to nodes of that type.

Selecting a partner type will display connections to nodes of that type in the list.

Partner Type

● To network node	○ To LEN node
○ To end node (or unknown node type)	○ To host

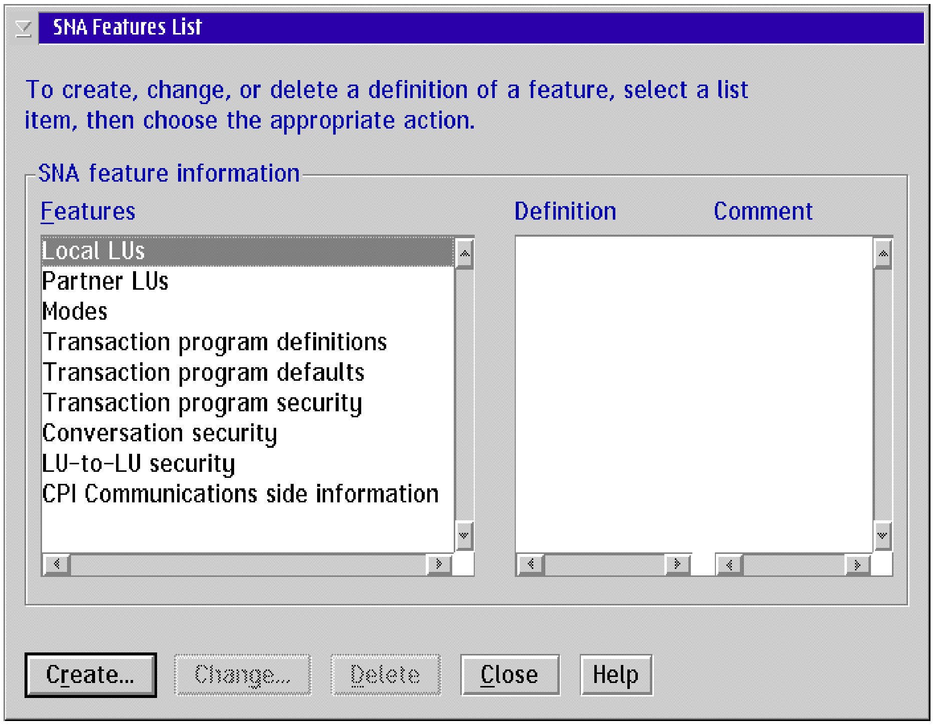
Link Name	Adapter	Adapter Number
LINK00S3	Token-ring or other LAN types	0

Comment

Figure 84. SNA Connections List Panel

Here you can select a partner type and click **Create** or **Change** to define the type of connection you want. The partner node types are more of a guide than a requirement, since you can select **to network node** and then **Solicit SSCP session**, and end up with the same result as selecting **to host** and then **APPN support**. Our example defines an NN-to-NN connection. On an EN-to-NN connection it is usual to define the connection at the EN end only since the EN is responsible for choosing the NN(s) from which it may require network services.

The final native SNA-related option you may wish to configure in a Sockets environment is the SNACKETS mode. This can be defined by selecting **SNA features** from the configuration list, as shown in Figure 85.



SNA Features List

To create, change, or delete a definition of a feature, select a list item, then choose the appropriate action.

SNA feature information

Features

- Local LUs
- Partner LUs
- Modes
- Transaction program definitions
- Transaction program defaults
- Transaction program security
- Conversation security
- LU-to-LU security
- CPI Communications side information

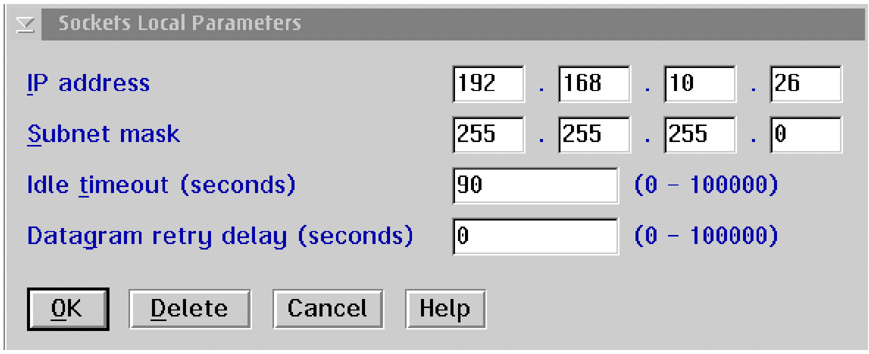
Definition

Comment

Figure 85. SNA Features Panel

Select **Modes** and **Create** and enter the parameters you require. Usually only the class of service is important.

Now you must turn to the unique Sockets over SNA configuration steps. The first panel is the **Sockets over SNA Local Parameters** panel as selected from the Configuration List and shown in Figure 86 on page 95.



Sockets Local Parameters

IP address: 192 . 168 . 10 . 26

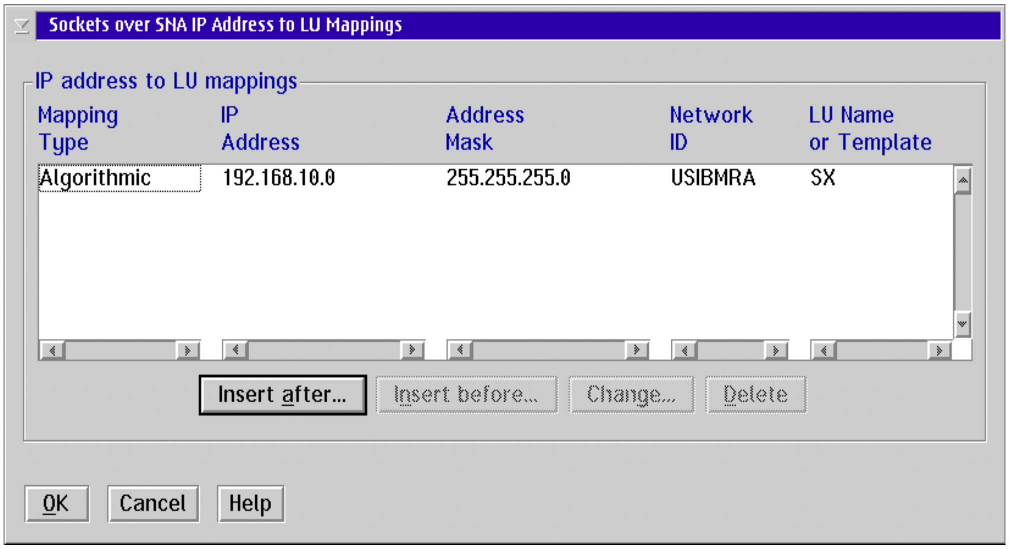
Subnet mask: 255 . 255 . 255 . 0

Idle timeout (seconds): 90 (0 - 100000)

Datagram retry delay (seconds): 0 (0 - 100000)

Figure 86. Sockets over SNA Local Parameters

Here you configure the IP address and subnet mask of the sna0 interface. Next, the IP to SNA address mapping needs to be defined using the appropriate selection (**Sockets over SNA IP address to LU mappings**) from the Configuration List. Figure 87 shows the mapping panel.



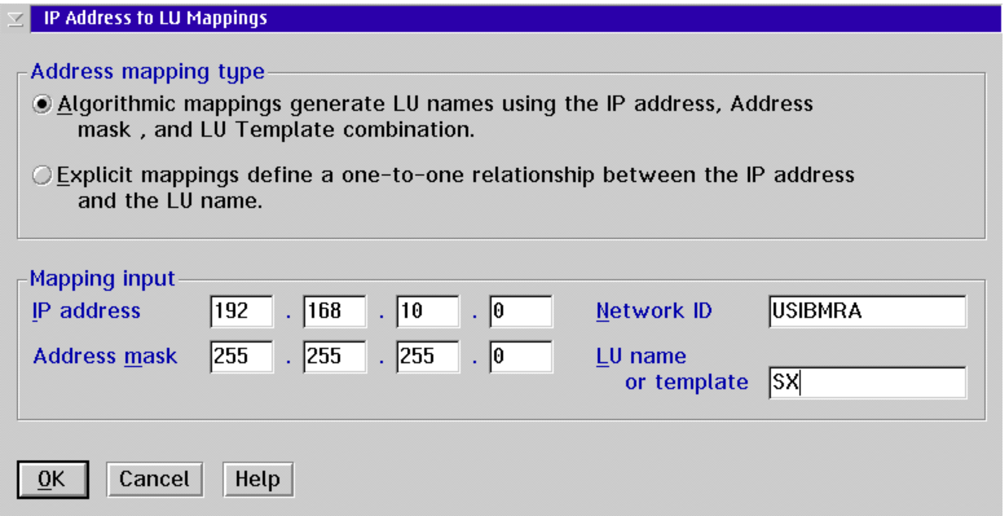
Sockets over SNA IP Address to LU Mappings

IP address to LU mappings

Mapping Type	IP Address	Address Mask	Network ID	LU Name or Template
Algorithmic	192.168.10.0	255.255.255.0	USIBMRA	SX

Figure 87. Sockets over SNA IP Address to LU Mapping Panel

On this panel you select a mapping entry (if there is one) and click **Insert after**, **Insert before** or **Change**. You are then presented with the details panel (Figure 88 on page 96) where an individual mapping entry can be defined.



IP Address to LU Mappings

Address mapping type

● Algorithmic mappings generate LU names using the IP address, Address mask , and LU Template combination.

○ Explicit mappings define a one-to-one relationship between the IP address and the LU name.

Mapping input

IP address 192 . 168 . 10 . 0

Address mask 255 . 255 . 255 . 0

Network ID USIBMRA

LU name or template SX

Figure 88. Address Mapping Details Panel

You have two methods of defining a mapping on this panel:

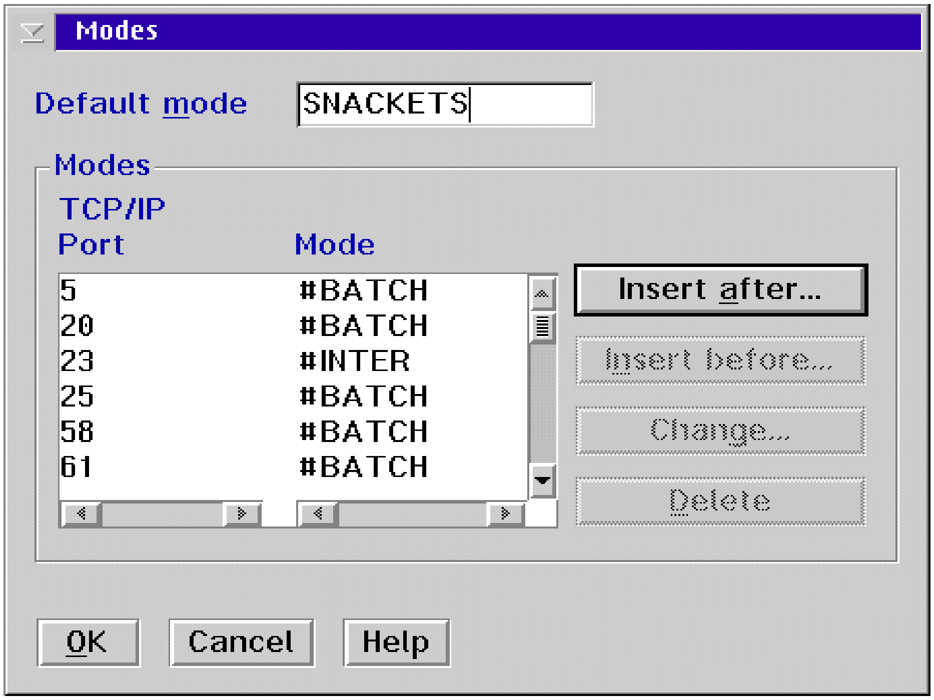
1. The recommended method for networks of more than a handful of nodes is algorithmic mapping. You enter an IP network address, a subnet mask, an SNA network ID and an LU name prefix. When AnyNet is called upon to translate an IP address to an LU name, it will determine the network address portion of the IP address, convert it to an SNA network ID with the defined LU name prefix, use the algorithm to calculate an LU name suffix from the host portion of the IP network, and append the LU suffix to the LU prefix to produce a fully qualified LU name. Algorithmic mapping eliminates the need to define all the remote nodes with which this node is to communicate.

2. The other method is explicit mapping. Here you define an IP host address and a fully qualified (complete with network ID) SNA LU name, and AnyNet maps directly from one to the other. Explicit mapping is most effective if a very small number of nodes are using Sockets over SNA, or if you are initially setting up the network and wish to test it with just a few nodes. Be aware, however, that if you use explicit mapping you will have to define every node to every other node in this way.

It is recommended that you use only one LU template for each SNA subnetwork for the following reasons:

- A minimum of entries is needed in the IP-LU mapping table of each node to represent all other Sockets over SNA nodes.
- New nodes can be added to the network without requiring any changes to the mapping tables of existing nodes.

The process of defining the mode to be used by AnyNet Sockets over SNA is similar to that in CS/NT. Select **Sockets over SNA modes** to obtain the panel shown in Figure 89 on page 97.



Default mode SNACKETS

TCP/IP Port	Mode
5	#BATCH
20	#BATCH
23	#INTER
25	#BATCH
58	#BATCH
61	#BATCH

Figure 89. Sockets over SNA Modes

The Modes panel allows you to associate a specific mode (and therefore a specific SNA class of service and thus a priority) with each IP port used by the Sockets application. The default mode is used for all communication not associated with one of the defined ports. Thus you can give Telnet (usually Port 23) traffic a higher priority than file transfer.

To use SNACKETS as the default mode (as shown) you must have previously defined it to CS/2 using the **SNA features** dialog. BLANK is the default mode.

You may also need to define some Sockets over SNA routes, but that panel is described in “AnyNet Sockets over SNA CS/2 Gateway” on page 97 since it is more likely to be encountered in a gateway configuration.

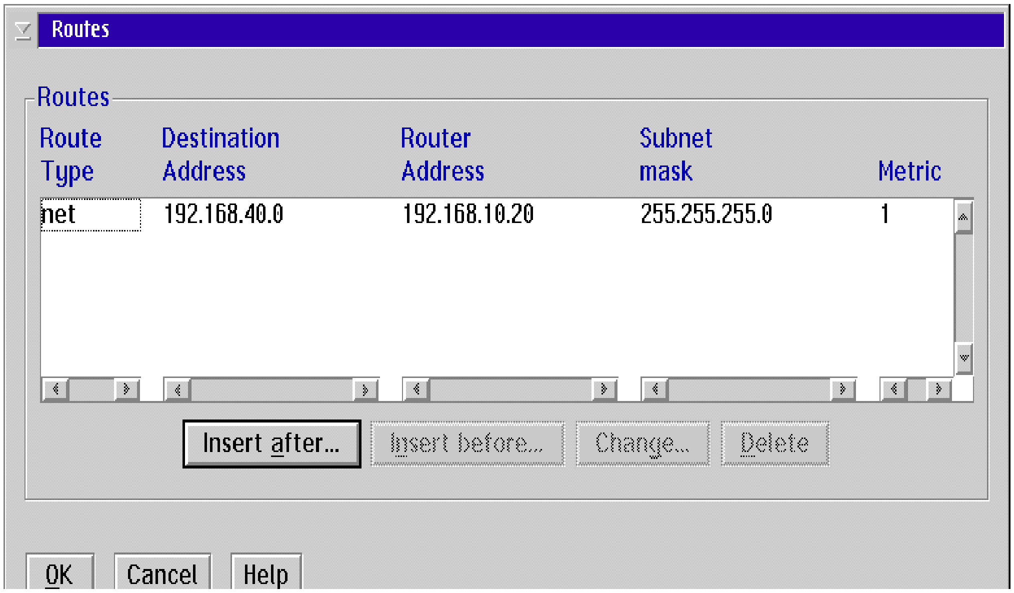
After you have finished configuring CS/2 AnyNet Sockets over SNA for this first time, the setup process will copy additional modules from the installation files and ask you to reboot. There is no requirement to install additional device drivers because the Sockets over SNA driver (AFINET.SYS) is part of MPTS and is always installed as a prerequisite to TCP/IP or to CS/2.

3.6 AnyNet Sockets over SNA CS/2 Gateway

As with CS/NT, the configuration of a gateway node is very similar to that of an access node. The gateway node has to be able to communicate with the native IP transport as well as the SNA transport. The main differences between the configuration processes are likely to be in the route definition area and especially in the backup and load balancing area.

One important consideration for a gateway node is that it must act as an IP router. Therefore, you must specify **IP forwarding** in the TCP/IP configuration notebook for your OS/2 machine. In TCP/IP Version 4 Release 1 this is a check box on the Routing tab in the notebook.

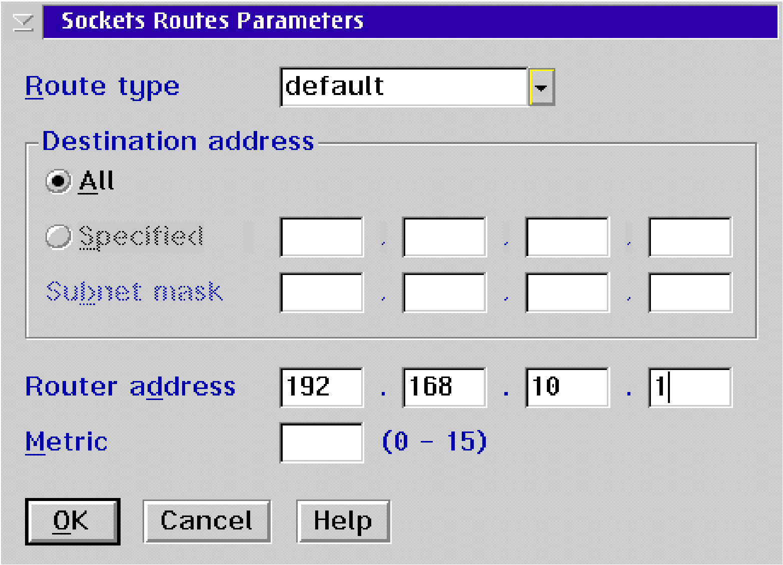
Figure 90 shows the Routes panel that you get when you select **Sockets over SNA routes** from the Configuration List.



Route Type	Destination Address	Router Address	Subnet mask	Metric
net	192.168.40.0	192.168.10.20	255.255.255.0	1

Figure 90. Sockets over SNA Routes Panel

If you select one of the **Insert** options or **Change** an existing route, you get the route parameters panel as in Figure 91.



Route type: default

Destination address

● All

○ Specified: ___ . ___ . ___ . ___

Subnet mask: ___ . ___ . ___ . ___

Router address: 192 . 168 . 10 . 1

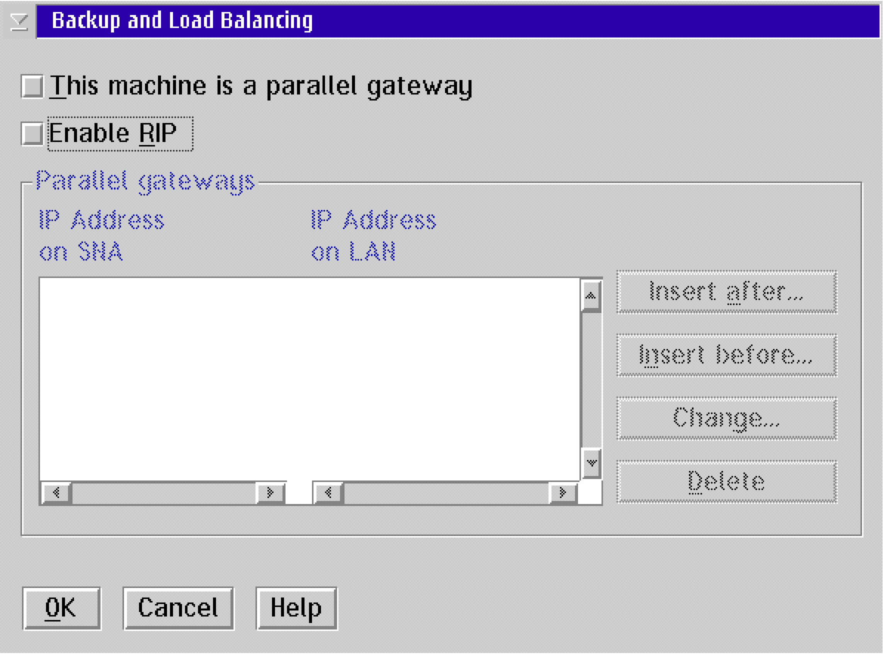
Metric: ___ (0 - 15)

Figure 91. Sockets over SNA Routes Parameters

On this panel you have the option of defining one of four types of route (default, net, subnet or host) with appropriate destination addresses, router addresses and subnet masks. The route types are:

- Default. Data is routed through the gateway or router whose IP address is specified in the Router Address field.
- Host. The address specified in the Destination Address field is the IP address of a particular host. Data is routed through the router whose IP address is specified in the Router Address field.
- Net. The address specified in the Destination Address field is the IP address of a remote network. Data is routed through the router whose IP address is specified in the Router Address field. If a conflict occurs between a host and a network route, the host route is used.
- Subnet. The address specified in the Destination Address field is the IP address of a remote subnetwork. Data is routed through the router whose IP address is specified in the Router Address field. If a conflict occurs between a host and a subnetwork route, the host route is used.

The final configuration option, that labeled **Sockets over SNA backup and load balancing**, is only relevant to the gateway setup. Selecting this option gives you the panel shown in Figure 92.



Backup and Load Balancing

□ This machine is a parallel gateway

□ Enable RIP

Figure 92. Routing and Parallel Gateways

The **Enable RIP** check box on this panel (disabled by default) allows Sockets over SNA to discover IP routes dynamically using the RIP protocol (in other words, to act exactly as an IP router). This can save you defining routes explicitly in the Routes panel.

The **parallel gateway** check box allows you to configure two or more Sockets over SNA gateways to service the same site. This has two advantages:

1. Backup, so that network service can survive the loss of one gateway
2. Better performance through load balancing

If you define parallel gateways they use RIP to monitor each other's status, so the **Enable RIP** check box is automatically enabled. You have to define the SNA and IP addresses of the gateways to each other using the **Insert** or **Change** buttons on the panel.

3.7 Working Scenarios

We tested the Sockets over SNA function using Communications Server for OS/2, Communications Server for Windows NT (in both access and gateway configurations) and CS for OS/390 (which acts only as an access node) in various combinations. Figure 93 on page 101 shows the network we used.

We had two CS for OS/390 systems in a Parallel Sysplex, two CS/NT PCs, one PC running CS/2 and one PC running Personal Communications for Windows NT (and therefore a subset of CS/NT which supports the Sockets over SNA access node) as shown. All were connected to a token-ring, the OS/390 hosts sharing the same OSA card (they used different SAPs but the same MAC address for SNA traffic).

For each node, the diagram shows the MAC address, the sna0 IP address (all had the same subnet mask 255.255.255.0) and, in the case of the PCs, the lan0 IP address.

Connectivity between the OS/390 hosts was via the cross-system coupling facility (XCF), and therefore both IP and SNA communication was available. XCF uses the same common DLC connection manager as multipath channel (MPC), and thus can be shared between the VTAM and TCP/IP components of CS for OS/390.

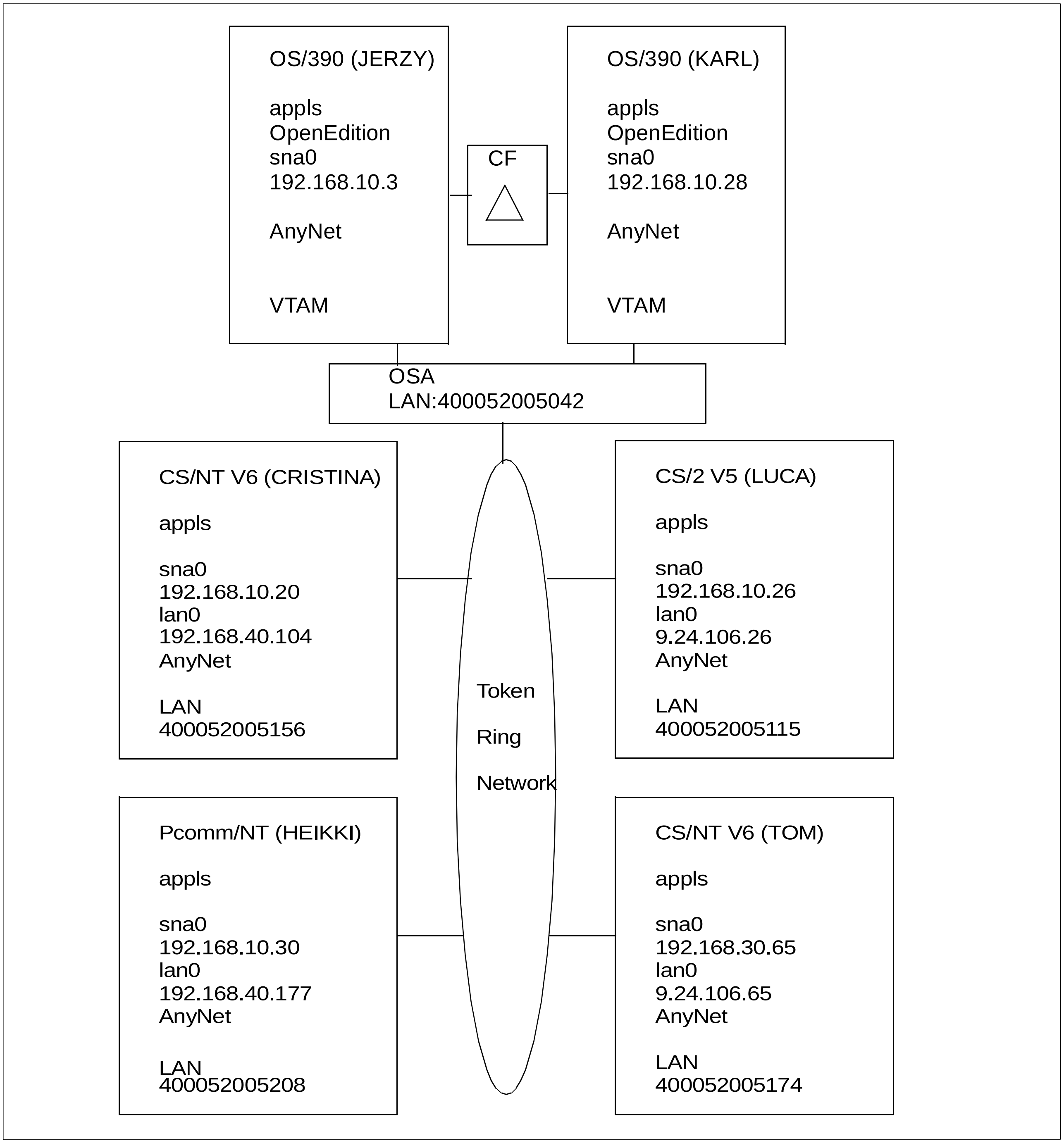


Figure 93. Sockets over SNA - Test Environment

3.7.1 Sockets over SNA between OS/390 Hosts

In our first example, two OS/390 hosts were connected using Sockets over SNA. The hosts were both members of a Parallel Sysplex, and thus had APPN connections to each other across the MVS cross-system coupling facility.

The two mainframes could equally well have been standalone systems, located a distance from each other and connected via a larger APPN network. Equally, the network could have been based on subarea SNA, with cross domain and/or SNI connections. Sockets over SNA requires no more than some LU 6.2 sessions, and does not concern itself with what form of SNA transport those sessions use. Here we used APPN simply because it is there in a Parallel Sysplex without your having to create any definitions.

Figure 94 shows the configuration of our OS/390-to-OS/390 connection.

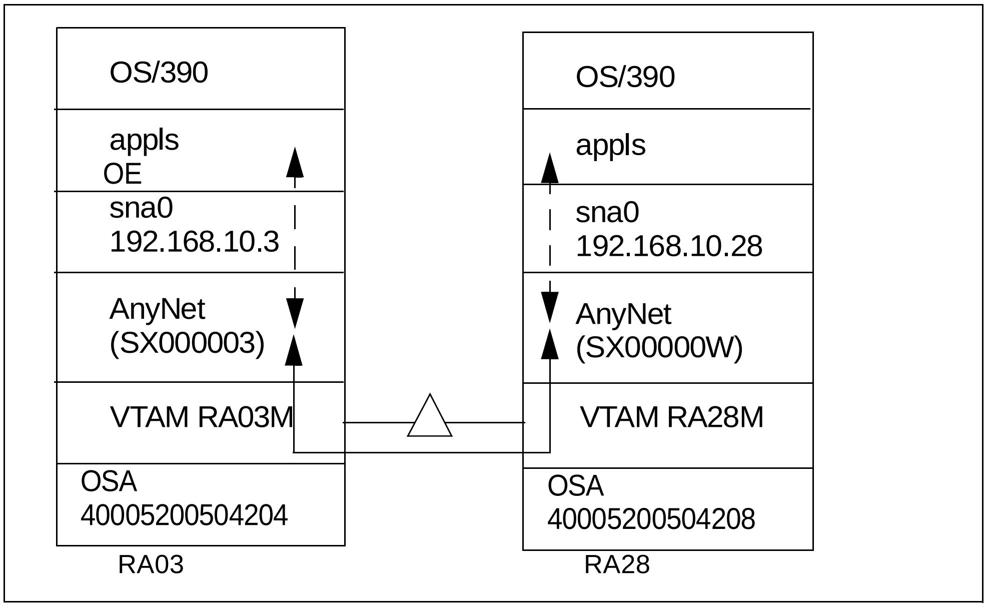


Figure 94. OS/390-to-OS/390 Configuration

The first thing we had to do was to modify the BPXPRMxx definitions on each system to support Sockets over SNA. The particular definitions required for Sockets over SNA are shown in the extract from RA03's definitions (BPXPRM03) in Figure 95.

```
026591 NETWORK DOMAINNAME(AF_INET)
026592         DOMAINNUMBER(2)
026593         MAXSOCKETS(10000)
026594         TYPE(CINET)
026595         INADDRANYPORT(4000)
026596         INADDRANYCOUNT(2000)
...........................................................
...........................................................
026661 SUBFILESYSTYPE NAME(ANYNET03)
026662                TYPE(CINET)
026663                ENTRYPOINT(ISTOEPIT)
```

Figure 95. BPXPRM03 Definitions

BPXPRM28 was modified in a similar fashion on RA28, and then both systems were re-IPLed. Any change to these UNIX Systems Services definitions requires a re-IPL of OS/390.

Next we defined application major nodes for VTAM to recognize the Sockets over SNA application LUs (SX000003 and SX00000W in this example). Figure 96 illustrates the one on RA28M.

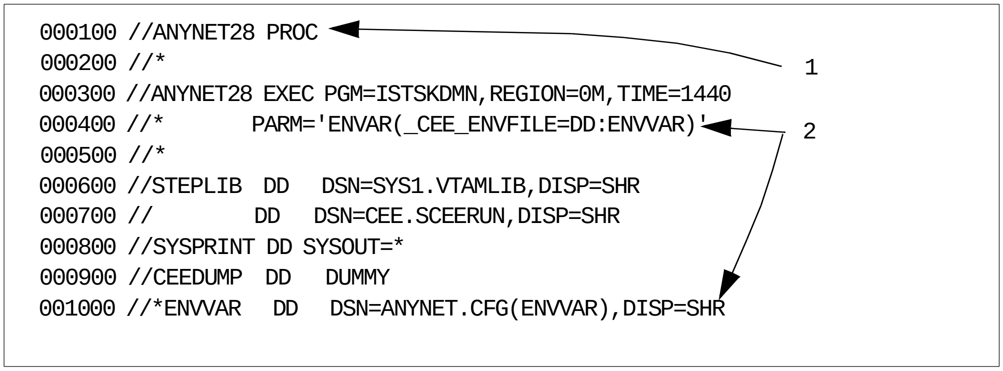
```
SOS28A1 VBUILD  TYPE=APPL
SX*    APPL    APPC=YES,                                               X
               PARSESS=YES,                                            X
               DSESLIM=10,                                             X
               DMINWNL=5,                                              X
               DMINWNR=5,                                              X
               AUTOSES=0,                                              X
               AUTH=(ACQ,PASS),                                        X
               OPERCNOS=ALLOW,                                         X
               ATNLOSS=ALL
```

Figure 96. Application Definition for Sockets over SNA on RA28M

On both systems we used model application definitions to save having to think about the LU names that the AnyNet algorithm would generate. We knew that our LU name prefix would be SX so we simply defined SX* as the application name.

The application major node caters for the local Sockets over SNA application; we did not define the remote application (on the other host) because VTAM is perfectly capable of finding that dynamically. If we enter an IP address the mapping algorithm will devise an LU name and then VTAM will search the network to discover the location of the named LU.

Next we activated the application major nodes on each host, then started the Sockets over SNA applications themselves. Figure 97 shows the JCL we used on RA28.



```
//ANYNET28 PROC
//*
//ANYNET28 EXEC PGM=ISTSKDMN,REGION=0M,TIME=1440
//*        PARM='ENVAR(_CEE_ENVFILE=DD:ENVVAR)'
//*
//STEPLIB  DD   DSN=SYS1.VTAMLIB,DISP=SHR
//         DD   DSN=CEE.SCEERUN,DISP=SHR
//SYSPRINT DD SYSOUT=*
//CEEDUMP  DD   DUMMY
//*ENVVAR   DD   DSN=ANYNET.CFG(ENVVAR),DISP=SHR
```

Figure 97. JCL to Start Sockets over SNA

The JCL itself is straightforward, but please note the following:

1. The procedure name (ANYNET28 on RA28) must match the name defined in the NAME keyword in the SUBFILESYSTEM statement in the BPXPRM28 member of PARMLIB. Figure 95 on page 102 is the equivalent RA03 member showing ANYNET03 as the procedure name.
2. If you code AnyNet environment variables (we did not) you can specify to OS/390 where to look for them on the PARM keyword. This points to the

ENVVAR DD statement which in turn points to the actual data set with the environment variable. In our example both of these definitions are commented out.

When this procedure had been started Sockets over SNA was running, but not yet available for processing requests. We now ran the address mapping program to define the IP address-to-SNA LU relationship. Without this definition Sockets over SNA cannot perform its task. Figure 98 shows the JCL used to run the mapping program.

```
//SOSSKMAP PROC
//*
//SKMAP    EXEC PGM=ISTSKMAP,REGION=1M,TIME=1440,                   1
//*        PARM='GET'
//        PARM='ADD 192.168.10.0 255.255.255.0 USIBMRA SX'
//*
//STEPLIB  DD   DSN=SYS1.VTAMLIB,DISP=SHR
//         DD   DSN=CEE.SCEERUN,DISP=SHR
//SYSPRINT DD SYSOUT=*
```

Figure 98. JCL for Address Mapping Program

Note the way of specifying the algorithmic mapping (1) via the execution parameters. Here we tell AnyNet that the IP subnetwork 192.168.10 (where both of our Sockets over SNA interfaces are) is to be mapped to LU names of the form USIBMRA.SX..... The algorithm does the rest, mapping 192.168.10.3 (RA03) to SX000003 and 192.168.10.28 (RA28) to SX00000W. Note that entering PARM=GET allows you to display details of the existing mapping tables.

Next, we ran the interface program (see Figure 99 for the JCL) to define the sna0 interface to OS/390. It is usual to run the address mapping and the interface programs within the same job.

```
//SKIFC    EXEC PGM=ISTSKIFC,REGION=1M,TIME=1440,
//        PARM=('SNA0 192.168.10.28 NETMASK 255.255.255.0')
//*
//STEPLIB  DD   DSN=SYS1.VTAMLIB,DISP=SHR
//         DD   DSN=CEE.SCEERUN,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*EEDUMP  DD   DUMMY
//*ENVVAR   DD   DSN=ANYNET.CFG(ENVVAR),DISP=SHR
```

Figure 99. JCL for Interface Mapping Program

The interface program, ISTSKIFC, is used to define which IP address Sockets over SNA must use on the virtual interface sna0. This example is for RA28. Once this has been run Sockets over SNA is ready to accept Sockets calls and IP traffic.

We tested the connectivity between the two hosts using the OPING command. PING (an MVS shell command) no longer works with Sockets over SNA, but can still be used for native IP connectivity testing. Figure 100 shows the OPING command and the response when we used it from RA03 to send a message to RA28.

```
oping 192.168.10.28__
....................
....................
 CS/390 V2R5: Pinging host 192.168.10.28
Ping #1 response took 0.142 seconds.
```

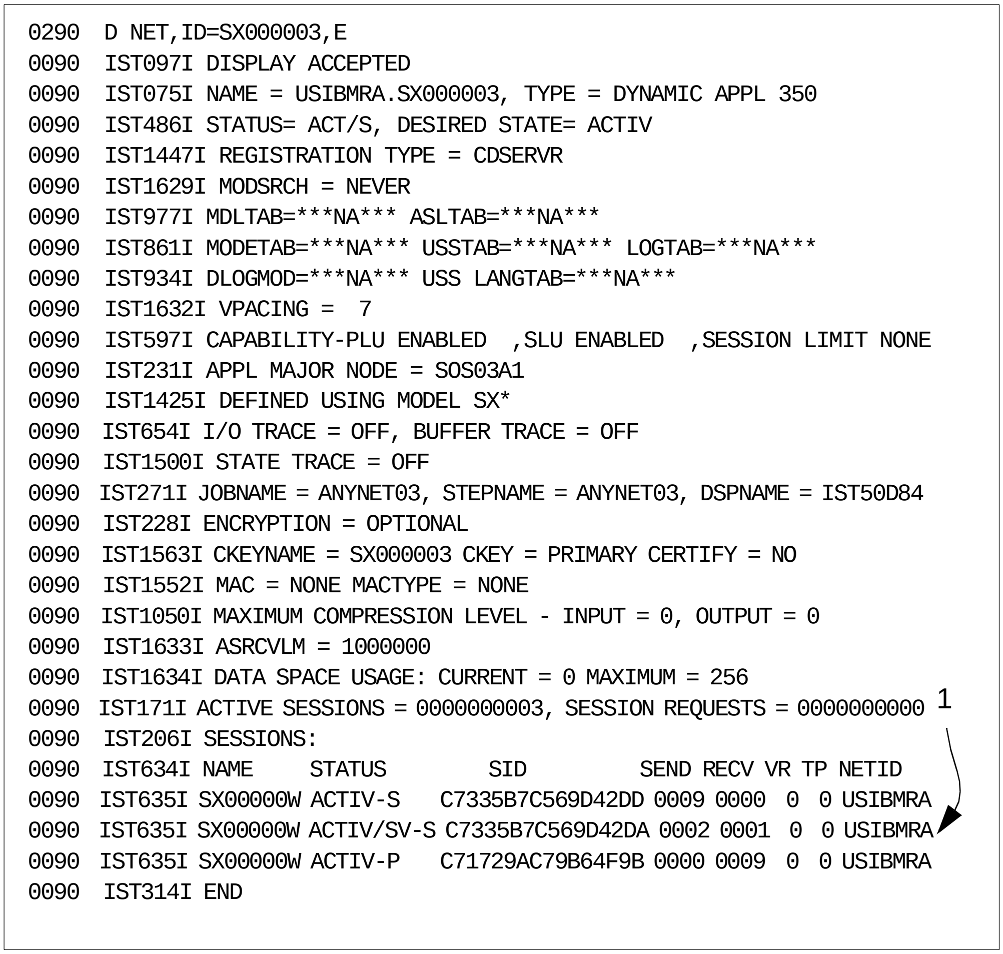
Figure 100. OPING between RA03 and RA28

You can issue various VTAM commands to see which sessions have been established as a result of your TCP/IP activities. If you do not know which LU name was generated by Sockets over SNA, you can start with a command that displays everything about the Sockets over SNA application, such as the display of the major node shown in Figure 101.

```
0290  D NET,ID=SOS03A1,E
0090  IST097I DISPLAY ACCEPTED
0090  IST075I NAME = SOS03A1, TYPE = APPL SEGMENT 287
0090  IST486I STATUS= ACTIV, DESIRED STATE= ACTIV               1
0090  IST360I APPLICATIONS:
0090  IST080I SX*      CONCT      SX000003 ACT/S
0090  IST314I END
```

Figure 101. Sockets over SNA Major Node Display

Here (on RA03) you can see (1) that VTAM has created an application named SX000003 which must be the algorithmically generated equivalent of 192.168.10.3. To see what the partner applications are, you can display the LU name SX000003 itself as in Figure 102 on page 106.

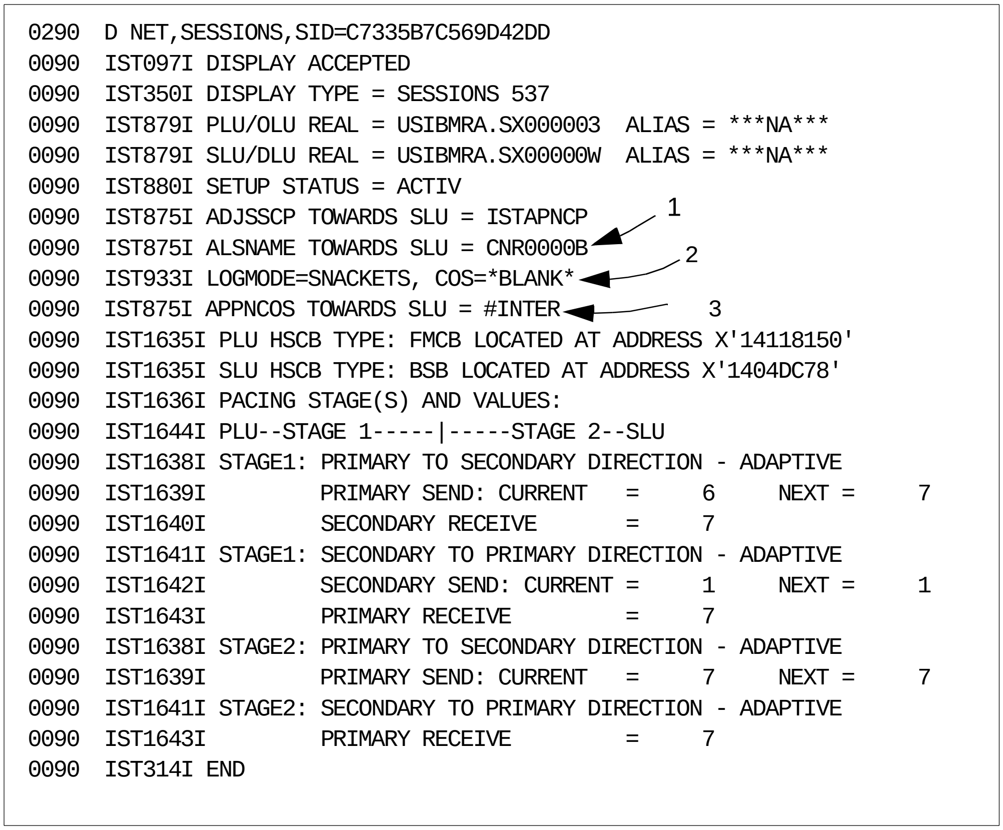


```
0290  D NET,ID=SX000003,E
0090  IST097I DISPLAY ACCEPTED
0090  IST075I NAME = USIBMRA.SX000003, TYPE = DYNAMIC APPL 350
0090  IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
0090  IST1447I REGISTRATION TYPE = CDSERVR
0090  IST1629I MODSRCH = NEVER
0090  IST977I MDLTAB=***NA*** ASLTAB=***NA***
0090  IST861I MODETAB=***NA*** USSTAB=***NA*** LOGTAB=***NA***
0090  IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
0090  IST1632I VPACING =  7
0090  IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT NONE
0090  IST231I APPL MAJOR NODE = SOS03A1
0090  IST1425I DEFINED USING MODEL SX*
0090  IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
0090  IST1500I STATE TRACE = OFF
0090  IST271I JOBNAME = ANYNET03, STEPNAME = ANYNET03, DSPNAME = IST50D84
0090  IST228I ENCRYPTION = OPTIONAL
0090  IST1563I CKEYNAME = SX000003 CKEY = PRIMARY CERTIFY = NO
0090  IST1552I MAC = NONE MACTYPE = NONE
0090  IST1050I MAXIMUM COMPRESSION LEVEL - INPUT = 0, OUTPUT = 0
0090  IST1633I ASRCVLM = 1000000
0090  IST1634I DATA SPACE USAGE: CURRENT = 0 MAXIMUM = 256
0090  IST171I ACTIVE SESSIONS = 0000000003, SESSION REQUESTS = 0000000000
0090  IST206I SESSIONS:
0090  IST634I NAME     STATUS         SID          SEND RECV VR TP NETID
0090  IST635I SX00000W ACTIV-S    C7335B7C569D42DD 0009 0000  0  0 USIBMRA
0090  IST635I SX00000W ACTIV/SV-S C7335B7C569D42DA 0002 0001  0  0 USIBMRA
0090  IST635I SX00000W ACTIV-P    C71729AC79B64F9B 0000 0009  0  0 USIBMRA
0090  IST314I END
```

Figure 102. Display of Sockets over SNA Application

You can see here that SX000003 has three SNA sessions (1), all with SX00000W (which can only be 192.168.10.28). The session labeled ACTIV/SV is the management (CNOS) session used to control the others. The other two, one in each direction (ACTIV-S and ACTIV-P) are the pair of LU 6.2 sessions used to carry the IP traffic. PING is a raw Sockets application (not a stream-type application) and therefore uses such a pair of sessions.

You can display further details of each of these sessions using the `DISPLAY SESSIONS` command as shown in Figure 103 on page 107.



```
0290  D NET,SESSIONS,SID=C7335B7C569D42DD
0090  IST097I DISPLAY ACCEPTED
0090  IST350I DISPLAY TYPE = SESSIONS 537
0090  IST879I PLU/OLU REAL = USIBMRA.SX000003  ALIAS = ***NA***
0090  IST879I SLU/DLU REAL = USIBMRA.SX00000W  ALIAS = ***NA***
0090  IST880I SETUP STATUS = ACTIV
0090  IST875I ADJSSCP TOWARDS SLU = ISTAPNCP
0090  IST875I ALSNAME TOWARDS SLU = CNR0000B
0090  IST933I LOGMODE=SNACKETS, COS=*BLANK*
0090  IST875I APPNCOS TOWARDS SLU = #INTER
0090  IST1635I PLU HSCB TYPE: FMCB LOCATED AT ADDRESS X'14118150'
0090  IST1635I SLU HSCB TYPE: BSB LOCATED AT ADDRESS X'1404DC78'
0090  IST1636I PACING STAGE(S) AND VALUES:
0090  IST1644I PLU--STAGE 1-----|-----STAGE 2--SLU
0090  IST1638I STAGE1: PRIMARY TO SECONDARY DIRECTION - ADAPTIVE
0090  IST1639I         PRIMARY SEND: CURRENT   =     6     NEXT =     7
0090  IST1640I         SECONDARY RECEIVE       =     7
0090  IST1641I STAGE1: SECONDARY TO PRIMARY DIRECTION - ADAPTIVE
0090  IST1642I         SECONDARY SEND: CURRENT =     1     NEXT =     1
0090  IST1643I         PRIMARY RECEIVE         =     7
0090  IST1638I STAGE2: PRIMARY TO SECONDARY DIRECTION - ADAPTIVE
0090  IST1639I         PRIMARY SEND: CURRENT   =     7     NEXT =     7
0090  IST1641I STAGE2: SECONDARY TO PRIMARY DIRECTION - ADAPTIVE
0090  IST1643I         PRIMARY RECEIVE         =     7
0090  IST314I END
```

Figure 103. Session Details Display for Sockets over SNA

This display shows you, for example:

1. That the session is using HPR (from the CNR... prefix for the link station).
2. That the mode used is SNACKETS (the default mode, since PING does not use any port, let alone a port associated with a particular mode).
3. That the APPN class of service in use is #INTER, which runs at high priority.

In addition to ISTSKMAP and ISTSKIFC, other utility programs are provided with Sockets over SNA. One of these is ISTSKNST, and with it you can display:

- The entries in the routing table (option -r)
- Socket information (option -s)
- Statistics on concurrent Sockets calls (option -d)

Those familiar with the `netstat` or `onetstat` commands will probably recognize ISTSKNST. Figure 104 on page 108 shows the JCL necessary to execute it. It can also be executed from the TSO command prompt or the UNIX Systems Services shell.

```
//SKNST    EXEC PGM=ISTSKNST,REGION=1M,TIME=1440,
//         PARM=('-R -S -D')
//*
//STEPLIB  DD   DSN=SYS1.VTAMLIB,DISP=SHR
//         DD   DSN=CEE.SCEERUN,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*
```

Figure 104. ISTSKNST JCL

Other utilities available are:

- ISTSKTRC, to trace Sockets over SNA activity
- ISTSKRTE, to define a routing table for Sockets over SNA. This is the equivalent of the Routes panel in CS/2 or CS/NT. It allows you to define routes to remote destinations if your network includes a Sockets over SNA gateway. OS/390 itself does not support acting as such a gateway.

3.7.2 Sockets over SNA between OS/390 and CS/NT

Our second test, illustrated in Figure 105 on page 109, was to establish communication between a CS/NT workstation and an OS/390 host. This time the two nodes were linked by an APPN connection over a token-ring.

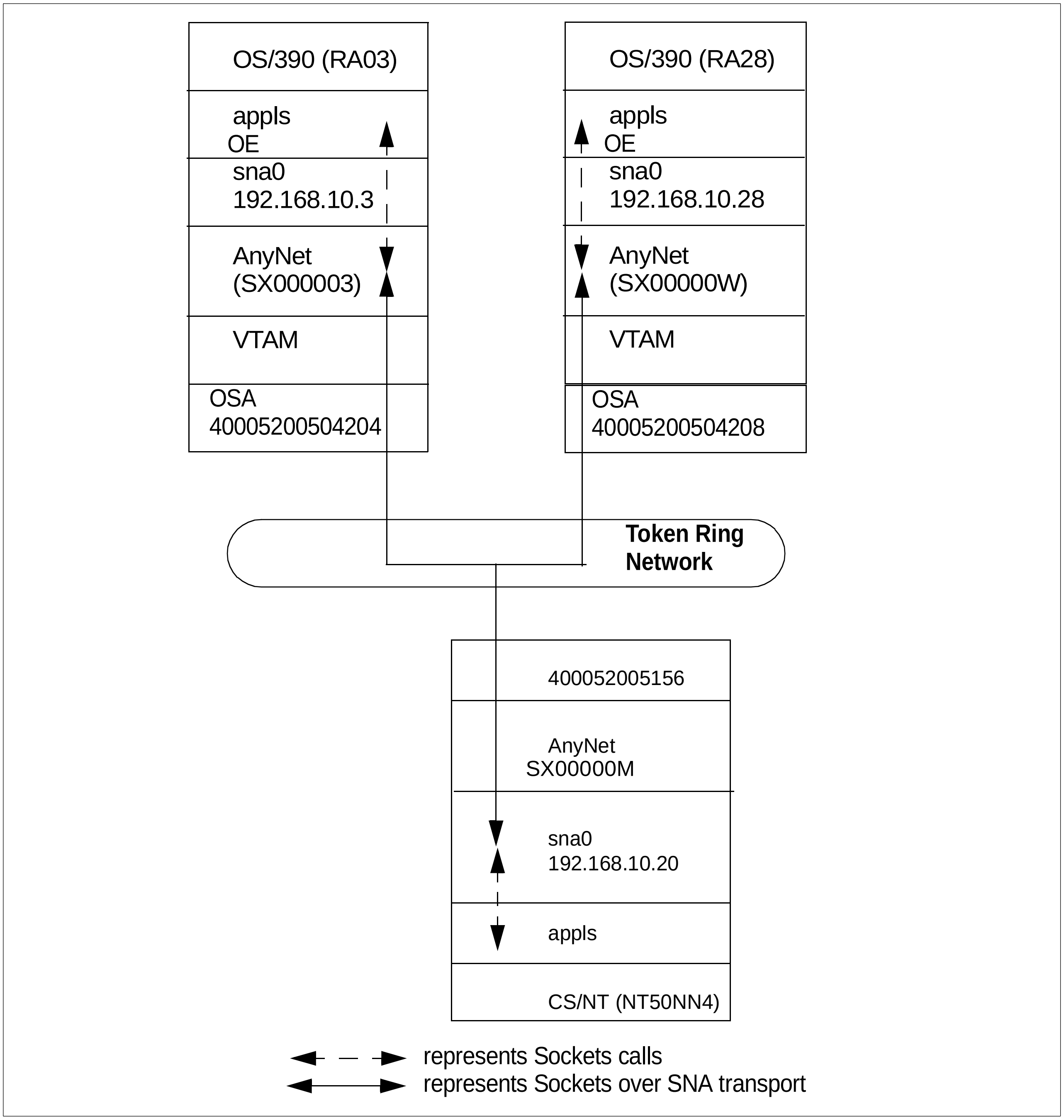


Figure 105. CS/NT to OS/390 Access Nodes

The configuration of the OS/390 node RA28 was exactly the same as in "Sockets over SNA between OS/390 Hosts" on page 101. The IP-to-SNA mapping and VTAM's dynamic definition of remote resources between them catered for the new LU name that we now had to talk to.

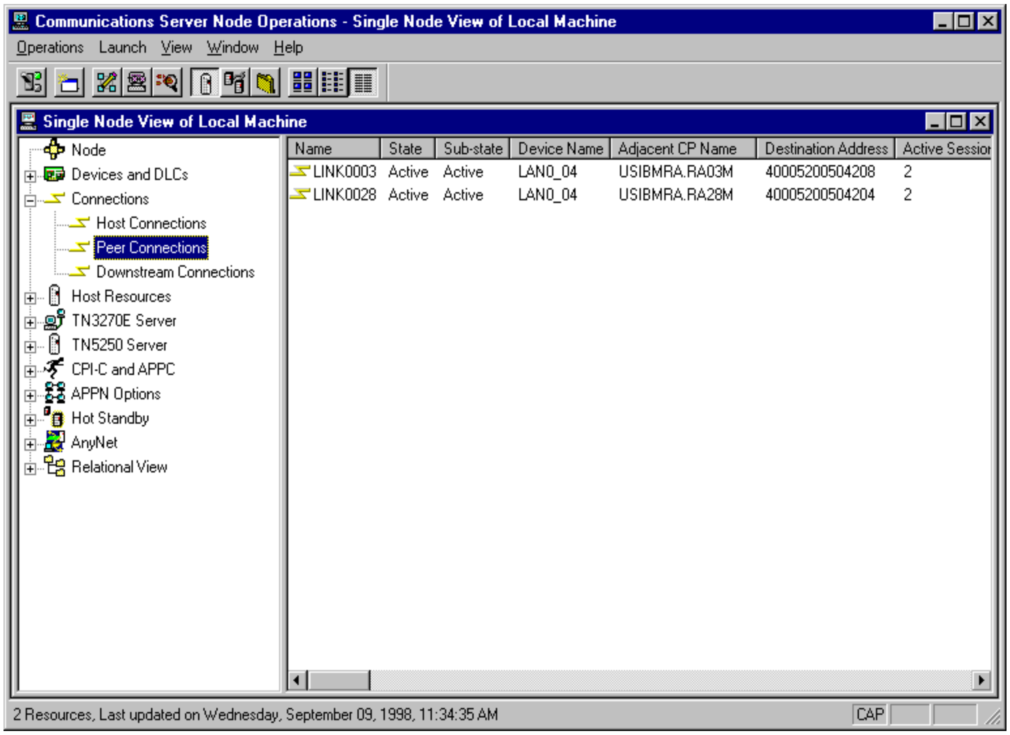
The CS/NT configuration was as follows:

- Local Node (see Figure 65 on page 79):
  - CP name is USIBMRA.NT50NN4
  - Node type is network node

  - Local node ID is 05D/05156
- LAN Connections (two; see Figure 67 on page 80):
  - Destination address is 400052005042 for both
  - Remote SAP is 04 on one connection and 08 on the other
  - APPN support is on for both
- Sockets over SNA local (see Figure 70 on page 82):
  - IP address is 192.168.10.20
  - Subnet mask is 255.255.255.0
- Sockets over SNA LUs (see Figure 72 on page 83):
  - Mapping type is Generate LU names
  - IP address is 192.168.10.0
  - Subnet mask is 255.255.255.0
  - SNA network ID is USIBMRA
  - LU template is SX
- Sockets over SNA routes (see Figure 77 on page 88):
  - None defined because there were no gateways involved
- Sockets over SNA modes (see Figure 74 on page 85):
  - Default mode is SNACKETS
  - Port/mode definitions were left unchanged

We did not restart any part of the OS/390 system RA03 because we had changed nothing. To verify the operation of AnyNet between the two nodes, we started the node NT50NN4 and performed both PING (OPING on OS/390) and APING (the SNA equivalent of PING) transactions between the two nodes. The OPING displays seen from RA03 were very similar to those observed in the previous scenario, simply because Sockets over SNA and VTAM are not aware of the platform on which the partner node, the partner SNA LU and the partner IP stack are running. We therefore illustrate this scenario with some displays performed on CS/NT.

Figure 106 on page 111 shows the SNA link stations that CS/NT has defined. These are obtained by selecting the **Peer Connections** option from the SNA Node Operations menu. This option can be found under both the **Relational View** and the **Connections** menu items.



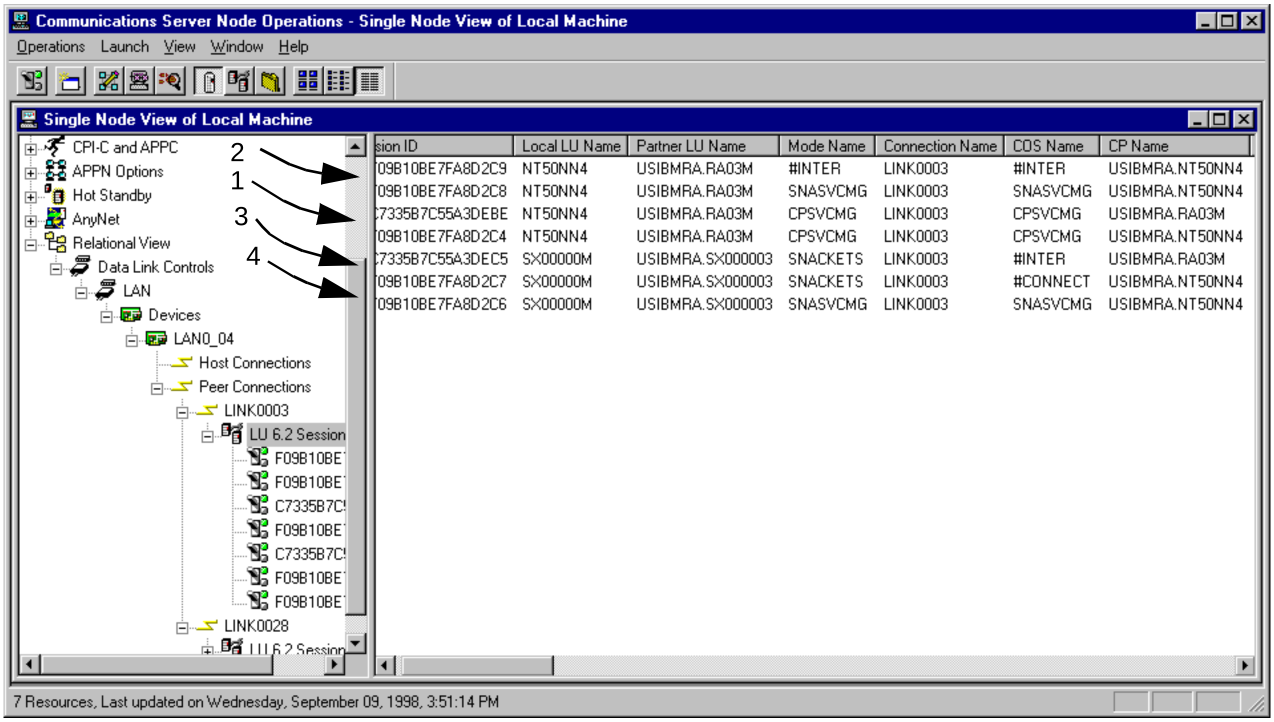
Name	State	Sub-state	Device Name	Adjacent CP Name	Destination Address	Active Sessior
LINK0003	Active	Active	LAN0_04	USIBMRA.RA03M	40005200504208	2
LINK0028	Active	Active	LAN0_04	USIBMRA.RA28M	40005200504204	2

2 Resources, Last updated on Wednesday, September 09, 1998, 11:34:35 AM

Figure 106. CS/NT Logical Links

The two connections are the token-ring links to RA03 and RA28 respectively; only the second of these was used in this test.

Figure 107 on page 112 shows the LU 6.2 sessions that were established as a result of our activity.



sion ID	Local LU Name	Partner LU Name	Mode Name	Connection Name	COS Name	CP Name
09B10BE7FA8D2C9	NT50NN4	USIBMRA.RA03M	#INTER	LINK0003	#INTER	USIBMRA.NT50NN4
09B10BE7FA8D2C8	NT50NN4	USIBMRA.RA03M	SNASVCMG	LINK0003	SNASVCMG	USIBMRA.NT50NN4
7335B7C55A3DEBE	NT50NN4	USIBMRA.RA03M	CPSVCMG	LINK0003	CPSVCMG	USIBMRA.RA03M
09B10BE7FA8D2C4	NT50NN4	USIBMRA.RA03M	CPSVCMG	LINK0003	CPSVCMG	USIBMRA.NT50NN4
7335B7C55A3DEC5	SX00000M	USIBMRA.SX000003	SNACKETS	LINK0003	#INTER	USIBMRA.RA03M
09B10BE7FA8D2C7	SX00000M	USIBMRA.SX000003	SNACKETS	LINK0003	#CONNECT	USIBMRA.NT50NN4
09B10BE7FA8D2C6	SX00000M	USIBMRA.SX000003	SNASVCMG	LINK0003	SNASVCMG	USIBMRA.NT50NN4

7 Resources, Last updated on Wednesday, September 09, 1998, 3:51:14 PM

Figure 107. LU 6.2 Sessions on CS/NT

You can see the two CP-CP sessions (1) which you would expect to see between adjacent APPN network nodes when CP-CP support has been enabled. The two sessions (2) at the top of the display are the result of the SNA APING transaction issued on NT50NN4. There is a CNOS session with mode SNASVCMG and a user data session with mode #INTER.

The PING transaction has also resulted in multiple sessions; (4) is the CNOS session for the user data sessions (again, two parallel sessions, one contention winner for each partner) (3). These sessions are using the SNACKETS mode as expected, but one of them is using the APPN class of service #INTER (high priority) and the other is using #CONNECT (medium priority). This is because the SNACKETS mode is defined differently on the two nodes. On VTAM it takes the class of service #INTER and on CS/NT we have defined it with the class of service #CONNECT. Because the class of service is chosen by the node at the primary end of the session, each partner node's contention winner session uses the class of service defined at that node. This can easily result in the Sockets over SNA sessions (and therefore the IP traffic on them) taking different routes as well as having different priorities. It is recommended that you make the mode and COS definitions consistent across your SNA network, subarea as well as APPN.

Finally, just to show what a Sockets over SNA PING transaction looks like on CS/NT, please see Figure 108 on page 113.

```
C:\IBMCS\sdk\win32\samples\ping 192.168.10.3

Pinging 192.168.10.3 with 32 bytes of data:

Reply from 192.168.10.3: bytes=32 time=60ms TTL=255
Reply from 192.168.10.3: bytes=32 time=50ms TTL=255
Reply from 192.168.10.3: bytes=32 time=100ms TTL=255
Reply from 192.168.10.3: bytes=32 time=50ms TTL=255
```

Figure 108. PING Transaction on CS/NT

This is remarkably similar to a native TCP/IP PING. Sockets over SNA is just another IP interface with much the same characteristics as lan0 or any other port.

3.7.3 Sockets over SNA, CS/NT to CS/2

In our third test we connected a CS/NT node to a CS/2 node directly over a token-ring, both nodes acting as Sockets over SNA access nodes. The definitions on CS/NT were as follows:

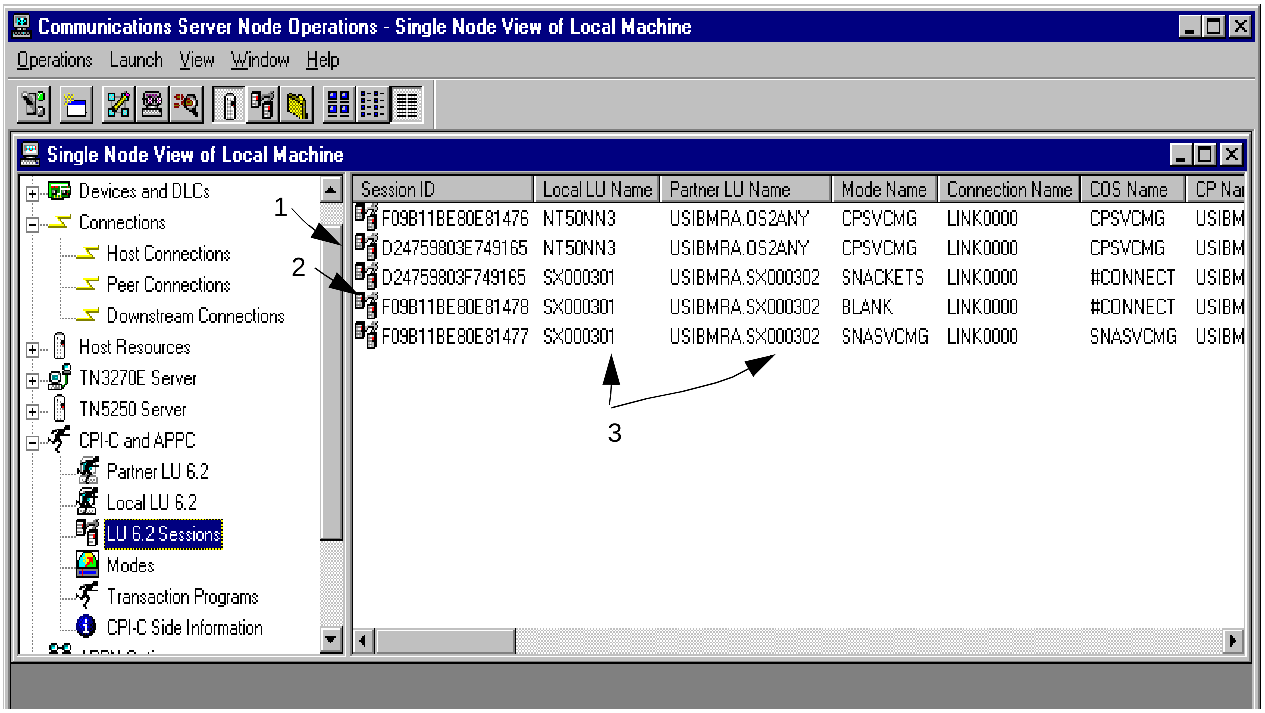
- Local Node (see Figure 65 on page 79):
  - CP name is USIBMRA.NT50NN3
  - Node type is network node
  - Local node ID is 05D/05128
- LAN Connection (see Figure 67 on page 80):
  - Destination address is 400052005115
  - Remote SAP is 04
  - APPN support is on
- Sockets over SNA local (see Figure 70 on page 82):
  - IP address is 128.109.140.1
  - Subnet mask is 255.255.224.0
- Sockets over SNA LUs (see Figure 72 on page 83):
  - Mapping type is Generate LU names
  - IP address is 128.109.140.0
  - Subnet mask is 255.255.224.0
  - SNA network ID is USIBMRA
  - LU template is SX
- Sockets over SNA routes (see Figure 77 on page 88):
  - None defined because there were no gateways involved
- Sockets over SNA modes (see Figure 74 on page 85):
  - Default mode is SNACKETS
  - Port/mode definitions were left unchanged

The definitions on CS/2 were as follows:

- Local Node (see Figure 82 on page 92):
  - CP name is USIBMRA.OS2ANY
  - Node type is network node
  - Local node ID is 05D/05115

- LAN Connection:
  - Destination address is 400052005128
  - Remote SAP is 04
- Sockets over SNA local (see Figure 86 on page 95):
  - IP address is 128.109.140.2
  - Subnet mask is 255.255.224.0
- Sockets over SNA LUs (see Figure 88 on page 96):
  - IP address is 128.109.140.0
  - Subnet mask is 255.255.224.0
  - SNA network ID is USIBMRA
  - LU name prefix is SX
- Sockets over SNA routes (see Figure 91 on page 98):
  - None defined, no gateways present
- Sockets over SNA modes (see Figure 89 on page 97):
  - Default mode is BLANK

When we started these two nodes and performed PING transactions between them, a similar set of connections and sessions was observed to those in the last scenario (CS/NT to OS/390). In architectural terms (as opposed to product implementation), the main difference was the fact that the subnet mask was different and therefore the algorithmic mapping of IP addresses to LU names worked differently. Figure 109 shows the sessions established between the CS/2 and CS/NT nodes.



Session ID	Local LU Name	Partner LU Name	Mode Name	Connection Name	COS Name	CP Na
F09B11BE80E81476	NT50NN3	USIBMRA.OS2ANY	CPSVCMG	LINK0000	CPSVCMG	USIBM
D24759803E749165	NT50NN3	USIBMRA.OS2ANY	CPSVCMG	LINK0000	CPSVCMG	USIBM
D24759803F749165	SX000301	USIBMRA.SX000302	SNACKETS	LINK0000	#CONNECT	USIBM
F09B11BE80E81478	SX000301	USIBMRA.SX000302	BLANK	LINK0000	#CONNECT	USIBM
F09B11BE80E81477	SX000301	USIBMRA.SX000302	SNASVCMG	LINK0000	SNASVCMG	USIBM

Figure 109. Sessions between CS/2 and CS/NT

Points of interest in this figure are:

3. The top two sessions are the CP-CP sessions between the two network nodes.
4. The two data sessions are using different modes (as we had defined different default modes on the two PCs), but by coincidence they are using the same class of service (#CONNECT) and therefore the same priority.
5. The algorithmic mapping has resulted in LU names of SX000301 for 128.109.140.1 and SX000302 for 128.109.140.2. The host address parts of the two IP addresses here are 12.1 and 12.2 respectively.

A display of the IP configuration from the Windows prompt (Figure 110) is also interesting. It shows the dummy sna0 interface as an Ethernet adapter called zzAnyNet Gateway2.

```
C:\>ipconfig

Windows NT IP Configuration

Token Ring adapter IbmTok41:

        IP Address. . . . . . . . . : 192.168.40.104
        Subnet Mask . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . :

Ethernet adapter zzAnyNet Gateway2:

        IP Address. . . . . . . . . : 128.109.140.1
        Subnet Mask . . . . . . . . : 255.255.224.0
        Default Gateway . . . . . . : 128.109.140.1
```

Figure 110. Windows NT IP Configuration

On the OS/2 workstation a PING results in Figure 111, which is no different from a native IP PING.

```
[C:\]PING 128.109.140.1
PING 128.109.140.1: 56 data bytes
64 bytes from 128.109.140.1: icmp_seq=0. time=60. ms
64 bytes from 128.109.140.1: icmp_seq=1. time=30. ms

----128.109.140.1 PING Statistics----
2 packets transmitted, 2 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 30/45/60
```

Figure 111. PING from OS/2 Sockets over SNA

Again on OS/2, a `netstat -a` command to display the status of the IP interfaces resulted in Figure 112 on page 116.

```
[C:\]netstat -a
addr 127.0.0.1 Interface 9 mask 0xff000000 broadcast 127.0.0.1
Multicast addrs:
 224.0.0.1

addr 9.24.106.26 Interface 0 mask 0xffffe000 broadcast 9.24.127.255
Multicast addrs:
 224.0.0.1

addr 128.109.140.2 Interface 20 mask 0xffffe000 broadcast 0.0.0.0
addr 127.0.0.2 Interface 21 mask 0xff000000 broadcast 0.0.0.0
```

Figure 112. IP Interfaces on OS/2

The sna0 interface is seen as an additional connection on which broadcasts or multicasts are not to be performed. This is quite reasonable as such things are not possible on an SNA network.

We were also able to display the session information in graphical form from the CS/2 Subsystem Management program. As with CS/NT, this shows details such as packet counts, byte counts and pacing window sizes for each session.

3.7.4 Sockets over SNA, CS/NT to Native TCP/IP via Gateway

This section describes a connection between a Sockets over SNA node running CS/NT and a native TCP/IP workstation. The connection is made via a Sockets over SNA gateway implementation on CS/NT. Figure 113 on page 117 shows the configuration.

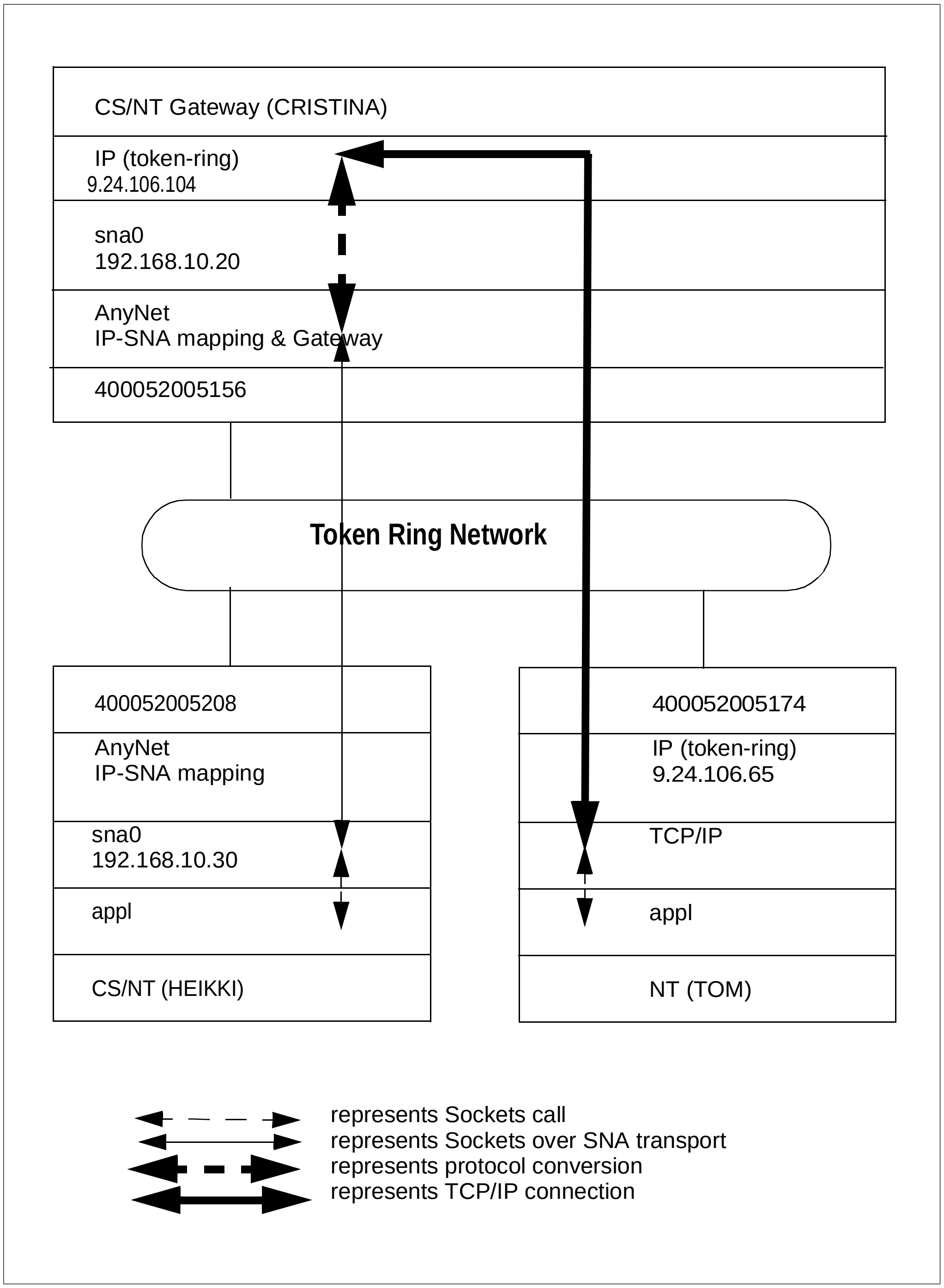


Figure 113. Single Sockets over SNA Gateway

In this configuration, an AnyNet Sockets over SNA access node (HEIKKI) is to communicate with a native TCP/IP node shown as TOM. The only way to do this is by means of a Sockets over SNA gateway which can pass through the IP datagrams while converting the SNA transport to IP transport and vice versa. This gateway has been implemented on a CS/NT node called CRISTINA.

The main addition to the definitions in this case comprises the routing information. Now that the IP network is somewhat larger than a few SNA nodes talking to each other, we must do more than simply translate IP addresses to LU names.

The configuration for the Sockets over SNA access node HEIKKI is as follows (we have omitted the SNA-only definitions to concentrate on the AnyNet parameters):

- Sockets over SNA local (see Figure 70 on page 82):
  - IP address is 192.168.10.30
  - Subnet mask is 255.255.255.0
- Sockets over SNA LUs (see Figure 72 on page 83):
  - Mapping type is Generate LU names
  - IP address is 192.168.10.0
  - Subnet mask is 255.255.255.0
  - SNA network ID is USIBMRA
  - LU template is SX
- Sockets over SNA routes (see Figure 77 on page 88):
  - Type is Default
  - Router address is 192.168.10.20
  - Direct connection is No

  The access node HEIKKI must be able to communicate with IP addresses outside its native subnetwork 192.168.10.0. However, it has no IP-to-LU mapping information for such addresses, simply because those addresses are outside the Sockets over SNA subnetwork. It needs to be told that these addresses must be reached via the gateway node CRISTINA which acts as the IP router for the 192.168.10.0 subnetwork.

  The router address is given as 192.168.10.20, which is in the Sockets over SNA subnetwork and can be contacted by SNA sessions after address translation.

  The destination address and subnet mask fields are not available for a default route; by its very nature a default route is used for all addresses not defined elsewhere.
- Sockets over SNA modes (see Figure 74 on page 85):
  - Default mode is SNACKETS
  - Port/mode definitions were left unchanged

The configuration of the native TCP/IP workstation TOM was as follows:

- TCP/IP Configuration
  - IP address of token-ring interface is 9.24.106.65
  - Subnet mask is 255.255.224.0
  - Gateway (default router) address is 9.24.106.1

  This is a mirror image of the Sockets over SNA definition on HEIKKI, except that it points to the native IP interface on its default router. This router should be able to find a route to the 192.168.10 subnetwork (in other words, to the gateway CRISTINA) by the usual IP means, namely RIP.

The configuration for the gateway node CRISTINA is as follows:

- TCP/IP Configuration

    - IP address of token-ring interface is 9.24.106.104
    - Subnet mask is 255.255.224.0
    - Gateway (default router) address is 9.24.106.1
    - IP Forwarding is enabled

- Sockets over SNA local (see Figure 70 on page 82):

    - IP address is 192.168.10.20
    - Subnet mask is 255.255.255.0

- Sockets over SNA LUs (see Figure 72 on page 83):

    - Mapping type is Generate LU names
    - IP address is 192.168.10.0
    - Subnet mask is 255.255.255.0
    - SNA network ID is USIBMRA
    - LU template is SX

- Sockets over SNA routes (see Figure 77 on page 88):

    - None required

  All IP addresses can be reached directly from this node, and all are in the same subnet as one or the other of the two interfaces. Addresses in the 192.168.10 subnetwork can be reached by SNA sessions, whereas addresses in the 9.24.96 subnetwork can be reached by a direct LAN connection.

- Sockets over SNA modes (see Figure 74 on page 85):

    - Default mode is BLANK
    - Port/mode definitions were left unchanged

We configured the three nodes as above, restarted them, and performed PING transactions between HEIKKI and TOM. After a successful connectivity test, we observed the usual set of SNA LU 6.2 sessions between HEIKKI and CRISTINA:

- Two CP-CP sessions between the control point LUs
- One CNOS session between the Sockets over SNA LUs
- Two data sessions between the Sockets over SNA LUs

Some displays relating to the gateway environment are shown in the last section of this chapter, “Connecting Two IP Networks Using CS/2 and CS/NT Gateways” on page 123.

3.7.5 Connecting Two SNA Networks Using CS/NT and CS/2 Gateways

Next, we implemented back-to-back Sockets over SNA gateways to connect two SNA (APPN) networks together using an IP network in the middle. Figure 114 shows this configuration.

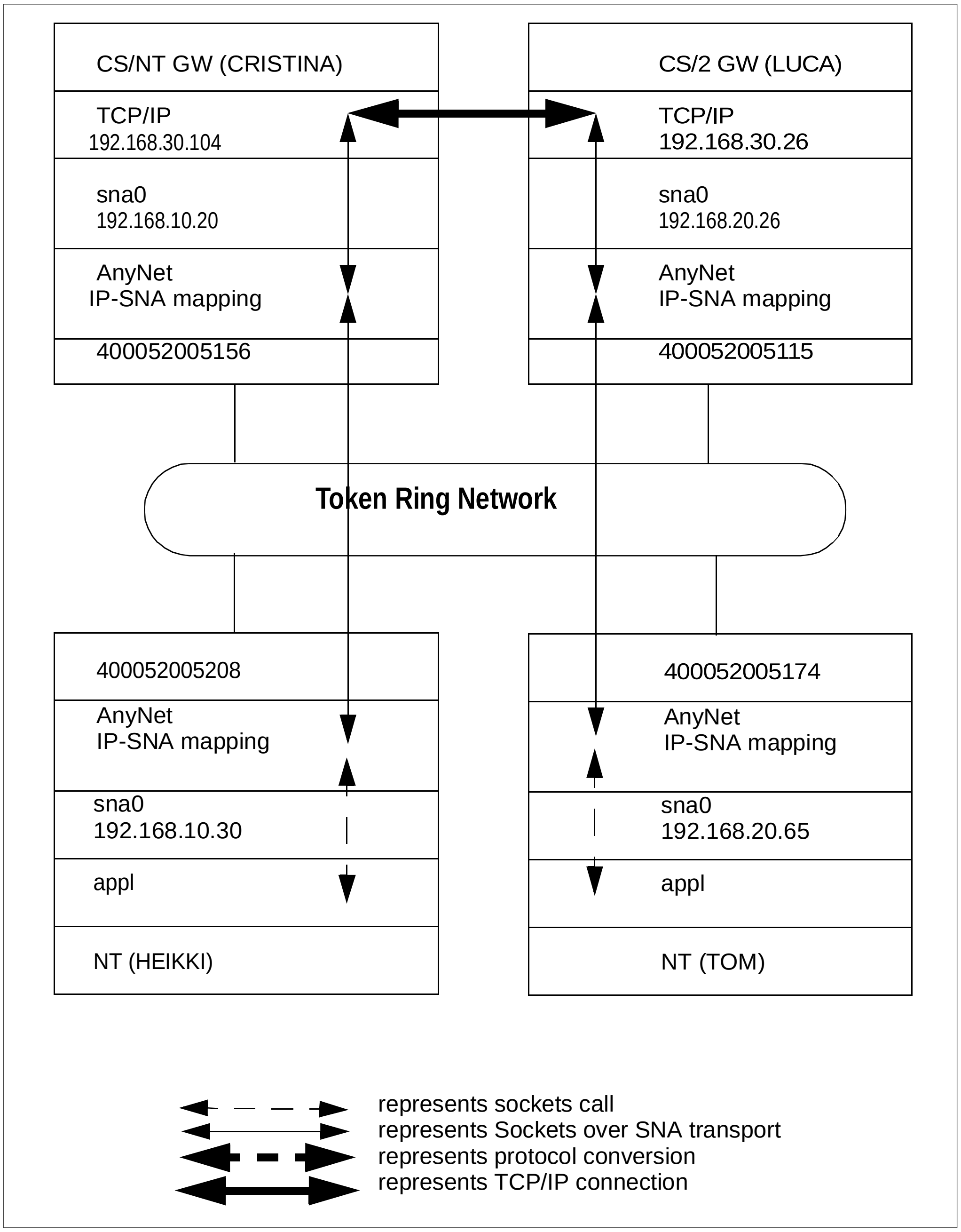


Figure 114. Multiple Sockets over SNA Gateways with IP Between

Here there are two Sockets over SNA access nodes, HEIKKI and TOM. Although physically they share the same token-ring, we have logically separated them so that only a native IP connection is available between their two subnetworks. To allow the IP traffic that starts out as Sockets over SNA datagrams to traverse a native IP transport network we need two gateway nodes (CRISTINA and LUCA) as shown. In this scenario we have implemented CRISTINA on CS/NT and LUCA on CS/2. In the configuration summaries that follow, we have again omitted the native SNA connections for clarity.

Note that we cannot put all the Sockets over SNA nodes in the same IP subnetwork here, as the SNA networks are disjoint. An attempt to contact an IP address in the other half of the Sockets over SNA network would result in IP-to-LU address translation and a fruitless search for an SNA LU that exists only in an unreachable SNA subnetwork. We therefore have three subnetworks:

- 192.168.10 for the network containing HEIKKI
- 192.168.20 for the network containing TOM
- 192.168.30 for the native IP network

The algorithmic address mapping, however, is the same in each disjoint SNA network. There is no danger of duplicate LU names in the same SNA network.

The configuration for the access node HEIKKI is as follows:

- Sockets over SNA local (see Figure 70 on page 82):
  - IP address is 192.168.10.30
  - Subnet mask is 255.255.255.0
- Sockets over SNA LUs (see Figure 72 on page 83):
  - Mapping type is Generate LU names
  - IP address is 192.168.10.0
  - Subnet mask is 255.255.255.0
  - SNA network ID is USIBMRA
  - LU template is SX
- Sockets over SNA routes (see Figure 77 on page 88):
  - Type is Default
  - Router address is 192.168.10.20
  - Direct connection is No

  As with the previous scenario, the access node HEIKKI needs at least this definition to communicate with IP addresses outside its native subnetwork 192.168.10.0. Such addresses now include those in the 192.168.20 subnetwork.
- Sockets over SNA modes (see Figure 74 on page 85):
  - Default mode is SNACKETS
  - Port/mode definitions were left unchanged

The configuration for the access node TOM, which is a mirror image of HEIKKI, is as follows:

- Sockets over SNA local (see Figure 70 on page 82):
  - IP address is 192.168.20.65
  - Subnet mask is 255.255.255.0
- Sockets over SNA LUs (see Figure 72 on page 83):
  - Mapping type is Generate LU names
  - IP address is 192.168.20.0
  - Subnet mask is 255.255.255.0
  - SNA network ID is USIBMRA
  - LU template is SX

- Sockets over SNA routes (see Figure 77 on page 88):
    - Type is Default
    - Router address is 192.168.20.26
    - Direct connection is No
- Sockets over SNA modes (see Figure 74 on page 85):
    - Default mode is BLANK
    - Port/mode definitions were left unchanged

The configuration of the gateway CS/NT node CRISTINA is:

- TCP/IP Configuration
    - IP address of token-ring interface is 192.168.30.104
    - Subnet mask is 255.255.255.0
    - IP Forwarding is enabled
- Sockets over SNA local (see Figure 70 on page 82):
    - IP address is 192.168.10.20
    - Subnet mask is 255.255.255.0
- Sockets over SNA LUs (see Figure 72 on page 83):
    - Mapping type is Generate LU names
    - IP address is 192.168.10.0
    - Subnet mask is 255.255.255.0
    - SNA network ID is USIBMRA
    - LU template is SX
- Sockets over SNA routes (see Figure 77 on page 88):
    - Type is Network
    - Destination address is 192.168.20.0
    - Destination mask is 255.255.255.0
    - Router address is 192.168.30.26
    - Direct connection is No

  Here we need routes to three separate subnetworks. Two of them (192.168.10 and 192.168.30) can be reached through the local interfaces (AnyNet and native IP respectively). The third one, 192.168.20, can only be reached through the partner gateway node LUCA whose address is 192.168.30.26.
- Sockets over SNA modes (see Figure 74 on page 85):
    - Default mode is BLANK
    - Port/mode definitions were left unchanged

The configuration of the gateway CS/2 node LUCA is:

- TCP/IP Configuration
    - IP address of token-ring interface is 192.168.30.26
    - Subnet mask is 255.255.255.0
    - IP Forwarding is enabled
- Sockets over SNA local (see Figure 86 on page 95):
    - IP address is 192.168.20.26
    - Subnet mask is 255.255.255.0
- Sockets over SNA LUs (see Figure 88 on page 96):

  - IP address is 192.168.20.0
  - Subnet mask is 255.255.255.0
  - SNA network ID is USIBMRA
  - LU name prefix is SX

- Sockets over SNA routes (see Figure 91 on page 98):

  - Type is Net
  - Destination address is 192.168.10.0
  - Destination mask is 255.255.255.0
  - Router address is 192.168.30.104
  - Metric is 2

  In CS/2 you do not define a direct connection explicitly. A direct route (one to another Sockets over SNA node in a different IP subnet) is defined by a metric of 0 and the address of the local interface. In our case we have an indirect route (to a subnetwork reachable only via a remote router) so we input the address of that router and a metric (number of hops to the destination) of 2.

- Sockets over SNA modes (see Figure 89 on page 97):

  - Default mode is BLANK

Again, we restarted all the nodes after reconfiguration and successfully established communication between all the IP subnetworks; in particular between HEIKKI and LUCA.

3.7.6 Connecting Two IP Networks Using CS/2 and CS/NT Gateways

For our final scenario, we implemented the opposite situation to that in “Connecting Two SNA Networks Using CS/NT and CS/2 Gateways” on page 120. Now we have (Figure 115 on page 124) two islands of IP connected via an SNA backbone. The Sockets over SNA function is now purely a gateway one, and there are no access nodes in the network.

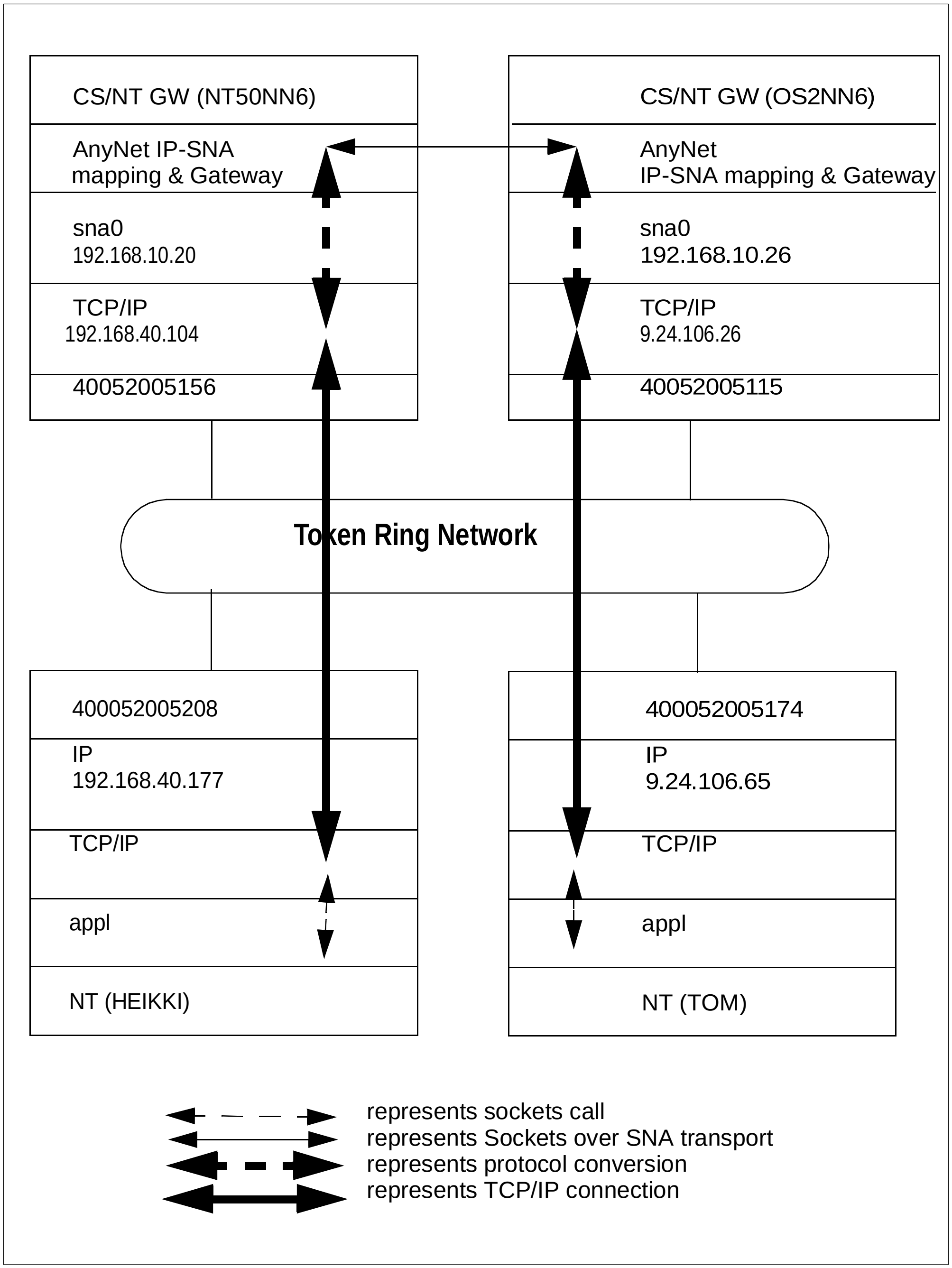


Figure 115. Multiple Sockets over SNA Gateways with SNA Between

The physical setup is exactly the same as before, but now the workstations HEIKKI and TOM are pure native TCP/IP and the connection between the gateways is SNA.

The configuration for the Windows NT workstation HEIKKI is as follows:

- TCP/IP Configuration
  - IP address of token-ring interface is 192.168.40.177
  - Subnet mask is 255.255.255.0
  - Route to 192.168.10.0 (subnet mask 255.255.255.0) is via 192.168.40.104

  - Route to 9.24.96.0 (subnet mask 255.255.224.0) is via 192.168.40.104

  We used network routes in this scenario, rather than defining default routes to 192.168.40.104 which would have had the same effect.

The configuration for the workstation TOM is as follows:

- TCP/IP Configuration
  - IP address of token-ring interface is 9.24.106.65
  - Subnet mask is 255.255.224.0
  - Route to 192.168.40.0 (subnet mask 255.255.255.0) is via 9.24.106.26
  - Route to 192.168.10.0 (subnet mask 255.255.255.0) is via 9.24.106.26

The configuration for the CS/NT gateway NT50NN6 is:

- TCP/IP Configuration
  - IP address of token-ring interface is 192.168.40.104
  - Subnet mask is 255.255.255.0
  - IP Forwarding is enabled
- Sockets over SNA local (see Figure 70 on page 82):
  - IP address is 192.168.10.20
  - Subnet mask is 255.255.255.0
- Sockets over SNA LUs (see Figure 72 on page 83):
  - Mapping type is Generate LU names
  - IP address is 192.168.10.0
  - Subnet mask is 255.255.255.0
  - SNA network ID is USIBMRA
  - LU template is SX
- Sockets over SNA routes (see Figure 77 on page 88):
  - Type is Network
  - Destination address is 9.24.96.0
  - Destination mask is 255.255.224.0
  - Router address is 192.168.10.26
  - Direct connection is No

  This gateway has no direct connections via Sockets over SNA to nodes in a different IP subnetwork. It has, however, an indirect route to the IP network 9.24.96.0 (which contains the 9.24.106 addresses), reachable via its Sockets over SNA partner 192.168.10.26.
- Sockets over SNA modes (see Figure 74 on page 85):
  - Default mode is SNACKETS
  - Port/mode definitions were left unchanged

Finally, the configuration of the CS/2 gateway OS2NN6 was:

- TCP/IP Configuration
  - IP address of token-ring interface is 9.24.106.26
  - Subnet mask is 255.255.224.0
  - IP Forwarding is enabled
- Sockets over SNA local (see Figure 86 on page 95):
  - IP address is 192.168.10.26

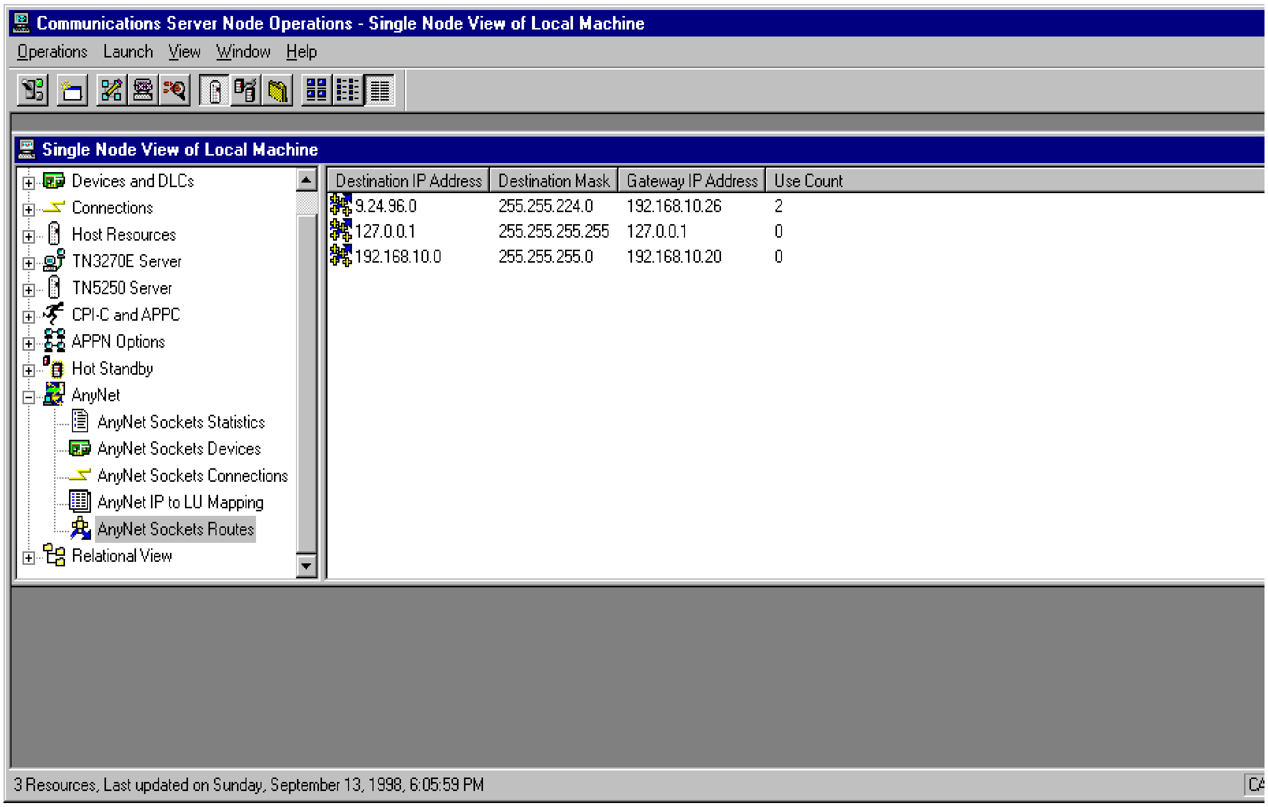
  - Subnet mask is 255.255.255.0

- Sockets over SNA LUs (see Figure 88 on page 96):
  - IP address is 192.168.10.0
  - Subnet mask is 255.255.255.0
  - SNA network ID is USIBMRA
  - LU name prefix is SX

- Sockets over SNA routes (see Figure 91 on page 98):
  - Type is Net
  - Destination address is 192.168.40.0
  - Destination mask is 255.255.255.0
  - Router address is 192.168.10.20
  - Metric is 1

- Sockets over SNA modes (see Figure 89 on page 97):
  - Default mode is SNACKETS

As usual, we established sessions between the two gateway nodes by sending some IP traffic between the native IP workstations. Some displays taken on the CS/NT workstation show the status of the Sockets over SNA environment. For example, Figure 116 illustrates the Sockets over SNA routes. This display can be found in the Node Operations menu by selecting **AnyNet** and then **AnyNet Sockets Routes**.



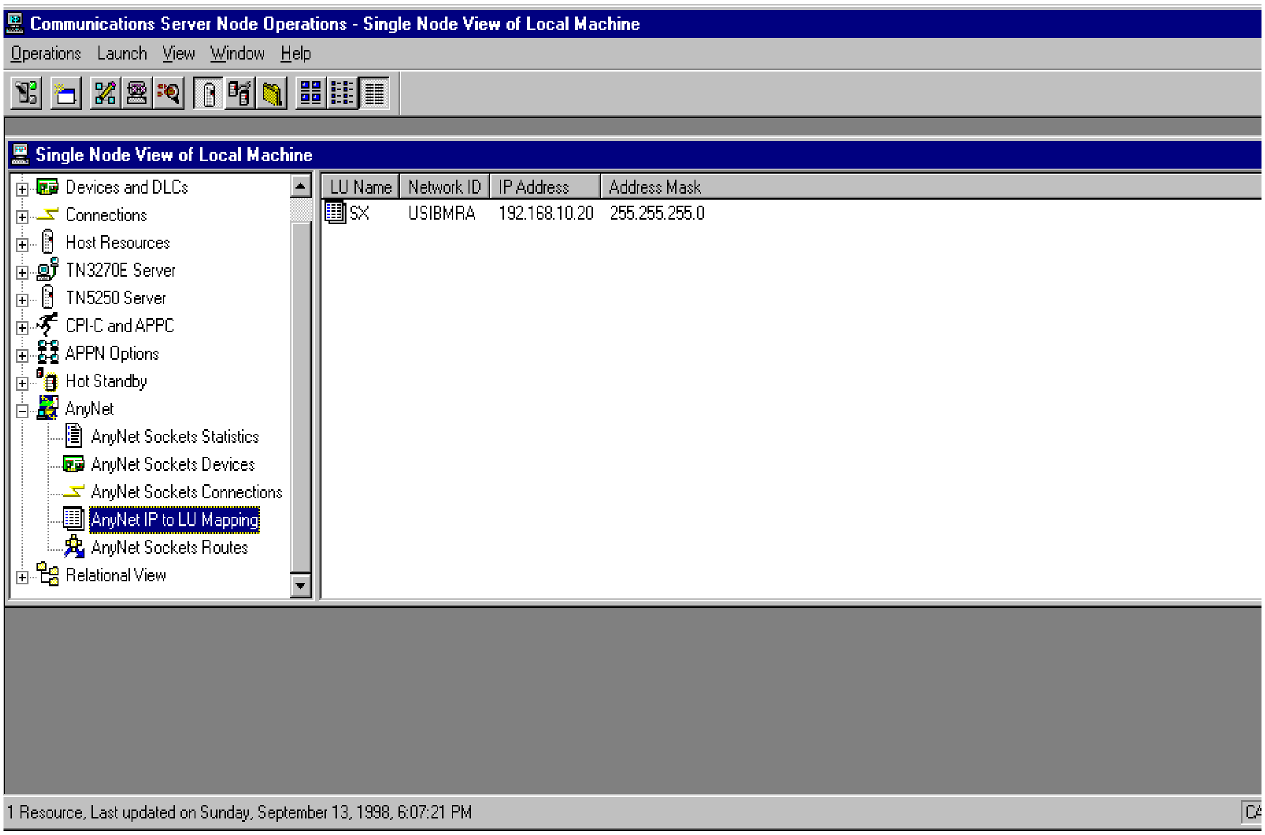
Destination IP Address	Destination Mask	Gateway IP Address	Use Count
9.24.96.0	255.255.224.0	192.168.10.26	2
127.0.0.1	255.255.255.255	127.0.0.1	0
192.168.10.0	255.255.255.0	192.168.10.20	0

3 Resources, Last updated on Sunday, September 13, 1998, 6:05:59 PM

Figure 116. Sockets over SNA Routes Display

The first route (to 9.24.96) was the one we defined in the CS/NT configuration. The other two were dynamically defined; the local loopback address and the direct route to the local Sockets over SNA subnetwork 192.168.10.

Figure 117 shows the display of the IP-to-LU mapping table, again reached from the **AnyNet** option in the Node Operations menu.



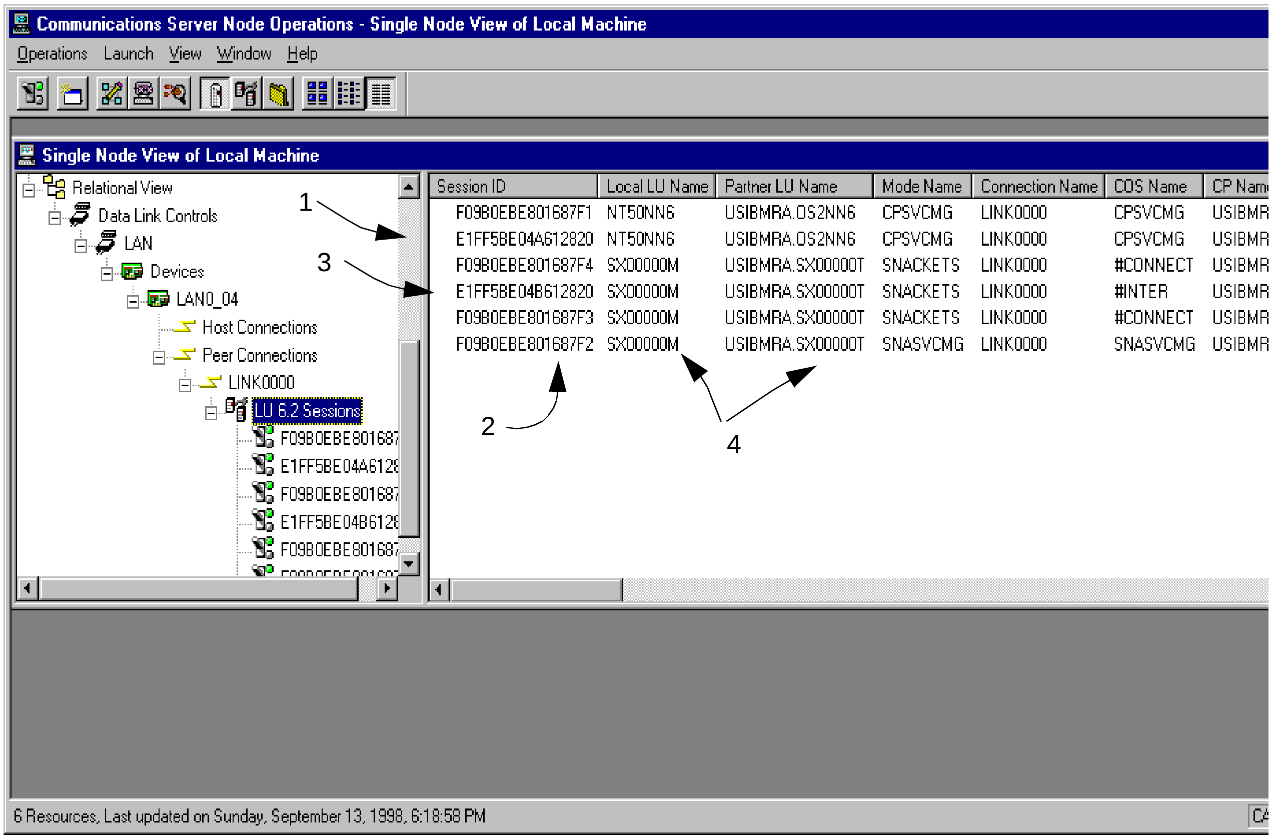
LU Name	Network ID	IP Address	Address Mask
SX	USIBMRA	192.168.10.20	255.255.255.0

1 Resource, Last updated on Sunday, September 13, 1998, 6:07:21 PM

Figure 117. LU Mapping Table Display

This display contains exactly what we defined in the configuration panels.

Figure 118 on page 128 shows the SNA sessions active after the IP traffic had been sent through the network.



Session ID	Local LU Name	Partner LU Name	Mode Name	Connection Name	COS Name	CP Nam
F09B0EBE801687F1	NT50NN6	USIBMRA.OS2NN6	CPSVCMG	LINK0000	CPSVCMG	USIBMR
E1FF5BE04A612820	NT50NN6	USIBMRA.OS2NN6	CPSVCMG	LINK0000	CPSVCMG	USIBMR
F09B0EBE801687F4	SX00000M	USIBMRA.SX00000T	SNACKETS	LINK0000	#CONNECT	USIBMR
E1FF5BE04B612820	SX00000M	USIBMRA.SX00000T	SNACKETS	LINK0000	#INTER	USIBMR
F09B0EBE801687F3	SX00000M	USIBMRA.SX00000T	SNACKETS	LINK0000	#CONNECT	USIBMR
F09B0EBE801687F2	SX00000M	USIBMRA.SX00000T	SNASVCMG	LINK0000	SNASVCMG	USIBMR

6 Resources, Last updated on Sunday, September 13, 1998, 6:18:58 PM

Figure 118. SNA Sessions between Gateways

We see from this display:

1. There are, as usual, two CP-CP sessions between the gateway nodes.
2. There is one CNOS session (as always) between the nodes. This was initiated from the CS/NT side because the workstation HEIKKI was the first to issue a PING.
3. There are three data transfer sessions; one (initiated by OS2NN6) uses the #INTER class of service and the others (initiated by NT50NN6) uses #CONNECT. The mode definitions are not consistent between the two machines.

   There are two data transfer sessions shown with NT50NN6 as the contention winner. This is probably because more than one PING was sent to more than one destination. Sessions for PING and other datagrams (one session at a time) are set up when the first packet is sent out to a given destination, and deallocated after a period of inactivity. Sessions for stream-type connections are set up (one or two depending on the partner nodes' capabilities) and taken down at the same times as the stream socket connection is set up and terminated.
4. The algorithmic mapping has translated 192.168.10.20 to USIBMRA.SX00000M and 192.168.10.26 to SX00000T.

Chapter 4. AnyNet SNA over IP

AnyNet SNA over IP fully meets the requirement of transporting any SNA session across a TCP/IP network. It does not rely on the API to follow certain standards as does Sockets over SNA. This is because SNA over IP implementations use the native IP stack as a transport mechanism and the native SNA stack for the API, whereas Sockets over SNA implementations use their own Sockets interfaces and do not require an IP stack except in the gateway configuration.

However, SNA over IP has one major drawback, in that the SNA logical connection cannot be APPN or subarea INN. It can be one of two things:

1. A low entry networking (LEN) connection, which is supported in both subarea and APPN environments but permits no control sessions and therefore no network dynamics.
2. A peripheral subarea connection between a workstation and an SNA gateway.

Therefore, a certain amount of predefinition is always required in the SNA over IP environment:

- No search requests can be sent and thus partner LU locations must be predefined at the originating node. Although an LU name can be translated into a partner IP address and that address can be discovered dynamically, the node must first be told how to distinguish between native SNA partner LUs and AnyNet SNA over IP LUs.
- Dependent LU sessions between workstation and gateway require some extra predefinition, since the LU name (and therefore the IP address) is not known at either of these nodes; only the host VTAM knows this.
- Dependent LU sessions between workstation (or gateway) and host require the use of DLUR (this is the only case in which DLUR is supported without requiring an APPN path between DLUR and DLUS).
- Dependent LU sessions must always take the same path as the DLUR/S pipe (the SNA over IP logical link), because there is no way of identifying the DLUR node's network connections to the session partner.

Also, HPR is not supported across the IP portion of the path because route setup cannot flow across a LEN link or a peripheral subarea link.

AnyNet SNA over IP is therefore a suitable solution only where the SNA traffic is limited and the above factors are not major issues. Enterprise Extender is a much better solution in most environments.

AnyNet SNA over IP is currently implemented as part of the Communications Server product family which comprises:

- SecureWay Communications Server for OS/390 (access node and gateway)
- OS/400 (access node only)
- Communications Server for AIX (access node only)
- Communications Server for OS/2 (access node and gateway)
- Communications Server for Windows NT (access node and gateway)

In addition, AnyNet access node function is available with Personal Communications for OS/2 and Windows. As with Sockets over SNA, older products have been available packaged in various ways over the years.

4.1 AnyNet SNA over IP Overview

AnyNet SNA over TCP/IP provides IP transport to applications written to various SNA interfaces. Using SNA over TCP/IP, an application program designed to use an SNA API can communicate with another SNA application program over a TCP/IP transport network by mapping an LU-LU session to a TCP connection.

Using MPTN terminology, a native SNA transport user is able to use the non-native TCP/IP transport provider.

All SNA over TCP/IP implementations use the installed TCP/IP application as the base. Thus SNA over TCP/IP supports all attachments supported by the underlying TCP/IP stack. When using SNA over TCP/IP, SNA application programs can access an IP network without modification.

Each SNA over IP implementation works in the same way: it takes SNA request units destined for the partner LU and sends them over the logical connection that represents the IP network. Therefore, the same LU communication types are supported over AnyNet SNA over IP as are supported over LEN or peripheral connections, with the addition of the special case of dependent LU requester. This special case is necessary to allow dependent LU sessions, without which the application of SNA over IP would be very limited.

Because SNA over IP must cause the IP network to look like an SNA connection, it must translate SNA LU names to IP addresses. Thus a session request for a given LU results in:

1. Mapping the partner LU to an IP address
2. Setting up a TCP connection (if one does not already exist) between this node and the partner node
3. Treating the TCP connection as an adjacent link station over which the session data flows.

The configuration information that you need to define for AnyNet SNA over IP comprises:

- Whether to route the data using SNA transport or using TCP/IP transport
- How to determine the IP address from the fully qualified LU name

On the TCP/IP side of the process, you have to designate a special port (TCP and UDP) for use by AnyNet SNA over IP. This enables the TCP/IP stack to route SNA over IP packets to AnyNet while native IP packets are sent to their proper destination. The port number normally used is 397, the well-known port for SNA over IP. Both TCP and UDP are required because AnyNet uses UDP datagrams for expedited SNA data even if the normal data is flowing on a TCP connection.

The translation process for LU names to IP addresses can be done in one of two ways:

- Explicitly. You define the name of each LU with which this node will establish a session, and the IP address that corresponds to the LU.

- Lookup. You define an IP domain name to which the SNA LU name and network ID are prefixed. The resulting IP host name is used to find the corresponding IP address in the normal IP manner. The normal manner consists of either looking up the name in the local hosts file, or contacting the domain name server to resolve the name.

For *outbound* session requests (from the SNA application to the IP network), the mapping process starts with SNA network-qualified LU names which are reformatted into TCP/IP fully qualified domain names and then mapped to IP addresses. For example, an LU name USIBMRA.RA39M may be mapped to the IP host name ra39m.usibmra.raleigh.ibm.com and from there to the IP address known by the domain name server.

For *inbound* session requests (from the IP network to the SNA application), the SNA LU name is passed in the MPTN connect command when the TCP connection is set up. No mapping is needed as it has already been done by the session partner.

When configuring an AnyNet SNA over IP environment, you should make sure that the domain name suffix used to translate LU names to IP host names is unique. This will prevent any conflict between your SNA LUs and real IP hosts in the IP network. Also, be aware that the maximum length for a fully qualified IP host name is 255 characters (as defined in RFC 1034). Thus you must ensure that the combination of network ID, LU name, suffix and intervening periods does not exceed this length.

4.2 SNA over IP in SecureWay Communications Server for OS/390

Unlike AnyNet Sockets over SNA, the SNA over IP variety of AnyNet requires *both* the VTAM and TCP/IP products to be active. VTAM accepts its API calls as usual over the Record or APPC interfaces, but then uses the UNIX Systems Services Sockets API to exchange the data with TCP/IP instead of dealing with it itself. Figure 119 shows the structure of a CS for OS/390 node that uses SNA over IP.

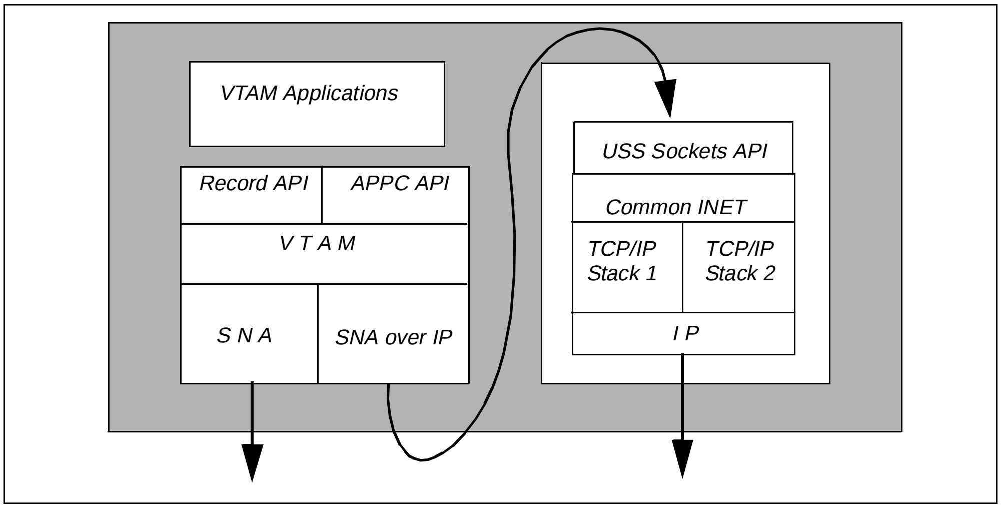


Figure 119. Structure of AnyNet/MVS SNA over IP

Since SecureWay Communications Server for OS/390 Release 5, the integration between the SNA and TCP/IP sides of MVS has increased to the extent that you no longer need special definition considerations on the TCP/IP side for AnyNet SNA over IP. To the TCP/IP stack AnyNet is just another Sockets application and the only thing you need to ensure is that the well-known port 397 is available for use by AnyNet. You can change this port number as long as you keep it consistent throughout your network; changing it from the default is not recommended.

CS for OS/390 supports multiple TCP/IP stacks on the same MVS image, but you do not define which one is to be used for a connection. The UNIX Systems Services Sockets interface invokes the Common INET function which determines the best TCP/IP stack to use for a particular connection. Note that this can result in the VTAM connection becoming inoperative if just one of several TCP/IP stacks is stopped, even if VTAM is using another stack for its AnyNet operations. We expect that this problem will be resolved by APAR OW37897.

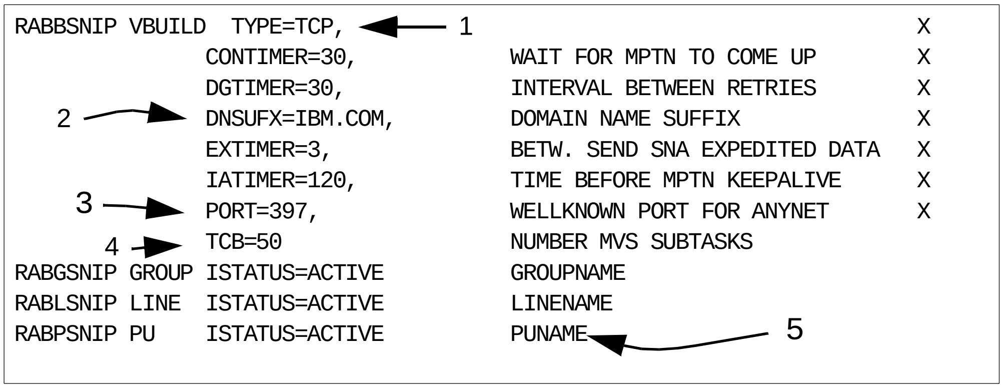
4.2.1 Definitions and Setup

VTAM uses the UNIX Systems Services Sockets API to invoke TCP/IP functions. Therefore, it needs authority to perform these functions. For example, if RACF is your installed security system you need to ensure that VTAM has a RACF user ID with a UNIX Systems Services segment.

Two main sets of definitions are required for AnyNet SNA over IP in OS/390:

1. The major node that defines the IP network connections as SNA link stations
2. The definitions of the remote LUs that will be reached via AnyNet SNA over IP

A new VTAM major node, the TCP/IP major node, defines the AnyNet connections to the IP network. AnyNet SNA over IP supports only one concurrent TCP/IP connection, thus only one TCP/IP PU (link station) may be active at any one time. Figure 120 shows an example of a TCP/IP major node.



```
RABBSNIP VBUILD  TYPE=TCP,                                          X
                CONTIMER=30,           WAIT FOR MPTN TO COME UP     X
                DGTIMER=30,            INTERVAL BETWEEN RETRIES     X
                DNSUFX=IBM.COM,        DOMAIN NAME SUFFIX           X
                EXTIMER=3,             BETW. SEND SNA EXPEDITED DATA X
                IATIMER=120,           TIME BEFORE MPTN KEEPALIVE   X
                PORT=397,              WELLKNOWN PORT FOR ANYNET    X
                TCB=50                 NUMBER MVS SUBTASKS
RABGSNIP GROUP  ISTATUS=ACTIVE         GROUPNAME
RABLSNIP LINE   ISTATUS=ACTIVE         LINENAME
RABPSNIP PU     ISTATUS=ACTIVE         PUNAME
```

Figure 120. TCP/IP Major Node

In this example:

1. The TYPE=TCP keyword identifies this as a TCP/IP major node.
2. The DNSUFX keyword identifies the IP domain that will contain the SNA over IP hosts that correspond to SNA LUs. If the lookup form of address mapping

is used (as it normally should be), then the LU name and network ID are prefixed to this value to form an IP host name.

3. The PORT keyword, which defaults to 397, allows you to change the TCP/UDP port number used by AnyNet.
4. VTAM uses MVS subtasks to access TCP/IP, each subtask being able to handle 120 sessions. If you plan to have more than 1200 SNA over IP sessions (the default for TCB is 10), then you can increase this value up to a maximum of 99.
5. The PU represents the SNA link station used to access the IP network. This is the PU name that must be associated with remote LU definitions for use by AnyNet SNA over IP. More than one PU can be defined (in the same or in separate TCP/IP major nodes) for use under various circumstances, but only one may be active at a time.

In addition, there are several timer values used by AnyNet SNA over IP:

- If no data has been received from a particular remote IP host (AnyNet SNA over IP node) within a given time, VTAM sends an MPTN keepalive message to make sure the connection still exists. This given time is defined by the IATIMER keyword.
- If VTAM is to send expedited data over the AnyNet connection, it first tries to send it as a normal packet on the TCP connection associated with the session. If no response is received within a suitable time, it next attempts to send it as a UDP datagram. This suitable time is defined by the EXTIMER keyword.
- VTAM also uses UDP datagrams for session termination requests and keepalive MPTN messages. If no response is received to a UDP datagram within a certain time, VTAM retries up to four more times. This certain time is defined by the DGTIMER keyword. After a total of five UDP datagrams VTAM gives up trying.
- The remaining timer, CONTIMER, is used by VTAM to await the MPTN (SNA over IP) connection after the TCP connection has been set up.

AnyNet SNA over IP supports cross-network connections as well as same-network. This is no problem architecturally because a LEN connection can have different network IDs at each end. However, you must make sure that VTAM will allow a cross-network link station to be connected over the AnyNet link. The XNETALS start option or PU keyword controls this. Be aware that if you code nothing the default is for cross-network connections to be forbidden.

The TCP/IP major node is new to AnyNet SNA over IP, but the way you define partner (remote) LUs is the traditional VTAM way. Independent and dependent remote LUs are defined as follows:

- Independent LUs, *always* LEN connected for SNA over IP, are defined as cross-domain resources (CDRSCs) in a CDRSC major node. The adjacent link station in the CDRSC definition points to the PU name you defined in the TCP/IP major node. Figure 121 on page 134 shows an example that corresponds to the TCP/IP major node shown above.

  Two alternatives are available to you if you do not wish to predefine LEN partner LUs:

- If the remote LU will always initiate the session, VTAM can dynamically create a CDRSC for it provided that the dynamic LU (DYNLU) option has been set to YES for the AnyNet PU. DYNLU can be coded as a VTAM start option or on the PU statement.
- The VTAM session management exit can be used to associate a link station name with a target LU. If your naming convention is reasonably consistent you can identify the AnyNet SNA over IP LUs in the exit and associate the TCP/IP PU with them.

- Dependent LUs, *always* DLUR resources for SNA over IP in CS for OS/390, are defined in switched major nodes, or by the use of the dynamic definition exits ISTEXCCS and/or ISTEXCSD. There is no special consideration for these because they are reached via AnyNet. As with normal DLUR resources you identify them by node ID, and if you want VTAM to establish the connection you need to identify the DLUR control point name as well. The DLUR CP itself is an independent LU and must be defined as in Figure 121. Figure 122 on page 135 shows a definition of a switched major node for use with AnyNet SNA over IP.

```
**********************************************************************
         VBUILD  TYPE=CDRSC
**********************************************************************
WTR05221 CDRSC ALSLIST=RABPSNIP,ALSREQ=YES
RAIAC    CDRSC ALSLIST=RABPSNIP,ALSREQ=YES
RAPAC    CDRSC ALSLIST=RABPSNIP,ALSREQ=YES
**********************************************************************
```

Figure 121. Independent LU Definition for AnyNet SNA over IP

In this CDRSC major node we define three remote LUs all of which are to be reached via SNA over IP. The ALSLIST keyword points to the PU name in the TCP/IP major node (Figure 120 on page 132) in each case.

Note that the LUs (CDRSCs) defined here need not, themselves, be capable of LEN or AnyNet. What matters is that they can be *reached* by means of the SNA over IP LEN connection. If the connection partner is an AnyNet SNA over IP gateway, a remote LU reachable through that gateway must be defined as above. In that case you must ensure that the remote LU is resolved to the same IP address as the partner gateway; the TCP connection will then be set up to the gateway node and the session request will be forwarded by the gateway into the SNA network.

The ALSREQ keywords are not strictly necessary, but will help to reduce unnecessary searching if your VTAM has APPN as well as LEN capability. VTAM will always search the APPN network for a target resource before looking at a LEN connection, unless ALSREQ=YES is coded. If you know that AnyNet resources are reachable only via IP, coding ALSREQ=YES on their definitions will eliminate the fruitless APPN search.

```
RABSSNIP VBUILD TYPE=SWNET
ISNIPJ01 PU    ADDR=01,                                                 X
               IDBLK=05D,  <-------  1                                  X
               IDNUM=05221,                                             X
               DISCNT=NO,                                               X
               ISTATUS=ACTIVE,                                          X
               MAXDATA=1033,                                            X
               PACING=0,                                                X
               PUTYPE=2,                                                X
               DLOGMOD=D4C32XX3,                                        X
               MODETAB=ISTINCLM,                                        X
               USSTAB=US327X,                                           X
               VPACING=0
*                                    2
*      PATH  PID=1,                              required for VTAM    X
*            DLURNAME=WTR05221,                  to initiate activation X
*            DLCADDR=(1,C,INTPU),                of a PU              X
*            DLCADDR=(2,X,05D05221)
*
*
ISNIPJL1 LU    LOCADDR=2
ISNIPJL2 LU    LOCADDR=3
ISNIPJL3 LU    LOCADDR=4
ISNIPJL4 LU    LOCADDR=5
```

Figure 122. Switched Major Node for DLUR LUs with AnyNet SNA over IP

If the remote LU named WTR05221 is in fact a DLUR-capable control point, Figure 122 shows the definitions needed on VTAM for the dependent LUs on WTR05221. In the definitions as they stand, the only identification VTAM has for the PU is the node ID (1). If WTR05221 contacts VTAM over AnyNet, this identification is sent in a REQACTPU request on the DLUR/S LU 6.2 sessions; VTAM does not need to know the identity of the DLUR node before this moment. If, however, VTAM is to initiate the contact and activation of the dependent resources itself, it must have the DLUR name defined. The commented PATH statement (2) is the way to do this. The DLURNAME keyword identifies the DLUR node, whose location must be defined in a CDRSC major node as shown above. Once the DLUR/S sessions have been established, the DLCADDR keywords are used by VTAM to identify and activate the dependent PU and LUs.

4.2.2 Address Mapping

AnyNet SNA over IP does not use the algorithmic mapping that is possible with Sockets over SNA. The only way to relate LU names to IP addresses is to convert them to IP names and then use DNS or local lookup to translate the IP host names into IP addresses.

The first stage in this process is shown in Figure 120 on page 132; the DNSUFX keyword in the TCP/IP major node defines an IP domain to which the SNA resources are considered to belong. In our example the DNSUFX is ibm.com (the default if no DNSUFX is coded is sna.ibm.com). If, then, VTAM receives a session request for a fully qualified LU name USIBMRA.ARCTOS, the IP host name generated will be arctos.usibmra.ibm.com. If no network ID is supplied VTAM will use its own (native) network ID unless XNETALS=YES and you have coded a NETID keyword on the TCP/IP major node. In this case the coded NETID is used for the address resolution.

The second stage of address mapping is to translate the host name to an IP address. VTAM issues a gethostbyname() call to the Sockets API to obtain the IP address. TCP/IP then tries to resolve the name using DNS, and looks in the HOSTS.LOCAL file if DNS cannot resolve the name.

4.2.3 Gateway Considerations

Because of the structure of an AnyNet SNA over IP node, the gateway configuration does not require special treatment or definition. Any VTAM is capable of passing data through from one interface to another, and SNA over IP is just one particular link station over which sessions may flow. Since the SNA over IP connection is LEN, VTAM regards resources on that connection as owned by itself. Therefore, sessions passing through the gateway do not treat it as an intermediate routing node and the gateway may be an APPN end node.

If there are multiple SNA over IP gateways between an SNA network and an IP network, they have no knowledge of each other as gateways. The choice of routing a session to one gateway or another follows the normal route selection mechanisms in the originating network.

4.3 AnyNet SNA over IP for CS/2

The structure of an AnyNet SNA over IP node running CS/2 is essentially the same as that of a CS for OS/390 node (Figure 119 on page 131). The SNA stack has an additional logical connection available to it in addition to the real SNA links; this connection comprises a Sockets API to the adjacent TCP/IP stack which looks after the TCP/IP transport.

SNA over IP supports the same APIs as native SNA: the Communications Server APPC and CPI-C APIs (for LU 6.2 sessions) and the LU Application (LUA) API for dependent LU sessions. LUA is used by PComm/2; thus PComm sessions can take advantage of AnyNet SNA over IP.

The gateway configuration is similar to the access node configuration except that additional SNA over IP links may need definition under some circumstances. To use the gateway configuration you need to be running the full Communications Server product, not the Access Feature subset. A gateway must be configured as an APPN network node, because it is to route sessions as an intermediate node. This does not apply to VTAM because VTAM acts as the resource owner on behalf of LEN-attached resources, whereas CS/2 acts as the network node server on behalf of such resources.

4.3.1 Dependent LU Support

AnyNet for CS/2 supports dependent LU communication to an SNA host using the dependent LU requester function. The DLUR/S sessions (LU 6.2 over a LEN connection) are set up in the same manner as any other LU 6.2 sessions, and the dependent LU activation flows are encapsulated within these sessions. When a session request is received for a dependent LU, CS/2 knows very little about it until a BIND appears on the LEN (SNA over IP) connection. All the logon, network searching, route selection and session parameter choosing are done by VTAM.

CS/2 can also act as an SNA gateway (this is not the same as an SNA over IP gateway) on behalf of other nodes with dependent LUs. In this case CS/2 maintains LUs on behalf of the downstream nodes, and the upstream connection to VTAM can be either native SNA (DLUR or more traditional forms), or SNA over IP (DLUR only).

If a downstream node is connected via AnyNet SNA over IP, CS/2 can be configured as:

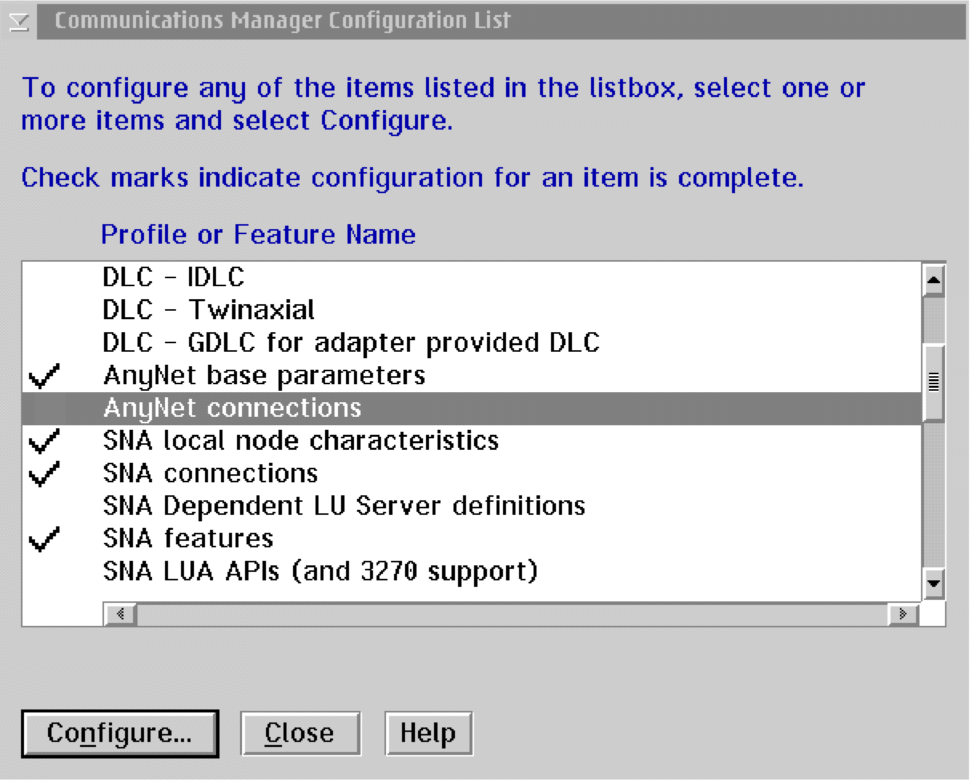
- An SNA gateway and AnyNet access node on behalf of the dependent LUs on the downstream node.
- An AnyNet gateway on behalf of the dependent or independent LUs on the downstream node. For the dependent LUs, this provides the same function as the case above.

If CS/2 is communicating with downstream dependent LUs over an IP network, the LU name to IP address mapping is useless because the LU names are not known; a dependent LU is named by its owning SSCP and identifies itself only by its local address. Therefore, AnyNet nodes on each side of such a connection must have the SNA over IP link predefined. They exchange resource activation protocols using UDP datagrams before embarking on real sessions.

4.3.2 Configuration and Definitions

Defining an SNA over IP configuration is the same as defining a normal CS/2 configuration, with the addition of one (and sometimes two) extra options. These are the **AnyNet base parameters** and the **AnyNet connections** options. They will be found in the Configuration List (Figure 123 on page 138) if you select **Options** and **Configure any profile or feature**.

If you are configuring an access node that has only IP connectivity to the network, you will not need to invoke the **SNA connections** option since the only external connection is via the IP network.



To configure any of the items listed in the listbox, select one or more items and select Configure.

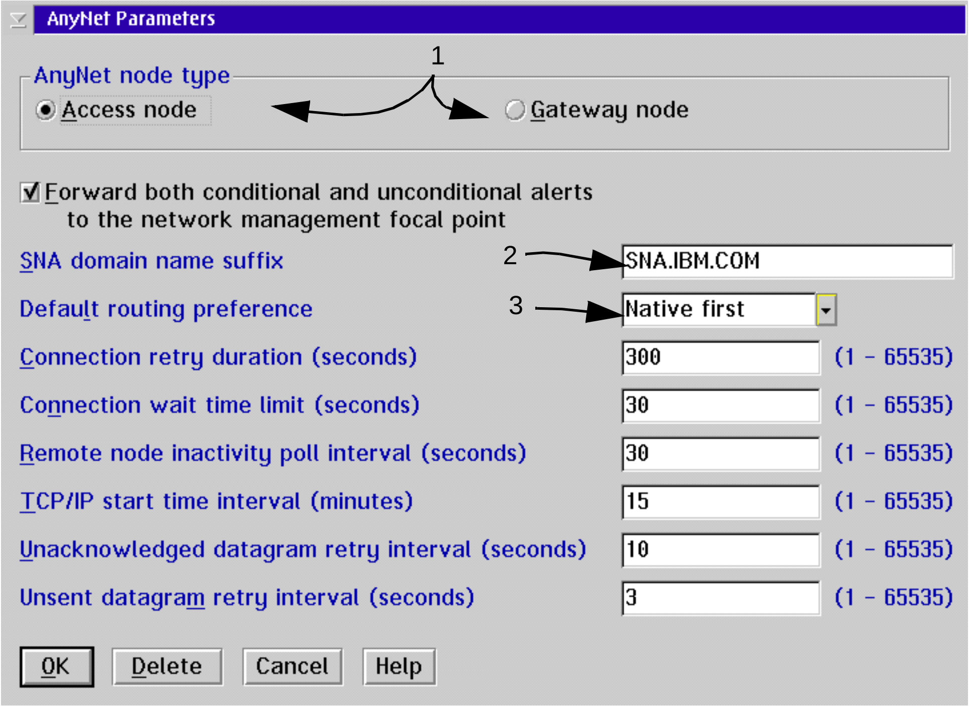
Check marks indicate configuration for an item is complete.

Profile or Feature Name

DLC - IDLC
DLC - Twinaxial
DLC - GDLC for adapter provided DLC
✓ AnyNet base parameters
AnyNet connections
✓ SNA local node characteristics
✓ SNA connections
SNA Dependent LU Server definitions
✓ SNA features
SNA LUA APIs (and 3270 support)

Figure 123. CS/2 Configuration List

The **AnyNet base parameters** option is always required in an SNA over IP configuration; selecting it and clicking **Configure** results in Figure 124 on page 139.



AnyNet node type
● Access node ○ Gateway node

☑ Forward both conditional and unconditional alerts to the network management focal point

SNA domain name suffix	SNA.IBM.COM	
Default routing preference	Native first	
Connection retry duration (seconds)	300	(1 - 65535)
Connection wait time limit (seconds)	30	(1 - 65535)
Remote node inactivity poll interval (seconds)	30	(1 - 65535)
TCP/IP start time interval (minutes)	15	(1 - 65535)
Unacknowledged datagram retry interval (seconds)	10	(1 - 65535)
Unsent datagram retry interval (seconds)	3	(1 - 65535)

Figure 124. AnyNet Base Parameters

The most interesting parameters available on this panel are:

1. Radio buttons allow you to specify the access node or the gateway configuration.
2. The domain name suffix, which defaults to sna.ibm.com, is appended to the LU name and network ID combination to create an IP host name. This name is resolved to an IP address using DNS or the file \mptn\etc\hosts, depending on the setting of the USE_HOSTS_FIRST environment variable in CONFIG.SYS.
3. You must tell AnyNet how to route traffic originating in the SNA network. There are four choices:
    - **Native first** means try SNA first.
    - **Non-native first** means try TCP/IP first.
    - **Native only** means try SNA only.
    - **Non-native only** means try TCP/IP only.

    This parameter defines the default routing preference for all session requests, and can be overridden for specified LUs. If you are configuring a gateway node this parameter should be set to **Native first** or **Native only**.

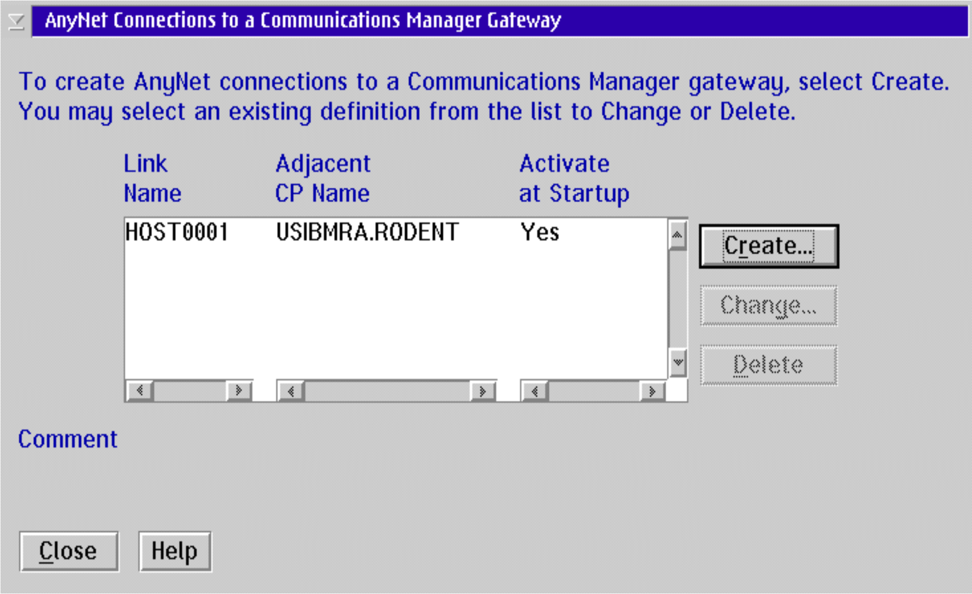
The remaining parameters on this panel comprise various timers (corresponding roughly with the CS for OS/390 timers), and an indication of whether you want AnyNet to forward SNA over IP alerts to your SNA network management focal point.

The AnyNet Connections definitions are required only if:

- This node is an access node and communicates with an SNA gateway using SNA over IP for dependent LU sessions.
- This node is an SNA gateway with explicit workstations defined, which also acts as an SNA over IP gateway for those workstations.

Under most circumstances a remote partner LU can be dynamically discovered by AnyNet across an IP network. On receiving a session request it translates the LU name into an IP address, contacts the IP address, and expects the remote node to accept a BIND for the specified LU name. However, if the SNA over IP gateway also acts as an SNA gateway (in other words, it pretends to be an SNA subarea node) on behalf of downstream SNA over IP nodes, the protocol is a little different. The gateway and the downstream nodes must exchange SNA activation flows (ACTPU, ACTLU and so on) before any session request is processed. Such connections must therefore be predefined to AnyNet.

To define an AnyNet SNA over IP connection *to* an SNA gateway *from* a downstream workstation, use the **AnyNet Connections** option from the Configuration List, as shown in Figure 125.

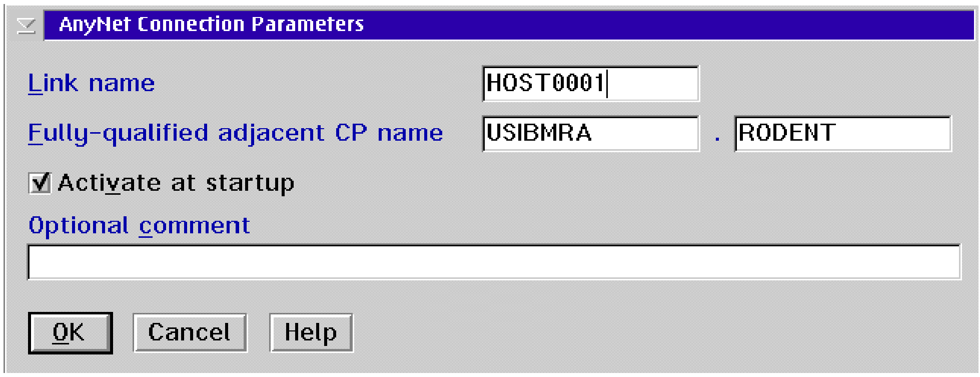


To create AnyNet connections to a Communications Manager gateway, select Create.
You may select an existing definition from the list to Change or Delete.

Link Name	Adjacent CP Name	Activate at Startup
HOST0001	USIBMRA.RODENT	Yes

Figure 125. AnyNet Connections Panel

Selecting **Create** from this panel prompts you (Figure 126 on page 141) to identify the adjacent CP name on this AnyNet link, to supply a link station name, and to specify whether the connection is to be automatically started.



Link name: HOST0001

Fully-qualified adjacent CP name: USIBMRA . RODENT

☑ Activate at startup

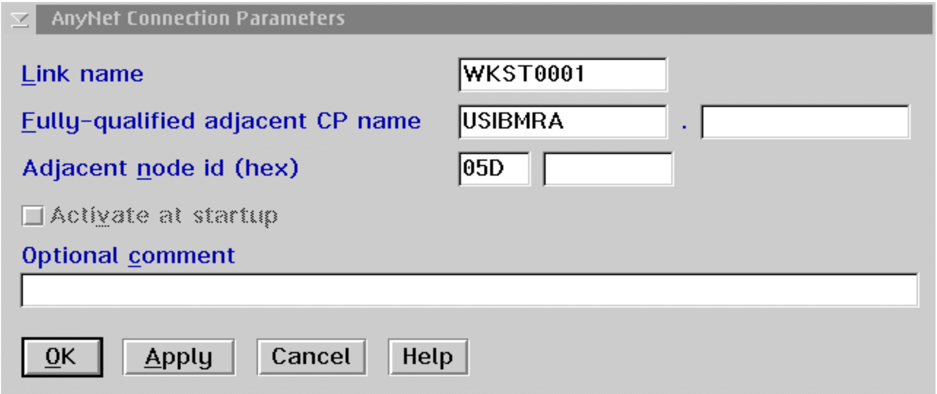
Optional comment:

Figure 126. AnyNet Connection Details

To define an AnyNet SNA over IP connection *from* an SNA gateway *to* a workstation, use instead the Gateway options from the Configuration List. There are three of these:

- **Gateway - Hosts and host LU pools** lets you define upstream connections and LU pools. In fact, this is an alternative way of defining AnyNet upstream connections if you simply select the AnyNet DLC.
- **Gateway - Implicit workstations** lets you define a template for use by LUs using the pools you defined in the Gateway-Hosts option.
- **Gateway - Explicit workstations** lets you define individual connections to downstream nodes.

For example, if you select **Explicit Workstations** you will see a panel listing the available links and the number of LUs defined on each. Initially these do not include the AnyNet SNA over IP connection, so select **Links** and select the **AnyNet** DLC type from the scroll bar on the subsequent panel. Now if you click **Create** you will see the panel shown in Figure 127.



Link name: WKST0001

Fully-qualified adjacent CP name: USIBMRA .

Adjacent node id (hex): 05D

□ Activate at startup

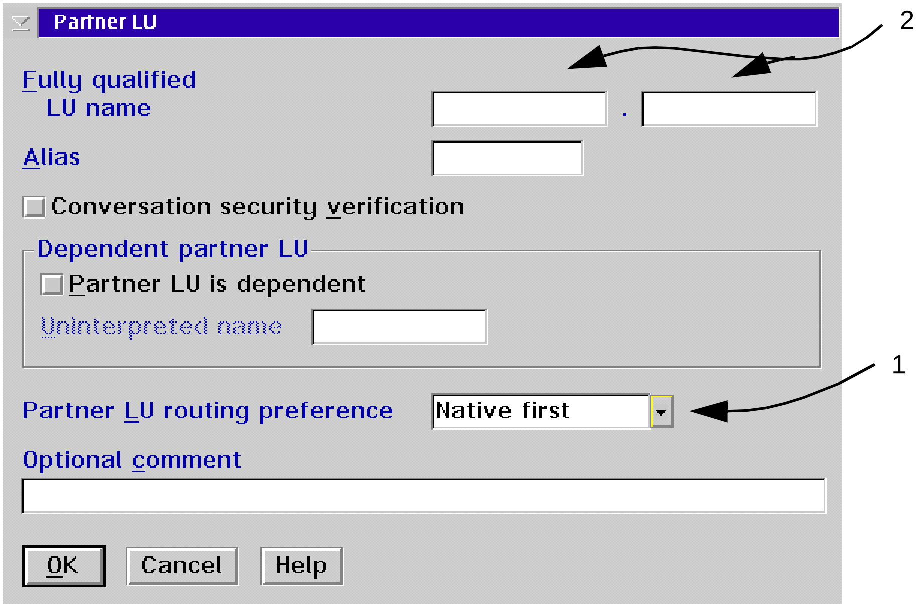
Optional comment:

Figure 127. AnyNet Downstream Connection

Here you can identify the adjacent node by its CP name (which will result in its being located using AnyNet address mapping and then DNS lookup), or by its node ID (if the partner is a type 2 node and does not have a CP name). If you use node ID then the IP host name generated by AnyNet comprises the numeric node ID followed by the domain suffix (for example, 05D05282.sna.ibm.com). Note that the only product which supports AnyNet over SNA using node ID identification is PComm for Windows 3.1.

Once you have defined the AnyNet downstream link, assigning LUs to it is the same as in native SNA configurations.

The final configuration step you may want to take is to define individual routing preferences for particular partner LUs. What you define here overrides the default routing preference from the AnyNet base parameters panel (Figure 124 on page 139). To define individual routing preferences, select **SNA Features** from the Configuration List, then **Partner LU** to see the panel shown in Figure 128.



Fully qualified LU name: .

Alias:

□ Conversation security verification

Dependent partner LU

□ Partner LU is dependent

Uninterpreted name:

Partner LU routing preference: Native first

Optional comment:

Figure 128. Partner LU Definitions

As you can see, this panel lets you change the AnyNet routing preference (1) for the remote LU whose name you define at (2).

4.4 AnyNet SNA over IP for CS/NT

The AnyNet SNA over IP capabilities of CS/NT are effectively the same as those of CS/2, with one subtle difference: a CS/NT AnyNet gateway supports only independent LU sessions. This means that you cannot combine the functions of an SNA gateway and an SNA over IP gateway. However, CS/NT does permit you to define an SNA gateway as an SNA over IP access node for its downstream workstations. Thus you can use SNA over IP for both dependent LU and independent LU sessions to your workstations, as long as the two types of sessions use different gateways. Figure 129 on page 143 illustrates the difference in the two approaches.

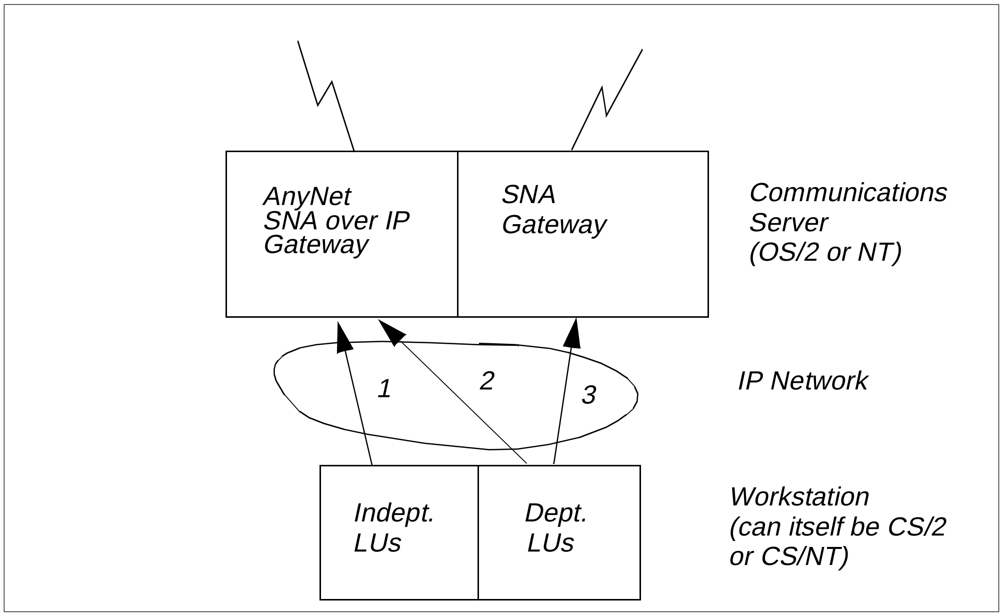


Figure 129. CS/2 and CS/NT AnyNet Difference

CS/2 permits all three types of connection (1,2,3) shown in the diagram, whereas CS/NT permits only 1 and 3.

The options available in CS/NT are therefore:

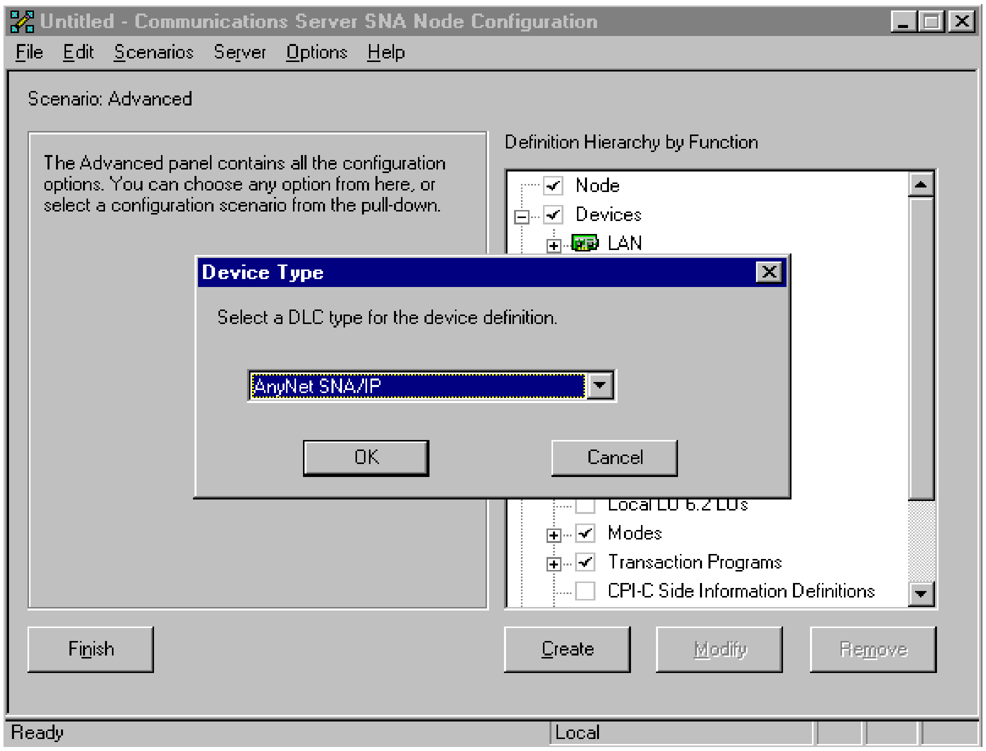
- SNA over IP Access Node using TCP/IP for:
  - LU 6.2 communication
  - Dependent LUs downstream, if this node is an SNA gateway
  - Dependent LUs upstream to host using DLUR
  - Dependent LUs upstream to SNA gateway node
- SNA over IP gateway using TCP/IP for:
  - Independent LU communication only

The same functions are available with the Access Feature and with PComm for Windows NT/95/98, with the usual exceptions:

- End node only
- No SNA gateway
- SNA over IP access node only

4.4.1 Configuration and Definitions

Remember that AnyNet SNA over IP is implemented using a virtual SNA logical connection that represents the IP network. Therefore, configuring an SNA over IP node is the same as configuring a native SNA node with the addition of one or two extra options. Define the local node, the real SNA ports (devices) and the real SNA connections (host resources or peer connections) as usual. Next you must define the virtual port to be used by AnyNet. Select **Devices**, **Create** and then **AnyNet SNA/ IP** as shown in Figure 130 on page 144.


Untitled - Communications Server SNA Node Configuration
File Edit Scenarios Server Options Help
Definition Hierarchy by Function
Node
Devices
LAN
Local LU 6.2 LUs
Modes
Transaction Programs
CPI-C Side Information Definitions
Finish
Create
Modify
Remove
Ready
Local
Device Type
OK
Cancel

Scenario: Advanced

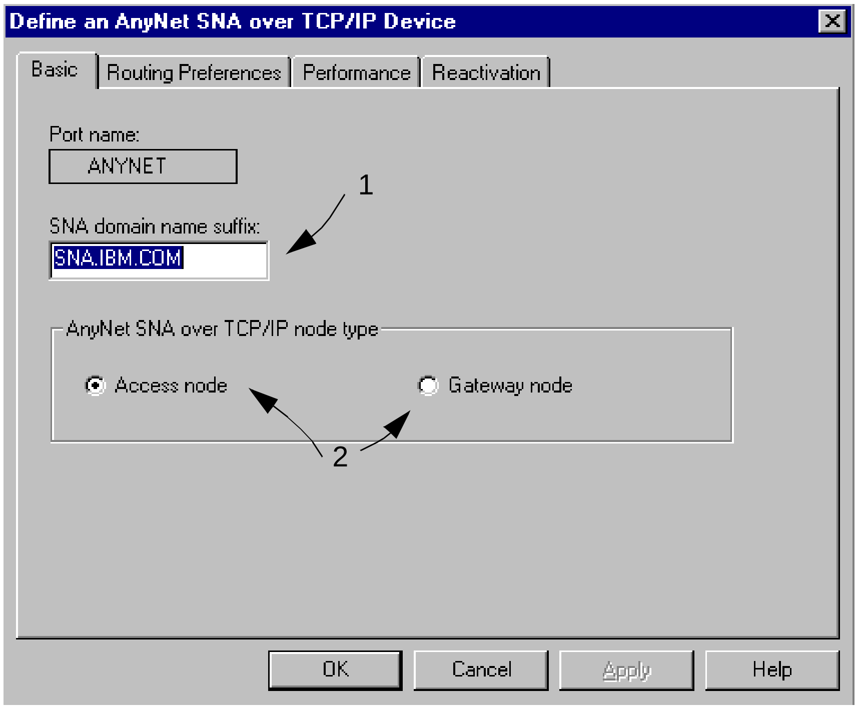
The Advanced panel contains all the configuration options. You can choose any option from here, or select a configuration scenario from the pull-down.

Select a DLC type for the device definition.

AnyNet SNA/IP

Figure 130. Selection of Device for AnyNet SNA/IP

The AnyNet SNA over IP Device Definition group of panels then appears. Figure 131 shows the basic definition parameters.



Port name: ANYNET

SNA domain name suffix: SNA.IBM.COM

AnyNet SNA over TCP/IP node type

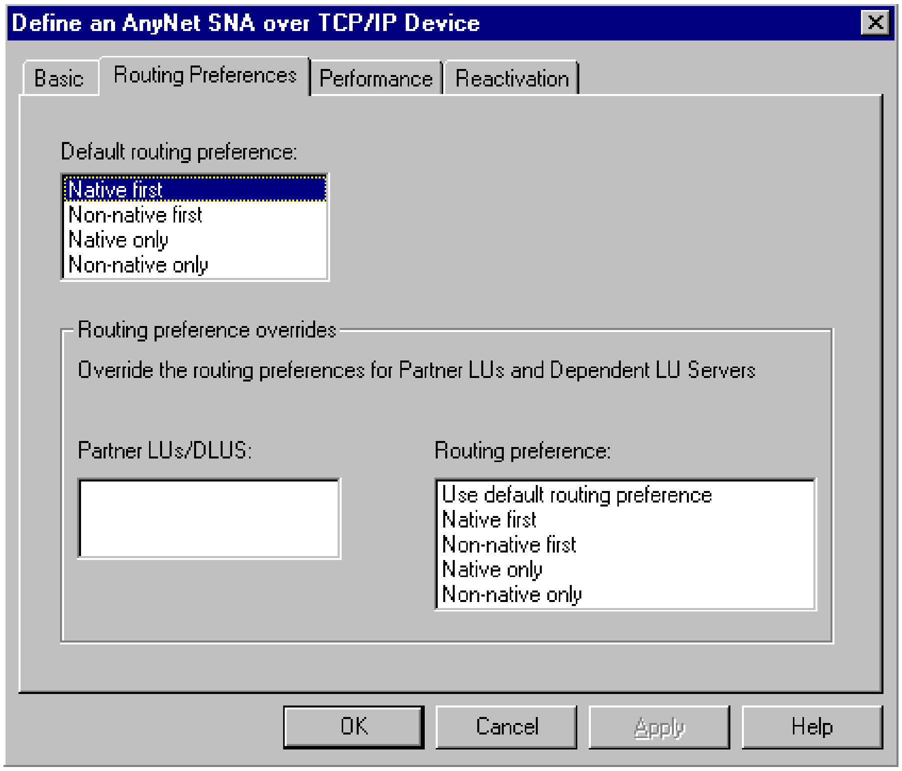
● Access node ○ Gateway node

Figure 131. AnyNet SNA over IP Basic Configuration

The options on this panel are:

1. The AnyNet domain name suffix, which is used to map LU names to IP addresses, defaults to sna.ibm.com.
2. You have the choice of defining an access node or a gateway node. Your SNA node must be a network node in order for an SNA over IP gateway to be defined.

The **Routing Preferences** tab (Figure 132) allows you to customize the routing parameters. Unlike in CS/2, the individual overrides for particular remote LUs are specified in the same panel as the default routing preferences. As with CS/2, you can prevent either SNA or IP from being searched, or you can favor one over the other, for a total of four choices.



Default routing preference:

- Native first
- Non-native first
- Native only
- Non-native only

Routing preference overrides

Override the routing preferences for Partner LUs and Dependent LU Servers

Partner LUs/DLUS:

Routing preference:

- Use default routing preference
- Native first
- Non-native first
- Native only
- Non-native only

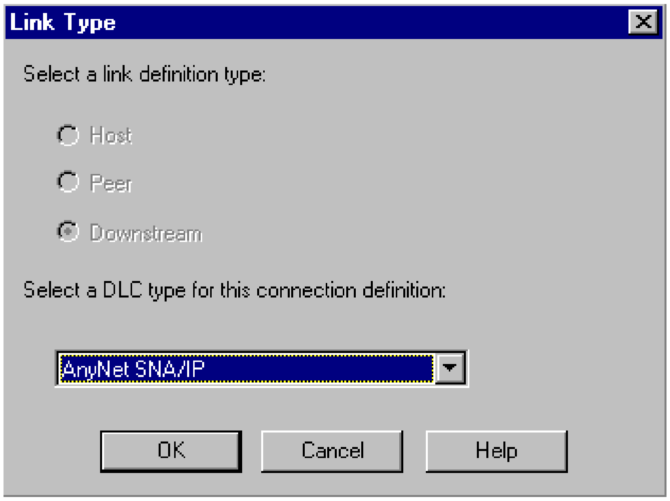
Figure 132. Routing Preferences Panel

In the example above the Partner LUs field is blank. When you define partner LUs to CS/NT (using **CPI-C and APPC** then **Partner LU 6.2 LUs**), these will appear in the box to allow you to select a routing preference for each one. The same applies to dependent LU servers, since they too are partner LU 6.2 LUs. Dependent LUs cannot appear in this field because either:

- They use DLUR protocols, in which case their DLU server's routing rules apply, or
- They use native protocols, in which case they are not known by name but need explicit connection definitions as described below.

As with CS/2, if you are not using native dependent LU connections in an SNA gateway configuration this is all you need to do. You have now defined sufficient parameters to CS/NT to enable it to find any remote SNA partner using native

SNA or TCP/IP. If, however, you are using an SNA gateway (on this node or on behalf of this node) then you need some more explicit connection definitions. To define an upstream connection to an SNA gateway, select **Host Resources**, **Host Connections** and **Create**. To define a downstream connection on an SNA gateway, select **Client Resources** and either **Implicit Client Templates** or **Explicit Client Connections**, followed by **Create**. In each case you will be offered the option of using the AnyNet DLC as your logical connection. To illustrate what you need to define, we show in Figure 133 what you see if you select **Explicit Client Connections**.



Select a link definition type:

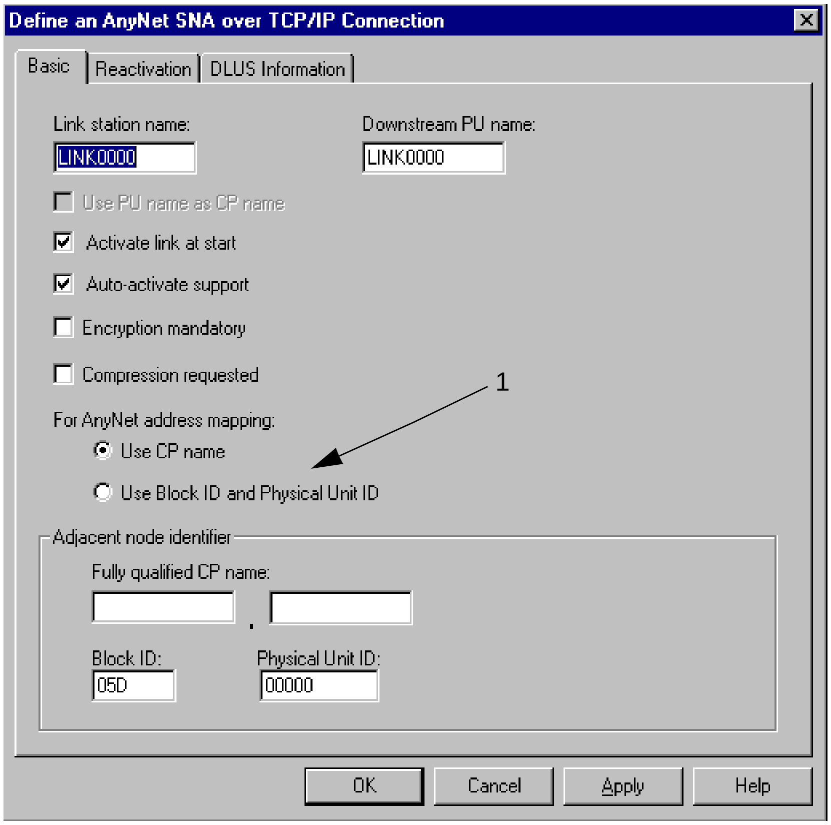
○ Host
○ Peer
● Downstream

Select a DLC type for this connection definition:

AnyNet SNA/IP

Figure 133. Downstream Connection - Creation

This is a downstream connection definition, for which we have selected **AnyNet SNA/IP**. Clicking **OK** gives you the basic connection definition panel as in Figure 134 on page 147. The whole purpose of this panel is to define an AnyNet SNA over IP connection to be used for resources for which you do not know the LU name, and therefore cannot use the normal AnyNet address mapping and lookup.



Define an AnyNet SNA over TCP/IP Connection

Link station name: LINK0000

Downstream PU name: LINK0000

□ Use PU name as CP name

☑ Activate link at start

☑ Auto-activate support

□ Encryption mandatory

□ Compression requested

For AnyNet address mapping:

● Use CP name

○ Use Block ID and Physical Unit ID

Adjacent node identifier

Fully qualified CP name: .

Block ID: 05D

Physical Unit ID: 00000

Figure 134. Downstream Connection - Basic

The downstream node could be a type 2 or a type 2.1 node, and could be identified by its CP name or by its node ID; therefore you are given the option (1) of which identification you wish to use. The default is by CP name, in which case the AnyNet name translation will be used, the partner node will be contacted using UDP, and MPTN protocols will then be used to activate the dependent resources. If you use the node ID (for PComm for Windows 3.1 workstations only), the mapping uses the full node ID as the unqualified host name, just as in CS/2.

Once you have defined a CP name or a node ID, you can assign a DLUS (if required) and dependent LUs to this connection just as in a native SNA configuration.

4.5 Working Scenarios

The working scenarios presented in this section are taken from the redbook *AnyNet: SNA over TCP/IP Installation and Interoperability,* GG24-4395. At the time that book was written there was no AnyNet support on any Windows platform except Windows 3.1, so the scenarios include OS/2 and MVS configurations only. The results described (and the figures we show) remain valid to this day, but we present the configurations by referring to the earlier sections of this chapter rather than by repeating the actual definitions used. In this way we hope to give you up-to-date information which you can use to match your own scenario to the ones we implemented.

All the workstations and OS/390 hosts we used were connected to a token-ring network over which only TCP/IP communication was used. To show the operation of an AnyNet SNA over IP gateway we had additional OS/390 hosts linked to the token-ring-attached ones by subarea or APPN SNA connections. We exercised both independent LU 6.2 and dependent DLUR LU communication using SNA over IP.

The SNA network ID used throughout the tests was USIBMRA.

4.5.1 SNA over IP between OS/390 Hosts

Figure 135 shows our first setup in which two OS/390 SNA over IP access nodes communicate with each other.

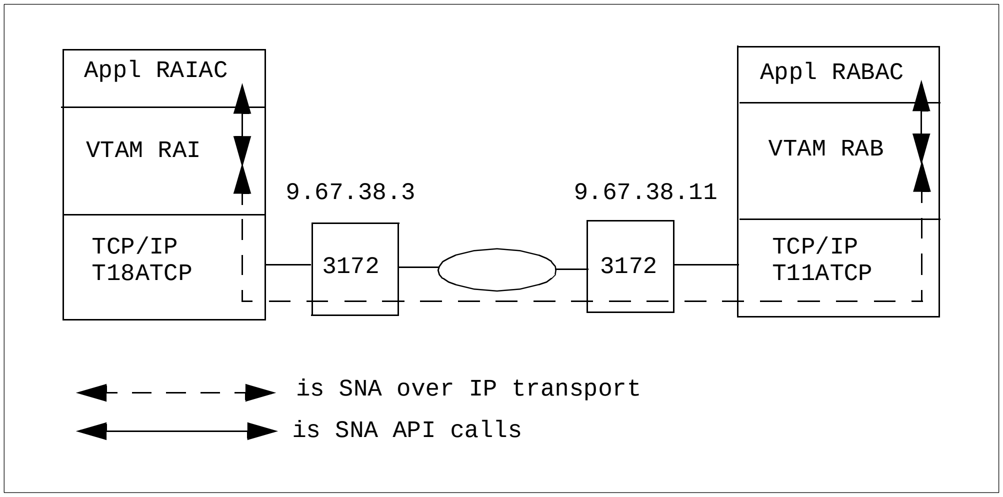


Figure 135. OS/390-to-OS/390 Configuration

We established LU 6.2 communication between two CICS systems, RAIAC and RABAC. They run on VTAMs RAI and RAB respectively; the companion TCP/IP stacks are called T18ATCP and T11ATCP. The TCP/IP stacks are in the same subnetwork and talk to each other via two 3172s as shown.

We first defined a TCP/IP major node on each VTAM system; the one on RAB is shown in Figure 136, the one on RAI being identical except for the resource names where I was substituted for B.

```
RABBSNIP VBUILD  TYPE=TCP,                                             X
               CONTIMER=30,            WAIT FOR MPTN TO COME UP        X
               DGTIMER=30,             INTERVAL BETWEEN RETRIES        X
               DNSUFX=IBM.COM,         DOMAIN NAME SUFFIX              X
               EXTIMER=3,              BETW. SEND SNA EXPEDITED DATA   X
               IATIMER=120,            TIME BEFORE MPTN KEEPALIVE      X
               PORT=397,               WELLKNOWN PORT FOR ANYNET       X
               TCB=50                  NUMBER MVS SUBTASKS
RABGSNIP GROUP ISTATUS=ACTIVE          GROUPNAME
RABLSNIP LINE  ISTATUS=ACTIVE          LINENAME
RABPSNIP PU    ISTATUS=ACTIVE          PUNAME
```

Figure 136. TCP/IP Major Node for RAB

Most of these parameters are in fact the defaults. The two important ones are the domain name suffix of ibm.com and the PU (logical link station) name of RABPSNIP.

Next we had to ensure that the LU name to IP address mapping would work correctly. There are four potential target LUs in the network (RAIAC, RABAC, RAI and RAB), any of which could be used as the destination of an LU 6.2 session request. Therefore, DNS or HOSTS.LOCAL (we used HOSTS.LOCAL because the network was very simple) must be able to resolve:

- raiac.usibmra.ibm.com to 9.67.38.3
- rai.usibmra.ibm.com to 9.67.38.3
- rabac.usibmra.ibm.com to 9.67.38.11
- rab.usibmra.ibm.com to 9.67.38.11

The other main VTAM definitions required to make this work were the CDRSC major nodes. These are required (unless the session management exit is coded to select the link station RA*PSNIP for them) because LEN-connected LUs cannot be dynamically discovered. Figure 137 shows the CDRSC major node on RAB.

```
*********************************************************************
         VBUILD  TYPE=CDRSC
*********************************************************************
RAIAC    CDRSC ALSLIST=RABPSNIP,ALSREQ=YES
*********************************************************************
```

Figure 137. CDRSC Major Node on RAB

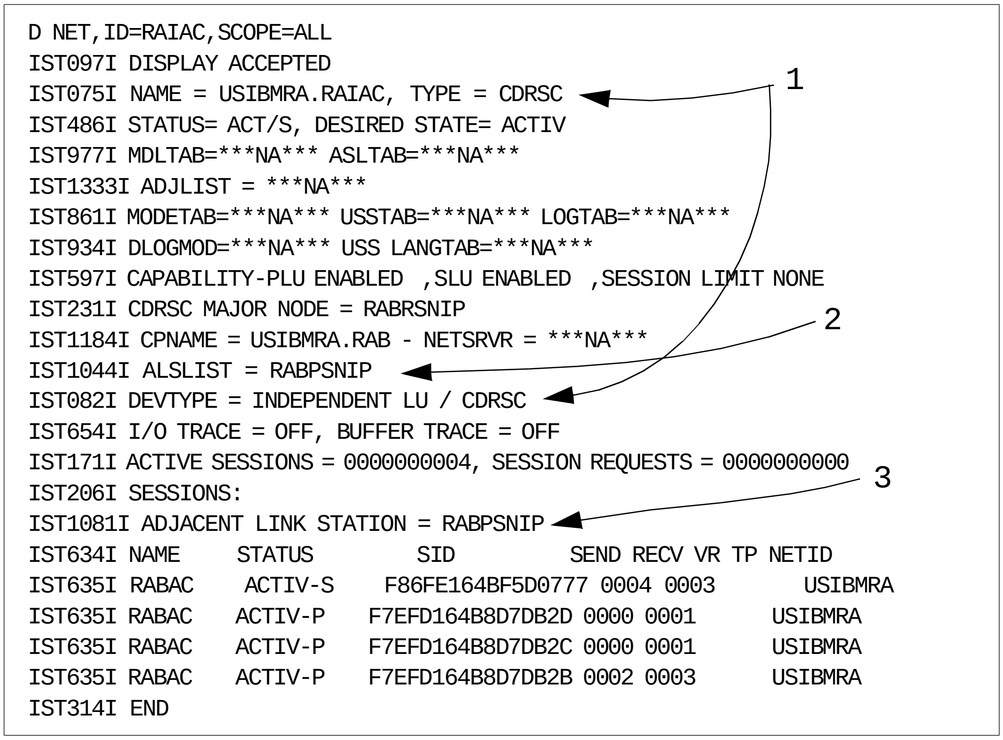
This definition associates RAIAC (the native network ID USIBMRA is the default) with the TCP/IP link station RABPSNIP. Thus, when VTAM receives a session request for RAIAC it will attempt to send a BIND on the link RABPSNIP. If this connection is not active AnyNet will establish a TCP connection and use MPTN protocols to set up the MPTN connection.

You will note the absence of RAI as a CDRSC in this definition. This was actually because we did not use APING to RAI for this test, preferring instead to set up CICS-to-CICS communication. Another reason might be if RAI always initiates the session; VTAM can define CDRSCs dynamically on an inbound session request. A third reason for omitting RAI might be because, under certain circumstances, VTAM can dynamically define a CDRSC for an adjacent control point LU (which RAI is here) for an *outbound* session request. The conditions under which VTAM does this are:

1. The start option CPCDRSC=YES is in effect.
2. The link station has CONNTYPE=LEN defined, either explicitly or from the start option.
3. The SME exit does not have the adjacent link station selection function active.

If *all three* of these are true then you do not need to define RAI as a CDRSC.

With these definitions the LU 6.2 sessions between the CICS applications were established correctly. We then displayed the remote LU, RAIAC, from RAB as shown in Figure 138 on page 150.

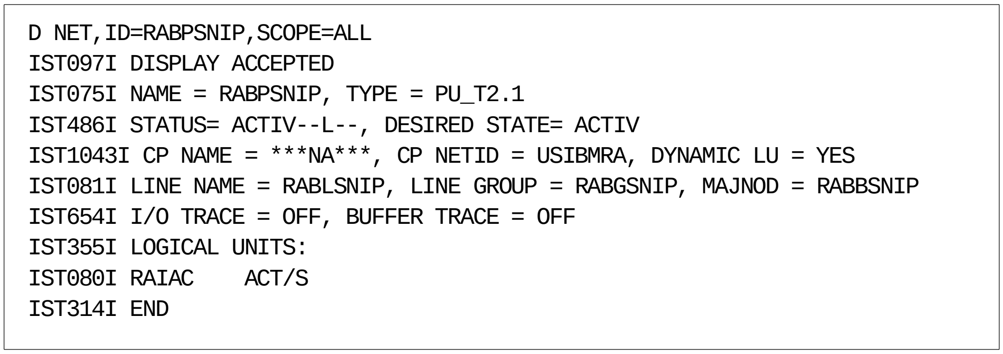


```
D NET,ID=RAIAC,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST075I NAME = USIBMRA.RAIAC, TYPE = CDRSC
IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
IST977I MDLTAB=***NA*** ASLTAB=***NA***
IST1333I ADJLIST = ***NA***
IST861I MODETAB=***NA*** USSTAB=***NA*** LOGTAB=***NA***
IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT NONE
IST231I CDRSC MAJOR NODE = RABRSNIP
IST1184I CPNAME = USIBMRA.RAB - NETSRVR = ***NA***
IST1044I ALSLIST = RABPSNIP
IST082I DEVTYPE = INDEPENDENT LU / CDRSC
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000004, SESSION REQUESTS = 0000000000
IST206I SESSIONS:
IST1081I ADJACENT LINK STATION = RABPSNIP
IST634I NAME     STATUS        SID          SEND RECV VR TP NETID
IST635I RABAC    ACTIV-S    F86FE164BF5D0777 0004 0003       USIBMRA
IST635I RABAC    ACTIV-P    F7EFD164B8D7DB2D 0000 0001      USIBMRA
IST635I RABAC    ACTIV-P    F7EFD164B8D7DB2C 0000 0001      USIBMRA
IST635I RABAC    ACTIV-P    F7EFD164B8D7DB2B 0002 0003      USIBMRA
IST314I END
```

Figure 138. Remote SNA over IP LU Display

Seen from RAB, RAIAC is an independent LU and a CDRSC (1), as we can expect from a LEN-connected resource. It can be reached via the link station RABPSNIP (2) which is the name of the PU defined in the TCP/IP major node (Figure 136 on page 148). All four sessions between RABAC and RAIAC are using this link station (3) as their path out of RAB to RAI.

We also displayed the status of the TCP/IP logical link station, as shown in Figure 139.



```
D NET,ID=RABPSNIP,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST075I NAME = RABPSNIP, TYPE = PU_T2.1
IST486I STATUS= ACTIV--L--, DESIRED STATE= ACTIV
IST1043I CP NAME = ***NA***, CP NETID = USIBMRA, DYNAMIC LU = YES
IST081I LINE NAME = RABLSNIP, LINE GROUP = RABGSNIP, MAJNOD = RABBSNIP
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST355I LOGICAL UNITS:
IST080I RAIAC    ACT/S
IST314I END
```

Figure 139. TCP/IP Link Station Display

This is the same as any LEN connection, with nothing to indicate that TCP/IP is being used to transport the sessions.

4.5.2 SNA over IP between OS/390 Hosts through a Gateway

Our second scenario extends the scope of the first by adding an extra SNA node (another VTAM, as it happens) to the network. Now the new node, USIBMRA.RA3, is reachable only by means of native SNA protocols and RAB has to act as an AnyNet SNA over IP gateway. Figure 140 shows the setup.

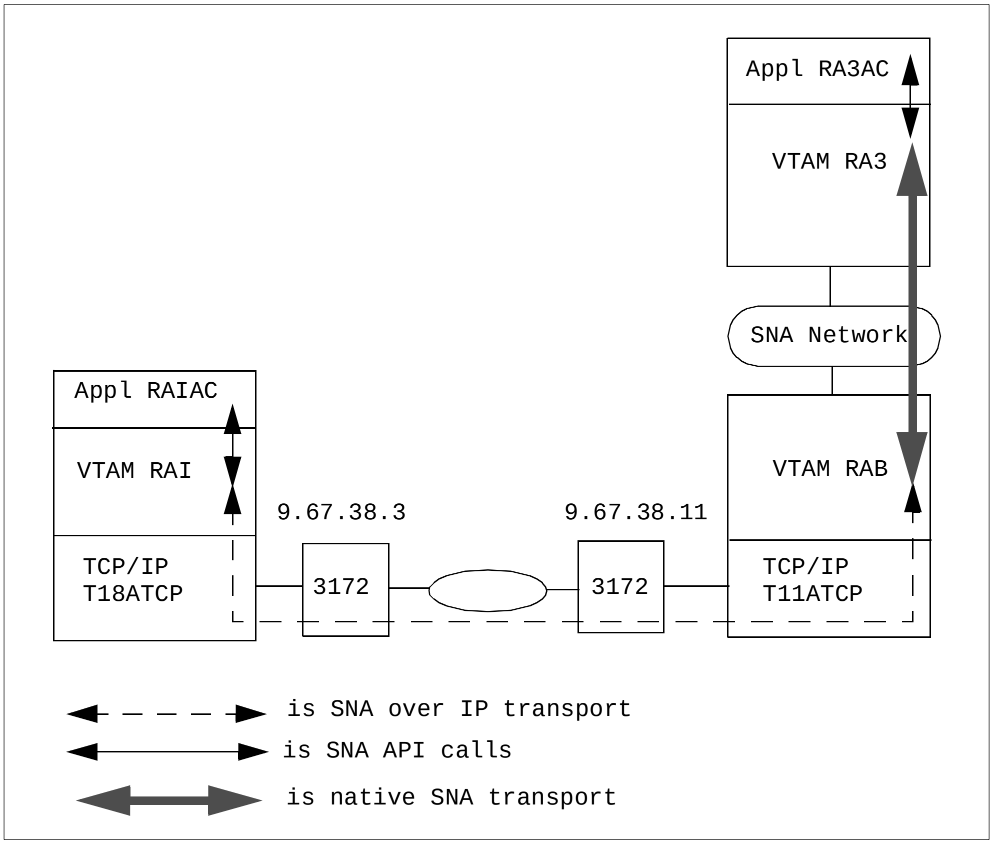


Figure 140. AnyNet MVS-to-MVS through a Gateway

Again we desired to establish LU 6.2 sessions, this time between RAIAC and RA3AC. The changed we needed to make to the definitions in the previous scenario were as follows:

- On RAI:

  Now RA3AC is a remote LU reachable via the gateway RAB. Therefore we had to add a new entry to the CDRSC major node, which now looks like Figure 141.

```
**********************************************************************
         VBUILD  TYPE=CDRSC
**********************************************************************
RABAC    CDRSC ALSLIST=RAIPSNIP,ALSREQ=YES
RA3AC    CDRSC ALSLIST=RAIPSNIP,ALSREQ=YES
**********************************************************************
```

Figure 141. CDRSC Major Node on RAI

RABAC and RA3AC are reached in the same way, namely across the IP network.

- On RAB:

  The connection between RAB and RA3 is native SNA, so the definition (if any) of RA3AC is not dependent on AnyNet. RA3AC is always a CDRSC to RAB, but it may be defined explicitly or dynamically depending on the configuration and the installation's policies.

- On RA3:

  The partner LU RAIAC is seen by RA3 as being owned by RAB, since it is actually LEN attached to RAB. Therefore, it is defined (if at all) to RA3 as a CDRSC on RAB.

- In the IP network:

  The only new circumstance that may impact the IP network is if RAI requests a session with RA3 or (as in our case) RA3AC. Since RAI now understands that RA3AC can be reached through the IP network (Figure 141 on page 151), the IP network must be configured so that ra3ac.usibmra.ibm.com is resolved (along with rabac) to RAB's IP address 9.67.38.11. Accordingly, we added precisely that entry to our HOSTS.LOCAL file on T18ATCP.

After making these changes we successfully established the LU 6.2 sessions between RAIAC and RA3AC. Displays of the partner LU (RA3AC) and the TCP/IP link station (RAIPSNIP) from RAI showed similar results to those seen in Figure 138 on page 150 and Figure 139 on page 150 respectively. To RAI:

- RA3AC is an independent LU and CDRSC reachable via the link station RAIPSNIP, which carries all the sessions between RAIAC and RA3AC.
- RAIPSNIP is a PU type 2.1 with RA3AC as a logical unit reachable through it.

4.5.3 SNA over IP between Hosts through a Pair of Gateways

Next, we extended our SNA network on the RAI side by adding a second VTAM node, RAP, and connecting it to RAI by means of an SNA connection. Figure 142 on page 153 shows the new configuration.

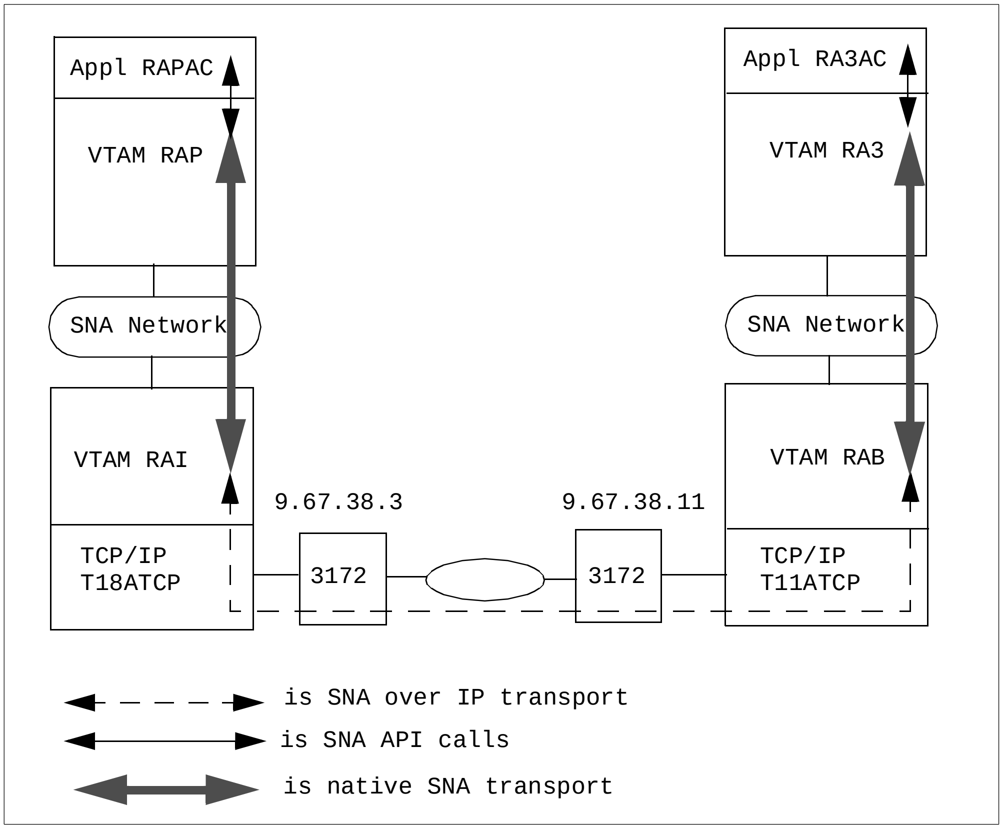


Figure 142. AnyNet MVS-to-MVS with Two Gateways

Now, to establish sessions between RAPAC (another CICS application on RAP) and the rest of the CICS systems we need to make the following changes:

- On RAP and RA3:

  No definitions are required if these VTAM nodes are able to locate partner resources dynamically, which is normally the case both in subarea and APPN environments. If you wish to define RAPAC and RA3AC to each other they are both seen as CDRSCs owned by RAI (as far as RAP is concerned) and by RAB (as seen by RA3). Between RAI and RAB the connection is LEN and the two parts of the SNA network (whether subarea or APPN) are unknown to each other.

- On RAI:

  RAPAC is defined (if at all) as a CDRSC owned by RAP. RA3AC is already defined from the previous scenario.

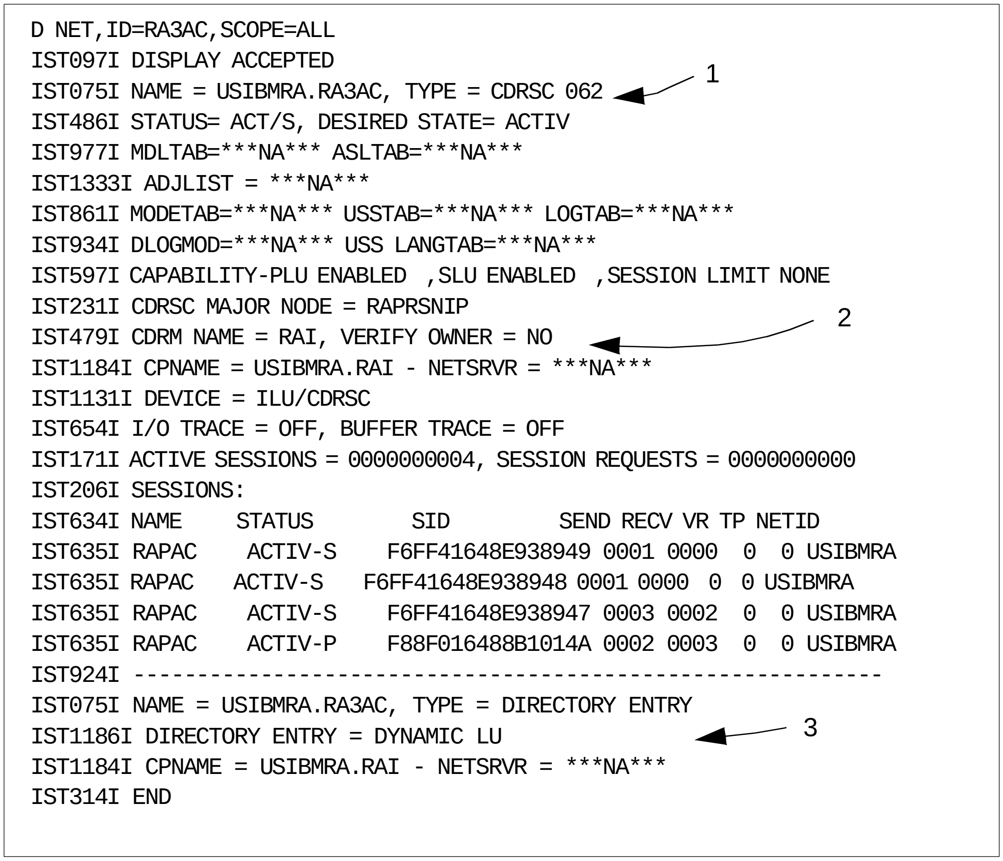
- On RAB:

  RAPAC must be defined as a CDRSC reachable via the TCP/IP link station RABPSNIP.

- In the IP network:

  The IP host rapac.usibmra.ibm.com must be associated with RAI's IP address, that is 9.67.38.3. We therefore added this entry to the HOSTS.LOCAL file on T11ATCP.

When we set up the LU 6.2 sessions between RAPAC and RA3AC, the displays of remote resources and link stations showed the expected results. One new display we took was to show RA3AC from RAP (a resource in one of the disjoint SNA networks from the other network). Figure 143 is the result.



```
D NET,ID=RA3AC,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST075I NAME = USIBMRA.RA3AC, TYPE = CDRSC 062
IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
IST977I MDLTAB=***NA*** ASLTAB=***NA***
IST1333I ADJLIST = ***NA***
IST861I MODETAB=***NA*** USSTAB=***NA*** LOGTAB=***NA***
IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT NONE
IST231I CDRSC MAJOR NODE = RAPRSNIP
IST479I CDRM NAME = RAI, VERIFY OWNER = NO
IST1184I CPNAME = USIBMRA.RAI - NETSRVR = ***NA***
IST1131I DEVICE = ILU/CDRSC
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000004, SESSION REQUESTS = 0000000000
IST206I SESSIONS:
IST634I NAME     STATUS         SID          SEND RECV VR TP NETID
IST635I RAPAC     ACTIV-S    F6FF41648E938949 0001 0000  0  0 USIBMRA
IST635I RAPAC    ACTIV-S    F6FF41648E938948 0001 0000  0  0 USIBMRA
IST635I RAPAC     ACTIV-S    F6FF41648E938947 0003 0002  0  0 USIBMRA
IST635I RAPAC     ACTIV-P    F88F016488B1014A 0002 0003  0  0 USIBMRA
IST924I -------------------------------------------------------------
IST075I NAME = USIBMRA.RA3AC, TYPE = DIRECTORY ENTRY
IST1186I DIRECTORY ENTRY = DYNAMIC LU
IST1184I CPNAME = USIBMRA.RAI - NETSRVR = ***NA***
IST314I END
```

Figure 143. Display of Remote Partner across Two Gateways

If you compare this with Figure 138 on page 150, you will note that:

1. The session partner is still a CDRSC, but
2. This is a real subarea CDRSC, since its owning VTAM is shown as another SSCP. The SNA network comprising RAP and RAI is, in fact a subarea network and RAP thinks that RA3AC resides on RAI. In subarea terms this is true; RA3AC is a LEN-connected LU owned by RAI.
3. RAP also has a directory entry for RA3AC showing RAI as the owner. This indicates that RAP is an APPN network node and is connected to RAI (another network node) by APPN methods as well as subarea.

4.5.4 SNA over IP between Workstation and OS/390

Our fourth configuration is a simpler one, comprising just a workstation node and an OS/390 node acting as SNA over IP access nodes to communicate with each other over an IP network. The workstation was actually running OS/2 at the time, but we refer to the CS/NT as well as the CS/2 configuration panels to show you how each of these needs to be set up in this environment. Figure 144 on page 155 shows the network described in this section.

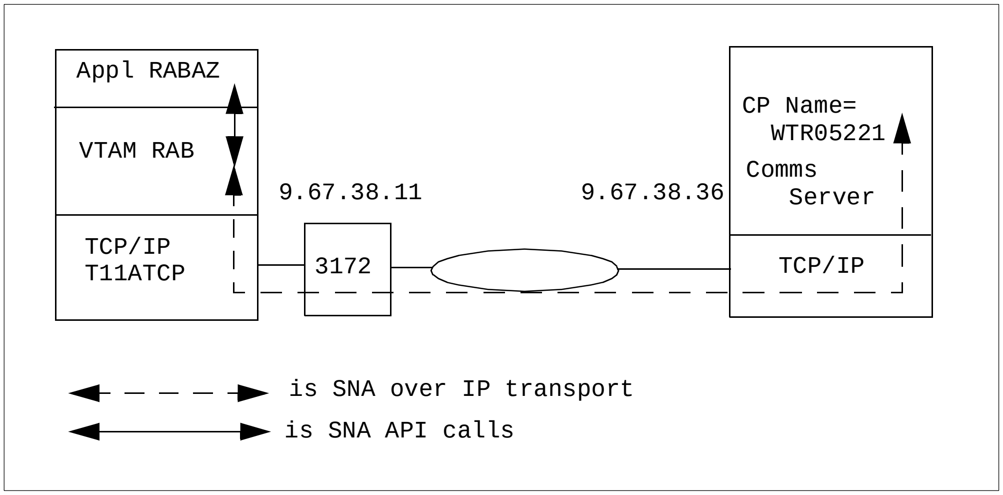


Figure 144. Workstation to OS/390 Access Nodes

Just to demonstrate that AnyNet SNA over IP works as well with the VTAM APPC API as with the record API, we used APPC/MVS (application named RABAZ) to communicate with the control point USIBMRA.WTR05221 on the workstation. We used the APING transaction to test the connectivity between the nodes (from VTAM V4R3 onwards VTAM itself can perform APING without the need for APPC/MVS).

The definitions on OS/390 were the same as in previous scenarios, except that we had to define WTR05221 to RAB as being yet another CDRSC associated with the link station RABPSNIP.

The definitions required on the Communications Server workstation are:

- SNA over IP domain name suffix is **ibm.com**.

  This is shown in Figure 124 on page 139 for CS/2 and in Figure 131 on page 144 for CS/NT.

- Node type is **access node**.

  This is shown in Figure 124 on page 139 for CS/2 and in Figure 131 on page 144 for CS/NT.

- Default routing preference is **Native first**.

  This parameter is not relevant because there is only one choice (non-native) in this example. The parameter is defined in Figure 124 on page 139 for CS/2 and in Figure 132 on page 145 for CS/NT.

- No individual routing preferences for partner LUs.

  If the workstation has been part of an SNA network and the only AnyNet resource was RABAZ, it would have been appropriate to define this as **Non-native first** or **Non-native only** in Figure 128 on page 142 for CS/2 or in Figure 132 on page 145 for CS/NT.

- No AnyNet connections explicitly defined.

  These are not required since no SNA gateway is involved. As this is an access node the appropriate definitions of the connection to the gateway would have been those in Figure 126 on page 141 for CS/2. On CS/NT you would select **Host Resources** and **Host Connections** and fill in panels similar (but not identical) to Figure 133 on page 146 and Figure 134 on page 147.

With these definitions we were able to execute APING transactions successfully between RABAZ and WTR05221.

4.5.5 SNA over IP between Workstation and OS/390 through a Gateway

Our fifth scenario is similar to the previous one (a connection between a workstation and an OS/390 host), but this time there is an OS/390 AnyNet SNA over IP gateway between the partner LUs. Figure 145 illustrates the configuration.

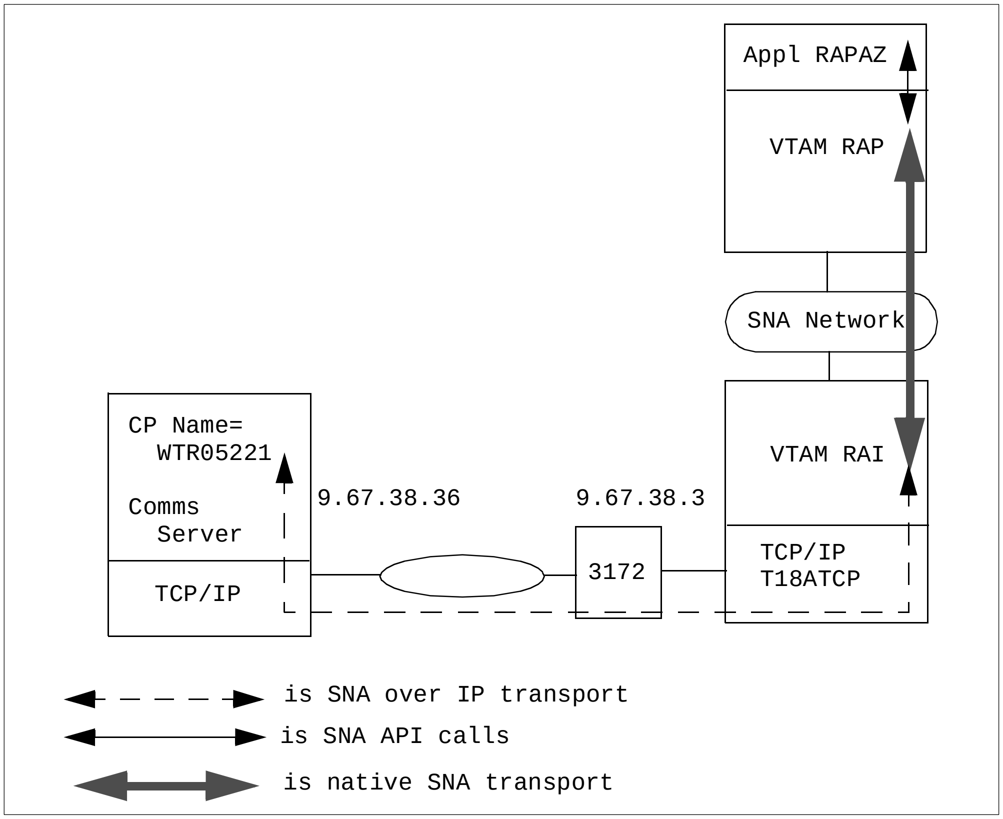


Figure 145. Workstation to OS/390 via Gateway

The OS/390 definitions were the same as previously, with the addition of a CDRSC WTR05221 defined to RAI as being on the TCP/IP link station RAIPSNIP (WTR05221 was added to Figure 141 on page 151). RAP should be able to find WTR05221 by sending a search request to RAI in the usual way.

The definitions required on the Communications Server workstation are:

- SNA over IP domain name suffix is **ibm.com** (Figure 124 on page 139 for CS/2 or Figure 131 on page 144 for CS/NT.
- Node type is **access node** (Figure 124 on page 139 for CS/2 or Figure 131 on page 144 for CS/NT).
- Default routing preference is **Native first** (Figure 124 on page 139 for CS/2 or Figure 132 on page 145 for CS/NT).

- No individual routing preferences for partner LUs.
- No AnyNet connections explicitly defined.

Having applied these changes, we performed APING transactions successfully between RAPAZ and WTR05221.

4.5.6 SNA over IP between Two Workstations

Our next scenario was to connect two Communications Server workstations directly together without any OS/390 node in between. For a change, we tried using DB2/2 transactions instead of the simple APING. However, such transactions are still LU 6.2 session users. Figure 146 illustrates this configuration.

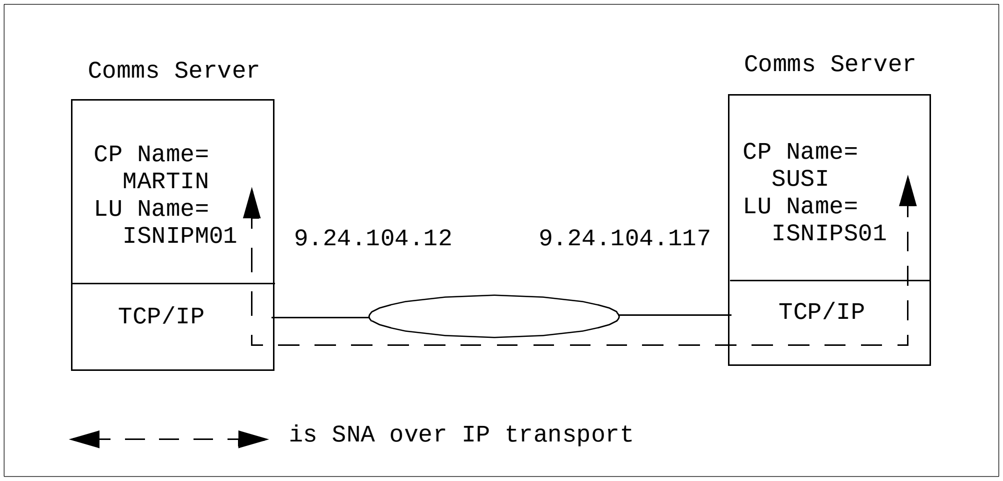


Figure 146. AnyNet SNA over IP between Workstations

Here we had the node USIBMRA.MARTIN set up as a DB2 client and the node called USIBMRA.SUSI was a DB2 server. The Communications Server definitions on both workstations were quite simple in AnyNet terms, but more complex in other areas to accommodate the DB2 transaction programming environment. The following definitions were needed:

- We defined a local LU for use by DB2. It is usual to utilize the control point LU for application traffic on workstations, but sometimes an application requires a separate LU for itself. The additional LUs were USIBMRA.ISNIPM01 on MARTIN and USIBMRA.ISNIPS01 on SUSI. To define an extra local LU, select **SNA Features** and **Local LUs** from the Configuration List panel on CS/2. On CS/NT, select **CPI-C and APPC** then **Local LU 6.2 LUs**.
- We defined transaction programs, mode names and CPI-C side information as required by DB2. All these definitions are accessible from the **SNA Features** option in CS/2 and from **CPI-C and APPC** in CS/NT.
- For the AnyNet domain name suffix (Figure 124 on page 139 in CS/2 or Figure 131 on page 144 for CS/NT) we specified **anynet.ibm.com**.
- For the default routing preferences (Figure 124 on page 139 in CS/2 or Figure 132 on page 145 in CS/NT) we left the default of **Native first**.

- We overrode this value for the individual partner LUs (as in Figure 128 on page 142 for CS/2 or Figure 132 on page 145 for CS/NT). We specified **Non-native only** for ISNIPM01 in SUSI and also for ISNIPS01 in MARTIN.
- Finally, we updated the domain name server (this time we used DNS instead of local lookup) for our IP domain. We had isnips01.usibmra.anynet.ibm.com resolved to 9.24.104.117 and isnipm01.usibmra.anynet.ibm.com resolved to 9.24.104.12.

Using these configurations we were able to issue DB2 transactions on MARTIN and have them executed on SUSI.

To check AnyNet's use of IP sockets, we issued the netstat command on one of the workstations. Note that the format of both the command and the output differs between TCP/IP implementations. Figure 147 is from TCP/IP for OS/2 Version 4 Release 1, where `netstat -s` is the appropriate form. On Windows NT similar information can be obtained using `netstat -a`.

```
[C:\]netstat -s
--------------------------------------------------------------------------
                              AF_INET Address Family:
                              Total Number of sockets 13

  SOCK   TYPE       FOREIGN          LOCAL         FOREIGN         STATE
                     PORT             PORT            HOST
======  =====      ==========      ==========      ==========    ========
     1 STREAM               0      telnet..23         0.0.0.0  LISTEN
     3 STREAM               0         ftp..21         0.0.0.0  LISTEN
     4  DGRAM               0       snmp..161         0.0.0.0  UDP
     5 STREAM               0           49152         0.0.0.0  LISTEN
     6 STREAM           49152           49153    9.24.104.117  ESTABLISH
     7 STREAM           49153           49152    9.24.104.117  ESTABLISH
     8  DGRAM               0               0         0.0.0.0  UDP
     9 STREAM               0       mptn..397         0.0.0.0  LISTEN
    11  DGRAM               0           13991         0.0.0.0  UDP
    12  DGRAM               0       mptn..397         0.0.0.0  UDP
    13 STREAM            1030       mptn..397     9.24.104.12  ESTABLISH
    14 STREAM            1031       mptn..397     9.24.104.12  ESTABLISH
  2048  DGRAM               0               0         0.0.0.0  UDP
--------------------------------------------------------------------------
                                AF_OS2 Address Family:
                              Total Number of sockets 0
```

Figure 147. Sockets Display from OS/2

The `netstat` command shows the status of the TCP/IP sockets in use. Note that AnyNet SNA over IP can use several; in this case it uses number 12 for the UDP datagram traffic, number 9 for an open TCP port (awaiting an AnyNet connection) and numbers 13-14 for TCP connections to a partner node. Before an AnyNet connection is established you will see only the UDP socket and the listening TCP socket.

4.5.7 SNA over IP Using Dependent LU Requester

For our final test we used dependent LU sessions running over an AnyNet SNA over IP connection. Figure 148 shows the network configuration for this scenario.

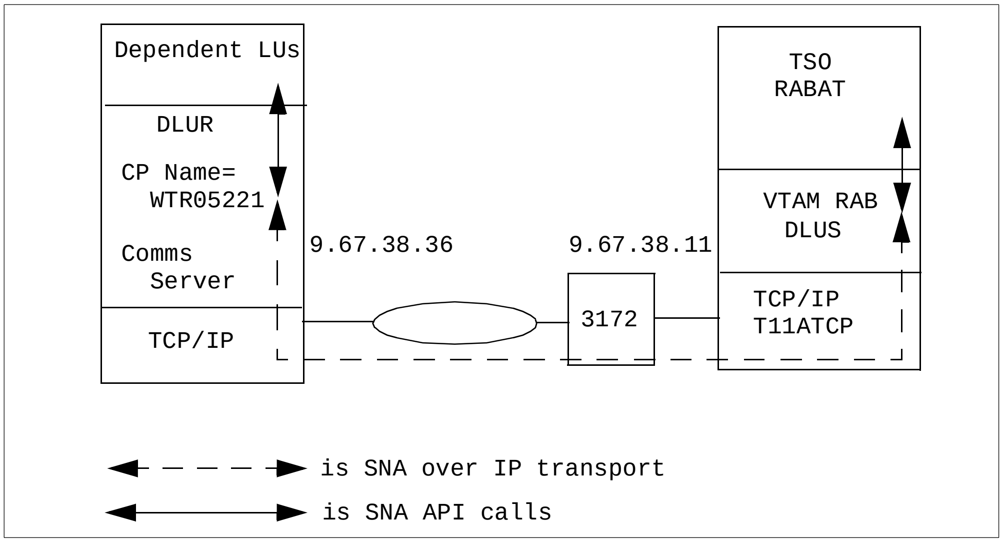


Figure 148. Dependent LU Sessions with SNA over IP

Here we have a dependent LU requester function defined on the workstation WTR05221, which will use VTAM RAB as its DLU server. We have four dependent LUs defined on the workstation which must be able to log on to an application in the SNA network. To demonstrate dependent LU sessions over an IP network we used the TSO application RABAT running on RAB.

There are two logical steps to this setup: configuration of the DLUR/S pipe (a pair of LU 6.2 sessions) and configuration of the dependent resources that will use that pipe. Exactly what you have to define to VTAM depends on which node(s) will initiate the DLUR/S pipe and the SSCP sessions for the dependent resources. In every case, however, VTAM needs to be an APPN network node to allow it to act as a dependent LU server.

The definitions required for this configuration are as follows:

- On the OS/390 host:

  If the workstation initiates the DLUR/S pipe, VTAM requires no definitions for this part of the setup. VTAM receives a BIND across the TCP/IP logical link whose origin LU is WTR05221 and whose mode is CPSVRMGR. This BIND is for WTR05221’s contention winner LU 6.2 session. The mode name tells VTAM that this is a DLUR/S request, and the incoming CP name tells VTAM that an LU called USIBMRA.WTR05221 is reachable via the link station RABPSNIP (which is where the BIND came from). Therefore VTAM can send out a BIND over the IP network to establish its own contention winner session for the DLUR/S pipe.

  If VTAM is to initiate the DLUR/S connection, it needs to be told two things: the name of the DLUR node and how to find it. Normally just the name of the DLUR node is sufficient, but in the AnyNet SNA over IP case the DLUR/S connection is LEN and therefore VTAM must be told how to locate WTR05221. Thus you need two definitions: a CDRSC definition and a switched link station definition. The former is accomplished by adding WTR05221 to a CDRSC major node such as that in Figure 141 on page 151, associating the CP LU name with the TCP/IP link station RABPSNIP. The switched link definition that tells VTAM the name of the DLUR is coded in a switched major node as in Figure 149.

```
RABSSNIP VBUILD TYPE=SWNET
ISNIPJ01 PU    ADDR=01,                                                X
               IDBLK=05D,                                              X
               IDNUM=05221,                                            X
               DISCNT=NO,                                              X
               ISTATUS=ACTIVE,                                         X
               MAXDATA=1033,                                           X
               PACING=0,                                               X
               PUTYPE=2,                                               X
               DLOGMOD=D4C32XX3,                                       X
               MODETAB=ISTINCLM,                                       X
               USSTAB=US327X,                                          X
               VPACING=0
       PATH  PID=1,                                                    X
             DLURNAME=WTR05221,                                       X
             DLCADDR=(1,C,INTPU),                     1               X
             DLCADDR=(2,X,05D05221)
ISNIPJL1 LU    LOCADDR=2
ISNIPJL2 LU    LOCADDR=3
ISNIPJL3 LU    LOCADDR=4
ISNIPJL4 LU    LOCADDR=5
```

Figure 149. Switched Major Node for DLUR

In this definition the DLUR is defined by the DLURNAME keyword in the PATH statement (1). The switched major node, however, defines more than just the DLUR node; it defines the dependent resources that are served by this DLUR node. Without the PATH statement it serves perfectly well as a set of definitions for a DLUR node that initiates the DLUR/S process.

If the workstation is to initiate the activation of the dependent resources (as opposed to activation of the DLUR/S sessions), then VTAM needs to be able to associate PU and LU definitions with a node ID. Therefore, you must do one of three things:

1. Define the dependent resources in a switched major node, as above. The keywords IDBLK and IDNUM on the PU statement identify the dependent internal PU in the workstation. This is *distinct* from the link station used by VTAM to contact the workstation itself, which is of course RABPSNIP and may or may not have any node ID associated with it. This was the course we adopted because our network was very simple.
2. Allow the configuration services exit ISTEXCCS to define the dependent PU and LUs dynamically. The exit uses the incoming node ID to tell VTAM how to define these resources.
3. Allow the dynamic dependent LU definition exit ISTEXCSD to define the resources. This one requires corresponding support on the workstation (CS/2 and CS/NT both have this support). This exit can define the LUs to VTAM based on information supplied by the workstation node, but the PU has to be defined by other means. These means can be a definition as above, the ISTEXCCS exit, or by VTAM itself without any assistance.

Note that the dependent PU and LUs can identify themselves by CP name instead of by node ID. This works properly only if the dependent resources are on a separate node attached downstream of the DLUR node.

If VTAM is to initiate the activation of the dependent PU and LUs, a switched major node is always required, and the PATH statement as in Figure 149 on page 160 is required within it. Note, however, that in this case the PU is identified *not* by the IDBLK and IDNUM keywords on the PU statement but by the DLCADDR keywords on the PATH statement.

In summary, the only strictly AnyNet-related consideration here is to make sure that the DLUR is reachable via the TCP/IP link station. Even that is not normally required because the workstation usually initiates the DLUR/S process.

- On the workstation:

  Once again, the only AnyNet-specific requirements are to ensure that the DLUS can be located correctly. From the previous setup, WTR05221 will try to locate USIBMRA.RAB using native (SNA) protocols first (and fail), followed by non-native (TCP/IP) protocols. The domain name suffix is already set to ibm.com so WTR05221 will look for rab.usibmra.ibm.com to set up its DLUR/S sessions.

  The dependent LUs are defined (both on CS/2 and CS/NT) just as are native dependent LUs, except that they are associated with a DLUS host name instead of with a real link station to a VTAM or an NCP. No additional configuration is required if the DLUS happens to be reached via AnyNet instead of via native APPN.

- In the IP network:

  If TCP/IP is to succeed we must ensure that the DNS (or the local HOSTS file) can map rab.usibmra.ibm.com to 9.67.38.11 and wtr05221.usibmra.ibm.com to 9.67.38.36. This must be done in both directions because each node will establish a TCP connection to the other.

No AnyNet definitions or address mappings are required for the dependent LUs or the OS/390 applications. A session request from a dependent LU travels over the DLUR/S pipe (on an LU 6.2 session) to VTAM, which searches the SNA network for the target application (TSO in our case). TSO sends a BIND to VTAM which forwards it across the AnyNet connection to the DLUR, where it knows the dependent LU resides. In the other direction, a session request from an application is handled by the DLUS VTAM and forwarded to the DLUR. Unlike the case of native APPN, all flows relating to the dependent LUs are across the one LEN link, and both DLUR and DLUS understand this. Once they know each other’s location they send all DLUR/S traffic to each other.

We configured the DLUR/S definitions on the workstation and on the host, and added the minimal extra requirements for AnyNet. We were then able to log on to TSO from dependent LUs using the 3270 emulator on the workstations. We took several displays of the dependent resources and their sessions, but there was hardly anything in them to give away the fact that they were using an IP network as their transport. As an example, please see Figure 150 on page 162 which is a display of the DLUR node from RAB.

```
D NET,ID=WTR05221,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST075I NAME = USIBMRA.WTR05221, TYPE = CDRSC 373
IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
IST977I MDLTAB=***NA*** ASLTAB=***NA***
IST1333I ADJLIST = ***NA***
IST861I MODETAB=***NA*** USSTAB=***NA*** LOGTAB=***NA***
IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT NONE
IST231I CDRSC MAJOR NODE = RABRSNIP
IST1184I CPNAME = USIBMRA.RAB - NETSRVR = ***NA***
IST1044I ALSLIST = RABPSNIP
IST082I DEVTYPE = INDEPENDENT LU / CDRSC
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000002, SESSION REQUESTS = 0000000000
IST206I SESSIONS:
IST1081I ADJACENT LINK STATION = RABPSNIP                   1
IST634I NAME    STATUS        SID          SEND RECV VR TP NETID
IST635I RAB      ACTIV/DL-S D96FD635E0901E31 000B 0000       USIBMRA
IST635I RAB      ACTIV/DL-P F7EFD164B8FDB804 0000 000E       USIBMRA
IST1355I PHYSICAL UNITS SUPPORTED BY DLUR USIBMRA.WTR05221         2
IST089I ISNIPJ01 TYPE = PHYSICAL UNIT    , ACTIV               3
IST314I END
```

Figure 150. Display of DLUR on AnyNet Link

Note especially:

1. The DLUR LU (control point) has sessions flowing over the TCP/IP link station RABPSNIP.
2. The sessions in question are the two parallel LU 6.2 sessions comprising the DLUR/S pipe.
3. This node is acting as a DLUR on behalf of the dependent PU ISNIPJ01, which we defined in Figure 149 on page 160.

A display of the TCP/IP link station (Figure 151) shows that both the CP, WTR05221, and the dependent LU ISNIPJL3 (defined in Figure 149 on page 160) are in session and using this link station.

```
D NET,ID=RABPSNIP,SCOPE=ALL
IST097I DISPLAY ACCEPTED
IST075I NAME = RABPSNIP, TYPE = PU_T2.1
IST486I STATUS= ACTIV--L--, DESIRED STATE= ACTIV
IST1043I CP NAME = ***NA***, CP NETID = USIBMRA, DYNAMIC LU = YES
IST081I LINE NAME = RABLSNIP, LINE GROUP = RABGSNIP, MAJNOD = RABBSNIP
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST355I LOGICAL UNITS:
IST080I WTR05221 ACT/S      ISNIPJL3 ACT/S
IST314I END
```

Figure 151. TCP/IP Logical Link Station with Dependent LU Session

Chapter 5. Data Link Switching

Data link switching is different in essence from Enterprise Extender and AnyNet SNA over IP. Data link switching maps *SNA transport* to IP transport, while the other technologies can effectively map *an SNA API* to IP transport. In fact, DLSw simply maps LLC2 transport to IP transport for particular SAPs, and the presence of SNA is merely incidental. Data link switching is something that happens in IP routers with an SNA connection outboard, whereas Enterprise Extender and AnyNet can be implemented at the endpoints of an SNA session. For this reason, a DLSw solution is never end-to-end and cannot provide the same high level of service as an end-to-end connection using the same technology throughout. In particular, each DLSw router constitutes a single point of failure whose loss can disrupt SNA sessions.

5.1 Data Link Switching Description

Data link switching is a widely-accepted standard and is described in RFCs 1434 and 1795 (Version 1) and RFC 2166 (Version 2). It is designed for both local and remote operation, and for carrying both NetBIOS and SNA traffic over an IP backbone. NetBIOS does not concern us here, so we concentrate on SNA. Moreover, local DLSw is essentially SDLC-to-LLC2 translation and does not use IP transport. Therefore, we will not consider local DLSw further and the rest of this chapter assumes remote DLSw for SNA over IP.

DLSw is supported by the current IBM router family which comprises the 2216, 2210, 2212, Network Utility, and 3746 MultiAccess Enclosure. We refer to these generically as *221X*.

An SNA node attached to a 221X uses remote DLSw to communicate with another SNA node attached to another 221X. SNA stations may be link-attached or LAN-attached to the 221Xs. The 221Xs, referred to as *DLSw partners*, must both be configured for DLSw. The partners communicate with each other across an IP network.

The essence of DLSw is to give a pair of IP routers at the edges of an IP network the appearance of a simple bridge. Thus SNA nodes connected to LANs at either side of the network can communicate with each other as if they were on the same LAN. Even if the SNA stations are link-attached, they are assigned an internal MAC address and this is used in all communications, being mapped to the correct DLC address at the appropriate points.

Remote DLSw supports the following SNA protocols:

- Peripheral subarea (type 2.0 to type 4/5 nodes)
- APPN or LEN (type 2.1 to type 2.1 nodes, including VTAM/NCP acting as type 2.1)
- INN subarea (type 4/5 to type 4/5 nodes)

DLSw can communicate with SNA stations across the following DLCs:

- IEEE 802.2 over token-ring, Ethernet or FDDI LANs, frame relay (RFC 1490/2427 bridged format), or ATM (LAN emulation or RFC 1483 bridged format)
- SDLC as primary, secondary or negotiable

- QLLC (X.25 PVC or SVC)
- ESCON or parallel channel using LSA

The LLC connections are terminated at the 221Xs. *Spoofing* is used to eliminate the transport of LLC acknowledgments across the WAN. This improves WAN utilization and prevents link-level timeouts due to delays in the WAN. Each SNA station receives acknowledgments from its locally-attached 221X, and not from its true SNA partner.

Since DLSw emulates a bridged LAN, LAN-attached SNA partner stations use the normal source routing information, including the hop counts, in the LLC2 frames to exchange data across a token-ring. However, the DLSw partners also terminate the hop counts locally. Thus the source station may be up to seven hops from the first 221X in the path and the receiving station may be up to seven hops from the last 221X in the path. On a native token-ring connection there may be a total of seven hops between the SNA partners.

For transport between the DLSw partners, the SNA frames are encapsulated in TCP packets and sent over a TCP connection. The route between two partners can contain IP routers that are not DLSw-capable.

5.1.1 How Data Link Switching Works

Like source route bridging, DLSw uses information in the data link layer header to determine the route for a packet. The DLSw function examines the destination MAC address and the destination service access point (DSAP) in the LLC2 header. SNA frames (on a token-ring or an Ethernet) always contain DSAPs that are multiples of 4 in the range X'04' to X'EC'. When you configure a 221X for DLSw, you must indicate which of the SNA SAPs you want DLSw to process.

Figure 152 on page 165 shows how SNA stations communicate with each other once a DLSw connection has been established. Device D1 on Ring 1 is communicating with station D2 on Ring 2. Both 221Xs are configured for DLSw, source route bridging, and IP. DLSw provides an LLC type 2 connection between D1 and the 221X on Ring 1 and an LLC type 2 connection between D2 and the 221X on Ring 2. The SNA frames are encapsulated in IP datagrams and sent across the IP network using a TCP connection. The path through the IP network may include intermediate routers, which do not have to support DLSw.

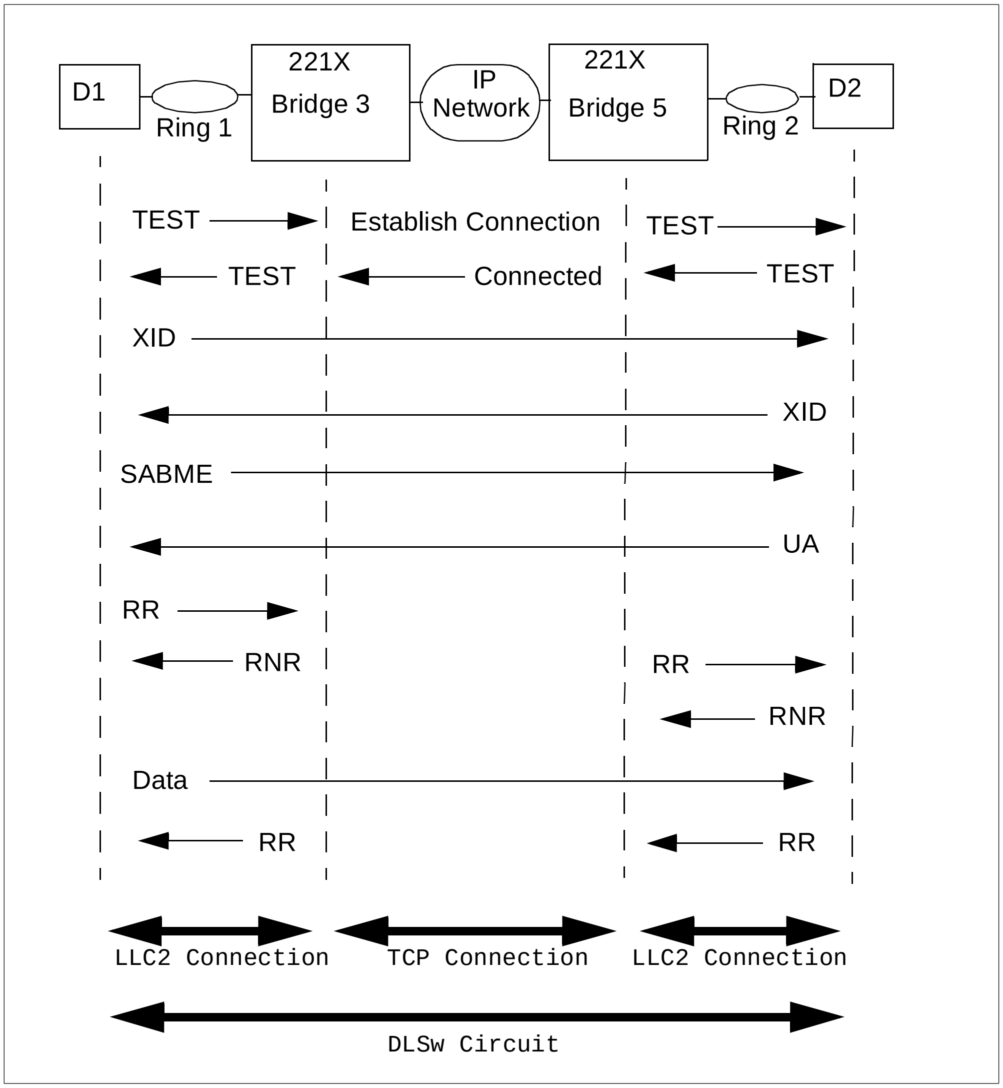


Figure 152. LLC and TCP Connections Using DLSw

The TEST frames are used by the 221Xs to establish the logical DLSw connection on the TCP connection between the routers. Next, the contents of the XID and SABME exchange are passed through the TCP connection because they need to be exchanged between the SNA partners. However, once the two SNA stations think that they have established an asynchronous balanced mode connection with SABME, all acknowledgments (RR and RNR) are terminated at the local router (spoofing). Only data frames are sent over the TCP connection. The complete end-to-end connection is referred to as a *DLSw circuit*. The routers perform flow control on a circuit basis, so that a misbehaving connection can be stopped to prevent it from flooding the WAN without affecting other DLSw circuits using the same port or the same TCP connection.

DLSw works by creating a *virtual ring*, which is a token-ring segment to which all remote DLSw partner devices appear to be connected. Figure 153 on page 166 illustrates the concept of a virtual ring in remote DLSw.

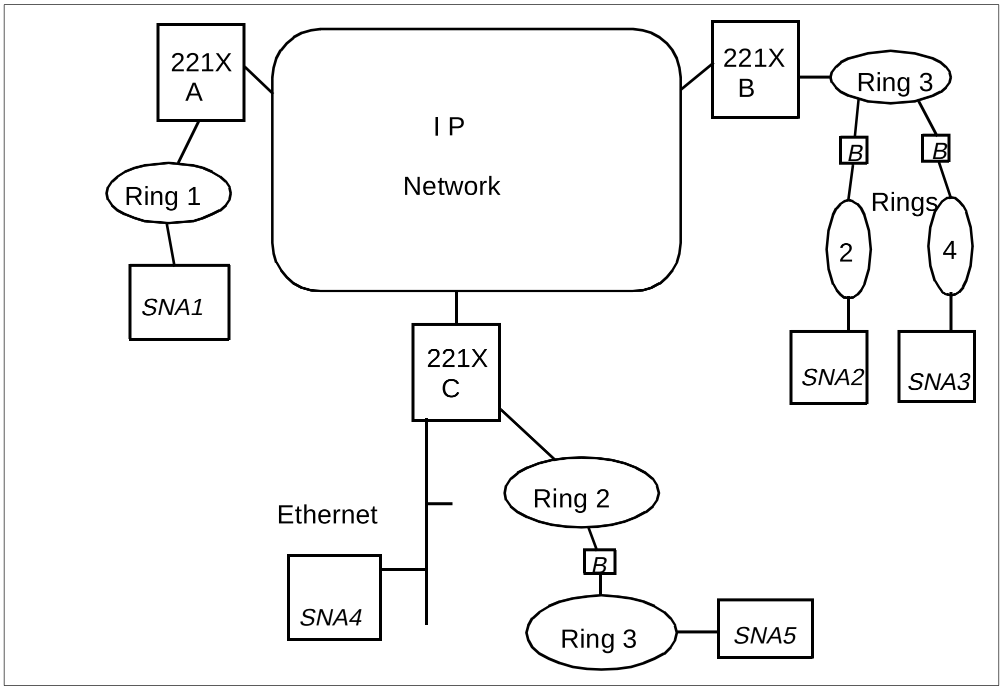


Figure 153. Physical DLSw Configuration

In this example, SNA1 to SNA5 are SNA stations connected to each other via various token-ring, Ethernet and DLSw links as shown. The small boxes labeled *B* are source route bridges, and the three DLSw routers are labeled A, B and C. The devices SNA2 and SNA3 share the same DLSw router, and their view of the whole network is the same. It is illustrated in Figure 154.

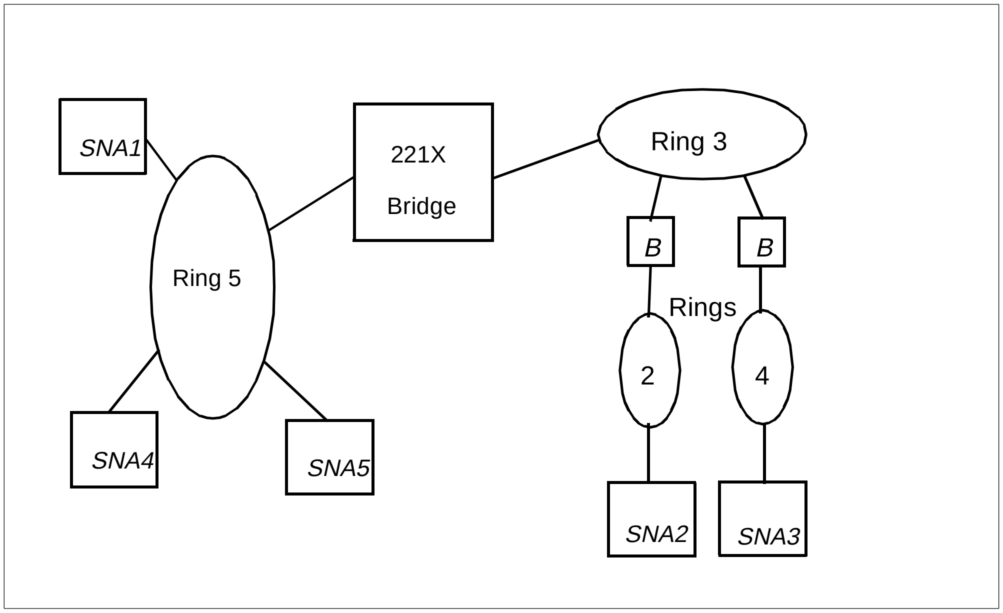


Figure 154. Virtual Ring in DLSw Configuration

The DLSw routers between them simulate a virtual token-ring numbered 5. This number must be configured in all the 221Xs, and must not be the same as any of the attached real rings to which it is apparently bridged. To the devices SNA2 and SNA3, the 221X labeled B appears to be a source-route bridge connecting them to Ring 5; and all the other SNA stations including the Ethernet one appear to be on Ring 5 itself. This method has the advantage that ring numbering schemes apply only to each group of bridged (*really* bridged) rings rather than to the network as a whole.

5.1.2 Circuit Establishment

As described above, a DLSw circuit is a logical connection between two end devices. A remote DLSw circuit consists of three portions:

- An LLC connection between a source device and a 221X configured for DLSw
- A TCP connection between that 221X and a partner 221X configured for DLSw
- An LLC connection between the partner 221X and the destination device

When you configure a 221X for DLSw, you can enter a list of its partner 221Xs, or you can define a common group which all the partners are to join. Connections to authorized active partners are established when the 221X is activated, or when a DLSw circuit is required, depending on how you have set up the router. These connections are maintained until the 221X leaves the network, or until the last circuit has been terminated. DLSw partners use port 2065 for receiving and port 2067 for sending all DLSw information.

The LLC connections, and their logical continuations across the TCP connections, are established when two end devices set up communication via 221Xs configured for DLSw. DLSw partners then exchange information needed to create and maintain their DLSw tables.

SNA stations on a LAN exchange TEST and XID frames when they first contact each other. These serve to determine the existence of the partner, the route to it, and the capabilities of the partner. During the XID flows the SAPs are exchanged. After the XID exchange the LLC2 connection is established by the sending of SABME from one partner to the other.

DLSw uses the TEST frames to determine the correct DLSw partner to choose and to establish the circuit. After the LLC2 connection has been set up (after the SABME response), DLSw checks the MAC and SAP addresses of subsequent frames and matches them to those received on the TEST and XID frames to determine on which circuit they should be sent. This works well until one of the partners decides to use HPR.

HPR normally uses an alternative SAP address (usually C8) for NLPs which do not require link-level error recovery. The HPR SAP address is passed in a CV 61 in the XID exchange, and not in the normal place in the IEEE 802.2 frame. Moreover, there may not be an LLC2 connection at all between the partners for DLSw to match the SAP to; the NLPs are sent as unnumbered information (UI) frames. Therefore, DLSw has no means of identifying such NLPs and either discards them or broadcasts them.

HPR is not supported on DLSw connections by the IBM router family. However, some SNA devices can utilize the same SAP for HPR traffic as for base APPN, at the cost of requiring link-level error recovery. Some non-IBM routers with vendor-specific extensions to DLSw can support HPR over DLSw circuits for such devices.

This is what happens when two SNA stations (let us call them A and B) contact each other across a DLSw connection. If A and B are connected to 221A and 221B respectively as DLSw partners:

1. Station A sends a TEST frame to find the destination device B. The TEST frame contains the addresses of both A (the source) and B (the destination).
2. 221A sees the TEST frame sent by A. If its DLSw routing table does not contain information about how to reach B, 221A sends a CANUREACH frame (a type of DLSw packet) to all its authorized partners (including 221B). The CANUREACH frame contains the information in the TEST frame and the source address of 221A.
3. When the DLSw partners receive the CANUREACH frame, they broadcast the TEST frame to all devices on their directly attached LANs.
4. B receives the TEST frame from 221B, and sends a response back to 221B. It may receive multiple copies of this frame if it happens to be connected to more than one DLSw router, in which case it will send a TEST response to each one.
5. When 221B receives the TEST response frame from B, it sends an ICANREACH frame (another type of DLSw-architected packet) to 221A. 221B also records in its DLSw table that the source address in the TEST frame (that of A) can be reached through 221A.
6. 221A receives the ICANREACH frame in response to its CANUREACH frame, and updates its DLSw table to identify 221B as being a way to reach the MAC address of B. It may have received more than one such response, in which case it registers all the responders in its DLSw table as being capable of reaching B. However, the first response received identifies the preferred route and will be used in the future unless something goes wrong.

The next time 221A receives a TEST frame that contains B's destination address, it uses the information in its DLSw table to send the CANUREACH frame directly to 221B. This eliminates the broadcast traffic required to send a CANUREACH to all of 221A's partners. Then, 221B broadcasts the TEST frame to B. Broadcasting this frame ensures that the LLC connection is established over the fastest path. Such broadcasts are contained within the LAN and do not generate WAN traffic.

If 221A receives multiple TEST frames containing a destination address that is not in its DLSw table, it will send only one CANUREACH frame for that destination address to all of its partners. 221A will wait two minutes before timing out a CANUREACH request. When the first ICANREACH response returns, 221A directs all the received frames to the router identified in the response frame.

5.2 Data Link Switching in the 221X Router Family

In this section we give a description of how the DLSw environment is configured in the 221X routers. They all contain basically the same code, so that even if the method of configuration differs (which it does in some cases) the principles are valid for all these routers. The routers all support remote DLSw over IP connections.

The configuration examples we give are based on the command-line interface available via the service port on the routers. The 3746 MAE, in particular, does not have such an interface and configuration is perfomed using the service processor console. The stand-alone routers (2216, 2212, 2210, Network Utility) can also be configured remotely using Telnet, or by using a GUI on a PC and downloading the result using SNMP.

5.2.1 DLSw Configuration

The DLSw function itself is defined using the `protocol dls` command which gives you a comprehensive range of options. However, certain prerequisites have to be met if you are going to configure DLSw successfully; not all of these are obvious:

- You have to define the physical machine configuration, including the SNA ports and the WAN ports over which DLSw will take place.
- You need to define link stations for both upstream and downstream SDLC connections. Downstream you define the stations that the router will poll, and upstream you define the stations that the router will pretend to be.
- You need to enable IP (no, this is not automatic) on the ports that will use it.
- Because DLSw emulates a bridge, you need to enable bridging on all LAN ports that will be used by stations communicating over DLSw circuits. By default the 221Xs enable transparent bridging, so you will have to turn this off and enable source-route bridging for token-ring ports. You will also have to enable DLSw on the LAN bridge ports. The 221Xs treats LAN bridge ports as distinct from the LAN interfaces associated with them.

  If you are planning to do real bridging as well as DLSw on the same port, you must ensure that frames destined for DLSw are not bridged. You do this by setting the protocol filters on all the non-DLSw bridge ports to reject frames destined for SAP 4 (and any other SAPs used by DLSw).
- If you want the DLSw partners to find each other automatically instead of having to be predefined, you can do this. If not all partner routers support automatic recognition, you can define some and leave others to contact each other dynamically. To use this function, you have to enable OSPF and then OSPF multicast on each router in a dynamic definition group.

Having set up the router to your requirements, you enter `protocol dls` and now have the opportunity to:

- Enable data link switching
- Set the source-route bridging virtual LAN segment number
- Define the group of dynamically recognized DLSw partners to which this router will belong, if appropriate
- Define DLSw partner IP addresses, if appropriate
- Define which SAPs DLSw is to recognize on received LAN frames
- Map the SDLC link stations (if there are any) to LAN MAC addresses and (if necessary) node IDs. SDLC-attached stations are bridged to the virtual token-ring in the same way as stations on a real LAN. If the SDLC-attached station is a type 2 node that does not supply its node ID on XID exchange, it must be given a node ID for use with the LLC2 protocol.

5.2.2 DLSw Operator Commands

When the 221X router is running, a wide variety of commands is available to display the status of its functions. To enter operator console mode, type `talk 5` at the basic prompt; then to access the DLSw functions type `protocol dls`. Now you will have the Data Link Switching Console available and you can enter commands such as these:

- `list tcp sessions` displays details of the TCP connections to partner DLSw routers.
- `list dls sessions` displays details of SNA connections (the 221X calls them sessions, but they are sessions only in the DLSw sense, not in the SNA sense). Source and destination MAC addresses (or SDLC station addresses) and partner DLSw router addresses are shown. Additional statistics on each SNA connection can be found by issuing the `list dls sessions detail` form of the command.
- `list sdlc sessions` shows details of the SDLC stations defined, the status of their connections, and their equivalent MAC addresses.

5.3 Working Scenarios

In our examples of DLSw configurations we have used 2210 routers throughout. As we stated above, the code in the 221X router family is the same in each box; therefore, the information you have to enter is the same even though it may be entered in different ways. We have also endeavored to produce a wide variety of connections in the examples: PPP or frame relay between routers, and Ethernet, token-ring and SDLC connections to the SNA nodes.

Note that the exact sequence of the configuration questions differs from release to release of the 221X code. The examples shown below may not correspond to your own environment.

5.3.1 Remote SDLC and Token-Ring over PPP to SDLC Host

Figure 155 on page 171 shows the configuration in this example.

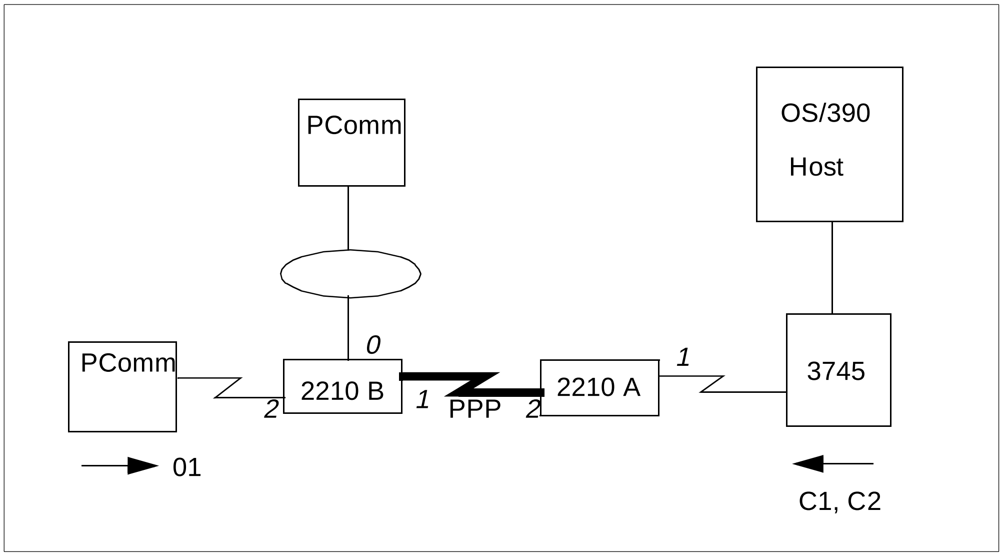


Figure 155. DLSw Connection via PPP to SDLC Host

In this scenario, there were two 2210s connected across the wide area network by a PPP link, and three SNA stations attached to the data link switching configuration as shown:

- The local or central site router, 2210A, was SDLC attached via serial port 1 to a 3745 running ACF/NCP V7R4. The PPP link to the partner 2210 was on serial port 2.
- The remote router, 2210B, had the PPP link on serial port 1 and an SDLC connection to a downstream 3270 emulation workstation on serial port 2. It also had a token-ring interface (port 0) through which a second 3270 emulation station was connected.
- The token-ring attached workstation had to be configured with a MAC address to represent the 3745. However, this MAC address was internal to the DLSw network, since the 3745 itself understood only SDLC. We had to map this MAC address to an SDLC station address at the 3745 end of the DLSw connection.
- The SDLC-attached workstation had an SDLC address of 01. This was mapped to an internal MAC address inside the DLSw network, which was in turn mapped to another MAC address and then an SDLC station address for the 3745's benefit.
- The 3745 itself was given SDLC addresses of C1 and C2 to poll the workstations. These addresses were defined only within the DLSw network, and mapped to the appropriate MAC addresses of the workstations (one real, one re-mapped to the SDLC address 01).

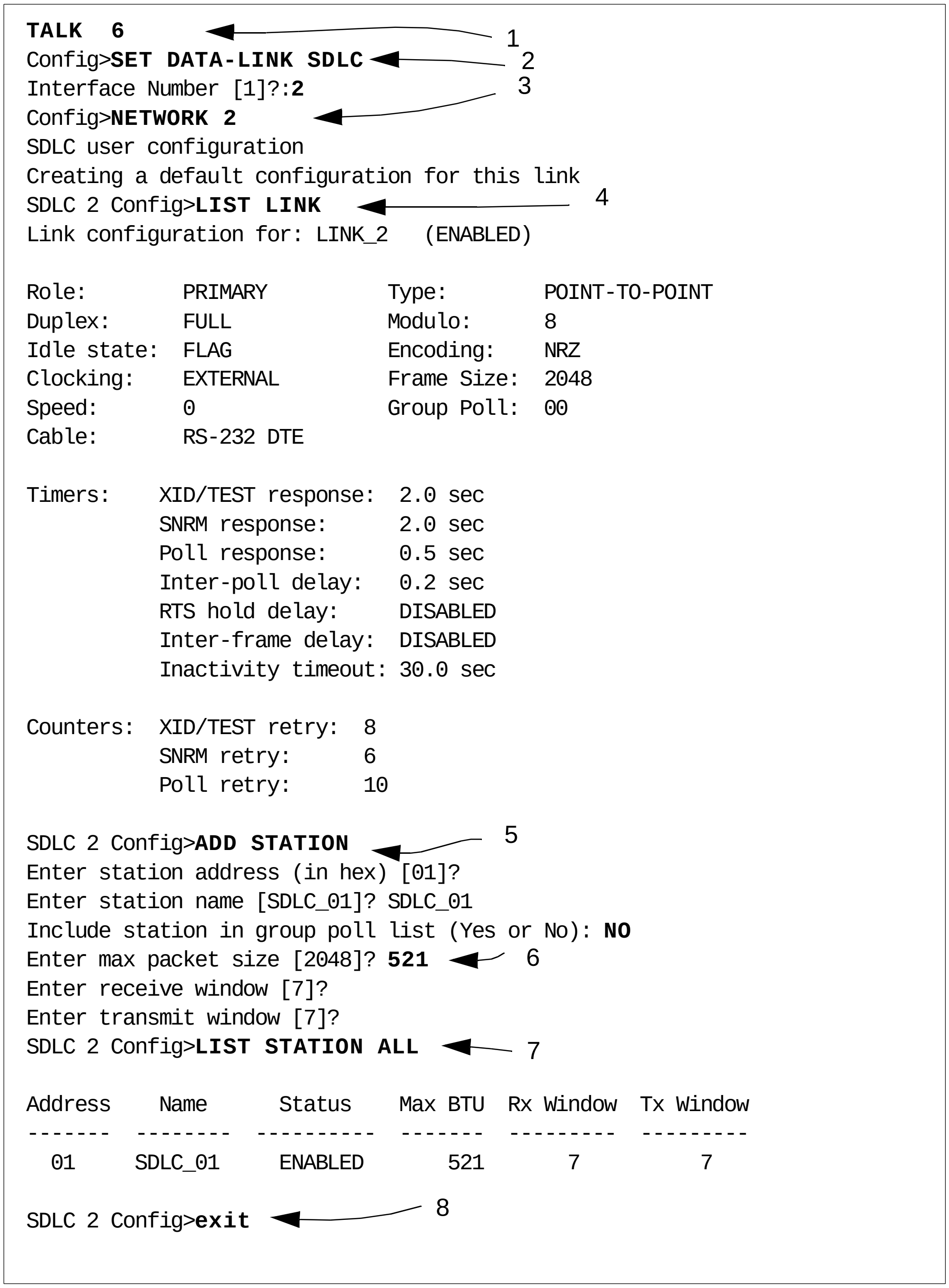
Both remote workstations were configured as PU Type 2.0 devices in this configuration. Neither the NCP nor the SDLC-attached station could send or receive XIDs. The token-ring attached station expected to exchange XID format 0 with what it thought was a 3745.

Our objective here was to establish 3270 sessions between the dependent LUs on each workstation and the host applications such as TSO and NetView. The potential benefits of such a configuration are that you can migrate from a multidropped SDLC network to a remote LAN configuration with minimal change to your host environment. The alternative would be to install remote LAN gateways, but such gateways do not usually have the multiprotocol capabilities of the 221X routers.

We have not shown much of the basic (non-DLSw-related) configuration for the 2210 routers. The following parameters were configured on 2210 B during the initial setup:

- Token-ring speed = 4 Mbps
- Token-ring connector type = STP
- Encapsulation for WAN 1 = PPP
- Cable type for WAN 1 = RS232 DCE
- Internal clock speed for WAN 1 = 56000 bps
- IP address of Interface 0 = 8.8.8.1 (subnet mask = 255.255.255.0)
- IP address of Interface 1 = 200.200.200.2 (subnet mask = 255.255.255.0)
- Dynamic routing = enabled
- OSPF = enabled

The remaining configuration parameters are described below. Figure 156 on page 173 shows the setup of the downstream SDLC port (port 2) on 2210 B.



```
TALK  6
Config>SET DATA-LINK SDLC
Interface Number [1]?:2
Config>NETWORK 2
SDLC user configuration
Creating a default configuration for this link
SDLC 2 Config>LIST LINK
Link configuration for: LINK_2   (ENABLED)

Role:         PRIMARY          Type:         POINT-TO-POINT
Duplex:       FULL             Modulo:       8
Idle state:   FLAG             Encoding:     NRZ
Clocking:     EXTERNAL         Frame Size:   2048
Speed:        0                Group Poll:   00
Cable:        RS-232 DTE

Timers:    XID/TEST response:  2.0 sec
           SNRM response:      2.0 sec
           Poll response:      0.5 sec
           Inter-poll delay:   0.2 sec
           RTS hold delay:     DISABLED
           Inter-frame delay:  DISABLED
           Inactivity timeout: 30.0 sec

Counters:  XID/TEST retry:  8
           SNRM retry:      6
           Poll retry:      10

SDLC 2 Config>ADD STATION
Enter station address (in hex) [01]?
Enter station name [SDLC_01]? SDLC_01
Include station in group poll list (Yes or No): NO
Enter max packet size [2048]? 521
Enter receive window [7]?
Enter transmit window [7]?
SDLC 2 Config>LIST STATION ALL

Address    Name      Status    Max BTU  Rx Window  Tx Window
-------  --------  ----------  -------  ---------  ---------
  01     SDLC_01     ENABLED      521       7          7

SDLC 2 Config>exit
```

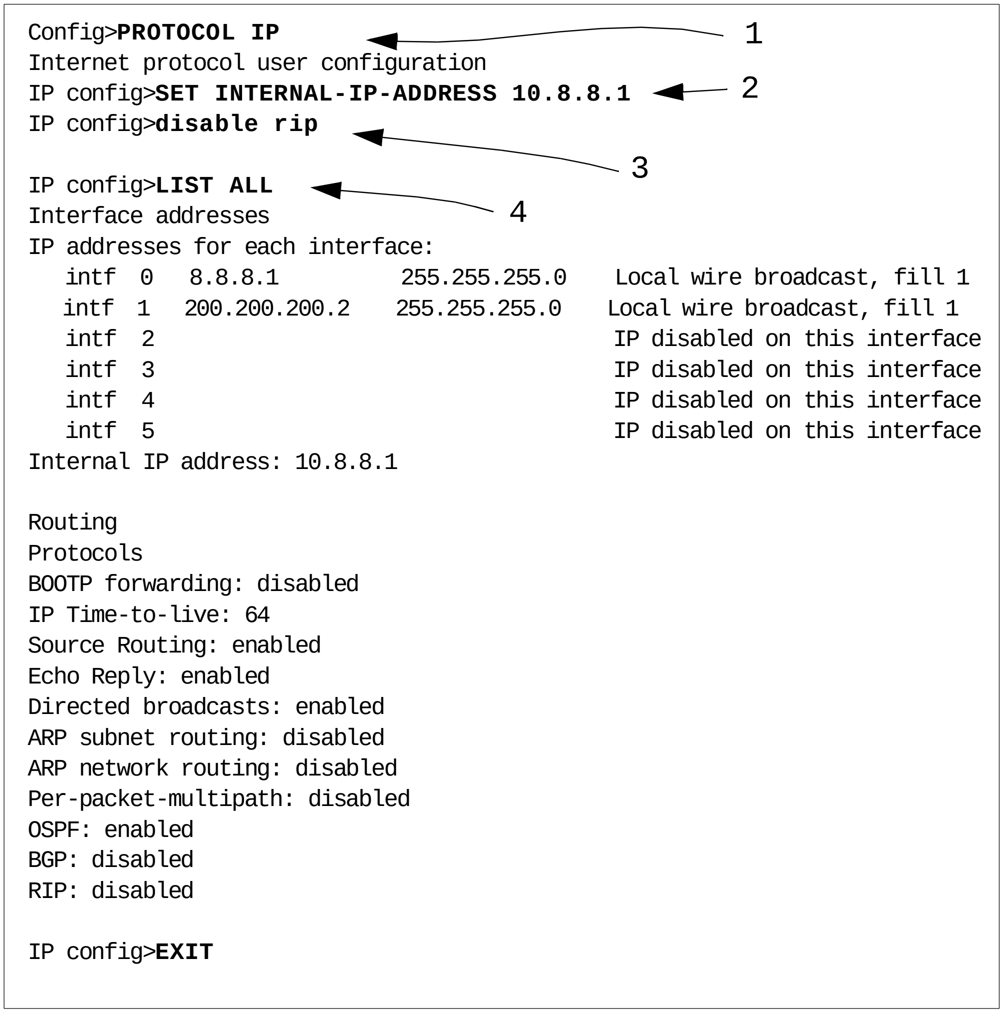
Figure 156. Downstream SDLC Port on 2210 B

In this part of the configuration:

1. We enter `talk 6` to invoke the configuration process.
2. We define port 2 as being an SDLC port (as a serial port, it could support one of a number of protocols).
3. We enter `network 2` to start the configuration process for the SDLC port 2. This defines the port with a default configuration.

4. To verify the configuration we enter `list all`. This link has been defined as point-to-point (the default). A multipoint link has a more complex set of configuration steps.
5. Now we need to add a link station that represents the 3270 emulation workstation. We shall define a station in the DLSw configuration (Figure 160 on page 177), so we do not need to add it here unless the default station parameters are unsatisfactory. The station address is correct so we leave it alone.
6. However, we need to code the maximum basic transmission unit (BTU) size this station will accept, as a type 2.0 station cannot supply this information on XID exchange. This is the equivalent of (and should be the same as) the MAXDATA keyword on the PU statement in the NCP.
7. We then check the configuration of the link station using another `list` command.
8. Finally we exit the SDLC port configuration.

Next, we need to configure IP on the 2210, as shown in Figure 157.



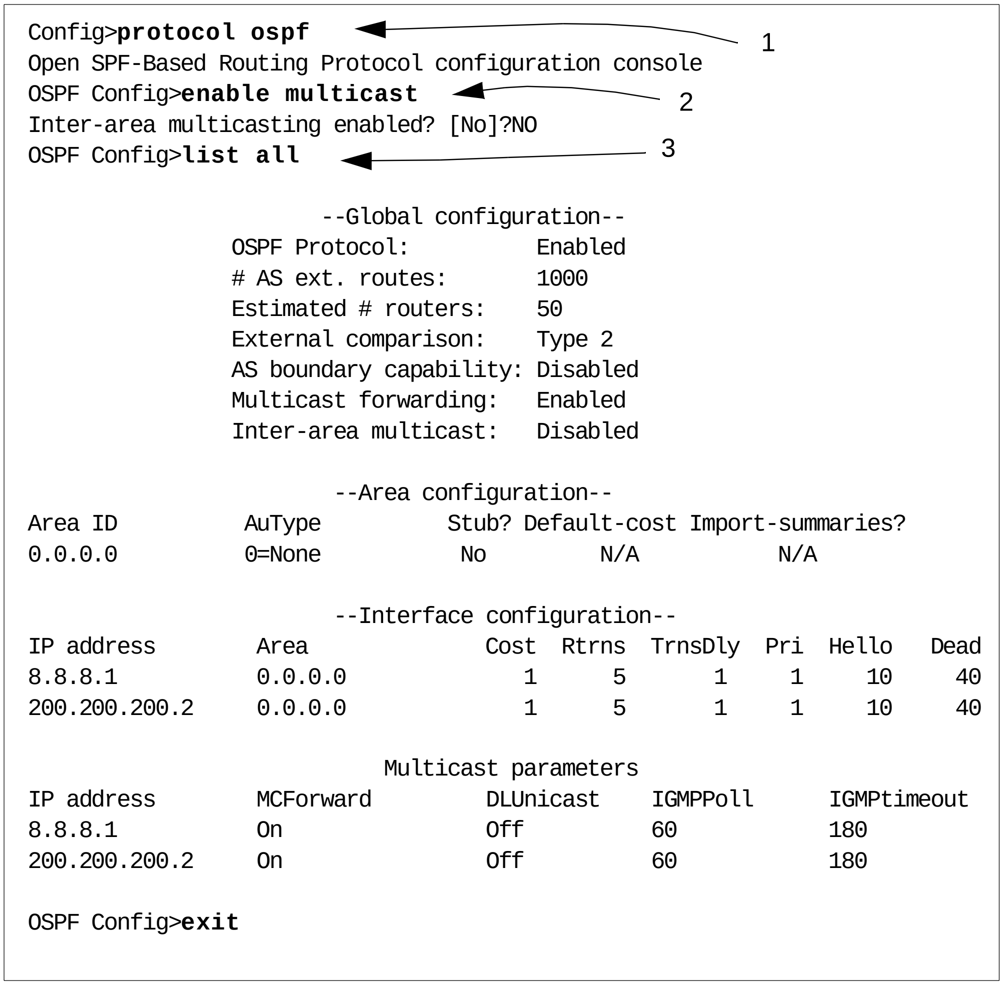
```
Config>PROTOCOL IP
Internet protocol user configuration
IP config>SET INTERNAL-IP-ADDRESS 10.8.8.1
IP config>disable rip

IP config>LIST ALL
Interface addresses
IP addresses for each interface:
   intf  0   8.8.8.1         255.255.255.0    Local wire broadcast, fill 1
   intf  1   200.200.200.2   255.255.255.0    Local wire broadcast, fill 1
   intf  2                                    IP disabled on this interface
   intf  3                                    IP disabled on this interface
   intf  4                                    IP disabled on this interface
   intf  5                                    IP disabled on this interface
Internal IP address: 10.8.8.1

Routing
Protocols
BOOTP forwarding: disabled
IP Time-to-live: 64
Source Routing: enabled
Echo Reply: enabled
Directed broadcasts: enabled
ARP subnet routing: disabled
ARP network routing: disabled
Per-packet-multipath: disabled
OSPF: enabled
BGP: disabled
RIP: disabled

IP config>EXIT
```

Figure 157. IP Configuration for Remote Router

In this configuration step:

1. We invoke `protocol IP` to begin customizing the IP specifications.
2. We enter the internal IP address to be used by the 2210 for data link switching. This address behaves like a VIPA address; it allows the 221X to establish and maintain DLSw TCP connections without depending on a particular interface to be active.
3. We disable RIP because we do not need it; all the routers in our network use OSPF.
4. We issue `list all` to check our IP setup.

Next, we would like the DLSw partners to find each other without predefinition, so we enable OSPF multicast as in Figure 158.

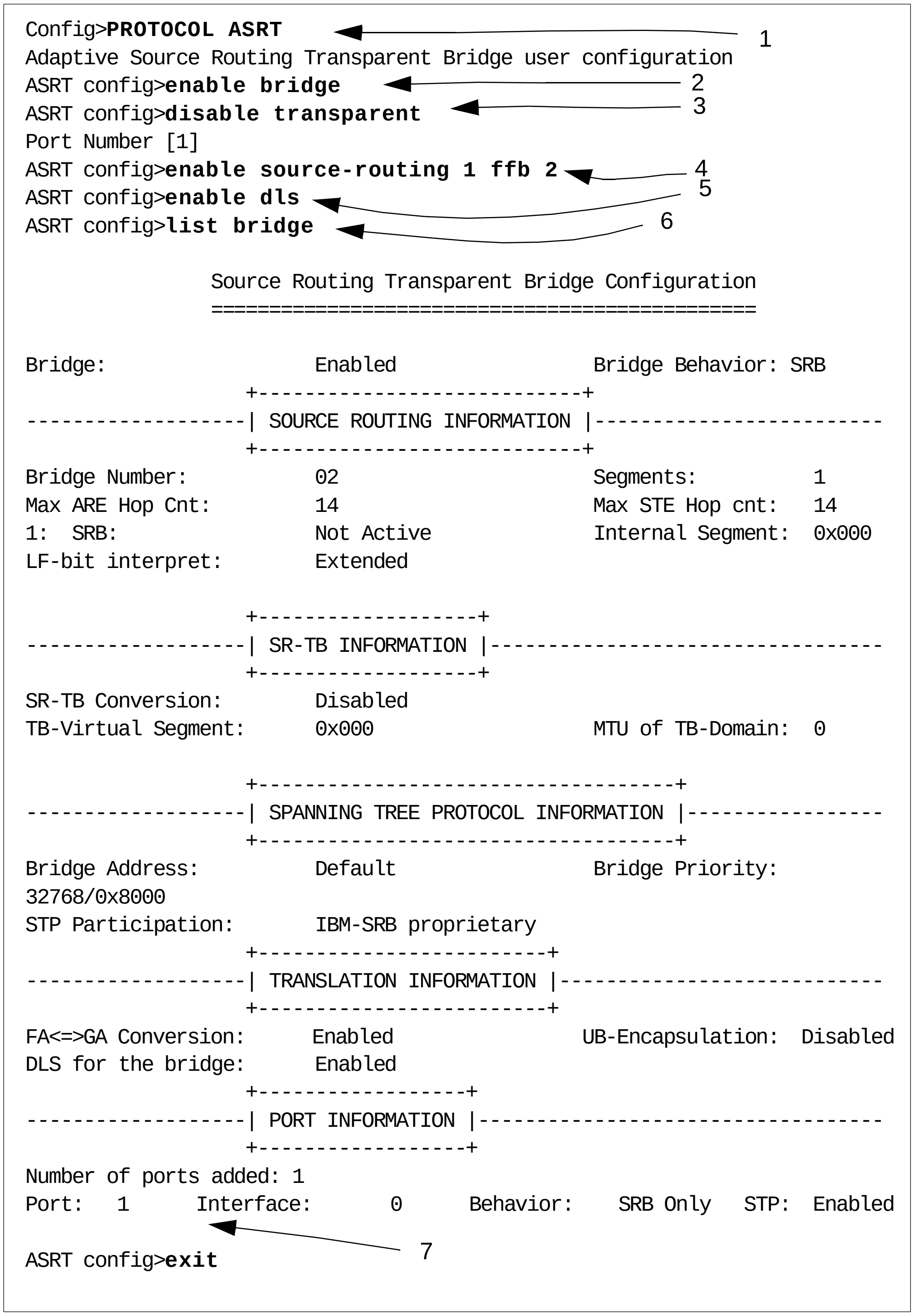


```
Config>protocol ospf
Open SPF-Based Routing Protocol configuration console
OSPF Config>enable multicast
Inter-area multicasting enabled? [No]?NO
OSPF Config>list all

                       --Global configuration--
                OSPF Protocol:          Enabled
                # AS ext. routes:       1000
                Estimated # routers:    50
                External comparison:    Type 2
                AS boundary capability: Disabled
                Multicast forwarding:   Enabled
                Inter-area multicast:   Disabled

                        --Area configuration--
Area ID          AuType          Stub? Default-cost Import-summaries?
0.0.0.0          0=None           No         N/A           N/A

                        --Interface configuration--
IP address        Area              Cost  Rtrns  TrnsDly  Pri  Hello   Dead
8.8.8.1           0.0.0.0              1      5        1    1     10     40
200.200.200.2     0.0.0.0              1      5        1    1     10     40

                            Multicast parameters
IP address        MCForward         DLUnicast    IGMPPoll      IGMPtimeout
8.8.8.1           On                Off          60            180
200.200.200.2     On                Off          60            180

OSPF Config>exit
```

Figure 158. OSPF Configuration on Remote Router

We enter `protocol ospf` (1) to invoke the OSPF configuration process, and simply say `enable multicast` (2). We then enter `list all` (3) to check the OSPF configuration.

The final step to be executed before configuring DLSw is to get the bridging setup right on all the real LAN ports. We have only one LAN port (interface 0) for which we must do this. Confusingly, the 221X assigns bridge port numbers independently from LAN interface numbers, so in this configuration interface 0 is known as bridge port 1. Please see Figure 159.



```
Config>PROTOCOL ASRT
Adaptive Source Routing Transparent Bridge user configuration
ASRT config>enable bridge
ASRT config>disable transparent
Port Number [1]
ASRT config>enable source-routing 1 ffb 2
ASRT config>enable dls
ASRT config>list bridge

                Source Routing Transparent Bridge Configuration
                ===============================================

Bridge:                  Enabled                Bridge Behavior: SRB
                   +----------------------------+
-------------------| SOURCE ROUTING INFORMATION |-------------------------
                   +----------------------------+
Bridge Number:           02                     Segments:          1
Max ARE Hop Cnt:         14                     Max STE Hop cnt:   14
1:  SRB:                 Not Active             Internal Segment:  0x000
LF-bit interpret:        Extended

                   +-------------------+
-------------------| SR-TB INFORMATION |----------------------------------
                   +-------------------+
SR-TB Conversion:        Disabled
TB-Virtual Segment:      0x000                  MTU of TB-Domain:  0

                   +-------------------------------------+
-------------------| SPANNING TREE PROTOCOL INFORMATION |-----------------
                   +-------------------------------------+
Bridge Address:          Default                Bridge Priority:
32768/0x8000
STP Participation:       IBM-SRB proprietary
                   +--------------------------+
-------------------| TRANSLATION INFORMATION |----------------------------
                   +--------------------------+
FA<=>GA Conversion:      Enabled               UB-Encapsulation:  Disabled
DLS for the bridge:      Enabled
                   +------------------+
-------------------| PORT INFORMATION |-----------------------------------
                   +------------------+
Number of ports added: 1
Port:   1     Interface:      0     Behavior:    SRB Only   STP:  Enabled

ASRT config>exit
```

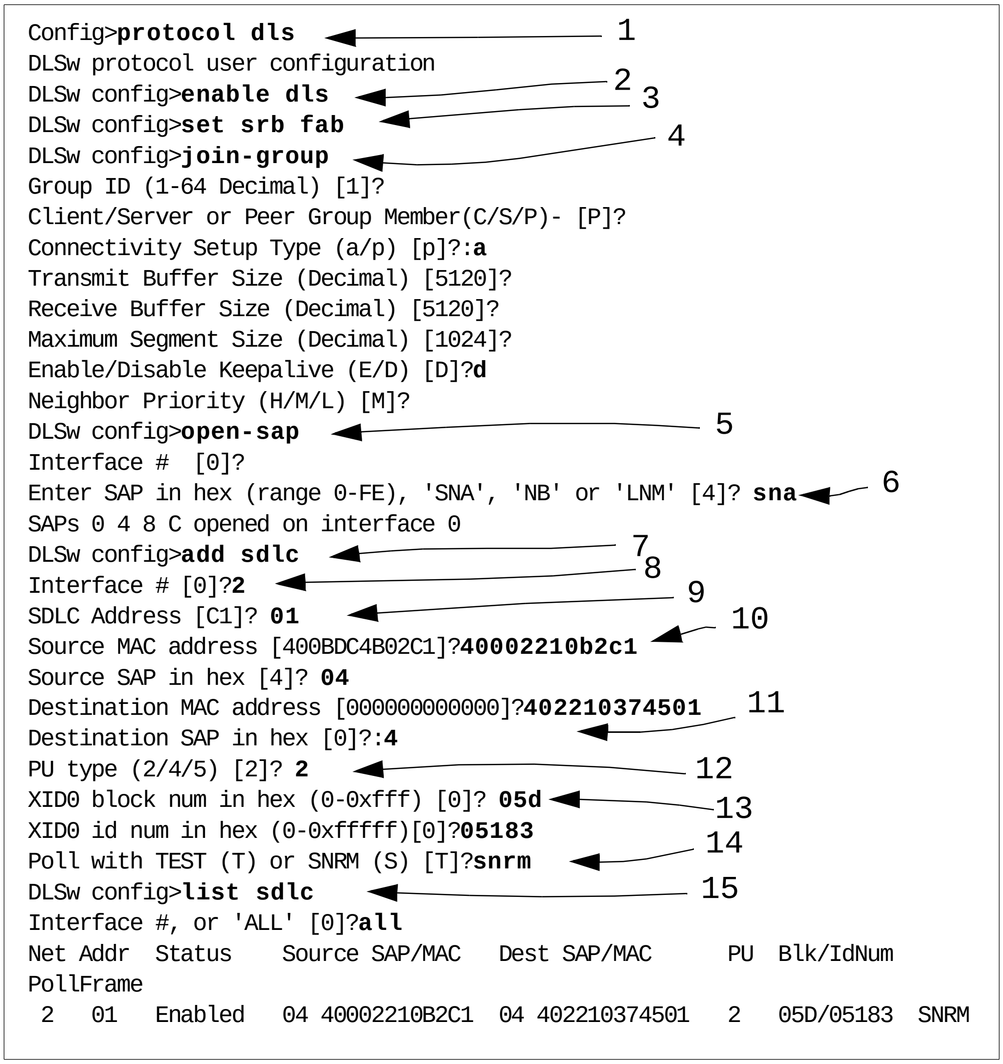
Figure 159. Bridging Configuration on Remote Router

This example shows that:

1. We enter `protocol asrt` to invoke the bridge configuration process.

2. We enable bridging, but by default the 221X creates a transparent bridge so we must make some modifications. It is not permitted for a token-ring bridge port using DLSw to have transparent bridging enabled.
3. We disable transparent bridging. We are prompted for the bridge port number, and leave the default as 1 (the only bridge port available).
4. We enable source-route bridging. We supply the port number (1), the number of the real ring segment (FFB), and the bridge number (2).
5. We enable data link switching on this bridge port.
6. We enter `list bridge` to check the bridging configuration.
7. Note at the bottom of the display that the 2210 tells you the relationship between the interface number and the bridge port number. This is the only sure way of determining this relationship.

Now we are ready for the actual DLSw configuration, as shown in Figure 160.



```
Config>protocol dls
DLSw protocol user configuration
DLSw config>enable dls
DLSw config>set srb fab
DLSw config>join-group
Group ID (1-64 Decimal) [1]?
Client/Server or Peer Group Member(C/S/P)- [P]?
Connectivity Setup Type (a/p) [p]?:a
Transmit Buffer Size (Decimal) [5120]?
Receive Buffer Size (Decimal) [5120]?
Maximum Segment Size (Decimal) [1024]?
Enable/Disable Keepalive (E/D) [D]?d
Neighbor Priority (H/M/L) [M]?
DLSw config>open-sap
Interface #  [0]?
Enter SAP in hex (range 0-FE), 'SNA', 'NB' or 'LNM' [4]? sna
SAPs 0 4 8 C opened on interface 0
DLSw config>add sdlc
Interface # [0]?2
SDLC Address [C1]? 01
Source MAC address [400BDC4B02C1]?40002210b2c1
Source SAP in hex [4]? 04
Destination MAC address [000000000000]?402210374501
Destination SAP in hex [0]?:4
PU type (2/4/5) [2]? 2
XID0 block num in hex (0-0xfff) [0]? 05d
XID0 id num in hex (0-0xfffff)[0]?05183
Poll with TEST (T) or SNRM (S) [T]?snrm
DLSw config>list sdlc
Interface #, or 'ALL' [0]?all
Net Addr  Status    Source SAP/MAC   Dest SAP/MAC      PU  Blk/IdNum
PollFrame
 2   01   Enabled   04 40002210B2C1  04 402210374501   2   05D/05183  SNRM
```

Figure 160. DLSw Configuration

The following points are worthy of note in this configuration:

1. We enter `protocol dls` to begin DLSw configuration.
2. We enable data link switching on the 2210.
3. We set the number of the virtual ring segment to FAB. This is the segment to which all LAN-attached SNA stations will appear to be bridged when establishing a connection over the DLSw network. It must be the same for all 221Xs in this DLSw group.
4. We tell the 2210 to join a group of DLSw partners. The group is identified by the number 1, and OSPF multicast will be used by the members to locate each other. The 221X can act in a client, a server or a peer capacity, and can take an active or a passive role in establishing the TCP connection with a partner.

   If the routers are configured for dynamic partner discovery, those configured as *client* will attempt to contact those configured as *server*, and vice versa. Those configured as *peer* will attempt to contact each other. An *active* router attempts to contact its partners at startup time and at intervals thereafter; a *passive* router does so only when a connection is required for a DLSw circuit. Here we define a peer group (of two routers) in which this station will be an active partner.
5. Next, we specify to which SAPs the DLSw function will respond on the LAN interfaces (0, the token-ring, in this example).
6. SNA normally uses SAP 04; the generic response `sna` to the question about SAPs results in 0, 4, 8 and C being placed in the list of SAPs supported.
7. Now we add the SDLC station to the DLSw support.
8. Interface 2 is the downstream serial port.
9. SDLC address 01 is the real polling address of the station we wish to define.
10. Next we define the MAC and SAP address to which this SDLC station will be mapped. If another SNA station on a LAN wishes to contact it this is the MAC address it must search for.
11. The remote MAC/SAP address is the partner SNA station to which this station (01) will talk. This is the virtual MAC address that represents the 3745 at the other end of the DLSw circuit. If the 3745 was LAN-attached this would be the real MAC address. It is not necessary to define this address for a type 2.0 SDLC station as the partner node will always set up the connection.
12. This is a PU type 2.
13. A type 2.0 node on a leased connection does not send a node ID to identify itself, so we must create one for it in case a LAN-attached partner wishes to contact it.
14. Similarly, a type 2.0 node expects SNRM as the first DLC frame to be sent to it.
15. When we have finished the SDLC configuration we enter `list sdlc` to check the definitions.

When the DLSw configuration has been defined to our satisfaction, we display it using the `list dls` command as shown in Figure 161 on page 179.

```
DLSw config>list dls
DLSw is                            ENABLED
LLC2 send Disconnect is            ENABLED
Dynamic Neighbors is               ENABLED

SRB Segment number                 FAB
MAC <-> IP mapping cache size      128
Max DLSw sessions                  1000
DLSw global memory allotment       141312
LLC per-session memory allotment   8192
SDLC per-session memory allotment  4096
NetBIOS UI-frame memory allotment  40960

Dynamic Neighbor Transmit Buffer Size  5120
Dynamic Neighbor Receive Buffer Size   5120
Dynamic Neighbor Maximum Segment Size  1024
Dynamic Neighbor Keep Alive            DISABLED
Dynamic Neighbor Priority              MEDIUM
DLSw config>exit
Config>
```

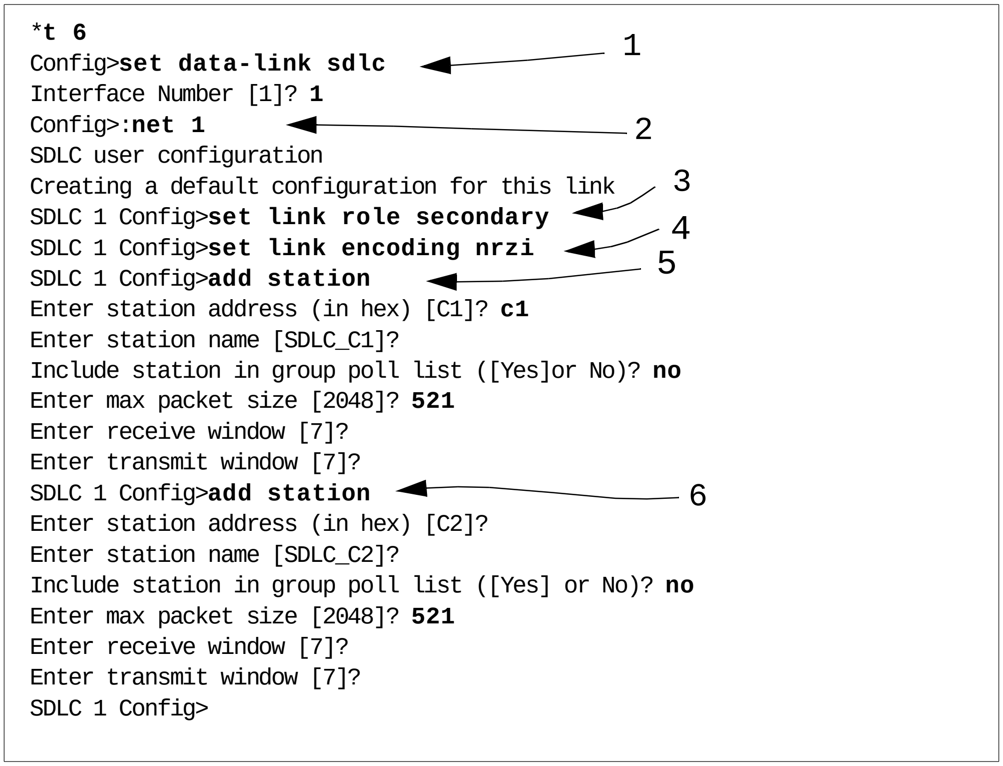
Figure 161. DLSw Configuration Display

After performing this configuration we restarted the 2210 to activate the definitions.

For the central site router 2210 A, the following parameters were configured to begin with:

- Encapsulation for interface 2 = PPP
- Cable type for interface 2 = RS232 DTE
- IP address of interface 1 = 200.200.200.1 (subnet mask 255.255.255.0)
- Internal IP address = 10.24.104.93 (subnet mask 255.255.255.0)
- Dynamic Routing = enabled
- OSPF = enabled

To configure the SDLC port (interface 1), we performed the actions shown in Figure 162 on page 180.



```
*t 6
Config>set data-link sdlc
Interface Number [1]? 1
Config>:net 1
SDLC user configuration
Creating a default configuration for this link
SDLC 1 Config>set link role secondary
SDLC 1 Config>set link encoding nrzi
SDLC 1 Config>add station
Enter station address (in hex) [C1]? c1
Enter station name [SDLC_C1]?
Include station in group poll list ([Yes]or No)? no
Enter max packet size [2048]? 521
Enter receive window [7]?
Enter transmit window [7]?
SDLC 1 Config>add station
Enter station address (in hex) [C2]?
Enter station name [SDLC_C2]?
Include station in group poll list ([Yes] or No)? no
Enter max packet size [2048]? 521
Enter receive window [7]?
Enter transmit window [7]?
SDLC 1 Config>
```

Figure 162. SDLC Configuration for Local Router

This is the primary end of an SDLC connection (the 2210 acts as the secondary), so the definitions are a little different:

1. We define interface 1 as an SDLC port.
2. We start to configure the SDLC function on interface 1.
3. The default for the link station role is primary (the 2210 acts as primary), as can be seen in the `list link` display in Figure 156 on page 173. Therefore, we define this connection as secondary. The NCP will see a multidrop line here, so it can only be the primary station.
4. On this connection the NCP uses NRZI encoding (the default is NRZ), so we must specify NRZI here.
5. We now add the C1 station (the SDLC-attached remote workstation; the connection between this C1 and the real remote station 01 will be made in the DLSw configuration section). We override the maximum BTU size to correspond with the NCP's defined MAXDATA and the remote station's actual capability.
6. We add the second (C2) station, which is the LAN-attached remote workstation. The connection between C2 and the actual MAC address of the workstation will be made at DLSw configuration time.

A display of the configuration of interface 1, together with its link stations, can be seen in Figure 163 on page 181.

```
SDLC 1 Config>list link
Link configuration for: LINK_1   (ENABLED)

Role:         SECONDARY        Type:        POINT-TO-POINT
Duplex:       FULL             Modulo:      8
Idle state:   FLAG             Encoding:    NRZI
Clocking:     EXTERNAL         Frame Size:  2048
Speed:        0                Group Poll:  00
Cable:        RS-232 DTE

Timers:    XID/TEST response:  2.0 sec
           SNRM response:      2.0 sec
           Poll response:      0.5 sec
           Inter-poll delay:   0.2 sec
           RTS hold delay:     DISABLED
           Inter-frame delay:  DISABLED
           Inactivity timeout: 30.0 sec

Counters:  XID/TEST retry:  8
           SNRM retry:      6
           Poll retry:      10

SDLC 1 Config>list station all

Address    Name      Status    Max BTU  Rx Window  Tx Window
-------  --------  ----------  -------  ---------  ---------
  C1     SDLC_C1     ENABLED      521       7          7
  C2     SDLC_C2     ENABLED      521       7          7
SDLC 1 Config>exit
```

Figure 163. Link and Station Display for Local Router

Next, we configure the IP protocol (Figure 164).

```
Config>:protocol ip
Internet protocol user configuration

IP config>:disable rip
IP config>set internal-ip-address 10.24.104.93

IP config>
```

Figure 164. IP Configuration for Local Router

The only difference between this and the remote 2210 is the internal IP address (10.24.104.93). Next we turn to OSPF, where the configuration is identical to that for the remote 2210 (Figure 158 on page 175).

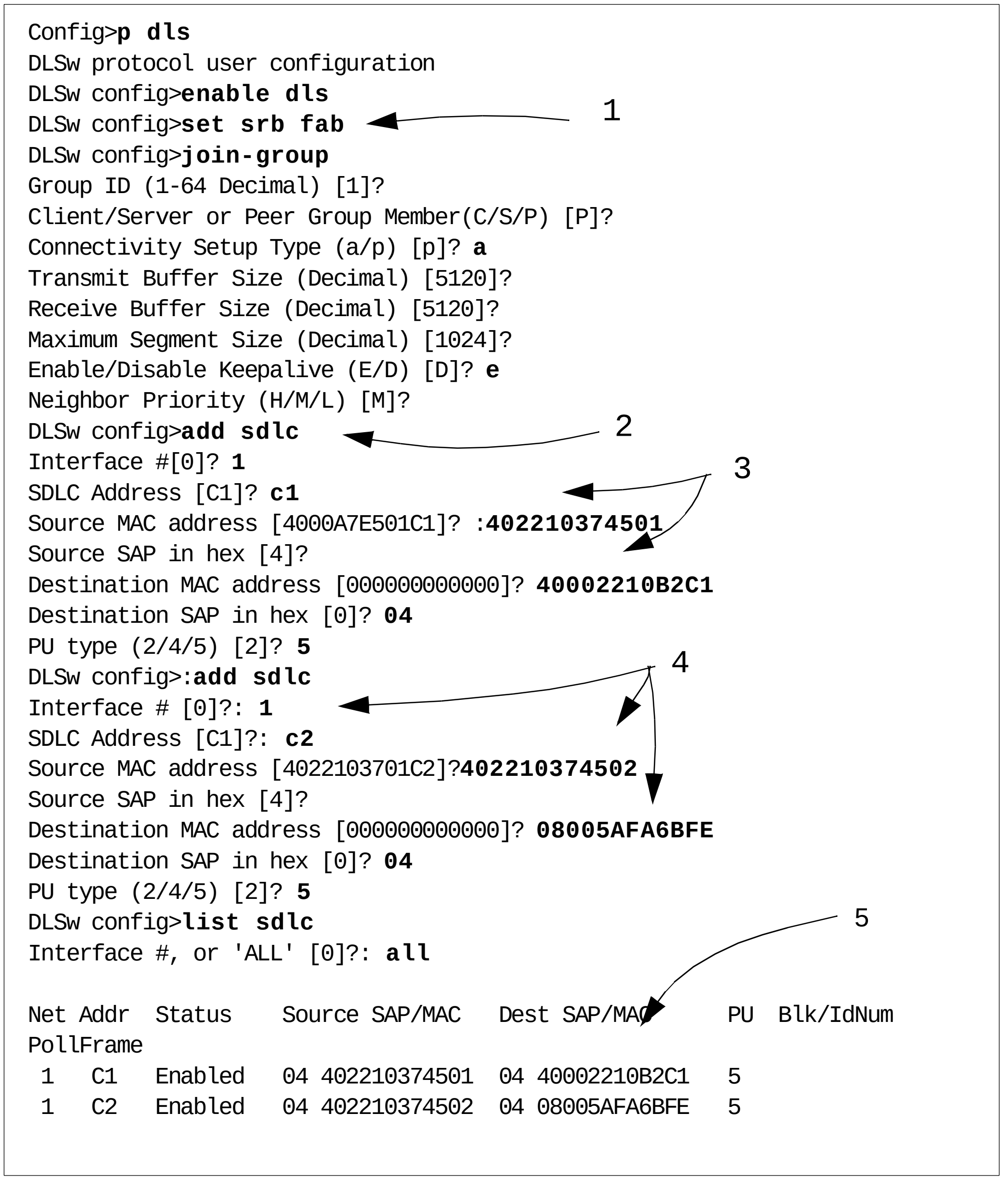
Next we configure bridging as shown in Figure 165 on page 182. There are no real LAN interfaces in operation, so we do not need to worry about source routing or transparent bridging or bridge ports. In fact, this bridge configuration step is not really necessary as there is no real LAN interface.

```
Config>protocol asrt
Adaptive Source Routing Transparent Bridge user configuration
ASRT config>enable bridge
ASRT config>enable dls
ASRT config>exit
```

Figure 165. Bridge Configuration for Local Router

For an SDLC connection, the command `add sdlc` has the same effect as `enable dls` in Figure 165 has for a LAN interface.

The next step is DLSw configuration itself, as shown in Figure 166.



```
Config>p dls
DLSw protocol user configuration
DLSw config>enable dls
DLSw config>set srb fab
DLSw config>join-group
Group ID (1-64 Decimal) [1]?
Client/Server or Peer Group Member(C/S/P) [P]?
Connectivity Setup Type (a/p) [p]? a
Transmit Buffer Size (Decimal) [5120]?
Receive Buffer Size (Decimal) [5120]?
Maximum Segment Size (Decimal) [1024]?
Enable/Disable Keepalive (E/D) [D]? e
Neighbor Priority (H/M/L) [M]?
DLSw config>add sdlc
Interface #[0]? 1
SDLC Address [C1]? c1
Source MAC address [4000A7E501C1]? :402210374501
Source SAP in hex [4]?
Destination MAC address [000000000000]? 40002210B2C1
Destination SAP in hex [0]? 04
PU type (2/4/5) [2]? 5
DLSw config>:add sdlc
Interface # [0]?: 1
SDLC Address [C1]?: c2
Source MAC address [4022103701C2]?402210374502
Source SAP in hex [4]?
Destination MAC address [000000000000]? 08005AFA6BFE
Destination SAP in hex [0]? 04
PU type (2/4/5) [2]? 5
DLSw config>list sdlc
Interface #, or 'ALL' [0]?: all

Net Addr  Status    Source SAP/MAC   Dest SAP/MAC      PU  Blk/IdNum
PollFrame
 1   C1   Enabled   04 402210374501  04 40002210B2C1   5
 1   C2   Enabled   04 402210374502  04 08005AFA6BFE   5
```

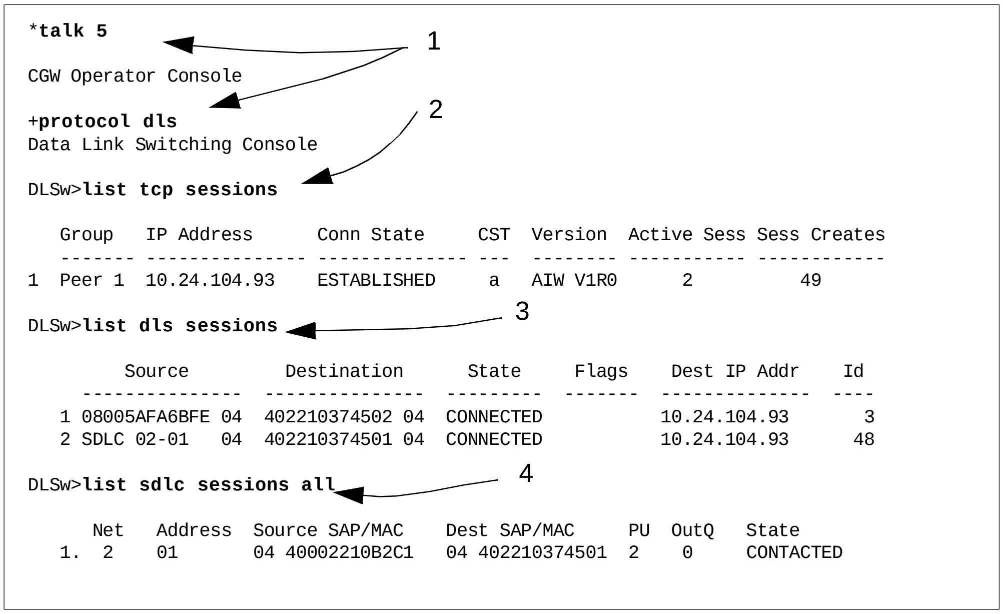
Figure 166. DLSw Configuration for Local Router

In this figure:

1. We enable DLSw and set the number of the virtual DLSw segment to FAB, as in the remote 2210.
2. We add an SDLC station known to DLSw to interface 1.
3. This station has address C1 (which the 3745 will poll). The 3745 side is represented by the MAC address 402210374501, while the remote side is represented by the MAC address 40002210B2C1. This address has been configured in the remote 2210 as being that of the SDLC-attached workstation (see Figure 160 on page 177). This connection is to a PU type 5 (type 4 and type 5 are interchangeable here).
4. The second station on this multidrop connection has address C2. The 3745 side of this one has MAC address 402210374502, and this is the address that is configured in the LAN-attached remote workstation as the destination MAC address. The remote side has MAC address 08005AFA6BFE, which is the real address of the remote workstation. This connection is also to a PU type 5.
5. Once again, a display confirms what we have just configured. This time there is no need to define a node ID on either link station. The remote workstations will not check the incoming node ID from VTAM, whereas VTAM needs the workstations' node IDs to identify the remote resources.

Once again, we restarted this node to activate the configuration. The host and workstation connections were activated, the DLSw circuits were established and communication was initiated.

The 221X routers have commmands available to tell you the status of the DLSw protocol functions. Figure 167 shows a display taken on the remote 2210, 2210B, after the network was activated.



```
*talk 5

CGW Operator Console

+protocol dls
Data Link Switching Console

DLSw>list tcp sessions

   Group   IP Address      Conn State     CST  Version  Active Sess Sess Creates
   ------- --------------- -------------- ---  -------- ----------- ------------
1  Peer 1  10.24.104.93    ESTABLISHED     a   AIW V1R0      2          49

DLSw>list dls sessions

         Source         Destination      State     Flags    Dest IP Addr    Id
     ---------------  ---------------  ---------  -------  --------------  ----
   1 08005AFA6BFE 04  402210374502 04  CONNECTED           10.24.104.93       3
   2 SDLC 02-01   04  402210374501 04  CONNECTED           10.24.104.93      48

DLSw>list sdlc sessions all

      Net   Address  Source SAP/MAC    Dest SAP/MAC     PU  OutQ   State
   1.  2    01       04 40002210B2C1   04 402210374501  2    0     CONTACTED
```

Figure 167. DLSw Display from Remote Router

In this figure:

1. Entering `talk 5` and `p dls` from the basic prompt gives you the DLSw console function.
2. The `list tcp sessions` command shows you the TCP connections to the DLSw partners. Here there is only one, to 2210A whose internal IP address is shown as the connection partner.
3. The `list dls sessions` command tells you the DLSw circuits that are active from this router. As expected, there is one between the SDLC-attached station and the 3745, and one between the LAN-attached workstation and the 3745. Both have the same DLSw partner.
4. The `list sdlc sessions` command gives details of the SNA connections on all attached SDLC interfaces.

Figure 168 shows the corresponding display on the local router taken at the same time.

```
*t 5

CGW Operator Console

+protocol dlsw

DLSw>list tcp sessions

   Group   IP Address      Conn State     CST  Version  Active Sess Sess Creates
   ------- --------------- -------------- ---  -------- ----------- ------------
1  Peer 1  10.8.8.1        ESTABLISHED     a   AIW V1R0      2          49

DLSw>list dls sessions all

         Source         Destination      State     Flags    Dest IP Addr    Id
     ---------------  ---------------  ---------  -------  -------------  ----
   1 SDLC 01-C1   04  40002210B2C1 04  CONNECTED           10.8.8.1         48
   2 SDLC 01-C2   04  08005AFA6BFE 04  CONNECTED           10.8.8.1         52

DLSw>list sdlc sessions

      Net   Address  Source SAP/MAC    Dest SAP/MAC     PU  OutQ   State
   1.  1    C1       04 402210374501   04 40002210B2C1  5    0     CONTACTED
   2.  1    C2       04 402210374502   04 08005AFA6BFE  5    0     CONTACTED
```

Figure 168. DLSw Display from Local Router

5.3.2 Remote Ethernet over PPP to Token-Ring Host

In the next scenario (Figure 169 on page 185), we show a DLSw configuration linking an Ethernet-attached workstation to a host, but this time the host is connected to the local router by means of a token-ring.

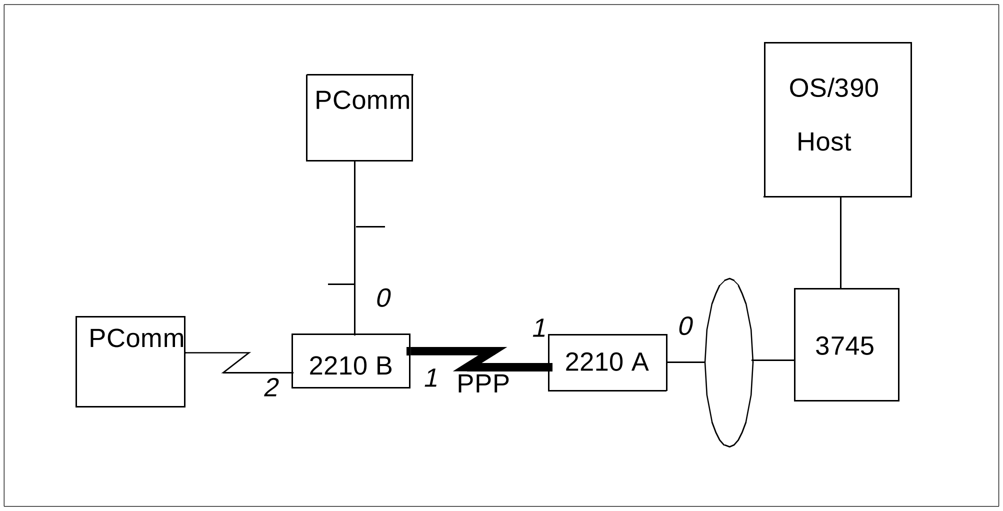


Figure 169. DLSw via PPP to LAN-attached Host

There are two major differences between this configuration and the last one: the remote LAN workstation is now on an Ethernet segment instead of a token-ring, and the host is connected to the network by a token-ring instead of an SDLC line. In this case:

- The central site router, 2210A, was attached to the token-ring by interface number 0. The PPP link to the partner 2210 was on serial port 1.
- The remote router, 2210B, had the PPP link on serial port 1 and an SDLC connection to a downstream 3270 emulation workstation on serial port 2. It also had an Ethernet port on interface 0.
- The Ethernet-attached workstation had to be configured with a MAC address to represent the 3745. This time the MAC address was the real 3745 TIC address.
- VTAM on the host was given the MAC addresses of the workstations; the real one for the Ethernet-attached station and a virtual one for the SDLC-attached station. The virtual one was mapped to the SDLC station address in 2210B, just as in the previous configuration. Note that Ethernet MAC addresses are specified to DLSw in non-canonical (token-ring) format.

Both remote workstations were configured as PU type 2.0 devices. The Ethernet-attached station and the NCP would exchange XIDs at activation time, but the NCP expected to exchange XIDs with the SDLC-attached station thinking it was on a LAN. 2210B was responsible for translating the XID information and the MAC address to and from the SDLC polling information.

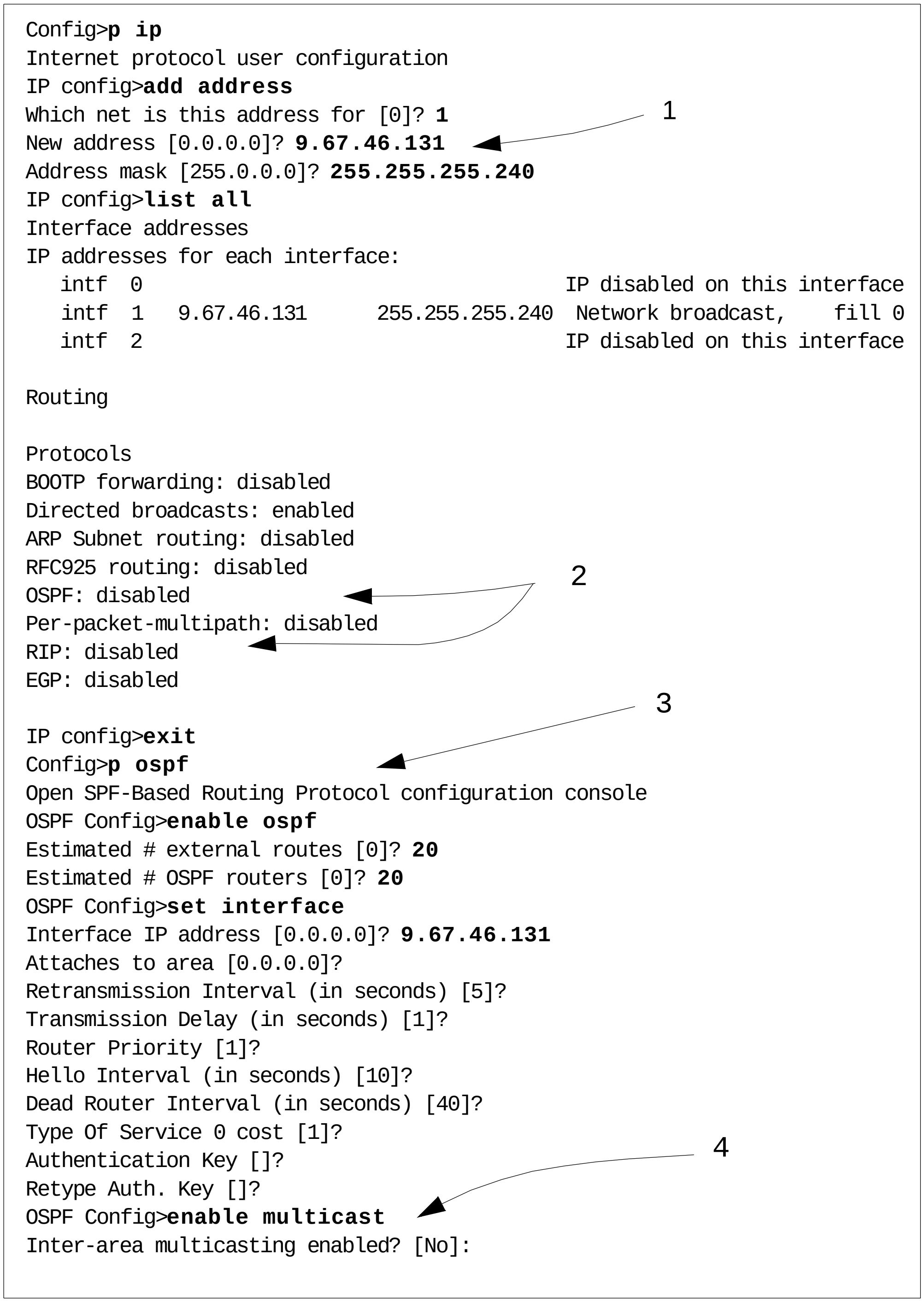
In this example we show only those parts of the configuration that are significantly different from the corresponding parts in “Remote SDLC and Token-Ring over PPP to SDLC Host” on page 170.

We start by configuring 2210A, the router at the host side of the network. Figure 170 on page 186 shows the physical configuration of the two interfaces.

```
*t 6
Config>net 0
Token-Ring interface configuration
TKR config>media unshielded
TKR config>set physical 40:00:22:10:A0:01
TKR config>list
Token-Ring configuration:

Packet size (INFO field): 2052
Speed:                    4 Mb/sec
Media:                    Unshielded

RIF Aging Timer:          120
Source Routing:           Enabled
MAC Address:              40002210A001
IPX interface configuration record missing

TKR config>exit
Config>net 1
Point-to-Point user configuration
PPP Config>set hdlc cable v35 dte
PPP Config>list all

Maximum frame size in bytes = 2048
Encoding: NRZ
Idle State: Flag
Clocking: External
Cable type: V.35 DTE
Internal Clock Speed: 0
Transmit Delay Counter: 0

LCP Parameters
CONFIG Request Tries:              20   CONFIG Nak Tries:
10
Terminate Tries:                  10   Retry Timer:                    3000

LCP Options
Max Receive Unit:               2048   Magic Number:                    Yes

NCP Parameters
CONFIG Request Tries:              20   CONFIG Nak Tries:
10
Terminate Tries:                  10   Retry Timer:                    3000

IPCP Options
IPCP Compression:                 None
IP Address:      Don't Send or Request
PPP Config>exit
```

Figure 170. Local Router Physical Interface Configuration

After configuring the token-ring and the PPP interfaces (0 and 1 respectively), we display their definitions to confirm that all the defaults are what we want them to be. Next, we configure IP and OSPF as shown in Figure 171 on page 187.

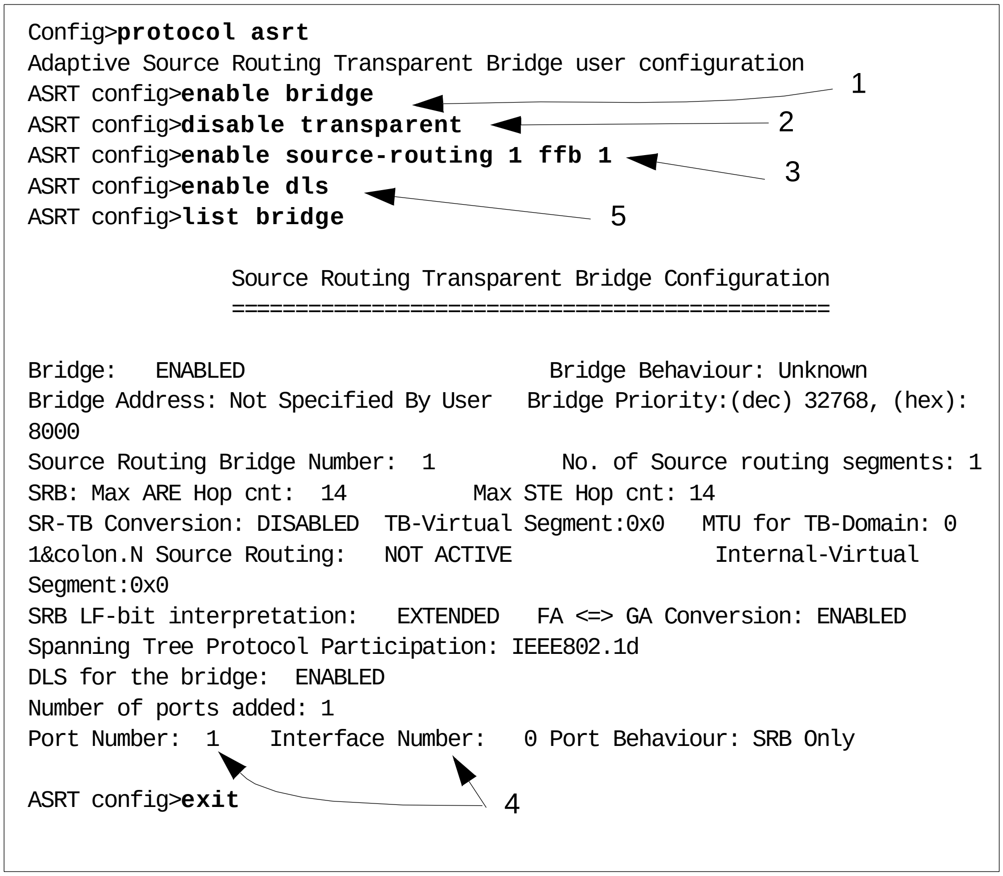


```
Config>p ip
Internet protocol user configuration
IP config>add address
Which net is this address for [0]? 1
New address [0.0.0.0]? 9.67.46.131
Address mask [255.0.0.0]? 255.255.255.240
IP config>list all
Interface addresses
IP addresses for each interface:
   intf  0                                    IP disabled on this interface
   intf  1   9.67.46.131      255.255.255.240  Network broadcast,    fill 0
   intf  2                                    IP disabled on this interface

Routing

Protocols
BOOTP forwarding: disabled
Directed broadcasts: enabled
ARP Subnet routing: disabled
RFC925 routing: disabled
OSPF: disabled
Per-packet-multipath: disabled
RIP: disabled
EGP: disabled

IP config>exit
Config>p ospf
Open SPF-Based Routing Protocol configuration console
OSPF Config>enable ospf
Estimated # external routes [0]? 20
Estimated # OSPF routers [0]? 20
OSPF Config>set interface
Interface IP address [0.0.0.0]? 9.67.46.131
Attaches to area [0.0.0.0]?
Retransmission Interval (in seconds) [5]?
Transmission Delay (in seconds) [1]?
Router Priority [1]?
Hello Interval (in seconds) [10]?
Dead Router Interval (in seconds) [40]?
Type Of Service 0 cost [1]?
Authentication Key []?
Retype Auth. Key []?
OSPF Config>enable multicast
Inter-area multicasting enabled? [No]:
```

Figure 171. IP and OSPF Configuration

We assign an IP address to the PPP interface 1 (1), but not to interface 0 because there is no IP traffic on the token-ring. By default all routing protocols are disabled (2), so we enable OSPF (3) and finally enable OSPF multicast (4) because we will be using dynamic partner discovery again.

Next we configure bridging, as in Figure 172 on page 188.

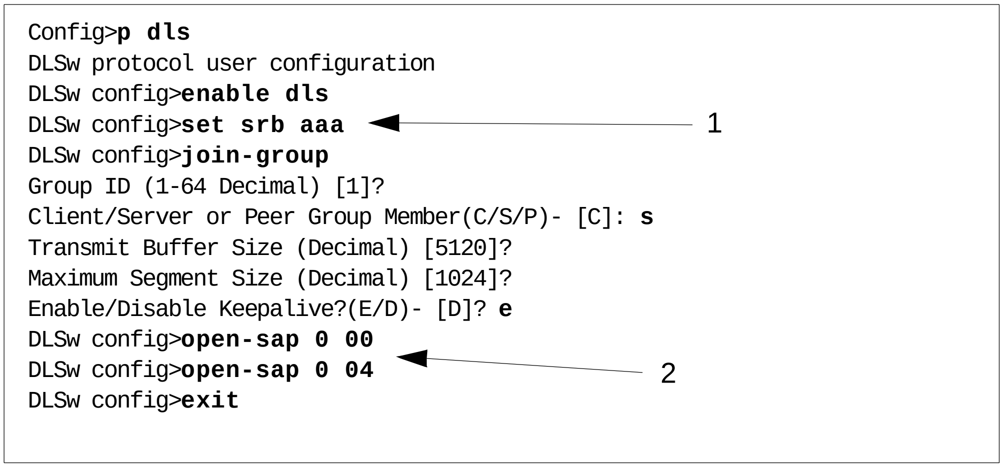


```
Config>protocol asrt
Adaptive Source Routing Transparent Bridge user configuration
ASRT config>enable bridge
ASRT config>disable transparent
ASRT config>enable source-routing 1 ffb 1
ASRT config>enable dls
ASRT config>list bridge

                Source Routing Transparent Bridge Configuration
                ===============================================

Bridge:   ENABLED                        Bridge Behaviour: Unknown
Bridge Address: Not Specified By User   Bridge Priority:(dec) 32768, (hex):
8000
Source Routing Bridge Number:  1          No. of Source routing segments: 1
SRB: Max ARE Hop cnt:  14          Max STE Hop cnt: 14
SR-TB Conversion: DISABLED  TB-Virtual Segment:0x0   MTU for TB-Domain: 0
1&colon.N Source Routing:   NOT ACTIVE                Internal-Virtual
Segment:0x0
SRB LF-bit interpretation:   EXTENDED   FA <=> GA Conversion: ENABLED
Spanning Tree Protocol Participation: IEEE802.1d
DLS for the bridge:  ENABLED
Number of ports added: 1
Port Number:  1    Interface Number:   0 Port Behaviour: SRB Only

ASRT config>exit
```

Figure 172. Bridge Configuration for Local 2210

We must enable bridging (1), disable transparent bridging (2) because this is a token-ring, and enable source-route bridging (3). The interface to be bridged is port 1 (interface 0, as the display at 4 confirms), the bridge number is 1 and the segment number of the token-ring is FFB. Finally we enable DLSw for this interface (5).

Now we turn to the DLSw configuration; see Figure 173.



```
Config>p dls
DLSw protocol user configuration
DLSw config>enable dls
DLSw config>set srb aaa
DLSw config>join-group
Group ID (1-64 Decimal) [1]?
Client/Server or Peer Group Member(C/S/P)- [C]: s
Transmit Buffer Size (Decimal) [5120]?
Maximum Segment Size (Decimal) [1024]?
Enable/Disable Keepalive?(E/D)- [D]? e
DLSw config>open-sap 0 00
DLSw config>open-sap 0 04
DLSw config>exit
```

Figure 173. DLSw Configuration for Local 2210

We enable DLSw and set the virtual segment number to AAA (1). Again we tell this 2210 to join a group of DLSw routers, this time as a server to which remote client routers will connect themselves. We allow SAPs 0 and 4 to be served by this DLSw function (2).

On the remote 2210B, we configure the serial links as shown earlier, and then the Ethernet interface as shown in Figure 174. As you can see, this is trivial and we need no customization at all to make it work.

```
*talk 6

Config>net 0
Ethernet interface configuration
ETH config>list
Connector type:           AUTO-CONFIG
No IPX interface configuration
IP Encapsulation:         ETHER

ETH config>exit
```

Figure 174. Ethernet Port Configuration

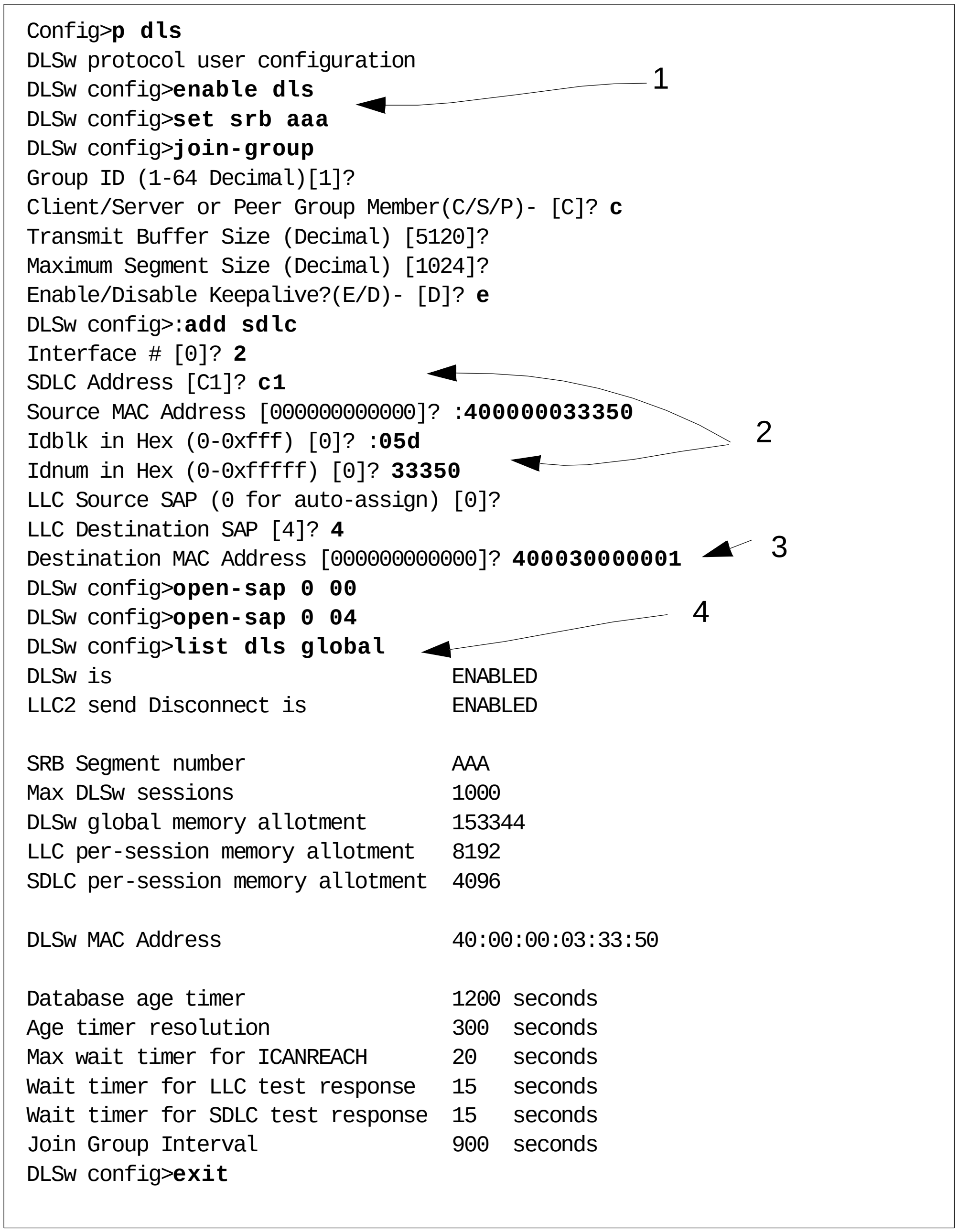
The IP and OSPF protocol configurations are similar to that for 2210A (Figure 171 on page 187). We assign the IP address 9.67.46.130 to interface 1 and enable OSPF multicast as before. The bridging configuration, however, is different as you can see in Figure 175 on page 190.

```
Config>p asrt
Adaptive Source Routing Transparent Bridge user configuration
ASRT config>enable bridge
ASRT config>enable dls                                        1
ASRT config>list bridge

                Source Routing Transparent Bridge Configuration
                ===============================================

Bridge:   ENABLED                        Bridge Behaviour: Unknown
Bridge Address: Not Specified By User   Bridge Priority:(dec) 32768, (hex):
8000
Source Routing Bridge Number:  0          No. of Source routing segments: 0
SRB: Max ARE Hop cnt:   0          Max STE Hop cnt: 0
SR-TB Conversion: DISABLED  TB-Virtual Segment:0x0   MTU for TB-Domain: 0
1&colon.N Source Routing:   NOT ACTIVE                Internal-Virtual
Segment:0x0
SRB LF-bit interpretation:   EXTENDED   FA <=> GA Conversion: ENABLED
Spanning Tree Protocol Participation: IEEE802.1d
DLS for the bridge:  ENABLED
Number of ports added: 1
Port Number:  1    Interface Number:   0 Port Behaviour: STB Only

ASRT config>exit
```

Figure 175. Bridge Definition for Ethernet 2210

We enable bridging and data link switching as before (1), but this time we fail to disable the default transparent bridging and we fail to enable source-route bridging. The result is recognized as the familiar transparent bridge by Ethernet stations.

Now we configure DLSw as in Figure 176 on page 191.

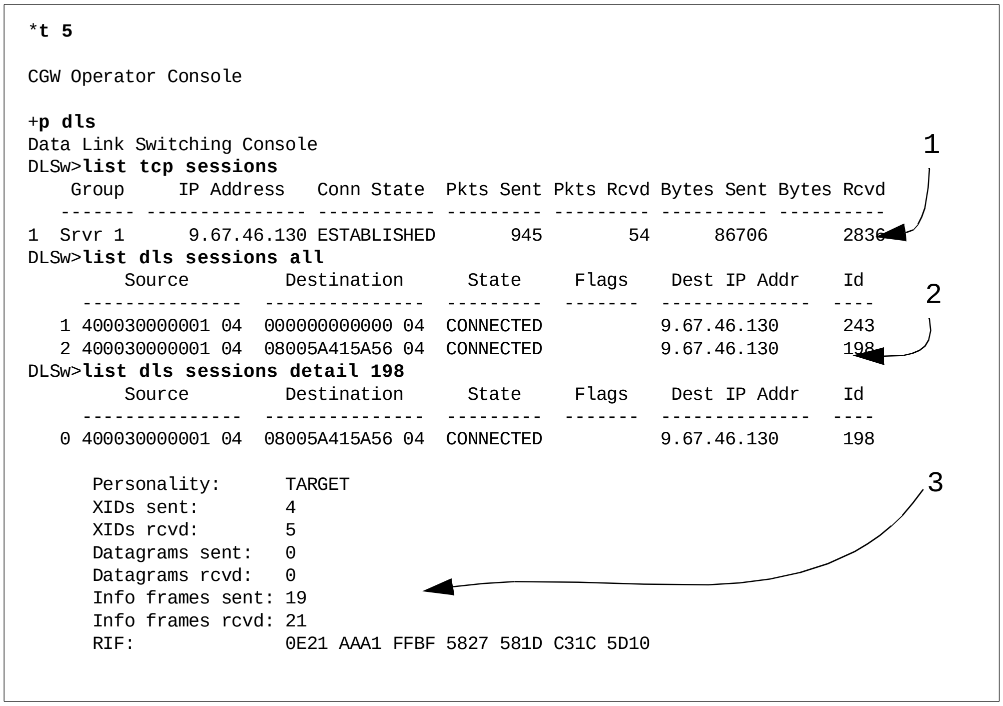


```
Config>p dls
DLSw protocol user configuration
DLSw config>enable dls
DLSw config>set srb aaa
DLSw config>join-group
Group ID (1-64 Decimal)[1]?
Client/Server or Peer Group Member(C/S/P)- [C]? c
Transmit Buffer Size (Decimal) [5120]?
Maximum Segment Size (Decimal) [1024]?
Enable/Disable Keepalive?(E/D)- [D]? e
DLSw config>:add sdlc
Interface # [0]? 2
SDLC Address [C1]? c1
Source MAC Address [000000000000]? :400000033350
Idblk in Hex (0-0xfff) [0]? :05d
Idnum in Hex (0-0xfffff) [0]? 33350
LLC Source SAP (0 for auto-assign) [0]?
LLC Destination SAP [4]? 4
Destination MAC Address [000000000000]? 400030000001
DLSw config>open-sap 0 00
DLSw config>open-sap 0 04
DLSw config>list dls global
DLSw is                           ENABLED
LLC2 send Disconnect is           ENABLED

SRB Segment number                AAA
Max DLSw sessions                 1000
DLSw global memory allotment      153344
LLC per-session memory allotment  8192
SDLC per-session memory allotment 4096

DLSw MAC Address                  40:00:00:03:33:50

Database age timer                1200 seconds
Age timer resolution              300  seconds
Max wait timer for ICANREACH      20   seconds
Wait timer for LLC test response  15   seconds
Wait timer for SDLC test response 15   seconds
Join Group Interval               900  seconds
DLSw config>exit
```

Figure 176. DLSw Configuration for Local 2210

We enable DLSw (1) and set the virtual ring number to AAA to correspond with that defined in the remote router. Then we make this router a client in the DLSw group. Next we define the SDLC-attached station to DLSw (2), giving it a virtual MAC address and node ID so that it can communicate with other devices that think it is on a LAN. We define the destination address for this station (3, the real MAC address of the 3745 TIC) and assign SAPs 0 and 4 to be serviced by DLSw in this router. Finally we check the DLSw configuration (4).

After activating these routers and the attached SNA devices, we were able to establish 3270 sessions between the workstation emulators and the host applications. Console displays on both 2210s showed what was going on in terms of DLSw. Figure 177 is a display taken on the local 2210A.



```
*t 5

CGW Operator Console

+p dls
Data Link Switching Console
DLSw>list tcp sessions
    Group     IP Address   Conn State  Pkts Sent Pkts Rcvd Bytes Sent Bytes Rcvd
   ------- -------------- ----------- --------- --------- ---------- ----------
1  Srvr 1      9.67.46.130 ESTABLISHED       945        54      86706       2836
DLSw>list dls sessions all
         Source         Destination      State     Flags    Dest IP Addr    Id
     ---------------  ---------------  ---------  -------  --------------  ----
   1 400030000001 04  000000000000 04  CONNECTED           9.67.46.130      243
   2 400030000001 04  08005A415A56 04  CONNECTED           9.67.46.130      198
DLSw>list dls sessions detail 198
         Source         Destination      State     Flags    Dest IP Addr    Id
     ---------------  ---------------  ---------  -------  --------------  ----
   0 400030000001 04  08005A415A56 04  CONNECTED           9.67.46.130      198

      Personality:      TARGET
      XIDs sent:        4
      XIDs rcvd:        5
      Datagrams sent:   0
      Datagrams rcvd:   0
      Info frames sent: 19
      Info frames rcvd: 21
      RIF:              0E21 AAA1 FFBF 5827 581D C31C 5D10
```

Figure 177. Display from Local LAN-attached 2210

In this figure:

1. Shows the DLSw TCP connection between 2210A and 2210B. Note that the DLSw connection is using the real IP interface address of the partner router; we did not configure an internal (VIPA) IP address on either 2210 in this example.
2. Shows the two DLSw circuits between the 3745 and the two remote stations. The first one is the SDLC station and the second is the Ethernet-attached station.
3. Is a more detailed display of the Ethernet station's connection. It shows the traffic statistics, and the routing information field used by DLSw to simulate a source-routing bridge to the 3745 attached to it.

A corresponding display on 2210B can be seen in Figure 178 on page 193.

```
*t 5

+p dls
Data Link Switching Console
DLSw>list tcp sessions
    Group     IP Address   Conn State  Pkts Sent Pkts Rcvd Bytes Sent Bytes Rcvd
   ------- --------------- ----------- --------- --------- ---------- ----------
1  Clnt 1      9.67.46.131 ESTABLISHED        50       486       2772      47528
DLSw>list dls sessions
         Source         Destination      State     Flags    Dest IP Addr    Id
     ---------------  ---------------  ---------  -------  --------------  ----
   1 000000000000 04  400030000001 04  CONNECTED           9.67.46.131       18
   2 08005A415A56 04  400030000001 04  CONNECTED           9.67.46.131      347
DLSw>list dls sessions detail 18
         Source         Destination      State     Flags    Dest IP Addr    Id
     ---------------  ---------------  ---------  -------  --------------  ----
   0 000000000000 04  400030000001 04  CONNECTED           9.67.46.131       18

      Personality:      ORIGINATOR
      XIDs sent:        2
      XIDs rcvd:        1
      Datagrams sent:   0
      Datagrams rcvd:   0
      Info frames sent: 18
      Info frames rcvd: 16
      RIF:
DLSw>list sdlc sessions
     Net     Addr     Src Sap  Dst Sap  Dest MAC           OutQ  State
   1.  2      C1        04       04     40:00:30:00:00:01   00   CONTACTED
```

Figure 178. DLSw Display on Remote 2210

This figure shows the detailed display of the SDLC station's circuit. There is no routing information field because there is no token-ring. Also, we show the SDLC session display for this station.

5.3.3 Data Link Switching over a Frame Relay Connection

In our third example we show DLSw in use where there are three routers in the DLSw group connected via a frame relay wide area network. Figure 179 on page 194 shows the configuration.

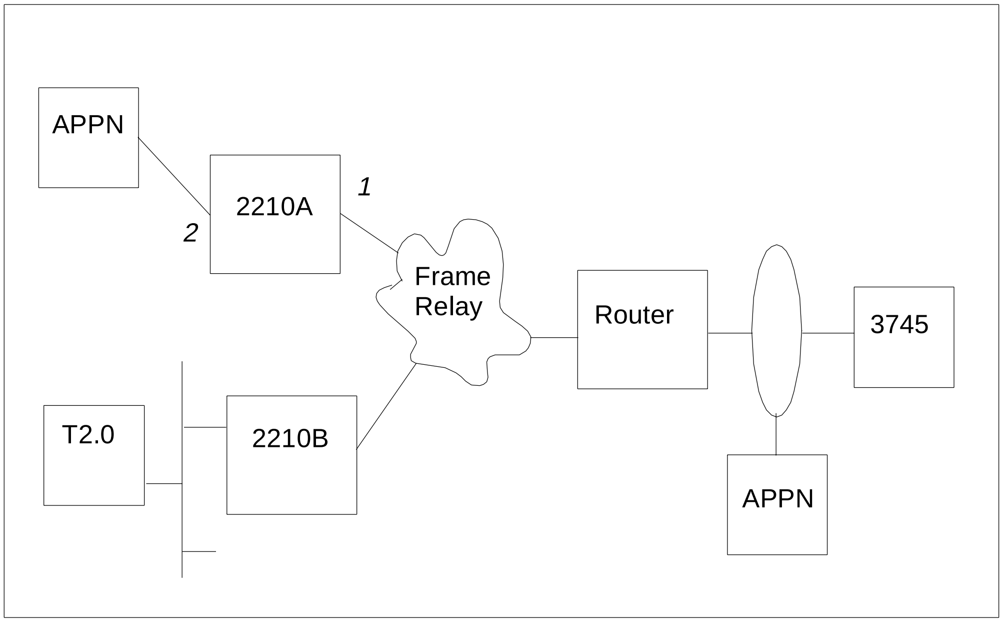
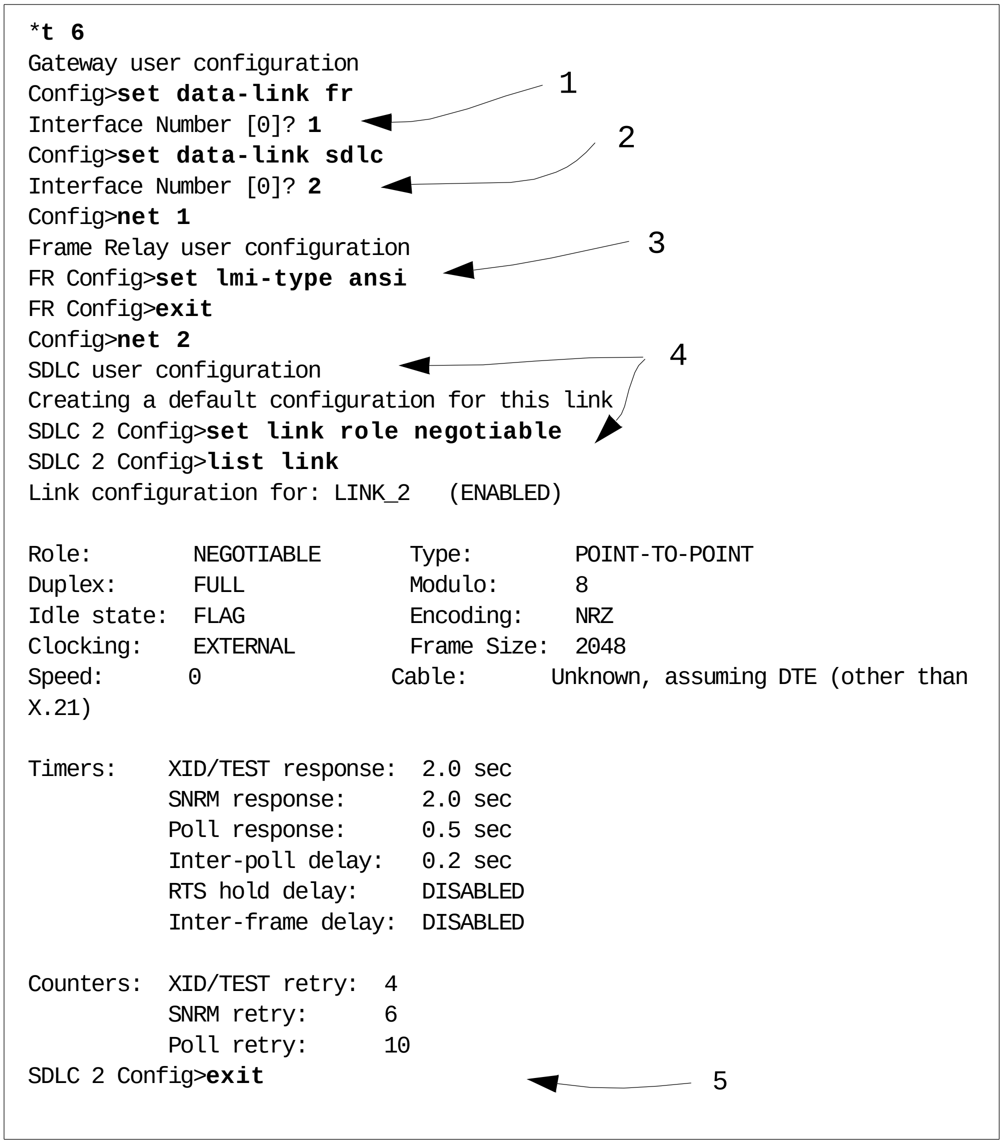


Figure 179. DLSw over Frame Relay

We show here only the configuration for the remote router 2210A, which has several features not previously described:

- It has an APPN station attached which will communicate with a partner APPN (type 2.1) node across the DLSw network. The SDLC connection to this is on interface 2.
- It has a frame relay WAN link (interface 1).
- It will not use dynamic DLSw partner discovery because the central router (actually an old 6611) does not support it.

Figure 180 on page 195 shows the configuration of the 2210's interfaces.



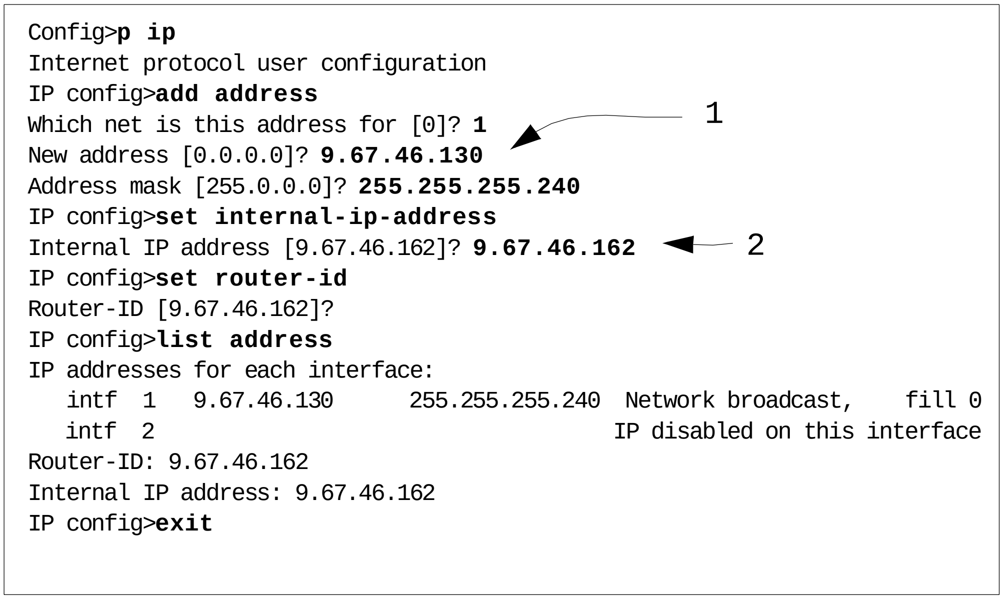
```
*t 6
Gateway user configuration
Config>set data-link fr
Interface Number [0]? 1
Config>set data-link sdlc
Interface Number [0]? 2
Config>net 1
Frame Relay user configuration
FR Config>set lmi-type ansi
FR Config>exit
Config>net 2
SDLC user configuration
Creating a default configuration for this link
SDLC 2 Config>set link role negotiable
SDLC 2 Config>list link
Link configuration for: LINK_2   (ENABLED)

Role:         NEGOTIABLE       Type:         POINT-TO-POINT
Duplex:       FULL             Modulo:       8
Idle state:   FLAG             Encoding:     NRZ
Clocking:     EXTERNAL         Frame Size:   2048
Speed:        0                Cable:        Unknown, assuming DTE (other than
X.21)

Timers:     XID/TEST response:  2.0 sec
            SNRM response:      2.0 sec
            Poll response:      0.5 sec
            Inter-poll delay:   0.2 sec
            RTS hold delay:     DISABLED
            Inter-frame delay:  DISABLED

Counters:   XID/TEST retry:  4
            SNRM retry:      6
            Poll retry:      10
SDLC 2 Config>exit
```

Figure 180. Frame Relay Router Configuration

In this example:

1. We specify that interface 1 is to use frame relay.
2. We also define interface 2 to be an SDLC link.
3. We perform a minimum of configuration on the frame relay connection; we simply state that the ANSI standards are to be used on the local management interface.
4. We configure the SDLC link to be negotiable. This tells the 2210 to expect type 2.1 nodes on this connection; in other words, an exchange of XID-3s is to be expected to establish the DLC connection rather than a SNRM sent from the SNA host.
5. We do not need to define the remote SDLC station. A station will be added by the 221X.

The IP configuration is next, as in Figure 181 on page 196.

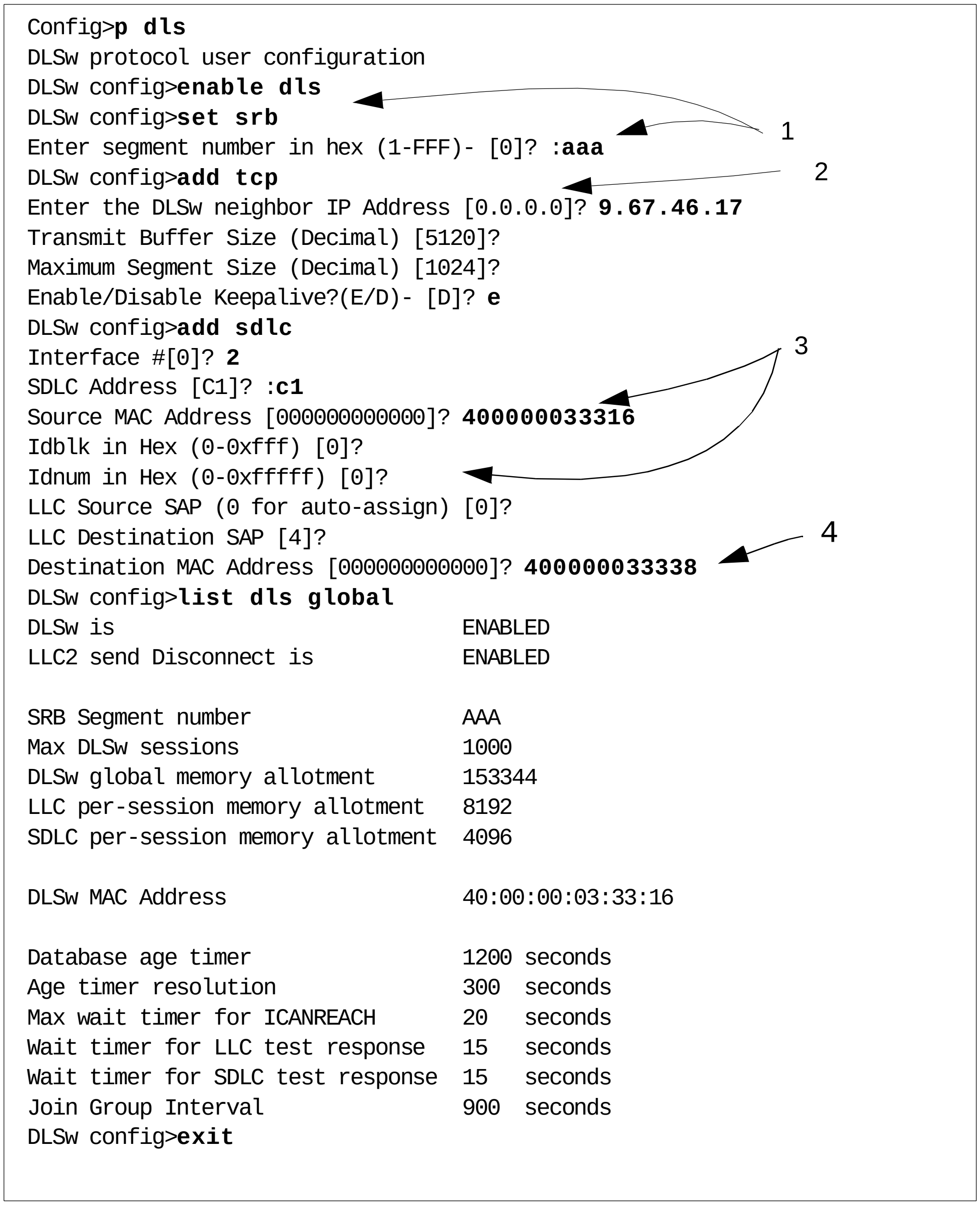


```
Config>p ip
Internet protocol user configuration
IP config>add address
Which net is this address for [0]? 1
New address [0.0.0.0]? 9.67.46.130
Address mask [255.0.0.0]? 255.255.255.240
IP config>set internal-ip-address
Internal IP address [9.67.46.162]? 9.67.46.162
IP config>set router-id
Router-ID [9.67.46.162]?
IP config>list address
IP addresses for each interface:
   intf  1   9.67.46.130      255.255.255.240  Network broadcast,    fill 0
   intf  2                                    IP disabled on this interface
Router-ID: 9.67.46.162
Internal IP address: 9.67.46.162
IP config>exit
```

Figure 181. IP Configuration

We assign the IP address 9.67.46.130 to the frame relay interface (1) and the internal IP address 9.67.46.162 to the router itself (2).

OSPF configuration follows, but this time we do not enable multicast. Lastly, we configure DLSw as in Figure 182 on page 197.



```
Config>p dls
DLSw protocol user configuration
DLSw config>enable dls
DLSw config>set srb
Enter segment number in hex (1-FFF)- [0]? :aaa
DLSw config>add tcp
Enter the DLSw neighbor IP Address [0.0.0.0]? 9.67.46.17
Transmit Buffer Size (Decimal) [5120]?
Maximum Segment Size (Decimal) [1024]?
Enable/Disable Keepalive?(E/D)- [D]? e
DLSw config>add sdlc
Interface #[0]? 2
SDLC Address [C1]? :c1
Source MAC Address [000000000000]? 400000033316
Idblk in Hex (0-0xfff) [0]?
Idnum in Hex (0-0xfffff) [0]?
LLC Source SAP (0 for auto-assign) [0]?
LLC Destination SAP [4]?
Destination MAC Address [000000000000]? 400000033338
DLSw config>list dls global
DLSw is                            ENABLED
LLC2 send Disconnect is            ENABLED

SRB Segment number                 AAA
Max DLSw sessions                  1000
DLSw global memory allotment       153344
LLC per-session memory allotment   8192
SDLC per-session memory allotment  4096

DLSw MAC Address                   40:00:00:03:33:16

Database age timer                 1200 seconds
Age timer resolution               300  seconds
Max wait timer for ICANREACH       20   seconds
Wait timer for LLC test response   15   seconds
Wait timer for SDLC test response  15   seconds
Join Group Interval                900  seconds
DLSw config>exit
```

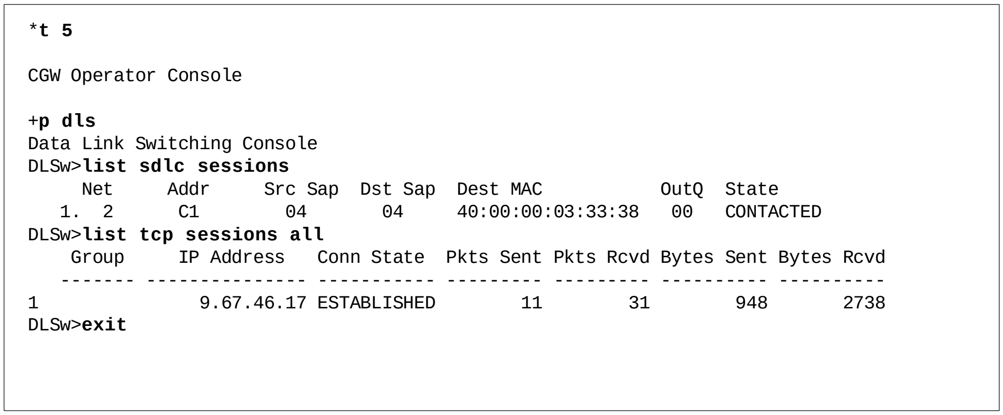
Figure 182. DLSw Configuration

We enable DLSw and set the DLSw virtual segment number to AAA (1). Next, we define an explicit connection to a partner DLSw node using the `add tcp` command (2). The partner's IP address is 9.67.46.17 (the central router).

Then we define the SDLC station (3) to DLSw, and assign it a MAC address of 400000033316 which is mapped to its SDLC station address. We do not define a node ID since this is a type 2.1 node which does not need it.

We also have to define the (virtual) MAC address of the partner SNA node (4); remember we are dealing with a leased connection where the partner node is expected to be on the other end of the link. Any frames sent out by this node will be automatically routed to the destination MAC address specified here, and thus sent to the partner router that has this MAC address defined as local.

After configuring the remainder of the routers and activating all the nodes in the network, we again displayed the DLSw status from the 2210 consoles. Figure 183 shows the status as seen from 2210A.



```
*t 5

CGW Operator Console

+p dls
Data Link Switching Console
DLSw>list sdlc sessions
     Net     Addr     Src Sap  Dst Sap  Dest MAC           OutQ  State
   1.  2      C1        04       04     40:00:00:03:33:38   00   CONTACTED
DLSw>list tcp sessions all
    Group     IP Address   Conn State  Pkts Sent Pkts Rcvd Bytes Sent Bytes Rcvd
   ------- --------------- ----------- --------- --------- ---------- ----------
1               9.67.46.17 ESTABLISHED        11        31        948       2738
DLSw>exit
```

Figure 183. DLSw Display for Frame Relay Router

Chapter 6. Telnet/3270

Strictly speaking, Telnet/3270 (TN3270) and its offshoots (Host On-Demand and Host Publisher) are not network-layer integration techniques. They take the protocol mapping a layer or two higher, going part of the way to full protocol conversion. However, they are by no means equivalent to full conversion since the Telnet/3270 clients must still have a comprehensive knowledge of parts of the SNA protocol.

By implementing Telnet/3270 you save some of the SNA protocol stack on the workstation, but you pay for it in loss of flexibility: TN3270 supports only one particular kind of SNA communication, that associated with 3270 dependent LUs (LU types 1, 2 and 3). LU types 0, 6.2 and others must still use other integration techniques. However, the 3270 protocol is the most common throughout the SNA world and TN3270 has significant benefits in the situation where you are trying to minimize the resources required on the workstations but to maximize the 3270 features available to those workstations.

TN3270 has a sister protocol, TN5250, which corresponds to the 5250 protocol used in the AS/400 world. The principles are much the same; here we discuss only the 3270 option.

6.1 TN3270 Description

Telnet/3270 is a development of the old Telnet protocol, which is described in RFCs 854 and 855. A Telnet client uses this protocol to access the resources on a Telnet server as if the client was directly attached to the server. TN3270 itself is described in RFCs 1041, 1576, 1646 and others. The TN3270 client has a TCP/IP stack with the TN3270 protocol. The TN3270 server acts as an SNA LU and translates the TCP/IP transport protocols to and from SNA protocols while leaving the data stream largely unchanged. Thus the data flows over a native SNA 3270 session between the TN3270 server and the application host.

The Telnet protocol is based on three principles:

- The Network Virtual Terminal (NVT) concept. An NVT is an imaginary device having a basic structure common to a wide range of real terminals. Each host maps its own terminal characteristics to those of an NVT, and assumes that every other host will do the same.
- A symmetric view of terminals and processes.
- Negotiation of terminal options. The principle of negotiated options is used by the Telnet protocol, because many hosts wish to provide additional services beyond those available with the NVT. The NVT has only a basic set of functions because it must cater for *all* hosts.

  Various options may be negotiated. Server and client use a set of conventions to establish the operational characteristics of their Telnet connection.

The two hosts begin by verifying their mutual understanding; once this initial negotiation is complete, they are capable of working at the minimum level implemented by the NVT. After this minimum understanding is achieved, they can negotiate additional options to extend the capabilities of the NVT to reflect more accurately the capabilities of the real hardware and the real applications in use. Because of the symmetric model used by Telnet, both the host and the client may propose additional options to be used.

The NVT has a printer (or display) and a keyboard, uses ASCII code, and emulates a simple half-duplex device operating in line-by-line mode. 3270 devices, however, behave rather differently from the old ASCII screens or printers and thus a TN3270 client and server must negotiate certain additional options if they are to succeed in communication. The most important of these are:

- Binary Transmission. 3270 uses EBCDIC data streams which are full of unprintable control characters, so TN3270 must allow any code to be transmitted.
- End of Record. 3270 data streams are block-oriented rather than line-oriented, so TN3270 must recognize the end of a block.
- Terminal Type. The client and server must negotiate not only the fact that they will exchange 3270 data streams, but also what model of 3270 screen they will emulate. Different models have different screen formats.

Once the basics have been negotiated there are still other things to get right, in particular the character translation. 3270 functions such as ATTN or SYSREQ must be mapped to particular combinations of keys on the client's keyboard.

6.1.1 TN3270 Enhancements

The basic TN3270 functions still leave some features of real 3270 SNA communication unsupported. These include:

- TN3270 does not support 328x printers.
- TN3270 cannot handle SNA BIND information. The BIND on a 3270 session includes information about screen formats and the ability to recognize the more advanced forms of the data stream.
- There is no support for the SNA positive/negative response process. This means, among other things, that end-to-end (TN3270 client to host application) response times cannot be measured.
- The 3270 ATTN and SYSREQ keys are not supported by all implementations.
- TN3270 cannot relate Telnet sessions to SNA LU names. This can make it difficult to manage a TN3270 environment.

To allow TN3270 devices to make use of all these advanced features of native 3270 communication, additions to the protocol were implemented. These additions are called TN3270 Enhancements (TN3270E), and are described in RFCs 1647 and 2355.

In order to use TN3270E, both the client and server must negotiate the TN3270 option when they are establishing their TN3270 connection. Once this has been agreed they then negotiate the TN3270E options which include:

- Device type (screen or printer, fixed or dynamic screen sizes, ability to support extended attributes)
- Printer data stream type (DSC or SCS)
- Device status information
- The passing of BIND information from server to client

- Positive/negative response exchanges (definite or exception responses, or no responses)
- Device name (and therefore the ability to influence the choice of SNA LU)

Because only TN3270E provides anything approaching the full capabilities of native SNA 3270 devices, we consider only TN3270E communication throughout the remainder of this chapter. The current IBM products that support TN3270E are as follows:

- TN3270E Client:
  - Personal Communications/3270 for OS/2
  - Personal Communications/3270 for Windows
  - Communications Server for AIX
- TN3270E Server:
  - SecureWay Communications Server for OS/390, Release 5 and above
  - Communications Server for OS/2, Version 4 Release 1 and above
  - Communications Server for Windows NT, Version 5 and above
  - The 221X router family (2216, 2212, 2210, 3746 MAE, Network Utility)

6.1.2 Positioning the TN3270E Server

The TN3270E server function has been implemented on a variety of platforms, and can be positioned almost anywhere in your network:

- On the OS/390 host, in CS for OS/390
- At the remote location, in CS/2 or CS/NT or a 2210/2212
- Somewhere in between, typically on a channel-attached 221X

The location of the server is the subject of much debate. In an ideal world the answer would depend on the distribution of your SNA applications, the relative costs of implementing TN3270E in various places, and the resilience of each option. We simply make the following observations:

- We assume that if you are implementing TN3270E then most of your workstations require access to 3270-type applications only.
- If all your 3270 applications are in the same place (same host or same Parallel Sysplex) then it probably makes sense to implement TN3270E in CS for OS/390. It can handle tens of thousands of connections, and you can achieve resilience by the judicious use of the Workload Manager and DNS-based functions of CS for OS/390. There is no requirement for SNA in the remote locations or in the backbone.
- If your 3270 applications are in the same location but not quite as closely coupled as in the first case, then TN3270E in a 2216 (or MAE, or the Network Utility which was designed for exactly this situation) may be the best solution. Optimum routing can be achieved because the traffic does not pass through intermediate hosts, and the Network Dispatcher can guarantee a measure of high availability.
- If your 3270 applications are scattered throughout your network, and even more so if there are non-3270 sessions, then it makes sense to implement TN3270E as near to the users' workstations as possible. That way optimum (now SNA) routing can be achieved, with SNA class of service between remote location and host application. If your backbone network is IP, then Enterprise Extender provides the perfect solution; any SNA traffic can be

carried, and the SNA routing can be as direct as you wish because of Enterprise Extender’s support for connection networks.

6.2 TN3270E in SecureWay Communications Server for OS/390

The Telnet server in CS for OS/390 supports line mode and basic TN3270 mode operation as well as TN3270E. It was completely rewritten in Release 5 of OS/390, and now takes full advantage of MVS multitasking capabilities. It uses the UNIX Systems Services Sockets API.

TN3270E in CS for OS/390 provides:

- Secure Sockets Support

  CS for OS/390 Release 6 introduced supports for the Secure Sockets Layer (SSL) standards. SSL provides secure data transmission between the TN3270 Server and an SSL-capable client. In an SSL session, any data on a secure port is encrypted using the SSL protocol before it is sent to the client. Data received from the client is decrypted before the data is sent to other processes (with TN3270, these other processes mean VTAM).

- Multiple Ports Support

  CS for OS/390 Release 6 also introduced the ability for the TN3270E Server to listen on multiple ports. This means you can define different security levels (basic or secure) or different configuration parameters, or both, for each port. Up to 255 ports are allowed.

- LU 0, 1, 2 and 3 Support

  LU type 0 represents a local non-SNA 3270 terminal and is often found on OS/390 hosts in its native form. Support for SNA printers means that no separate product and no separate operational considerations are required to print SNA data over the IP network.

- Unformatted System Services (USS) Support

  This provides the ability to display the VTAM USS messages on the TN3270E client. The actual USS table to be used can be selected via the configuration files. The TN3270E client can issue the VTAM logon commands in the same way as on a native SNA terminal.

- IP Address to LU Name Mapping (IP Filtering)

  This function provides the ability to select both an LU name and an application name for incoming TN3270E sessions. The selection may be made on the basis of a specific IP address, a group of IP addresses, a subnet, or the link name used to connect to the OS/390 host. The function makes the LU name and the application name predictable and controllable. In addition, CS for OS/390 Release 7 supports selection based on IP *host name*, or group of names, as well as IP address. With the increasing use of dynamic IP (where a given client is not tied to a specific IP address) this can be very beneficial in maintaining control over the mapping.

  You can also associate printer devices with display devices in the server, so the server knows where to direct a printer session that has been requested on behalf of a given client.

- Operator Console Commands.

VTAM and TCP/IP commands have been enhanced to provide additional information about the mapping between TN3270E connections and SNA sessions.

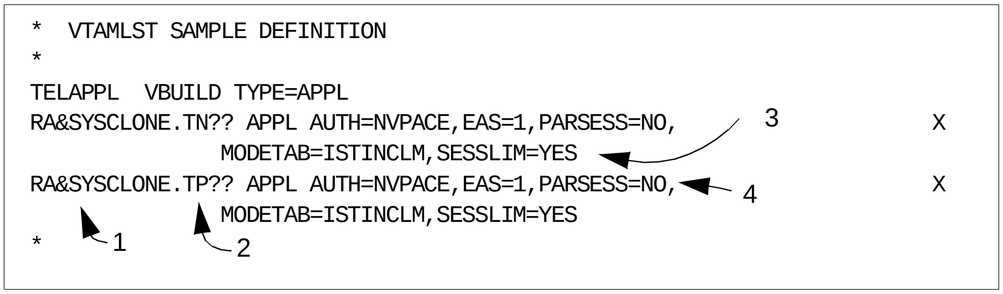
- Workload Manager Support

  CS for OS/390's TN3270E Server can register itself with the MVS workload manager as a member of a generic server group. This allows the domain name server to perform load balancing across the TN3270E instances in a sysplex.

6.2.1 Configuration and Definition

The TN3270E function in CS for OS/390 must implement SNA LUs in order to map TCP/IP connections to SNA sessions. VTAM sees these as application LUs, because they are located within the OS/390 host. In fact, the TN3270E Server looks to VTAM like a session manager application. The only difference is that the downstream terminal sessions are over TCP/IP instead of native SNA. The task of customizing the TN3270E environment, therefore, consists mainly of defining suitable applications to VTAM, the corresponding SNA LUs to TCP/IP, and the IP-to-SNA mapping to TCP/IP.

The VTAM definitions are coded in the VTAMLST data set as application major nodes. Typically they will be model (cloned) definitions, because this allows you to code them once and not have to worry about them again as the network grows. Figure 184 shows the VTAM definitions that we use for the ITSO TN3270E Server in Raleigh.



```
*  VTAMLST SAMPLE DEFINITION
*
TELAPPL  VBUILD TYPE=APPL
RA&SYSCLONE.TN?? APPL AUTH=NVPACE,EAS=1,PARSESS=NO,                    X
               MODETAB=ISTINCLM,SESSLIM=YES
RA&SYSCLONE.TP?? APPL AUTH=NVPACE,EAS=1,PARSESS=NO,                    X
               MODETAB=ISTINCLM,SESSLIM=YES
*
```

Figure 184. VTAM Definitions for TN3270E Server

The important points to note in this example are:

1. This definition is shared among all the members of our sysplex. The &SYSCLONE variable is replaced by the unique two-digit identification of the OS/390 system on which the VTAM reading it happens to be running. Thus all LU names are unique.

2. A single definition statement caters for a large number of TN3270E LUs. The question marks are wildcard characters, so that VTAM will accept any request from TCP/IP to open an ACB whose name has the form RAxxTNxx or RAxxTPxx. We have one statement for display clients and one for printers because we like to be able to distinguish them by their names.

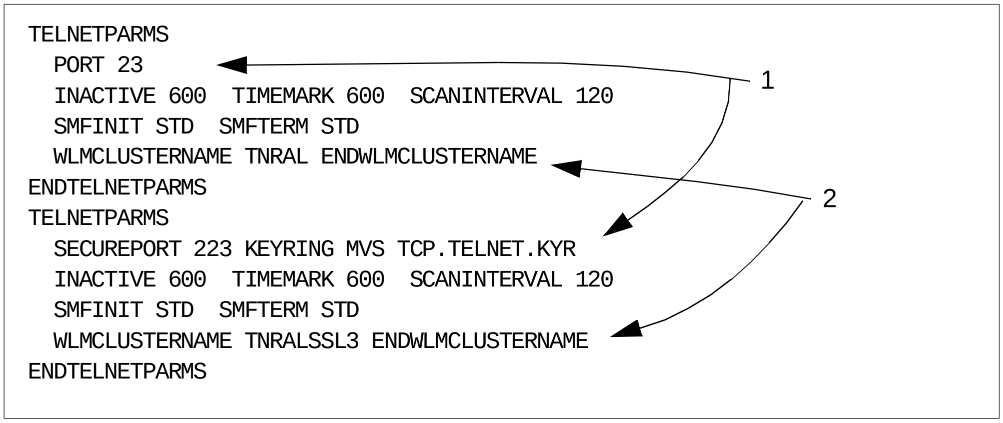
3. TN3270E accepts only one SNA session at a time on each LU, just as do real dependent LUs. To ensure that VTAM handles the LUs correctly, code SESSLIM=YES which limits VTAM to one session at a time for each LU.

EAS=1 merely reserves storage for one session; it does not limit session setup.

4. PARSESS=NO is the default (no parallel session support), and AUTH=NVPACE has no effect because these LUs always act as secondary LUs.
5. No LOGMODE entries have been coded. These are usually specified in the TCP/IP TN3270E definition file (in the TELNETDEVICE statement), and provided to VTAM on the logon request when a TN3270E client is connected.

The TN3270E Server is customized on the TCP/IP side in the profile data set. The definitions can be easily updated while TCP/IP is running, by issuing the VARY OBEYFILE command against the modified statements.

The main statements that define the TN3270E environment are the TELNETPARMS and BEGINVTAM blocks. TELNETPARMS is used to specify the Telnet ports and the environment as a whole, and BEGINVTAM is used to define the TN3270E LUs and their mapping requirements. Figure 185 shows our TELNETPARMS definition statements.



```
TELNETPARMS
  PORT 23
  INACTIVE 600  TIMEMARK 600  SCANINTERVAL 120
  SMFINIT STD  SMFTERM STD
  WLMCLUSTERNAME TNRAL ENDWLMCLUSTERNAME
ENDTELNETPARMS
TELNETPARMS
  SECUREPORT 223 KEYRING MVS TCP.TELNET.KYR
  INACTIVE 600  TIMEMARK 600  SCANINTERVAL 120
  SMFINIT STD  SMFTERM STD
  WLMCLUSTERNAME TNRALSSL3 ENDWLMCLUSTERNAME
ENDTELNETPARMS
```

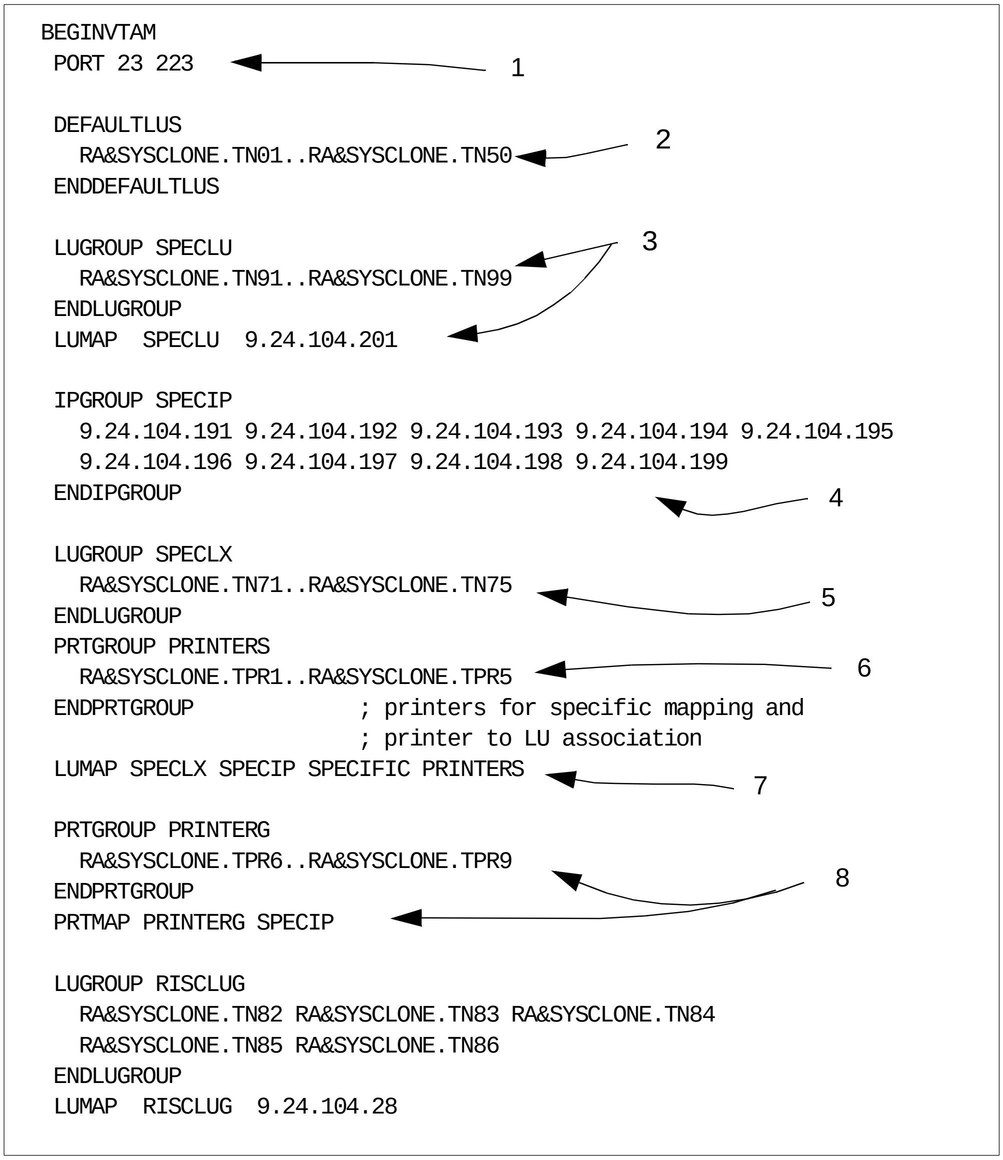
Figure 185. Telnet Definitions in TCP/IP

Note in this example:

1. Ports 23 (the default) and 223 have been reserved for use by Telnet. If you are using SSL, define the port(s) to be used via the SECUREPORT statement. You can define up to 255 listening ports for Telnet sessions.
2. The TNRAL and TNRALSSL3 parameters in the WLMCLUSTERNAME statements define the generic host names by which this Telnet server will register itself to WLM. These are the group names that clients must specify when they wish to connect to a member of the WLM-managed Telnet application group. Secure clients have a separate generic group to non-secure clients.

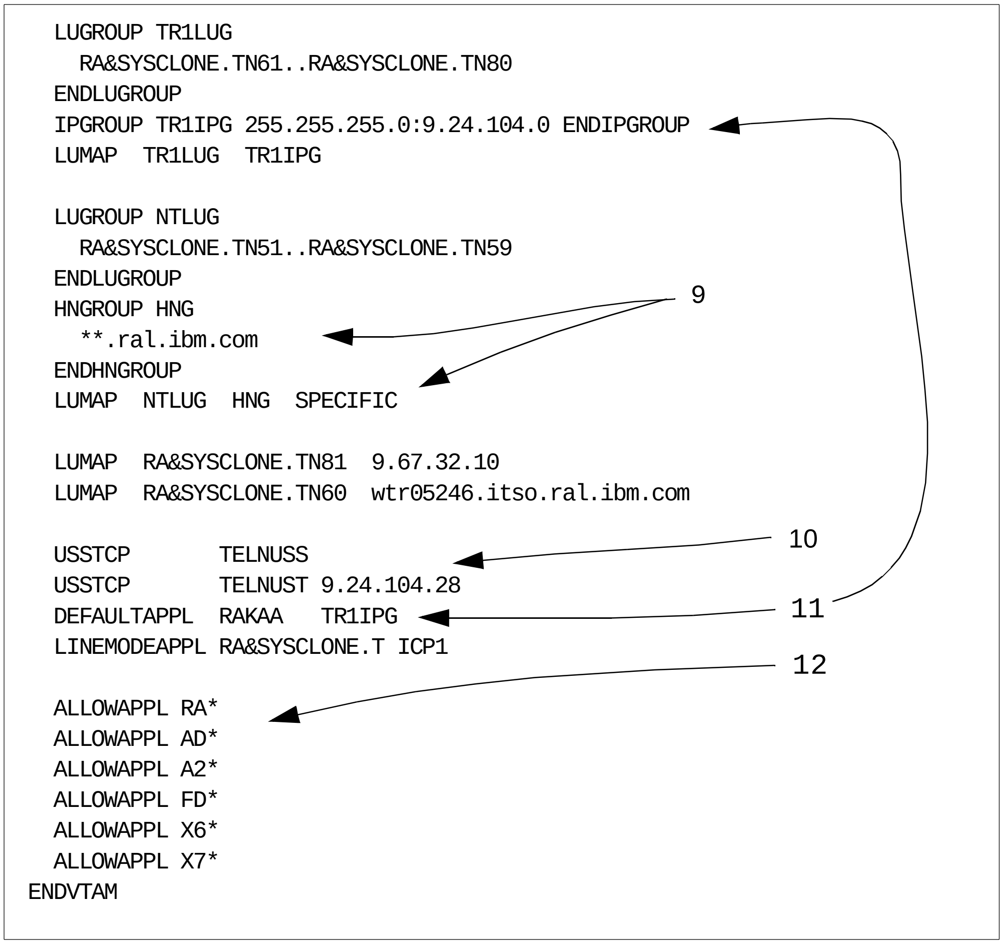
You can also specify a NOTN3270E keyword in TELNETPARMS, which tells the server never to negotiate the Telnet connection options up to TN3270E.

When you have defined the general Telnet characteristics in the TELNETPARMS statements, you can complete the work by defining the VTAM LUs and their characteristics in the BEGINVTAM statements. Our BEGINVTAM definitions from our TCP/IP profile are illustrated in Figure 186 and Figure 187 on page 206.



```
BEGINVTAM
 PORT 23 223

 DEFAULTLUS
   RA&SYSCLONE.TN01..RA&SYSCLONE.TN50
 ENDDEFAULTLUS

 LUGROUP SPECLU
   RA&SYSCLONE.TN91..RA&SYSCLONE.TN99
 ENDLUGROUP
 LUMAP  SPECLU  9.24.104.201

 IPGROUP SPECIP
   9.24.104.191 9.24.104.192 9.24.104.193 9.24.104.194 9.24.104.195
   9.24.104.196 9.24.104.197 9.24.104.198 9.24.104.199
 ENDIPGROUP

 LUGROUP SPECLX
   RA&SYSCLONE.TN71..RA&SYSCLONE.TN75
 ENDLUGROUP
 PRTGROUP PRINTERS
   RA&SYSCLONE.TPR1..RA&SYSCLONE.TPR5
 ENDPRTGROUP             ; printers for specific mapping and
                         ; printer to LU association
 LUMAP SPECLX SPECIP SPECIFIC PRINTERS

 PRTGROUP PRINTERG
   RA&SYSCLONE.TPR6..RA&SYSCLONE.TPR9
 ENDPRTGROUP
 PRTMAP PRINTERG SPECIP

 LUGROUP RISCLUG
   RA&SYSCLONE.TN82 RA&SYSCLONE.TN83 RA&SYSCLONE.TN84
   RA&SYSCLONE.TN85 RA&SYSCLONE.TN86
 ENDLUGROUP
 LUMAP  RISCLUG  9.24.104.28
```

Figure 186. BEGINVTAM Statements for TN3270E (Part 1)



```
  LUGROUP TR1LUG
    RA&SYSCLONE.TN61..RA&SYSCLONE.TN80
  ENDLUGROUP
  IPGROUP TR1IPG 255.255.255.0:9.24.104.0 ENDIPGROUP
  LUMAP  TR1LUG  TR1IPG

  LUGROUP NTLUG
    RA&SYSCLONE.TN51..RA&SYSCLONE.TN59
  ENDLUGROUP
  HNGROUP HNG
    **.ral.ibm.com
  ENDHNGROUP
  LUMAP  NTLUG  HNG  SPECIFIC

  LUMAP  RA&SYSCLONE.TN81  9.67.32.10
  LUMAP  RA&SYSCLONE.TN60  wtr05246.itso.ral.ibm.com

  USSTCP       TELNUSS
  USSTCP       TELNUST 9.24.104.28
  DEFAULTAPPL  RAKAA   TR1IPG
  LINEMODEAPPL RA&SYSCLONE.T ICP1

  ALLOWAPPL RA*
  ALLOWAPPL AD*
  ALLOWAPPL A2*
  ALLOWAPPL FD*
  ALLOWAPPL X6*
  ALLOWAPPL X7*
ENDVTAM
```

Figure 187. BEGINVTAM Statements for TN3270E (Part 2)

The principle to remember when defining these is that the SNA LU name is selected *after* the target application is known. The target application may be determined from these definitions, or from what the terminal user enters when presented with the sign-on (USS 10) screen. TN3270E can display a USS 10 screen *before* allocating an SNA LU. When the application name is known, the LU name is then determined from the client's IP address, the status of the target application (allowed, restricted, default) and the appropriate LUMAP statements. Please see *OS/390 eNetwork Communications Server: IP Configuration,* SC31-8513, for a full description of how TN3270E interacts with an IP client.

In this set of definitions:

1. The PORT statement defines the TN3270 ports to which this BEGINVTAM set of statements will apply.
2. The DEFAULTLUS statement specifies a range of SNA LUs (ACB names, in fact) that TN3270E will use to represent a client connection when the selected application or LUMAP definitions do not call for a specific set of LUs. Note the use of system symbolic definitions here, just as in the VTAM definitions (Figure 184 on page 203). The names defined here must correspond with those in the VTAM application major node. Here we select RAxxTN01 to RAxxTN50 to be used as default LUs, which leaves plenty of names for specific mappings (there is no need for the wildcard characters to be numeric).

3. Next, we define a pool of LUs called SPECLU. SPECLU now comprises the LUs named RAxxTN91 to RAxxTN99. We associate the IP host 9.24.104.201 with this LU pool. When this IP host connects to the server as a TN3270E client it will be assigned to one of the LUs in this pool, provided that the application selected is permitted to this client.

4. We define a group of IP clients called SPECIP, which comprises nine IP addresses as shown.

5. We define another pool of five LUs called SPECLX.

6. We define a pool of printer LUs called PRINTERS which contains the LUs names RAxxTPR1 to RAxxTPR5.

7. Then we tie the last three definitions together with another LUMAP. We associate the group of IP clients SPECIP with the pool of LUs SPECLX, so that if one of these clients requests a TN3270 connection it will be assigned a VTAM LU in this pool. This is a SPECIFIC mapping definition, so it is used when the client requests a specific device name. If this fails, an equivalent GENERIC (the default value) mapping definition is used if available.

   The same statement also maps the printer pool to the terminal pool.

8. A printer client group can also be assigned to a printer LU pool without any association with displays. Here we define a printer LU pool called PRINTERG and map SPECIP to it. When a printer in the SPECIP group connects to TN3270E without requesting a specific LU name it will be assigned to one of the LUs in PRINTERG.

9. You can also define a group of clients by IP host name. Here we create a group called HNG which comprises all hosts in the domain ral.ibm.com. We associate it with the LU pool NTLUG, provided the client specifies the LU name it wants.

10. The USSTCP statement associates a TN3270-specific USS message table with some or all of your TN3270E clients. Here, any client that does not have a default target application defined will be presented with the USS screen from the TELNUSS table. The exception is 9.24.104.28, which will receive the screen from the TELNUST table.

11. Clients in the group TR1IPG will not receive any USS messages, but will be logged straight on to the application RAKAA. Note the definition of TR1IPG, where we select the *whole* of the 9.24.104 subnetwork by prefixing its address with its subnet mask.

12. Finally, we define which SNA applications will be permitted access from the TN3270E clients. Once again, wildcards are used to specify a range of applications. The ALLOWAPPL and RESTRICTAPPL statements are used to control which users are allowed to log on to which applications using which LUs. Restricted applications are associated with a list of authorized users who must identify themselves by passwords.

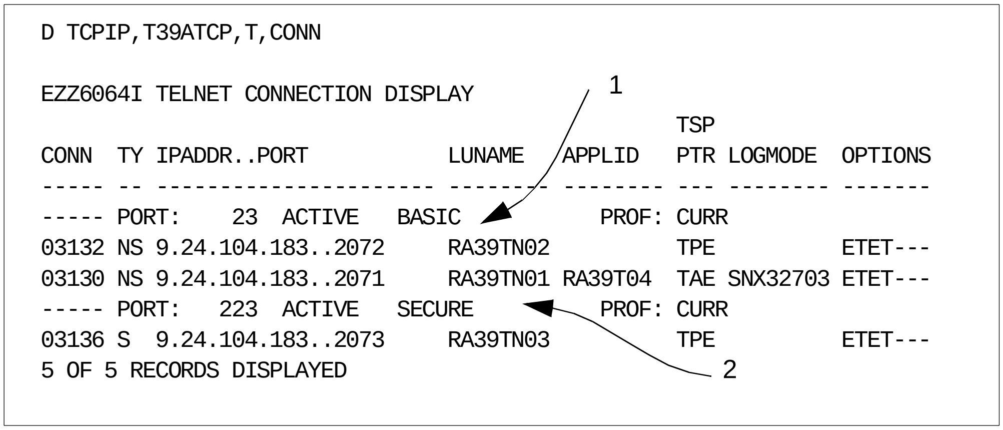
Not present in our definitions, but of some interest, is the TELNETDEVICE statement. This associates an SNA mode name with a device type (as defined in the TN3270 standards). The default is for TN3270E to use LU type 0 (non-SNA) modes.

For a comprehensive description of TN3270E in CS for OS/390, please see *OS/390 eNetwork Communications Server TCP/IP Implementation Guide Volume 1,* SG24-5227.

6.2.2 TN3270E Displays

Both VTAM and TCP/IP provide extra displays to enable you to understand how TN3270E is working. The potential complexity of the profile definitions has resulted in the availability of several commands to summarize them. It is possible to have a number of profiles, some of which override some parameters in some of the others, but none of which is completely redundant. You can display CS for OS/390's current view of, for example, the relationship between a given client and the potential partner applications.

Once TN3270E is running, other commands are available to display the status. As an example, Figure 188 shows the Telnet connection display taken on one of our TN3270E servers.



```
D TCPIP,T39ATCP,T,CONN

EZZ6064I TELNET CONNECTION DISPLAY
                                                 TSP
CONN  TY IPADDR..PORT          LUNAME   APPLID   PTR LOGMODE  OPTIONS
----- -- --------------------- -------- -------- --- -------- -------
----- PORT:    23  ACTIVE   BASIC            PROF: CURR
03132 NS 9.24.104.183..2072     RA39TN02          TPE          ETET---
03130 NS 9.24.104.183..2071     RA39TN01 RA39T04  TAE SNX32703 ETET---
----- PORT:   223  ACTIVE   SECURE           PROF: CURR
03136 S  9.24.104.183..2073     RA39TN03          TPE          ETET---
5 OF 5 RECORDS DISPLAYED
```

Figure 188. Telnet Connections Summary

This display shows a high-level view of what connections exist and what they are being used for:

1. Port 23 is being used for basic (non-secure) Telnet communication. It has two connections:
   - Number 3132 has client address 9.24.104.183 with remote port 2072. It is using LU RA39TN02 but is not presently connected to an application.
   - Number 3130 has the same client with a different remote port number (another emulator session). This one is using LU RA39TN01 and is in session with TSO (RA39T04) using mode SNX32703.
2. Port 223 has been defined as an SSL port. It has the same client connected but not in session.

To display the details of one of these connections, we enter the same command plus the connection identifier, as shown in Figure 189 on page 209.

```
D TCPIP,T39ATCP,T,CONN,CONN=3130

EZZ6065I TELNET CONNECTION DISPLAY
  CONN: 03130    IPADDR..PORT: 9.24.104.183..2071
  HOSTNAME: NO HOSTNAME
  CONNECTED: 17:20:56  02/10/1999  STATUS: SESSION ACTIVE
  PORT:    23  ACTIVE   BASIC           ACCESS: NON-SECURE
  PROFILE ID: CURR   PROFILE OPTIONS: --L-M--W--H--
  PROTOCOL: TN3270E  LOGMODE: SNX32703 DEVICETYPE: IBM-3278-3-E
    OPTIONS: ETET---  3270E FUNCTIONS: BSR--
  USSTABLE: TELNUST     HN/IPGROUP: EXACT IP ADDR
  LUNAME: RA39TN01      TYPE: TERMINAL
    LU/PRTGROUP: LU1         HN/IPGROUP: IP1
  APPLID: RA39T04
    DEFAULTAPPL HN/IPGROUP: **N/A**
    RESTRICTAPPL USERID: **N/A**
13 OF 13 RECORDS DISPLAYED
```

Figure 189. TN3270E Connection Details

This one gives you some additional history, such as which group the terminal belongs to, which TN3270E options were negotiated, and which USS table is in use.

On the VTAM side, a display of the application major node (Figure 190) shows which VTAM LUs are active (and therefore have TN3270E clients connected).

```
D NET,ID=TELAPPL,E

IST097I DISPLAY ACCEPTED
IST075I NAME = TELAPPL, TYPE = APPL SEGMENT
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST360I APPLICATIONS:
IST080I RA39TN?? CONCT      RA39TN01 ACT/S      RA39TN02 ACT/S
IST080I RA39TN04 ACT/S      RA39TP?? CONCT
IST314I END
```

Figure 190. TN3270E Application Major Node Display

This display shows the LUs defined in the model application major node shown in Figure 184 on page 203. There are three real LUs active. If we display one of them, RA39TN01, we see the output in Figure 191 on page 210.

```
D NET,ID=RA39TN01,E

IST097I DISPLAY ACCEPTED
IST075I NAME = USIBMRA.RA39TN01, TYPE = DYNAMIC APPL
IST486I STATUS= ACT/S, DESIRED STATE= ACTIV
IST1447I REGISTRATION TYPE = CDSERVR
IST1629I MODSRCH = NEVER
IST977I MDLTAB=***NA*** ASLTAB=***NA***
IST861I MODETAB=ISTINCLM USSTAB=***NA*** LOGTAB=***NA***
IST934I DLOGMOD=***NA*** USS LANGTAB=***NA***
IST1632I VPACING =  7
IST597I CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT 00000001
IST231I APPL MAJOR NODE = TELAPPL
IST1425I DEFINED USING MODEL RA39TN??
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I STATE TRACE = OFF
IST271I JOBNAME = T39ATCP, STEPNAME = T39ATCP, DSPNAME = ISTAC614
IST228I ENCRYPTION = OPTIONAL
IST1563I CKEYNAME = RA39TN01 CKEY = PRIMARY CERTIFY = NO
IST1552I MAC = NONE MACTYPE = NONE
IST1050I MAXIMUM COMPRESSION LEVEL - INPUT = 0, OUTPUT = 0
IST1633I ASRCVLM = 1000000                                       1
IST1634I DATA SPACE USAGE: CURRENT = 0 MAXIMUM = 0
IST1669I IPADDR..PORT 9.24.104.183..2079
IST171I ACTIVE SESSIONS = 0000000001, SESSION REQUESTS = 0000000000
IST206I SESSIONS:
IST634I NAME     STATUS         SID          SEND RECV VR TP NETID
IST635I RA39T08  ACTIV-P    C707961CAB61DE16 0000 0002       USIBMRA
IST314I END
```

Figure 191. TN3270E LU Details

There is a new line in this display (1), which tells you the IP address and the remote port number of the TN3270E client which is using this LU. You can get this information in reverse by issuing the DISPLAY ...IDTYPE=IPADDR command as shown in Figure 192.

```
D NET,IDTYPE=IPADDR,ID=9.24.104.183

IST097I DISPLAY ACCEPTED
IST1668I LUNAME            IPADDR..PORT
IST1670I USIBMRA.RA39TN01   9.24.104.183..2079
IST1670I USIBMRA.RA39TN02   9.24.104.183..2080
IST1670I USIBMRA.RA39TN04   9.24.104.183..2082
IST314I END
```

Figure 192. Display LU Details from IP Address

This display gives you the LU names being used by the TN3270E client with the stated IP address. If there is only one connection you will get the full details as shown in Figure 191.

6.3 TN3270E in Communications Server for OS/2

The TN3270E server function is available on Communications Server for OS/2 itself, but not on the Access Feature that is a subset of CS/2.

Since CS/2 supports only LU type 2 displays (not LU 0), the TN3270E implementation does the same. Major functions in TN3270 on CS/2 are:

- LU types 1, 2 and 3 using the RFC 1647 standards.
- SNA positive and negative responses.
- Handling of attention and system request keys.
- Grouping of LUs into classes, which simplifies user access and management of users. Both printer and display LUs can be defined as implicit (clients are allocated to any available LU in the pool) or explicit (clients require a specific device name).
- Association of printer LUs with display LUs.

TN3270E communication using SSL is not supported, nor is filtering (the association of particular clients to particular LUs). Both of these options are planned for implementation in the next release due in Spring 1999.

Setting up the TN3270E Server is less complex than on CS for OS/390. The owning VTAM is not in the same machine as the TN3270E server function, and the filtering and SSL options are (for now) absent.

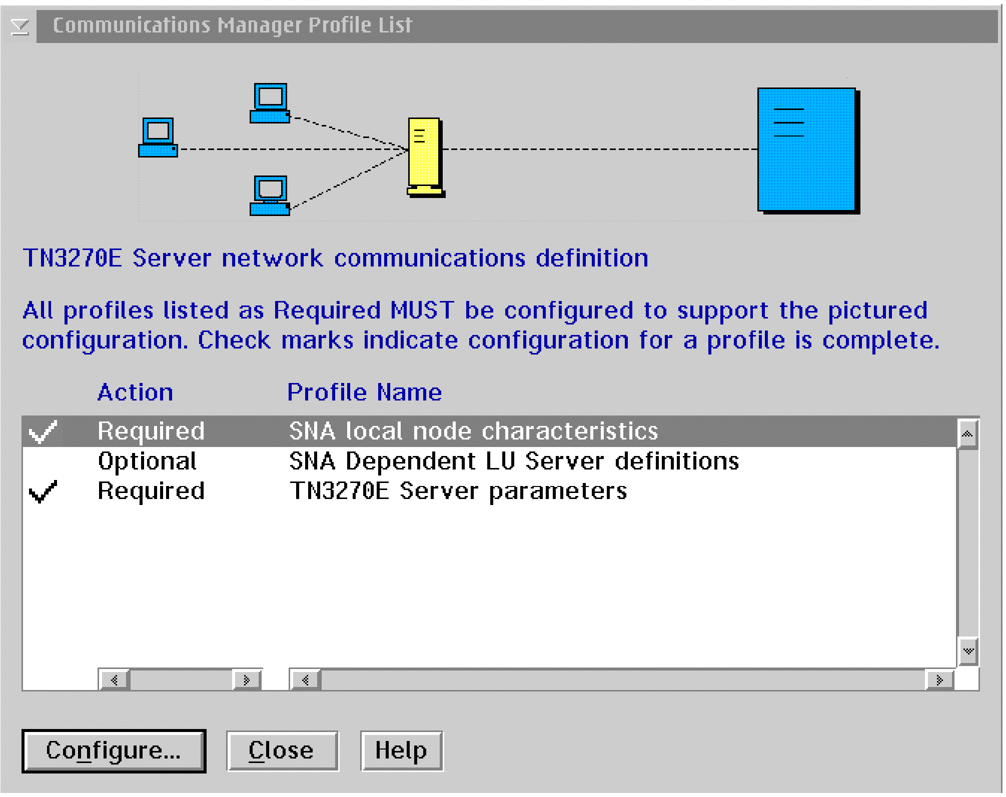
6.3.1 Configuration and Definition

Once you have configured your CS/2 machine for native SNA operation, the additional work required for TN3270E is not great. In essence you have to select one or more host links for TN3270E to use, define some LUs on those links, and allocate the LUs to TN3270E sessions. The host links used by TN3270E can be native or DLUR connections; the requirement is that SSCP sessions are supported over those connections.

There are two ways to invoke the TN3270E configuration from the Communications Manager Configuration Definition panel:

1. Click **TN3270E** and **Configure**.
2. Click **Options** and **Configure any profile or feature**.

In either case, the item on the Profile List you are looking for is called TN3270E Server parameters. Figure 193 on page 212 shows the Profile List when we selected the **TN3270E/Configure** option.



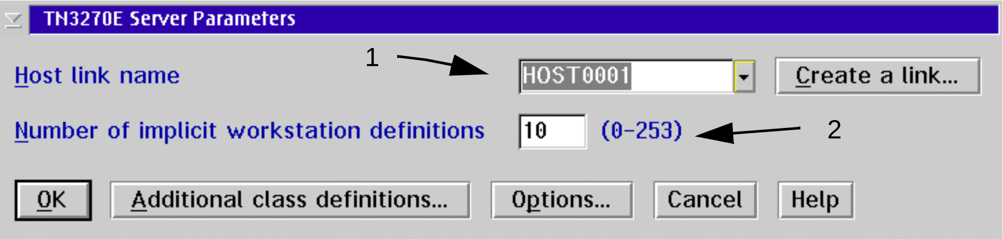
TN3270E Server network communications definition

All profiles listed as Required MUST be configured to support the pictured configuration. Check marks indicate configuration for a profile is complete.

	Action	Profile Name
✓	Required	SNA local node characteristics
	Optional	SNA Dependent LU Server definitions
✓	Required	TN3270E Server parameters

Figure 193. CS/2 Profile List for TN3270E

If you now select **TN3270E Server parameters** you are presented with the screen shown in Figure 194.



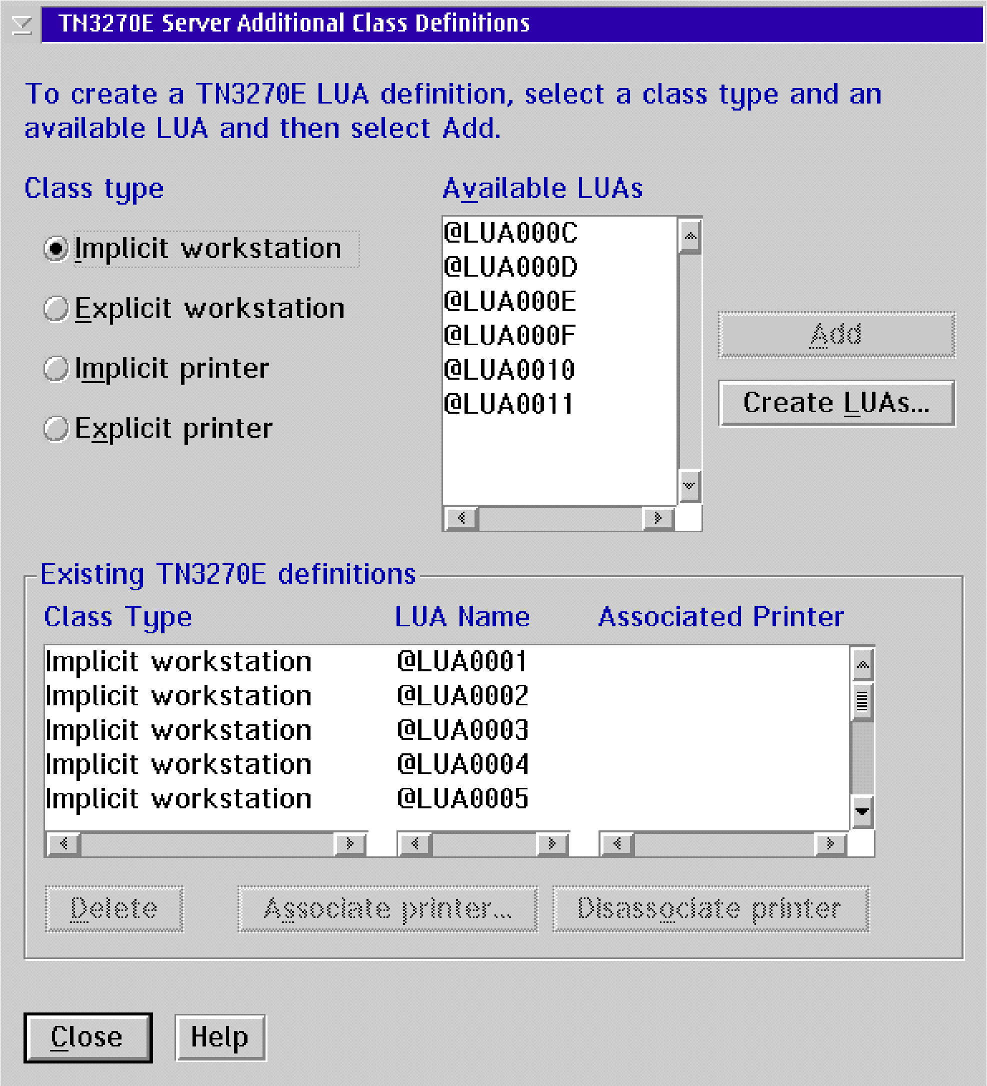
Host link name: HOST0001

Number of implicit workstation definitions: 10 (0-253)

Figure 194. TN3270 Server Parameters

This panel asks you for the name of the host connection (1). It presents you with a list of defined host or DLUR links and invites you to create a new one if you so wish. The **number of implicit workstations** field (2) defines the number of LU definitions that will be automatically created for TN3270 use on the chosen host link. These LUs will be used as a pool of default LUs for any TN3270E client that connects to this server without specifying an LU name. The LU name here is that known locally to CS/2; it may or may not correspond to the actual VTAM LU name depending on your VTAM definitions.

To create default printer LUs, or explicit display and printer LUs, click **Additional class definitions** to see the panel in Figure 195 on page 213.



TN3270E Server Additional Class Definitions

To create a TN3270E LUA definition, select a class type and an available LUA and then select Add.

Class type

● Implicit workstation

○ Explicit workstation

○ Implicit printer

○ Explicit printer

Available LUAs

@LUA000C
@LUA000D
@LUA000E
@LUA000F
@LUA0010
@LUA0011

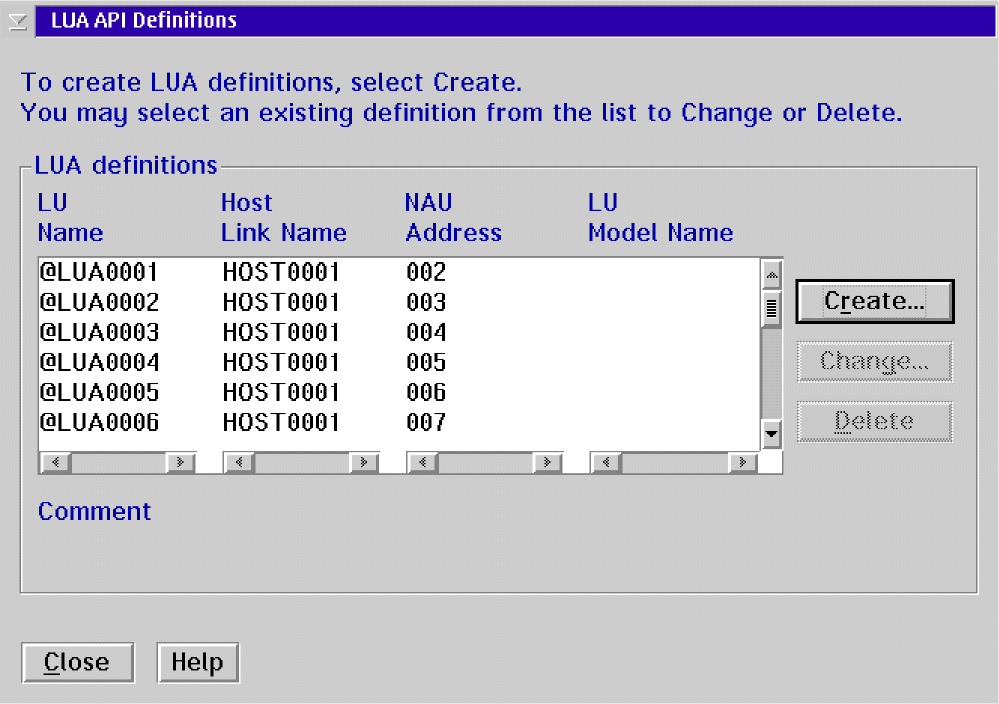
Existing TN3270E definitions

Class Type	LUA Name	Associated Printer
Implicit workstation	@LUA0001	
Implicit workstation	@LUA0002	
Implicit workstation	@LUA0003	
Implicit workstation	@LUA0004	
Implicit workstation	@LUA0005	

Figure 195. TN3270 Additional Class Definitions

The **Available LUAs** list is blank when you first see this panel; if you want to define printers or explicit LUs you must either delete some implicit workstations or create some new LUs using the **Create LUAs** button. The **Create LUAs** button leads to the panel shown in Figure 196 on page 214. This is the same panel as you get when you ask to create LUs on any host connection, regardless of whether they are TN3270 LUs or native LUs for use by PComm or similar products.

You can associate a printer LU with a display LU by selecting the display LU in the lower scroll bar and clicking **Associate printer**. This gives you a list of defined printers from which you choose.



LUA API Definitions

To create LUA definitions, select Create.
You may select an existing definition from the list to Change or Delete.

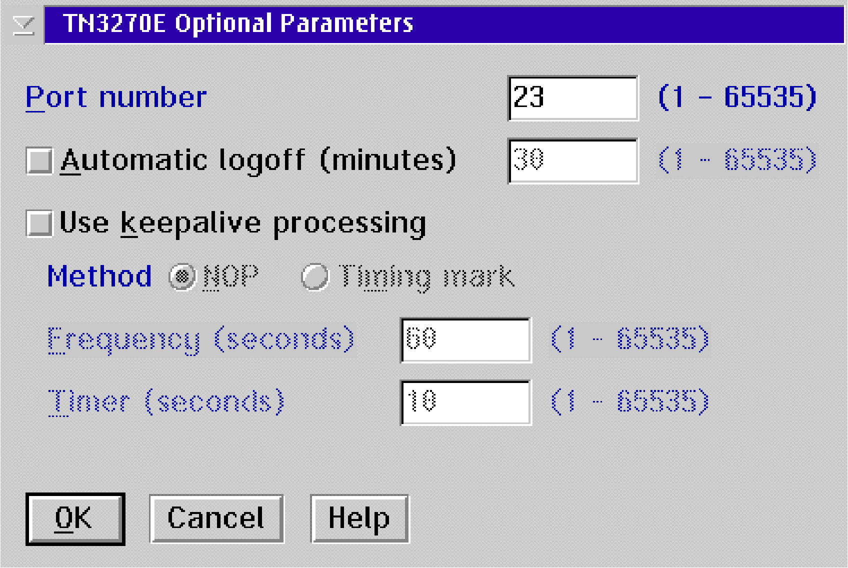
LUA definitions

LU Name	Host Link Name	NAU Address	LU Model Name
@LUA0001	HOST0001	002	
@LUA0002	HOST0001	003	
@LUA0003	HOST0001	004	
@LUA0004	HOST0001	005	
@LUA0005	HOST0001	006	
@LUA0006	HOST0001	007	

Comment

Figure 196. LU Creation Panel

The only other options to consider when defining TN3270E are those available from the **Options** button on the Server Parameters panel (Figure 194 on page 212). Figure 197 is displayed when you select **Options**.



TN3270E Optional Parameters

Port number 23 (1 - 65535)

☐ Automatic logoff (minutes) 30 (1 - 65535)

☐ Use keepalive processing

Method ● NOP ○ Timing mark

Frequency (seconds) 60 (1 - 65535)

Timer (seconds) 10 (1 - 65535)

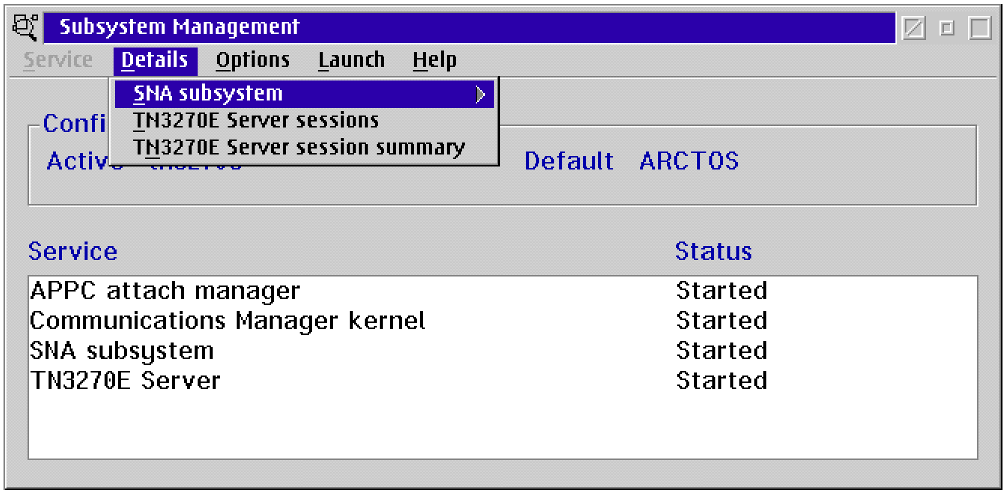
Figure 197. TN3270 Server Options

Here you can define the Telnet port number (default 23, the well-known port for Telnet) and how long the server is to wait with no traffic on a connection before terminating the connection. You can also ask the server to send regular keepalive messages to the client to check whether the connection is still up.

Be careful in OS/2 if you let the TN3270E port default to 23. The base Telnet server is often automatically started on port 23 using the INETD daemon; if this is so TN3270E will not work and your TN3270E clients will be connected in line-by-line mode. If you need to use base Telnet as well as TN3270E you must configure one of them, and all its clients, to use another port.

6.3.2 Operation

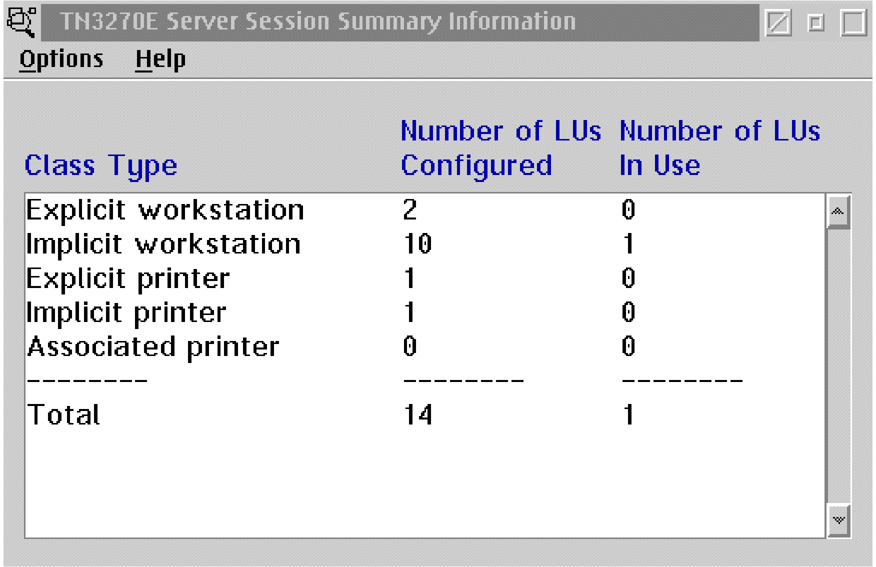
Once the TN3270E server function is started, it appears as an extra subsystem on the Subsystem Management panel, as seen in Figure 198. The Details pull-down now includes two new items, a session details and a session summary option for TN3270E.



Service	Status
APPC attach manager	Started
Communications Manager kernel	Started
SNA subsystem	Started
TN3270E Server	Started

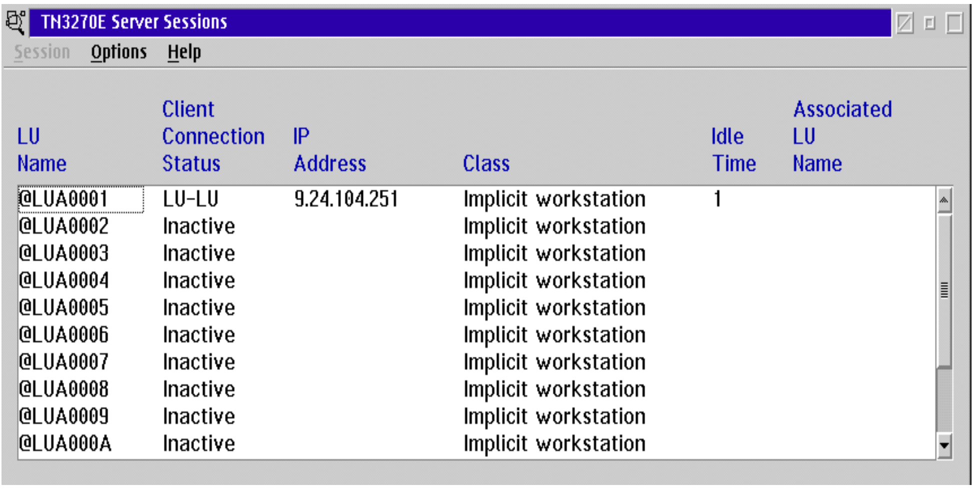
Figure 198. Subsystem Management Panel with TN3270E

Selecting TN3270E Server session summary shows the panel in Figure 199, whereas the TN3270E Server sessions option shows you Figure 200 on page 216.



Class Type	Number of LUs Configured	Number of LUs In Use
Explicit workstation	2	0
Implicit workstation	10	1
Explicit printer	1	0
Implicit printer	1	0
Associated printer	0	0
--------	--------	--------
Total	14	1

Figure 199. TN3270E Session Summary



LU Name	Client Connection Status	IP Address	Class	Idle Time	Associated LU Name
@LUA0001	LU-LU	9.24.104.251	Implicit workstation	1	
@LUA0002	Inactive		Implicit workstation		
@LUA0003	Inactive		Implicit workstation		
@LUA0004	Inactive		Implicit workstation		
@LUA0005	Inactive		Implicit workstation		
@LUA0006	Inactive		Implicit workstation		
@LUA0007	Inactive		Implicit workstation		
@LUA0008	Inactive		Implicit workstation		
@LUA0009	Inactive		Implicit workstation		
@LUA000A	Inactive		Implicit workstation		

Figure 200. TN3270E Server Session Details

This panel demonstrates that the client 9.24.104.251 (actually PComm/2 Version 4 Release 2) is connected to the LU with a local name of @LUA0001. It is in session with a host application; if it were not the connection status would be SSCP-LU.

6.4 TN3270E in Communications Server for Windows NT

CS/NT supports the following options in its TN3270E function:

- LU types 1, 2 and 3 using the RFC 1647 standards.
- SNA positive and negative responses.
- Handling of attention and system request keys.
- Categorization of LUs into classes, as with CS/2.
- Association of printer LUs with display LUs.
- Secure Sockets Layer support.
- IP filtering, that is the control of access to host LUs depending on the client IP addresses or host names.
- Server load balancing. This requires corresponding support in the client workstations, and allows TN3270E connections to be spread over multiple servers. PComm Version 4 Release 3 is an example of a client that supports this function.

The upstream host connection can be any suitable connection that supports dependent LUs with their SSCP sessions (native or DLUR).

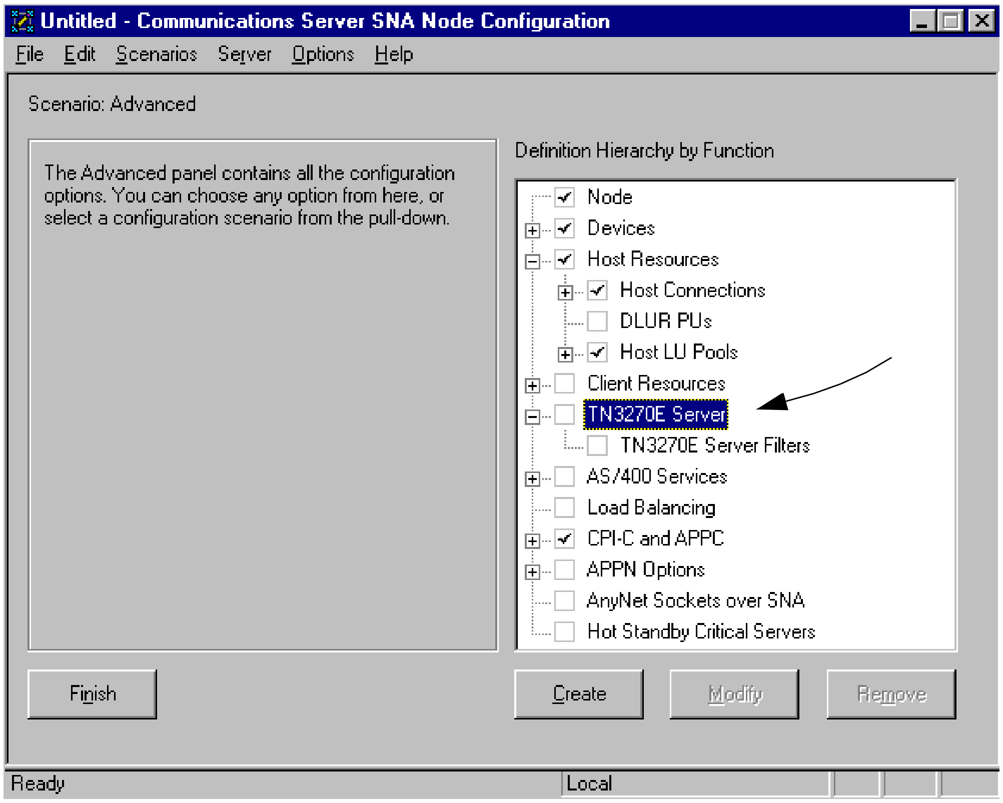
For a comprehensive guide to the TN3270E functions in CS/NT, please refer to *IBM Communications Server for Windows NT Version 5.0,* SG24-2099, and *IBM eNetwork Communications Server for Windows NT Version 6.0 Enhancements,* SG24-5232.

6.4.1 Configuration and Definitions

As with all CS/NT configurations, you can select either **TN3270E Server** from the list of basic configuration scenarios, or the **Advanced** check box to receive a menu of all the possible configuration steps. The principle is the same as in CS/2, but the details are different.

If you select **TN3270E Server**, a wizard appears which guides you through some basic steps to create a working TN3270E environment. If you want to review all the options or to configure some of the more advanced ones, you will need to invoke the **Advanced** configuration afterwards. Our examples are based on the **Advanced** version as we wish to show all the options available.

As with CS/2, we configure the native SNA setup first and then the TN3270E additions. The TN3270E Server option is now the one you want, as shown in Figure 201.



Untitled - Communications Server SNA Node Configuration

Scenario: Advanced

The Advanced panel contains all the configuration options. You can choose any option from here, or select a configuration scenario from the pull-down.

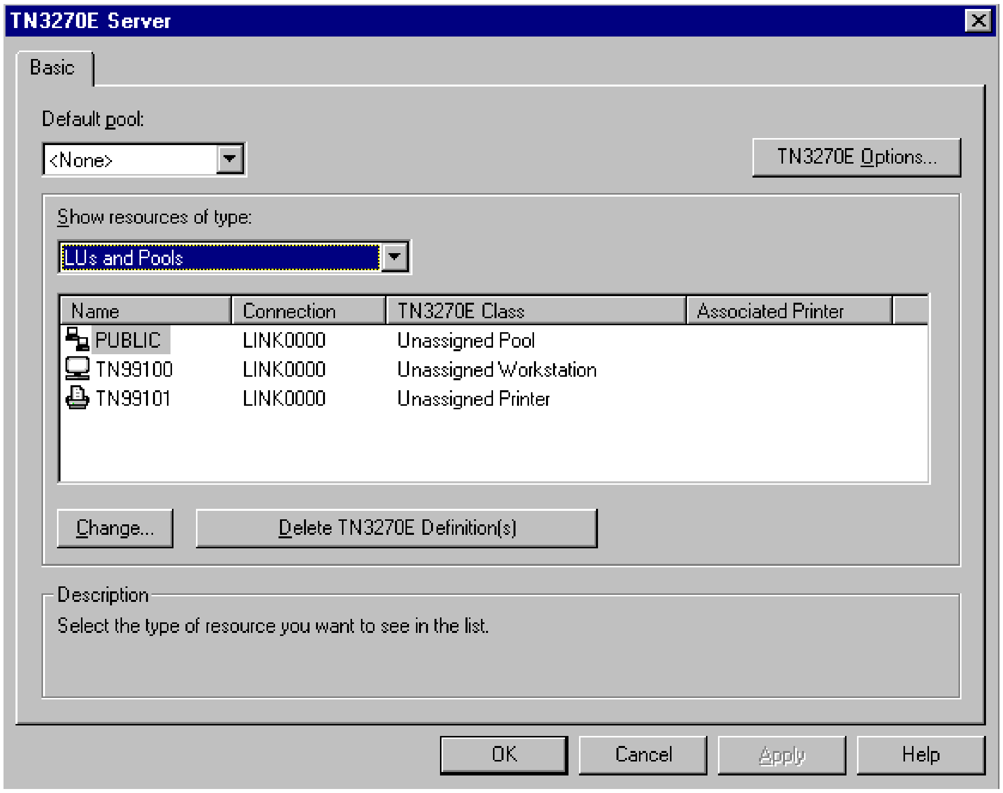
Definition Hierarchy by Function

- ☑ Node
- ☑ Devices
- ☑ Host Resources
  - ☑ Host Connections
  - □ DLUR PUs
  - ☑ Host LU Pools
- □ Client Resources
- □ TN3270E Server
  - □ TN3270E Server Filters
- □ AS/400 Services
- □ Load Balancing
- ☑ CPI-C and APPC
- □ APPN Options
- □ AnyNet Sockets over SNA
- □ Hot Standby Critical Servers

Ready | Local

Figure 201. CS/NT Configuration with TN3270E Option

The **TN3270E Server** selection expands into a main option and a **Filters** option. We select the main option and see Figure 202 on page 218. Here you define which host LUs are associated with TN3270E, and in what way.



Default pool: <None>

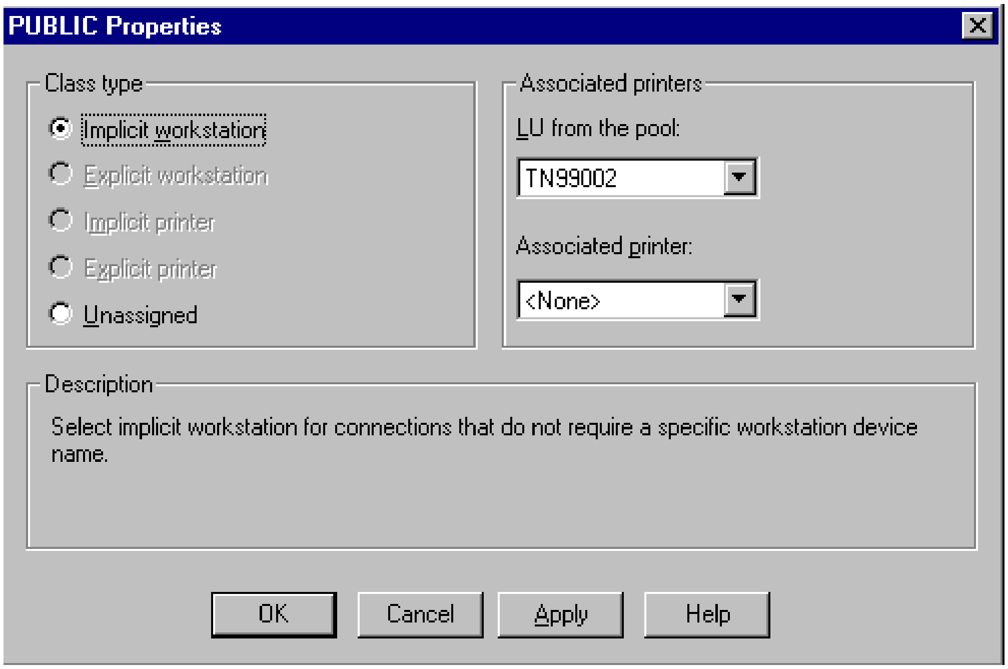
Show resources of type: LUs and Pools

Name	Connection	TN3270E Class	Associated Printer
PUBLIC	LINK0000	Unassigned Pool	
TN99100	LINK0000	Unassigned Workstation	
TN99101	LINK0000	Unassigned Printer	

Description
Select the type of resource you want to see in the list.

Figure 202. TN3270E Server Main Panel

In this example we have defined a host connection called LINK0000, on which are some twenty dependent LUs. We have a pool of display LUs called PUBLIC, a single display LU called TN99100, and a single printer LU called TN99101. The scroll bar labeled **Show resources of type** lets you see what kind of LUs and pools are available for use by TN3270E, and which of them have already been allocated for use by TN3270E. To assign a pool or an individual LU to TN3270E, select the appropriate LU or pool from the window and click **Change**. You will then see the Properties panel shown in Figure 203 on page 219.


PUBLIC Properties

Class type

● Implicit workstation
○ Explicit workstation
○ Implicit printer
○ Explicit printer
○ Unassigned

Associated printers

LU from the pool: TN99002

Associated printer: <None>

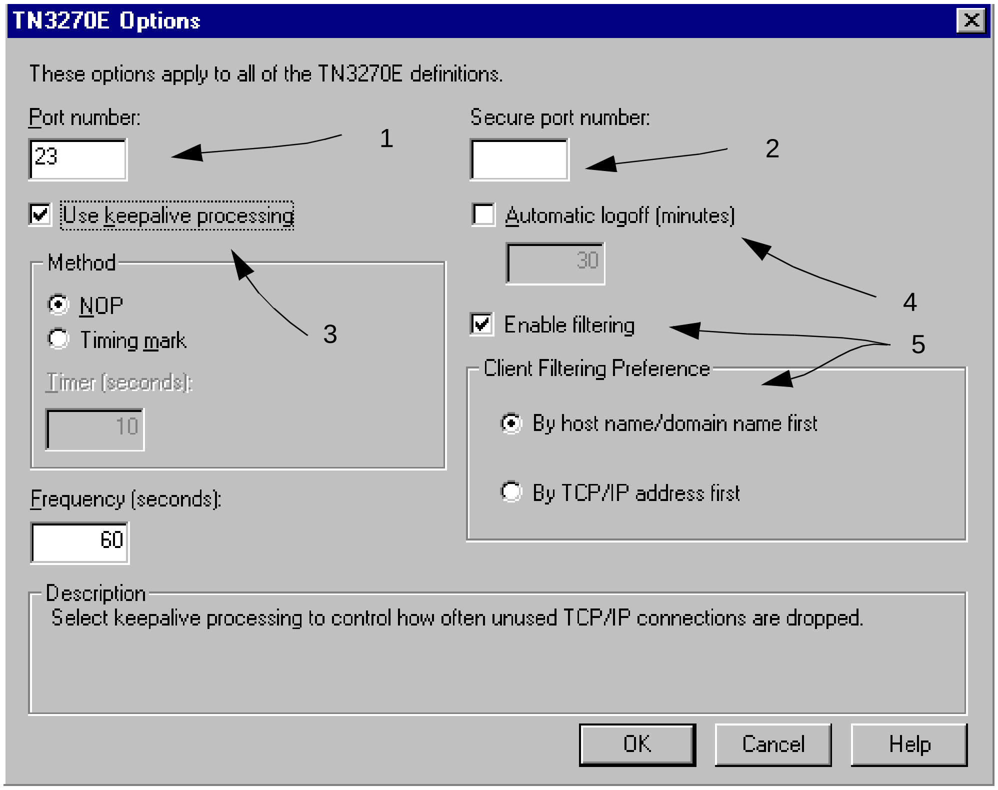
Description

Select implicit workstation for connections that do not require a specific workstation device name.

Figure 203. TN3270E LU Assignments

This panel lets you associate LU pools with implicit clients (of the same type as the pool LUs, display or printer) and LUs with explicit clients (again, the type must match). Here we have selected the **PUBLIC** pool so we are allowed only two choices: **Unassigned** (not for use by TN3270E) or **Implicit Workstation** (for use when the client display does not specify a device name). You can also associate a printer LU with a display LU. There is no requirement for both printer LU and display LU to be from a pool, or both individually defined.

When we are happy with our LU assignments, we click **OK** and return to the main TN3270E panel (Figure 202 on page 218). We then select **TN3270E Options** and see Figure 204 on page 220.



TN3270E Options

These options apply to all of the TN3270E definitions.

Port number: 23

Secure port number:

☑ Use keepalive processing

Method

● NOP

○ Timing mark

Timer (seconds): 10

Frequency (seconds): 60

□ Automatic logoff (minutes) 30

☑ Enable filtering

Client Filtering Preference

● By host name/domain name first

○ By TCP/IP address first

Description

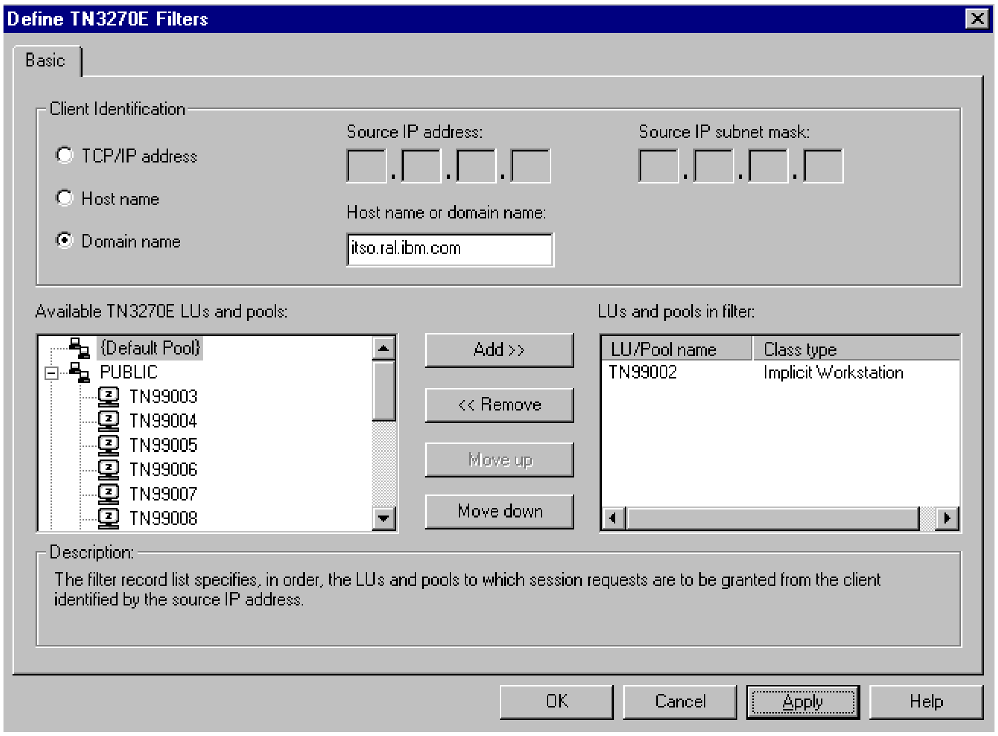
Select keepalive processing to control how often unused TCP/IP connections are dropped.

Figure 204. TN3270E Options Panel

The definitions available on this panel are:

1. The IP port number to be used for non-SSL TN3270 communication. As always, it defaults to 23 and you must ensure that it is not also in use for line-mode Telnet connections.
2. The IP port number to be used for SSL connections.
3. Whether to send regular keepalive packets to the client.
4. Whether to terminate the session automatically if there has been no activity on it for a certain time.
5. Whether to enable filtering, and if so whether the client's IP address or host name should be checked first to determine what filtering to perform.

The filtering scheme itself is invoked from the main menu (Figure 201 on page 217) by selecting **TN3270E Server Filters** and **Create**. Figure 205 on page 221 is the result.



○ TCP/IP address

○ Host name

● Domain name

Host name or domain name: itso.ral.ibm.com

{Default Pool}
PUBLIC
TN99003
TN99004
TN99005
TN99006
TN99007
TN99008

LU/Pool name	Class type
TN99002	Implicit Workstation

The filter record list specifies, in order, the LUs and pools to which session requests are to be granted from the client identified by the source IP address.

Figure 205. TN3270E Filters

The idea here is that you define a group of IP clients, select a group of LUs, and link them together to ensure that clients from the selected group are assigned only to LUs of the selected group.

LUs may be selected in three ways:

- By group.
- By individual LU.
- As the default pool (this option can be seen at the top of the Available LUs and Pools window). The default pool is that defined as such on the main TN3270E Server panel (Figure 202 on page 218). It allows you to change the default pool without having to redefine the filters.

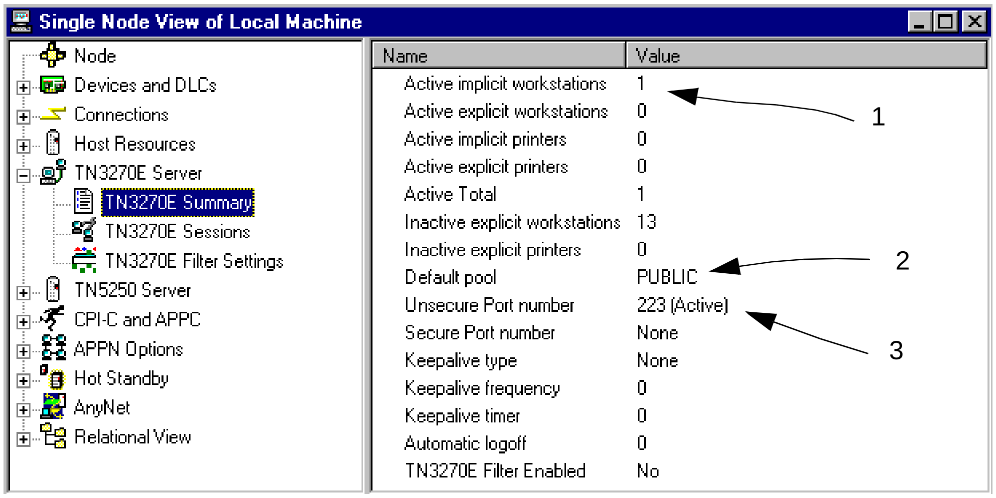
Clients may also be selected in three ways:

- By IP address. You can specify an individual client address or (by defining a suitable subnet mask) a group of clients in the same logical subnetwork.
- By host name.
- By domain name, thus selecting a whole domain’s worth of clients.

Be sure not to enable filtering on the Options panel, then forget to define any filters on the Filters panel. This will result in connections being denied with no obvious cause.

6.4.2 Operation

The Node Operations function of CS/NT contains some additional displays to monitor the operation of TN3270E. On its main panel, there is a **TN3270E Server** option which splits into three parts: summary, sessions and filters. Figure 206 shows the **TN3270E Summary** display on our system when one client had connected to our CS/NT server.



Name	Value
Active implicit workstations	1
Active explicit workstations	0
Active implicit printers	0
Active explicit printers	0
Active Total	1
Inactive explicit workstations	13
Inactive explicit printers	0
Default pool	PUBLIC
Unsecure Port number	223 (Active)
Secure Port number	None
Keepalive type	None
Keepalive frequency	0
Keepalive timer	0
Automatic logoff	0
TN3270E Filter Enabled	No

Figure 206. TN3270E Summary Display

You can see here that:

1. There is one active LU which is an implicit workstation; the client has connected to TN3270E without specifying a device (LU) name.
2. The LU pool from which the LU was taken is PUBLIC (in fact, the default LU pool in this particular setup).
3. The TCP port number in use is 223.

To see the details of this connection we first turn to the TN3270E Sessions panel (Figure 207).

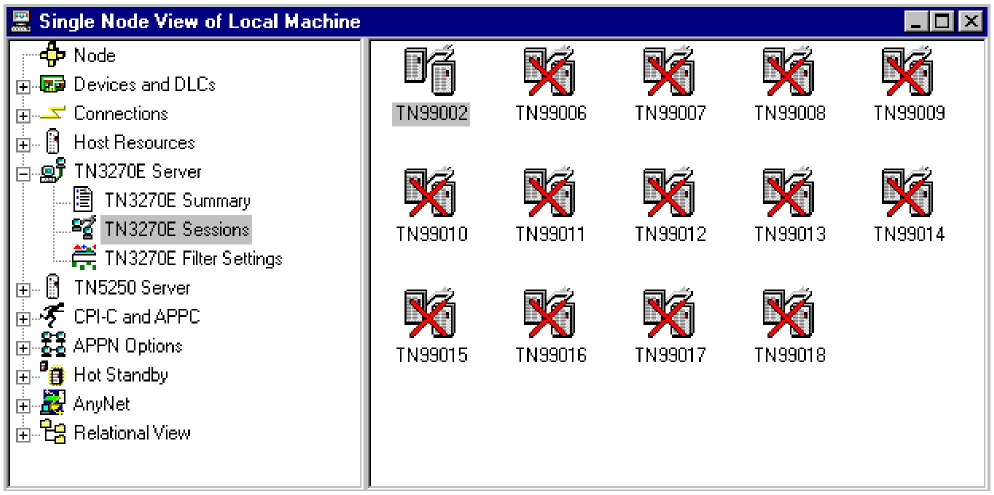
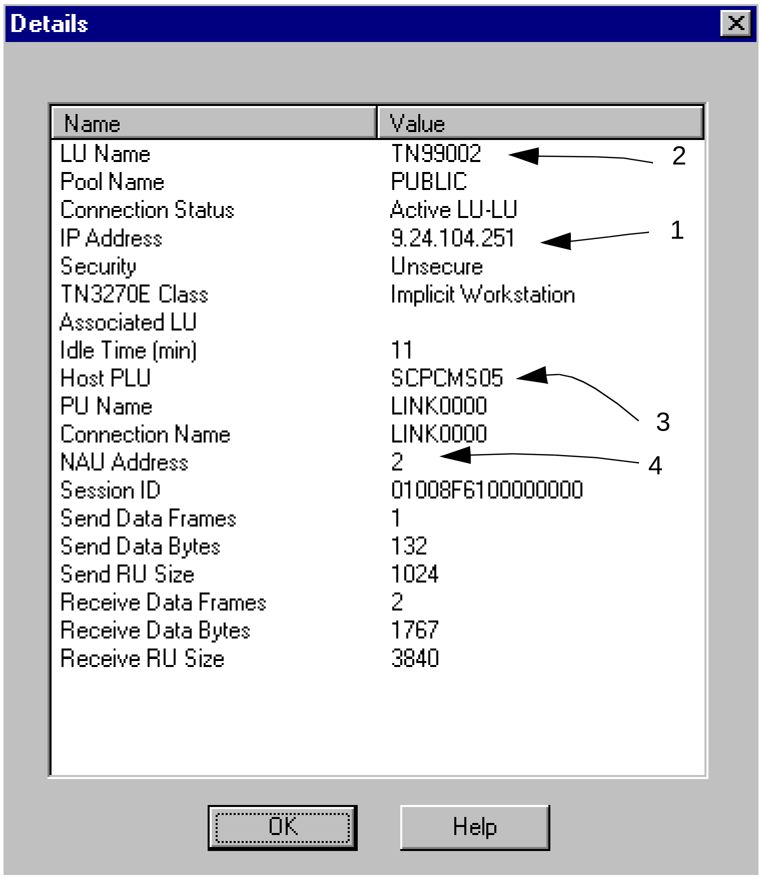


Figure 207. TN3270E Sessions Panel

This shows an icon representing each active LU defined to TN3270E, plus the inactive explicitly defined ones. The inactive implicit LUs are not shown, just as they do not appear in the summary in Figure 206. The inactive LUs are depicted by a red cross drawn through them.

Finally, we double-click the active LU icon (TN99002) to see the details of this connection. Please see Figure 208.



Name	Value
LU Name	TN99002
Pool Name	PUBLIC
Connection Status	Active LU-LU
IP Address	9.24.104.251
Security	Unsecure
TN3270E Class	Implicit Workstation
Associated LU	
Idle Time (min)	11
Host PLU	SCPCMS05
PU Name	LINK0000
Connection Name	LINK0000
NAU Address	2
Session ID	01008F6100000000
Send Data Frames	1
Send Data Bytes	132
Send RU Size	1024
Receive Data Frames	2
Receive Data Bytes	1767
Receive RU Size	3840

Figure 208. TN3270E Session Details

In this panel you can relate the TN3270E client (1) to the locally assigned device name (2). The actual VTAM LU name is not present, since it is optional for VTAM to send this information. However, the primary LU (application) name is shown at (3) as is the local LU address (4) so you can use VTAM commands to display this session from the OS/390 host. The PU and connection names shown are local and meaningless to VTAM.

6.5 TN3270E in the 221X Router Family

The TN3270E function in the 221X router family supports:

- LU types 1, 2 and 3 using the RFC 1647 standards.
- SNA positive and negative responses.
- Handling of attention and system request keys.
- Categorization of LUs into classes (pools).
- Association of printer LUs with display LUs.

- IP filtering, that is the mapping of clients to host LUs in order to control access.
- Multiple ports for use by TN3270E. Each port can be associated with its own set of LUs and pools, so that the client can access a different set of host sessions by connecting to a different port.
- Host access via native subarea connections or DLUR connections. Subarea connections are supported on token-ring, Ethernet, frame relay or ESCON connections as applicable to the router in question. DLUR connections are supported on any APPN link including DLSw, PPP and ATM.

There is always a line-mode Telnet server in a 221X (the remote console), so you need to be careful about the use of port numbers for TN3270E use.

The TN3270E code, together with the prerequisite APPN code, is not automatically loaded into the 221X unless you are using the Configuration Program (the remote configuration tool which generates a configuration file to be downloaded). If you are using Telnet or an ASCII emulator on the service port, you need to enter the following commands to get the right code loaded into the box:

```
load add package appn
load add package tn3270e
```

Unless you do both of these you will not be able to configure a subarea link, let alone the TN3270E Server.

6.5.1 Configuration of TN3270E

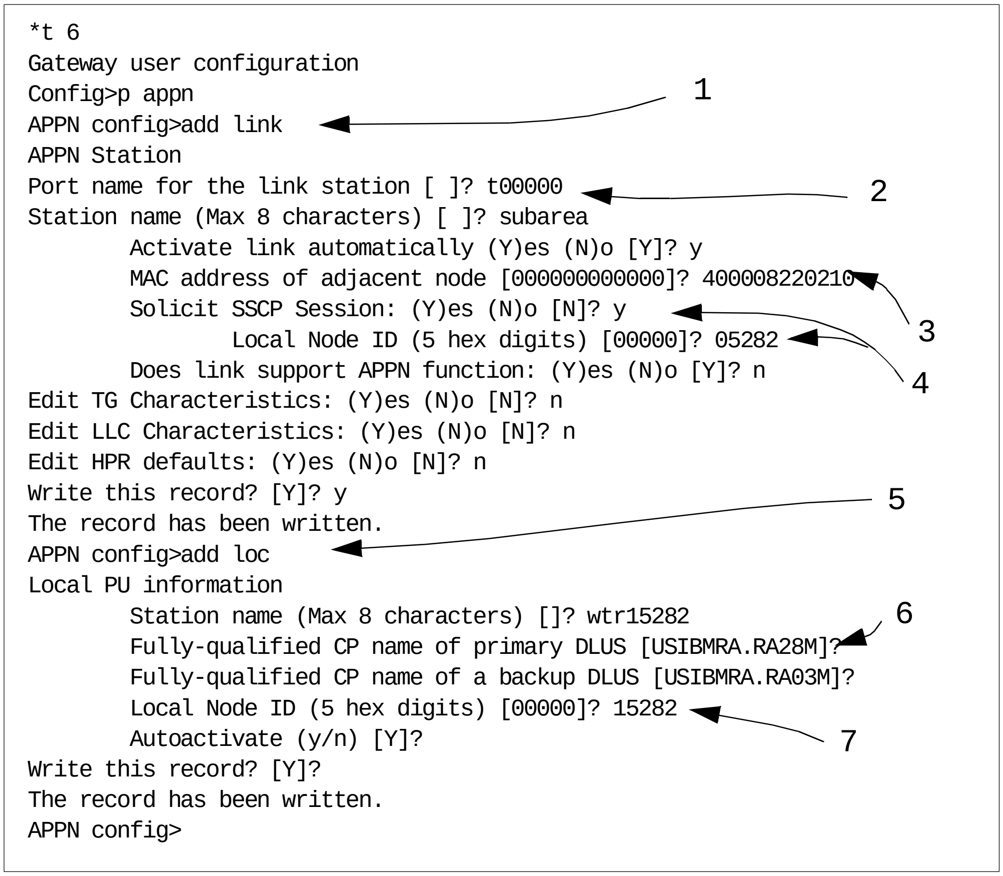
Whether you will be using subarea or APPN/DLUR connectivity to the SNA host, you need to configure APPN on the 221X. Then, to define a host connection, you need to do one of the following:

- For subarea connection, you configure a link station (**add link** in the APPN Configuration process) and make sure that you answer **yes** to **Solicit SSCP-PU session**.
- For DLUR connection, you configure a local PU (**add loc** in the APPN Configuration process) and give it the network name of the DLUS VTAM.

Having done this, you tell the TN3270E configuration process the name of this link station or PU to identify the host connection to be used. The TN3270E configuration process itself is invoked by typing `TN3270` from the APPN Configuration prompt.

Figure 209 on page 225 shows the APPN configuration relevant to TN3270E in our test environment. The physical interfaces, the APPN node and the real APPN connections have been defined, and we need to add some host links as the first stage of the definition. We shall add a subarea link on a token-ring port and a DLUR link.

To reach the APPN Configuration prompt, enter `t 6` followed by `p appn` from the base (asterisk) prompt.



```
*t 6
Gateway user configuration
Config>p appn
APPN config>add link
APPN Station
Port name for the link station [ ]? t00000
Station name (Max 8 characters) [ ]? subarea
        Activate link automatically (Y)es (N)o [Y]? y
        MAC address of adjacent node [000000000000]? 400008220210
        Solicit SSCP Session: (Y)es (N)o [N]? y
                Local Node ID (5 hex digits) [00000]? 05282
        Does link support APPN function: (Y)es (N)o [Y]? n
Edit TG Characteristics: (Y)es (N)o [N]? n
Edit LLC Characteristics: (Y)es (N)o [N]? n
Edit HPR defaults: (Y)es (N)o [N]? n
Write this record? [Y]? y
The record has been written.
APPN config>add loc
Local PU information
        Station name (Max 8 characters) []? wtr15282
        Fully-qualified CP name of primary DLUS [USIBMRA.RA28M]?
        Fully-qualified CP name of a backup DLUS [USIBMRA.RA03M]?
        Local Node ID (5 hex digits) [00000]? 15282
        Autoactivate (y/n) [Y]?
Write this record? [Y]?
The record has been written.
APPN config>
```

Figure 209. APPN Host Link Configuration

In this display:

1. We enter `add link` to define a subarea host connection.
2. We supply the name of the port. This one is interface 0, a token-ring port.
3. We supply the MAC address of the host (a 3745 in this case) to which the 221X is connected.
4. We enter `y` to request SSCP-PU and SSCP-LU sessions from the host. The 221X only understands APPN connections, so what we have defined is an APPN link with SSCP session support. If the VTAM host happens not to be configured for APPN, this will be a LEN connection with dependent LUs. Both the 221X and older versions of VTAM (back to V3R2) are happy with this setup.

   If we want SSCP sessions we also need a node ID to identify this station to VTAM. The ID number is defined here, but the ID block is fixed at 077 for the 221X.

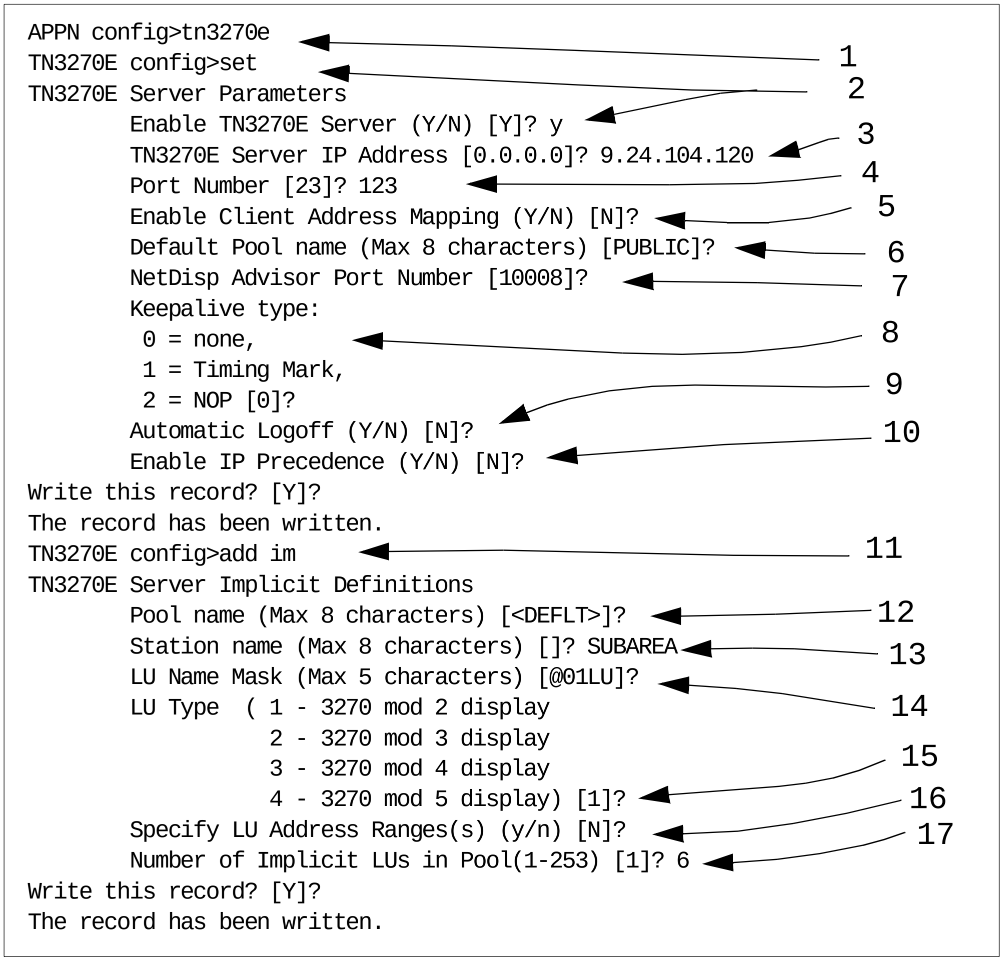
5. We also enter `add loc` to define a DLUR/S connection. The other main use of DLUR/S on the 221X is as an SNA gateway on behalf of downstream SNA nodes. In that case the PU that VTAM sees is a real SNA PU. For TN3270 the PU is internal to the 221X, so it must be defined in this manner.

   `Add loc` will not work unless you have enabled DLUR at the node level using `set dlur`.

6. We ensure that the new internal PU will be connected to the appropriate VTAM network node DLU server when required.

7. We need to identify this PU to VTAM. A downstream SNA node will usually have its own node ID, but the internal SNA PU must have the node ID defined here.

Now we have the host connections defined, we enter TN3270 to configure the server itself, as shown in Figure 210.



```
APPN config>tn3270e
TN3270E config>set
TN3270E Server Parameters
        Enable TN3270E Server (Y/N) [Y]? y
        TN3270E Server IP Address [0.0.0.0]? 9.24.104.120
        Port Number [23]? 123
        Enable Client Address Mapping (Y/N) [N]?
        Default Pool name (Max 8 characters) [PUBLIC]?
        NetDisp Advisor Port Number [10008]?
        Keepalive type:
         0 = none,
         1 = Timing Mark,
         2 = NOP [0]?
        Automatic Logoff (Y/N) [N]?
        Enable IP Precedence (Y/N) [N]?
Write this record? [Y]?
The record has been written.
TN3270E config>add im
TN3270E Server Implicit Definitions
        Pool name (Max 8 characters) [<DEFLT>]?
        Station name (Max 8 characters) []? SUBAREA
        LU Name Mask (Max 5 characters) [@01LU]?
        LU Type  ( 1 - 3270 mod 2 display
                   2 - 3270 mod 3 display
                   3 - 3270 mod 4 display
                   4 - 3270 mod 5 display) [1]?
        Specify LU Address Ranges(s) (y/n) [N]?
        Number of Implicit LUs in Pool(1-253) [1]? 6
Write this record? [Y]?
The record has been written.
```

Figure 210. TN3270 Server Configuration

As with the other TN3270E implementations, we define some TN3270E options such as the port number, then we define some LUs, and lastly we associate clients, LUs and printers in an appropriate fashion. In the display above:

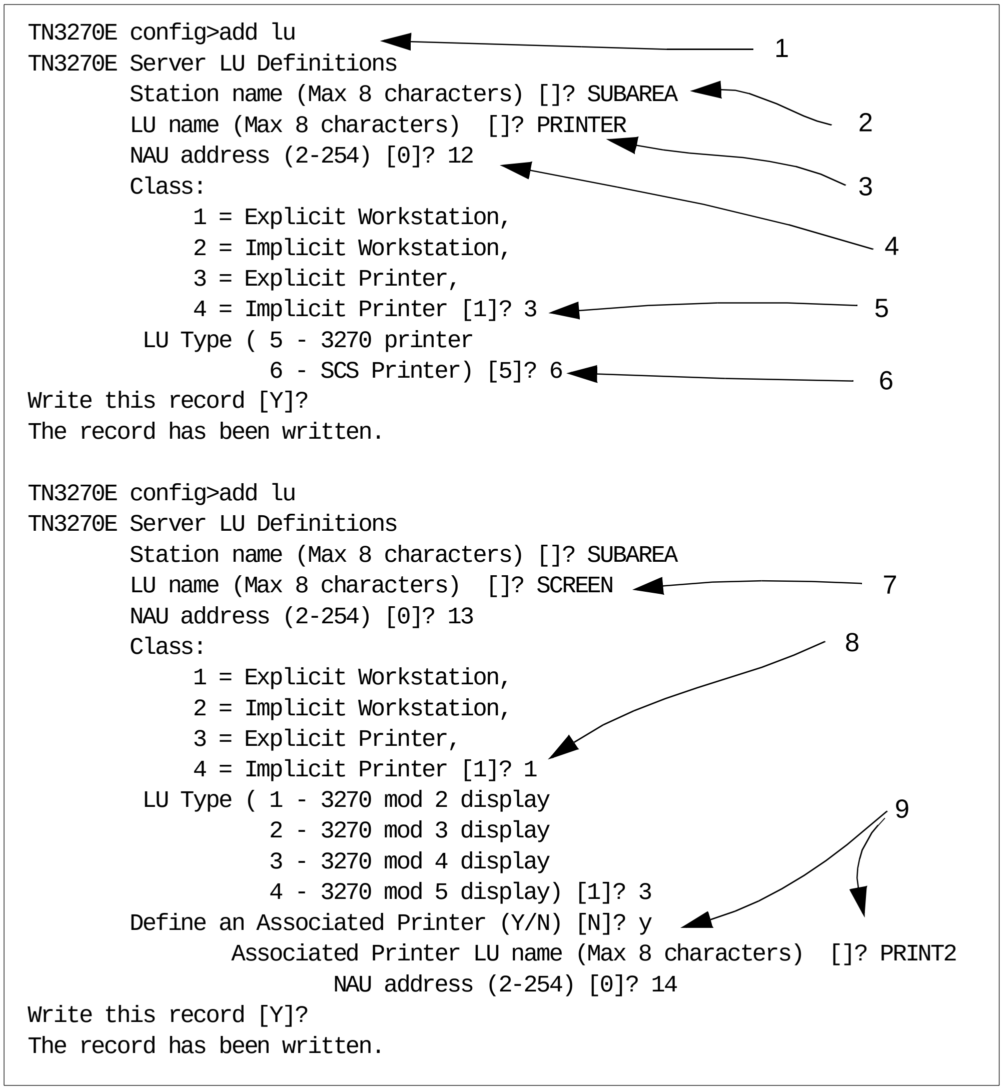
1. We enter `tn3270e` to begin the server configuration process.

2. We type `set` to set the server parameters, and then enable the TN3270E Server.

3. The TN3270E function has its own IP address. It is recommended that a real interface address is used; use the 221X internal IP address only if you need to be able to access the server through an alternative port if the primary one is down.

4. We change the TN3270E port number to 123 to avoid clashes with the line-mode Telnet server. We actually used Telnet to configure this particular machine, a 2216. If you wish to define additional TN3270E ports, use the `add port` command from the TN3270E prompt.
5. The default is not to filter IP clients. If you want to define filters, use the `add map` command from the TN3270E prompt.
6. The default LU pool works the same way as in CS/NT. When you specify a default pool here, you can change its name via the `set` command here without needing to redefine all your implicit and explicit LUs.
7. The 221X can interoperate with a network dispatcher to provide load balancing over multiple TN3270E Servers. The advisor function is the process that communicates server status to the network dispatcher, and here we can change the port number used by the advisor.
8. The sending of keepalive messages to the client is an option.
9. As with CS/2 and CS/NT, automatic termination of host sessions after a period of inactivity can be defined.
10. The 221X allows you to set the priority on TN3270E packets by using the precedence bits in the IP header.
11. When we have defined the basic TN3270E parameters, we must define some host LUs for the clients to use. The simplest way is to define an implicit LU pool via the `add im` command.
12. We let the LUs we are about to define be in the default pool. The `set` process has already given the pool a name of PUBLIC.
13. We must associate a host link (a PU, in other words) with these LUs. For this pool we choose the subarea connection previously defined in Figure 209 on page 225. Our alternative would be the DLUR PU called WTR15282 that was defined in the same panel.
14. We give the pool of LUs some locally-known LU names.
15. These LUs are to be 24x80 display screens. The answer to this question affects what the 221X sends to VTAM as the device type for use by the self-defining dependent LU exit (ISTEXCSD). The device types sent are 3270002 to 3270005 for displays and SCSP or 3270P for SCS or DSC printers respectively.
16. We let the 221X decide what local (NAU) addresses to give the LUs. By default it starts at 2 and goes upwards.
17. We define six LUs in the pool.

This is enough for a client to be able to connect to the subarea host, but we also define some explicit LUs as in Figure 211 on page 228.



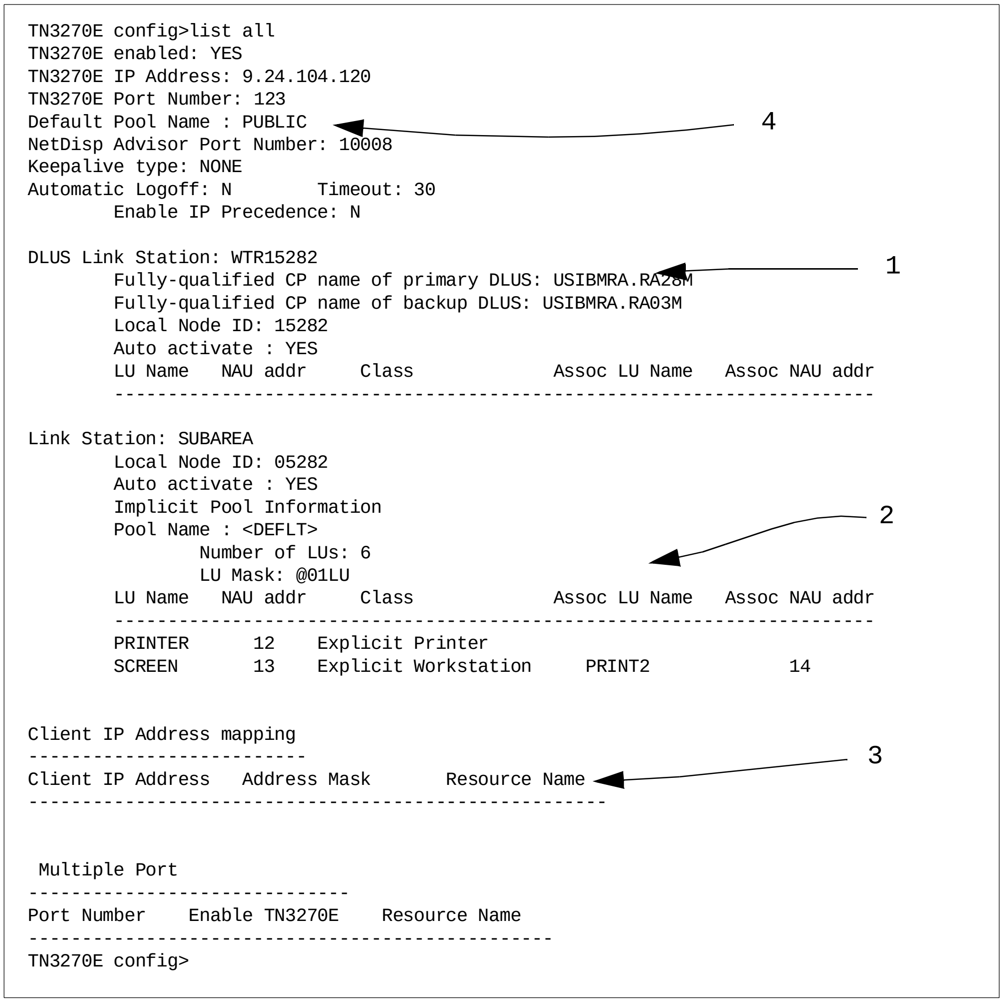
```
TN3270E config>add lu
TN3270E Server LU Definitions
        Station name (Max 8 characters) []? SUBAREA
        LU name (Max 8 characters)  []? PRINTER
        NAU address (2-254) [0]? 12
        Class:
             1 = Explicit Workstation,
             2 = Implicit Workstation,
             3 = Explicit Printer,
             4 = Implicit Printer [1]? 3
         LU Type ( 5 - 3270 printer
                   6 - SCS Printer) [5]? 6
Write this record [Y]?
The record has been written.

TN3270E config>add lu
TN3270E Server LU Definitions
        Station name (Max 8 characters) []? SUBAREA
        LU name (Max 8 characters)  []? SCREEN
        NAU address (2-254) [0]? 13
        Class:
             1 = Explicit Workstation,
             2 = Implicit Workstation,
             3 = Explicit Printer,
             4 = Implicit Printer [1]? 1
         LU Type ( 1 - 3270 mod 2 display
                   2 - 3270 mod 3 display
                   3 - 3270 mod 4 display
                   4 - 3270 mod 5 display) [1]? 3
        Define an Associated Printer (Y/N) [N]? y
                Associated Printer LU name (Max 8 characters)  []? PRINT2
                        NAU address (2-254) [0]? 14
Write this record [Y]?
The record has been written.
```

Figure 211. Explicit LU Definition

We use the `add lu` command (1) to define an LU explicitly. We assign it to a host connection (2) and give it a local name (3) and a local address (4). This LU is to be an explicit printer (5) using LU type 1 (SCS, 6) data streams.

We also define an explicit display (7, 8) on the same link. An explicit display LU can be associated with a printer; (9) defines the name and local address of that printer.

Having defined our TN3270E configuration, we issue `list all` from the TN3270E prompt to see Figure 212 on page 229.



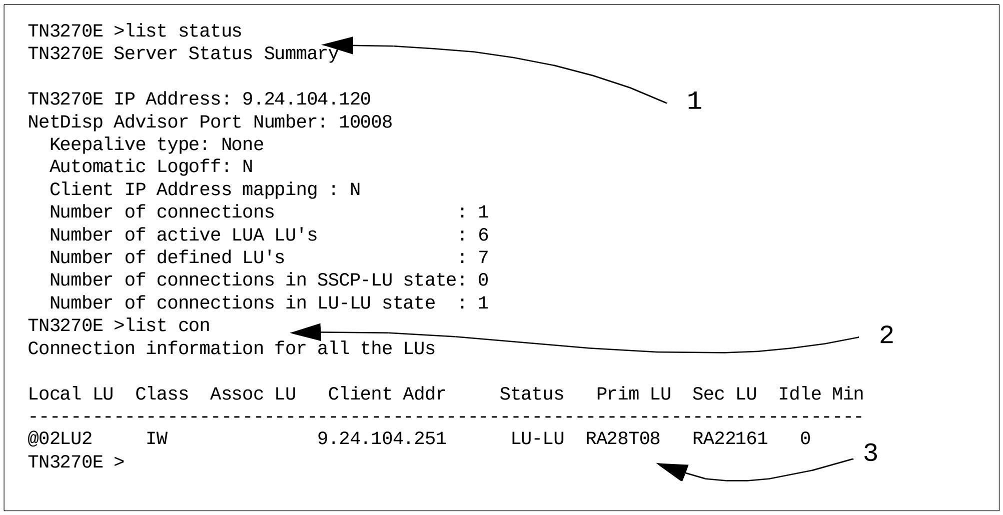
```
TN3270E config>list all
TN3270E enabled: YES
TN3270E IP Address: 9.24.104.120
TN3270E Port Number: 123
Default Pool Name : PUBLIC
NetDisp Advisor Port Number: 10008
Keepalive type: NONE
Automatic Logoff: N        Timeout: 30
        Enable IP Precedence: N

DLUS Link Station: WTR15282
        Fully-qualified CP name of primary DLUS: USIBMRA.RA28M
        Fully-qualified CP name of backup DLUS: USIBMRA.RA03M
        Local Node ID: 15282
        Auto activate : YES
        LU Name   NAU addr     Class             Assoc LU Name   Assoc NAU addr
        ----------------------------------------------------------------------

Link Station: SUBAREA
        Local Node ID: 05282
        Auto activate : YES
        Implicit Pool Information
        Pool Name : <DEFLT>
                Number of LUs: 6
                LU Mask: @01LU
        LU Name   NAU addr     Class             Assoc LU Name   Assoc NAU addr
        ----------------------------------------------------------------------
        PRINTER      12    Explicit Printer
        SCREEN       13    Explicit Workstation     PRINT2            14


Client IP Address mapping
--------------------------
Client IP Address   Address Mask       Resource Name
------------------------------------------------------



 Multiple Port
------------------------------
Port Number    Enable TN3270E    Resource Name
-------------------------------------------------
TN3270E config>
```

Figure 212. TN3270E Configuration Display

There are two host connections available for TN3270E to use. The DLUR PU called WTR15282 (1) has no implicit or explicit LUs defined on it yet. The subarea connection SUBAREA (2) has six LUs in the default pool (PUBLIC, as (4) tells us) and two explicit LUs as shown. There is no filtering in operation (3).

6.5.2 Operation of TN3270E

To display the status of the TN3270E function while the 221X is running, use `t 5` followed by `p appn` and then `tn3270e`. You then have a `list` command available which can tell you the status of the server itself, the connections, the pools, the mapping and so on. Figure 213 on page 230 shows a connection display after a client has established a TN3270E session to a host application.



```
TN3270E >list status
TN3270E Server Status Summary

TN3270E IP Address: 9.24.104.120
NetDisp Advisor Port Number: 10008
  Keepalive type: None
  Automatic Logoff: N
  Client IP Address mapping : N
  Number of connections                : 1
  Number of active LUA LU's            : 6
  Number of defined LU's               : 7
  Number of connections in SSCP-LU state: 0
  Number of connections in LU-LU state  : 1
TN3270E >list con
Connection information for all the LUs

Local LU  Class  Assoc LU   Client Addr     Status   Prim LU  Sec LU  Idle Min
------------------------------------------------------------------------------
@02LU2     IW              9.24.104.251      LU-LU  RA28T08   RA22161   0
TN3270E >
```

Figure 213. TN3270E Connection Display

We issue the `list status` command (1) to receive a summary of the connections and the server options. We then display a list of the connections (2) using `list con`. Our single TN3270E client, 9.24.104.251, is using the LU whose local name is @02LU2 and is in session with TSO application RA28T08. Note (3) that the VTAM LU name, RA22161, is also displayed. VTAM has sent this information to the 2216 on a 0E control vector in the ACTLU request.

6.6 Host On-Demand

Host On-Demand (HOD) is Telnet/3270 with a difference. It utilizes a separate function (the HOD server) which acts as a TN3270E client to the TN3270 Server, but as a Web server to the real client on the user’s workstation. The client is actually a Java applet downloaded from the HOD server to a Web browser. Host On-Demand is thus an ideal solution to 3270 application access for users who require it on a casual basis, or who use Web browsers extensively and therefore are quite familiar with the principles. It also means that a TN3270 client can be implemented on any workstation that supports a reasonably current level of Web browser. This probably covers just about any type of workstation imaginable.

The HOD server must reside on the same machine as a Web browser, but the TN3270E Server need not be in the same place. Figure 214 on page 231 illustrates the concept of Host On-Demand.

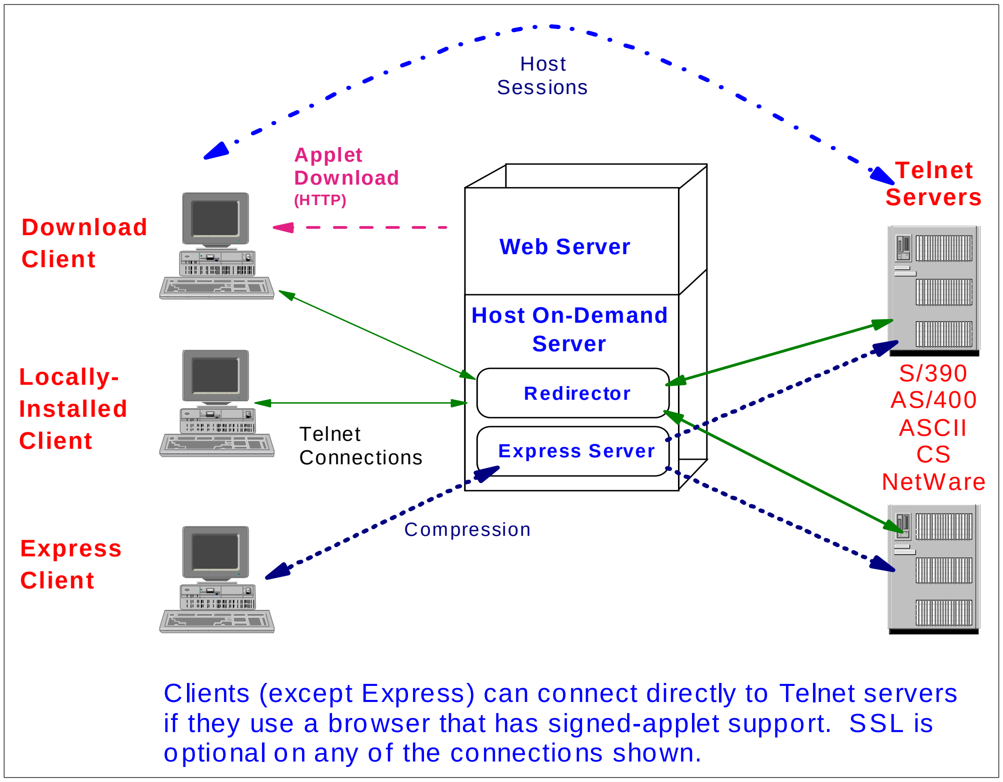


Figure 214. Host On-Demand

The box in the middle of the diagram represents the Web server and HOD server, which must run together. There are three types of client that can be implemented on the user's workstation:

- The Download client is the basic implementation; the Java applet is downloaded from the HOD server when the user invokes the correct URL.
- The Locally Installed client is used when the network connection is very slow or unreliable; the client code is installed permanently on the client and loaded from the hard disk instead of being downloaded when required for a session.
- The Express client uses compression techniques to speed up the process; this is done through additional software on both client and HOD server machines.

On the server side, the TN3270E Server can be in the same machine as the HOD server, or separate. If it is separate, a function called the Redirector on the HOD server routes the connection to the TN3270E Server itself.

Connections using HOD can use SSL if both the client and the TN3270E Server support it.

6.6.1 Host On-Demand Implementations

The requirements to run HOD on a client are quite simple: a Web browser and a Java virtual machine. Thus almost any up-to-date workstation with Internet access can reach 3270 applications using HOD. If you need to use a local or an express client, however, the code is operating-system dependent (there is more to it than just Java) and is available only on the Windows 95, 98 and NT platforms.

For full TN3270E support, Version 3 of the HOD server is required. This is implemented on the following IBM platforms:

- OS/390 UNIX Systems Services
- OS/400, Version 4 Release 2 or later
- OS/2 Warp and Warp Server, Version 4 or later
- AIX, Version 4 Release 2 or later
- Sun Solaris
- HP-UX
- Windows NT, Version 4 with SP3 or later
- Novell Netware, Version 4

The HOD server also requires Java, which comes with the HOD server code for Windows NT but is obtainable separately for the other platforms.

HOD server Version 3 supports the following major functions:

- TN3270, TN5250 and VT emulation as well as TN3270E
- Session security through the Secure Sockets Layer (SSL) protocol
- Server-based management of user configurations
- Support for up to 26 concurrent sessions
- Telnet redirection
- LU and LU pool definition (explicit and implicit classes)
- Support for attention and system request functions
- 3270 host printing
- Bookmarking of sessions in several ways
- Macro record/play, with prompts and waits, just as PComm

6.7 Host Publisher

To the user with Web access and occasional requirements for 3270-based applications, HOD is an ideal solution except for one thing: the screen presentation is that of a 3270 emulator, in other words a green screen with a row of indicators on the bottom line and (most of the time) text-only presentation. Many Web users prefer their applications to provide scroll bars, check boxes, and little pictures by way of illustration; in other words a more natural GUI rather than the traditional 3270 interface.

Host Publisher addresses this requirement. In principle it works the same way as HOD except that it translates the 3270 data streams to Web-familiar presentation instead of native 3270 presentation. The current Host Publisher technology is more general than just 3270-to-Web-via-Java, and both the technology and the products which implement it are still under development. Therefore, we present only a brief summary of its operation.

The components of a Host Publisher server, which need not be together on the same machine, are:

- A Web server, which talks to the client browser
- A component which recognizes certain strings in the URL and directs requests to a page server
- Page servers, which deliver the Web-friendly pages to the Web server
- Integration components, which connect the page servers to the applications acting as the true servers (in our case, 3270 applications)

We expect to see a new generation of Host Publisher implementations later in 1999.

Appendix A. Special Notices

This publication is intended to help networking profesionals understand and implement the various ways of integrating SNA and TCP/IP communication. The information in this publication is not intended as the specification of any programming interfaces that are provided by the eNetwork Communications Server family of products. See the PUBLICATIONS section of the IBM Programming Announcement for those products for more information about what publications are considered to be product documentation.

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

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

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. 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.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ACF/VTAM
AIX
APPN
DB2
IBM®
OS/2
OS/400
RACF
System/390
Advanced Peer-to-Peer Networking
AnyNet
CICS
ESCON
OpenEdition
OS/390
Parallel Sysplex
SecureWay
VTAM

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, Windows, Windows NT, and the Windows 95 logo are trademarks or registered trademarks of Microsoft Corporation.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix B. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook. Note that, whereas current versions of Communications Server publications use the brand name eNetwork, future ones will use the name SecureWay. We have used eNetwork throughout this appendix to reflect the current set of publications.

B.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 239.

- *3746, 2210, 2216, and 2220 Interconnectivity: Frame Relay and Related Functions,* SG24-2146
- *IBM 2210 Nways Multiprotocol Router and IBM 2216 Nways Multiaccess Connector Description and Configuration Scenarios - Volume II*, SG24-4956
- *IBM 2210 Nways Multiprotocol Router Description and Configuration Scenarios, Volume I*, SG24-4446
- *MSS Release 2.1, Including the MSS Client and Domain Client*, SG24-5231
- *3746 Nways Controller Models 900 and 950: Multiaccess Enclosure Primer*, SG24-5238
- *IBM 2216/Network Utility Host Channel Connections*, SG24-5303
- *Converging TCP/IP and SNA Networks: Web Access over SNA*, SG24-2101
- *OS/390 eNetwork Communications Server TCP/IP Implementation Guide Volume 1: Configuration and Routing*, SG24-5227
- *IBM Communications Server for Windows NT Version 5.0*, SG24-2099
- *IBM eNetwork Communications Server for Windows NT Version 6.0 Enhancements*, SG24-5232
- *IBM Communications Server for OS/2 Warp Version 4.0 Enhancements*, SG24-4587
- *IBM Communications Server for OS/2 Warp Version 4.1 Enhancements*, SG24-4916
- *IBM eNetwork Communications Server for OS/2 Warp Version 5.0 Enhancements*, SG24-2147
- *IBM Network Utility Description and Configuration Scenarios*, SG24-5289
- *Subarea to APPN Migration: HPR and DLUR Implementation*, SG24-5204
- *Inside APPN - The Essential Guide to the Next-Generation SNA*, SG24-3669
- *TCP/IP Tutorial and Technical Overview*, GG24-3376
- *AnyNet: SNA over TCP/IP Installation and Interoperability*, GG24-4395 (no softcopy available)

B.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

B.3 Other Publications

These publications are also relevant as further information sources:

- *6611 Network Processor Introduction and Planning Guide*, *Version 1 Release 4*, GK2T-0334
- *Multiprotocol Transport Networking (MPTN) Architecture: Technical Overview,* GC31-7073
- *Multiprotocol Transport Networking (MPTN) Architecture: Formats*, GC31-7074
- *OS/390 eNetwork Communications Server IP Configuration,* SC31-8513
- *OS/390 eNetwork Communications Server SNA Resource Definition Reference,* SC31-8565
- *OS/390 eNetwork Communications Server SNA Network Implementation Guide,* SC31-8563
- *OS/390 eNetwork Communications Server AnyNet: Guide to SNA over IP*, SC31-8578 (available in softcopy only, on CD-ROM SK2T-6012)
- *OS/390 eNetwork Communications Server AnyNet: Guide to Sockets over SNA,* SC31-8577 (available in softcopy only, on CD-ROM SK2T-6012)

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** `http://www.redbooks.ibm.com/`

  Search for, view, download or order hardcopy/CD-ROM redbooks from the redbooks web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this redbooks site.

  Redpieces are redbooks in progress; not all redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders via e-mail including information from the redbooks fax order form to:

	e-mail address
In United States	usib6fpl@ibmmail.com
Outside North America	Contact information is in the “How to Order” section at this site: `http://www.elink.ibmlink.ibm.com/pbl/pbl/`

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: `http://www.elink.ibmlink.ibm.com/pbl/pbl/`

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: `http://www.elink.ibmlink.ibm.com/pbl/pbl/`

This information was current at the time of publication, but is continually subject to change. The latest information for customer may be found at `http://www.redbooks.ibm.com/` and for IBM employees at `http://w3.itso.ibm.com/`.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and redbooks by accessing the IBM Intranet Web site at `http://w3.itso.ibm.com/` and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may also view redbook. residency, and workshop announcements at `http://inews.ibm.com/`.

IBM Redbook Fax Order Form

Please send me the following:

Title	Order Number	Quantity

First name   Last name

Company

Address

City   Postal code   Country

Telephone number   Telefax number   VAT number

□ Invoice to customer number

□ Credit card number

Credit card expiration date   Card issued to   Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

List of Abbreviations

ACB	Access method Control Block
AIX	Advanced Interactive eXecutive
ANR	Automatic Network Routing
ANSI	American National Standards Institute
APAR	Authorized Program Analysis Report
API	Application Program Interface
APPC	Advanced Program-to-Program Communication
APPN	Advanced Peer-to-Peer Networking
ARB	Adaptive Rate-Based
ASCII	American Standard Code for Information Interchange
ATM	Asynchronous Transfer Mode
BTU	Basic Transmission Unit
CCU	Central Control Unit
CDRSC	Cross-Domain ReSourCe
CICS	Customer Information Control System
CNOS	Change Number Of Sessions
COS	Class Of Service
CP	Control Point
CPI-C	Common Programming Interface for Communications
CRC	Cyclic Redundancy Check
CS for OS/390	SecureWay Communications Server for OS/390
CS/2	SecureWay Communications Server for OS/2
CS/NT	SecureWay Communications Server for Windows NT
CTS	Common Transport Semantics
DCE	Data Circuit-terminating Equipment
DLC	Data Link Control
DLSw	Data Link Switching
DLUR	Dependent LU Requester
DLUR/S	Dependent LU Requester/Server
DLUS	Dependent LU Server
DM	Disconnected Mode
DNS	Domain Name Server
DSAP	Destination Service Access Point
DSC	Display Screen Compatibility
DTE	Data Terminal Equipment
EBCDIC	Extended Binary Coded Decimal Interchange Code
EMIF	ESCON Multiple Image Facility
EN	End Node
ESCON	Enterprise Systems CONnection
FDDI	Fiber Distributed Data Interface
FID	Format Identification
GUI	Graphical User Interface
HOD	Host On-Demand
HPR	High-Performance Routing
IANA	Internet Assigned Number Authority
IBM	International Business Machines Corporation
IC-TG	Interchange Transmission Group
ICMP	Internet Control Message Protocol
ICP	Interconnect Controller Program
IEEE	Institute of Electrical and Electronics Engineers
INN	Intermediate Network Node
IOCP	Input/Output Configuration Program
IP	Internet Protocol
IPX	Internetwork Packet eXchange
ISDN	Integrated-Services Digital Network
ISPF	Interactive System Productivity Facility
ISR	Intermediate Session Routing

ITSO	International Technical Support Organization
JCL	Job Control Language
LAN	Local Area Network
LDLC	Logical Data Link Control
LEN	Low Entry Networking
LLC	Logical Link Control
LPAR	Logical PARtition
LSA	Link Services Architecture
LU	Logical Unit
LUA	Logical Unit Application
MAC	Medium Access Control
MAE	MultiAccess Enclosure
MNPS	Multinode Persistent Sessions
MPC	MultiPath Channel
MPTN	MultiProtocol Transport Networking
MPTS	MultiProtocol Transport Services
MSS	Multiprotocol Switched Services
MVS	Multiple Virtual Storage
NAU	Network Accessible Unit
NCP	Network Control Program
NLP	Network Layer Packet
NN	Network Node
NRZ	Non-Return-to-Zero
NRZI	Non-Return-to-Zero Inverted
NVT	Network Virtual Terminal
OS/2	Operating System/2
OS/390	Operating System/390
OS/400	Operating System/400
OSA	Open Systems Adapter
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
PC	Personal Computer
PComm	Personal Communications
PFS	Physical File System
PPP	Point-to-Point Protocol
PU	Physical Unit
PVC	Permanent Virtual Circuit
QLLC	Qualified Logical Link Control
RACF	Resource Access Control Facility
RFC	Request For Comments
RIP	Routing Information Protocol
RNR	Request Not Ready
RR	Request Ready
RSCV	Route Selection Control Vector
RTP	Rapid Transport Protocol
SABME	Set Asynchronous Balanced Mode Extended
SAP	Service Access Point
SCS	SNA Character String
SDLC	Synchronous Data Link Control
SME	Session Management Exit
SNA	Systems Network Architecture
SNI	SNA Network Interconnect
SNMP	Simple Network Management Protocol
SNRM	Set Normal Response Mode
SSCP	System Services Control Point
SSL	Secure Sockets Layer
SVC	Switched Virtual Circuit
TCID	Transport Connection Identifier
TCP	Transmission Control Protocol
TG	Transmission Group
TN3270	Telnet/3270
TN3270E	Telnet/3270 Extended
TN5250	Telnet/5250
TRL	Transport Resource List
TRLE	Transport Resource List Element
TSO	Time Sharing Option
UDP	User Datagram Protocol
UI	Unnumbered Information
URL	Uniform Resource Locator
USS	Unformatted System Services
VIPA	Virtual Internet Protocol Address(ing)
VR-TG	Virtual Route-based Transmission Group

VTAM	Virtual Telecommunications Access Method
WAN	Wide Area Network
WLM	Workload Manager
XCA	eXternal Communications Adapter
XCF	Cross-System Coupling Facility
XID	eXchange IDentifier

Index

ITSO Redbook Evaluation

SNA and TCP/IP Integration
SG24-5291-00

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

- Use the online evaluation form found at http://www.redbooks.ibm.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
_ **Customer** _ **Business Partner** _ **Solution Developer** _ **IBM employee**
_ **None of the above**

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction __________

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

What other redbooks would you like to see published?

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

SG24-5291-00

Printed in the U.S.A.

SNA and TCP/IP Integration

SG24-5291-00