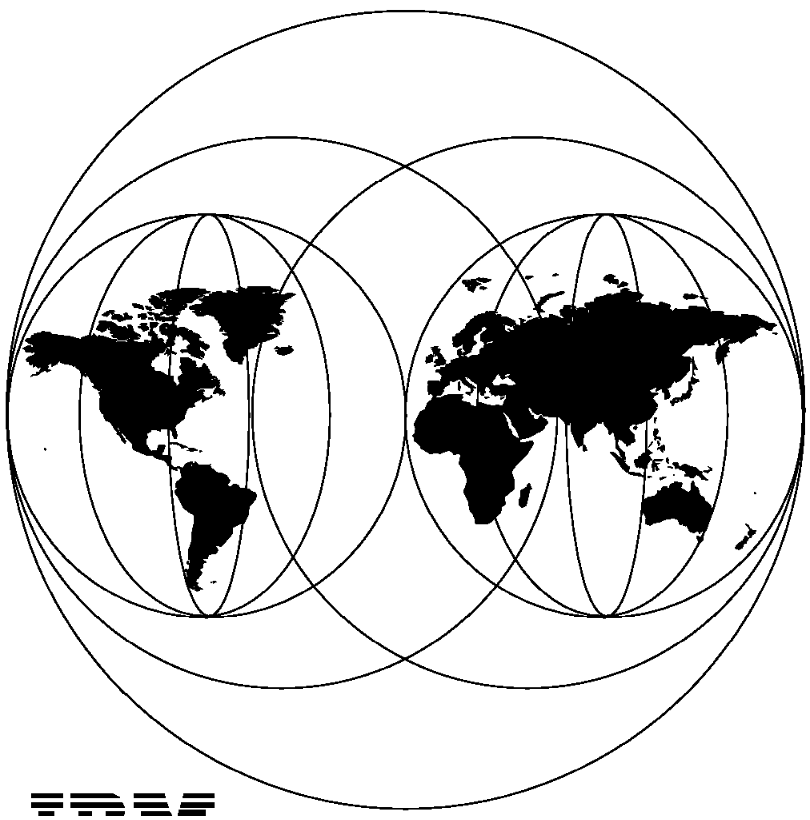
SG24-2143-00

Using NetDA/2 in an APPN Environment

November 1997



International Technical Support Organization
Raleigh Center

International Technical Support Organization

SG24-2143-00

Using NetDA/2 in an APPN Environment

November 1997

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 167.

First Edition (November 1997)

This edition applies to Version 1, Release Number 5 of Network Design and Analysis/2 (program number 5622-009) for use with the OS/2 operating system. The abbreviation NetDA/2 is used throughout the book.

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

Contents

Figures

Preface

This redbook will help users of NetDA/2 to employ the tool for APPN Networks. It describes how you can use the tool to define a network, design its topology, obtain the various route reports, both APPN and HPR, and perform simulation and capacity planning.

This redbook includes a real case scenario of how to modify APPN parameters in order to alter the flow of traffic, using NetDA/2 to corroborate its implementation.

Also included in this document is a description of how to use NetView APPN Accounting for gathering APPN traffic and how to apply it to NetDA/2 models.

The redbook should be regarded as a supplement to SG24-4225-03, *NetDA/2 V1R5 Design Tool Guide and Tutorial*.

How to Access Samples from This Book

Many of the sample programs and NIOs referenced in this document are available through the Internet.

Point your web browser to http://www.redbooks.ibm.com, click on **Downloads**, and click on **SG244225**. You will find there the files referenced in this book.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the Systems Management and Networking ITSO Center, Raleigh.

Alan Stoner was with IBM UK for 29 years, mainly in the SNA and networking areas. Since taking early retirement in 1994 he has worked with a number of clients using NetDA/2′s subarea routing and frame relay functions. He is a co-author of earlier redbooks.

Jorge Ferrari is a Senior ITSO Specialist in Network Design Tools at the Systems Management and Networking ITSO Center, Raleigh. He writes extensively and teaches IBM classes worldwide on the area of network design tools. Before joining the ITSO in 1993, he worked in the SNAP/SHOT and Network Design group in Raleigh, NC.

Jerzy Buczak is an IT Consultant at the Systems Management and Networking ITSO Center, Raleigh. 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 16 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.

Anton Smit is an Advisory IT Specialist in IBM South Africa. He has 10 years of experience in SNA and Network Management. Presently he is working in Product Support Services supporting WAN networking and networking management.

Thanks to the following people for their invaluable contributions to this project:

Juan Rodriguez
International Technical Support Organization, Raleigh Center

Irit Loy
IBM Israel

Mike Prendergast
IBM United Kingdom

Johnathan Harter
CS/390 Development, Research Triangle Park

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 179 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

For Internet users	http://www.redbooks.ibm.com
For IBM Intranet users	http://w3.itso.ibm.com

- Send us a note at the following address:

  redbook@vnet.ibm.com

Chapter 1. Introduction to APPN

There are two distinct architectures comprising SNA: subarea networking and Advanced Peer-to-Peer Networking (APPN). To the user and the application program they appear exactly the same. Within the network they are of course different, but VTAM (uniquely) supports both and can translate between them where necessary. Both subarea and APPN provide a similar range of network services to their end users, but APPN has more dynamics built into the architecture and requires much less predefinition.

Because subarea networking requires many more definitions (particularly routing definitions) than APPN, it is particularly labor-intensive when changes are made to the network. As a result, the risk of loss of availability due to errors in the changed definitions can be unacceptable. For this reason, one of the main benefits of APPN is an increase in network availability. This advantage becomes much more obvious in a rapidly changing, dynamic, modern enterprise-wide network with short maintenance windows.

A more recent extension to APPN is high-performance routing (HPR). In HPR, the intermediate session routing function is enhanced to be as simple as possible, rather like source routing in token-ring networks. This has two major advantages:

- Very little storage or processing power is required in intermediate nodes (nodes that simply route data through the network as opposed to nodes that contain the LUs that generate the data).
- The nodes at the endpoints of the HPR connection can easily switch the session route without disrupting the session. Thus if a node or a link on the session path fails, the user sees no more than an extended response time, provided of course that an alternate route is available. For the first time, HPR combines the predictable manageability of SNA with the nondisruptive rerouting that was always the hallmark of TCP/IP.

1.1 Review of Subarea Networking

Subarea networking was introduced with the first release of VTAM in 1974. The essential characteristics of subarea networking are:

- A subarea network is hierarchical and network services are centralized. VTAM is the highest level of node in the hierarchy, a network control program (NCP) in a communications controller is second, and everything else (such as terminal controllers and emulators) comes a poor third. Every LU in a subarea network is *owned* by a VTAM, in the sense that it has to request all its network services from the VTAM to which it is primarily connected. The entity in VTAM from which services are requested is known as the *system services control point* (SSCP).

  Several types of nodes are defined in the subarea networking hierarchy:

  - A type 5 node contains an SSCP. In effect, this means VTAM. A type 5 node can also perform the intermediate routing function.
  - A type 4 node can perform intermediate routing, but requires an SSCP to provide all network services. In practical terms, this means an NCP in a communications controller.

- A type 2 node is a peripheral node that cannot perform routing, and uses a simple subset of SNA to communicate with its adjacent routing node (type 4 or 5). This type of node is implemented by the older, low-function terminal controllers and emulators. LUs in such a node are always *dependent* LUs, since they are dependent on an SSCP to provide network services.
- A type 2.1 node is also a peripheral node, but uses a (less simple) subset of APPN to communicate with its adjacent routing node. As with a type 2 node, its subarea sessions must pass through its (single) attached routing node. LUs on such a node can be dependent or *independent*. An independent LU does not have an SSCP-LU session. If it needs an LU-LU session with a subarea partner, it simply sends a BIND across the connection and hopes that the node on the other side will do something about it. VTAM, of course, then uses subarea network services to locate the partner and set up the session. Independent LUs use only LU type 6.2 communication.
- A type 1 node is not often seen these days, being of even lower function than a type 2.

- The resource directory is distributed and partner location can be dynamic, without the need to predefine the owner of a resource.
- The network topology is distributed, but static. All paths in a subarea network must be manually defined, and route selection is performed by choosing from predefined lists of routes.
- Sessions are carried on logical paths called *virtual routes* (VRs). VTAM and NCP nodes establish and maintain these VRs. Routing and flow control in a subarea network is done using VRs as the basis. A VR can carry multiple sessions, and each session is assigned to a VR on the basis of a class of service (COS). A side effect of the VR system is that *only* VTAM and NCP nodes can act as intermediate routing nodes.
- In subarea networking, control information (such as session requests and searches) flows on sessions, just as application data flows on sessions. However, in a subarea network there are entities other than LUs for these purposes. The SSCP in VTAM maintains sessions with other SSCPs (known as SSCP-SSCP sessions), which are used to verify the location of a resource, to notify the partner of a session, and so on. The SSCP also maintains sessions with its owned LUs, over which those LUs request network services. These are, of course, SSCP-LU sessions. Finally, we must mention physical units (PUs), which again have SSCP-PU sessions with their owners. A PU represents the physical resource associated with an SNA node, and confines itself to exchanging management and control information with its SSCP.

A subarea network can be composed of independent *subnetworks*, each of which is distinguished by a unique network ID. This scheme is called SNA Network Interconnect (SNI), and is often used to link two or more distinct enterprises where neither wants to maintain the routing definitions for the other. SNI isolates the topologies of subarea networks from each other, but not the directories. SSCP-PU and SSCP-LU sessions cannot cross a network boundary, nor can VRs. LU-LU sessions can (of course), as can SSCP-SSCP sessions because they are used to locate resources.

An SNI internetwork boundary is *always* within an NCP.

1.2 APPN

APPN first appeared in 1986, and its first proper implementation was on the AS/400. APPN was SNA′s response to the rapid growth of workstations and peer-to-peer networking. Its major characteristics are:

- An APPN network is (as its name implies) organized on a peer-to-peer basis. In fact there *is* a hierarchy, but only to distinguish between network nodes (NNs, which provide all the network services) and end nodes (ENs, which provide only the API and rely on network nodes for the rest). Every node (EN or NN) can own LUs in an APPN network, and every node has a *control point* (CP) from which those LUs request network services. If the node is an EN, the control point must then communicate with an NN to get the request satisfied. Each EN has (at any one time) a single NN from which it requests services. This NN, which must be directly connected to its EN, is called the *network node server* (NNS) of the EN.

  Since there is (virtually) no hierarchy in APPN, there is not the same range of node types defined. Effectively all APPN nodes are type 2.1, and indeed this is how they appear to a subarea network.

- The resource directory is distributed and partner location is dynamic, without the need to predefine the owner of a resource. The NNs maintain the distributed directory in an APPN network. In a large network there are normally one or more *central directory servers* (CDSs), in which the distributed directory tends to be concentrated. The use of CDSs reduces search traffic because a target resource is likely to be found more quickly by asking a CDS than by asking all the other network nodes for its location.

- The network topology is distributed, and is maintained dynamically. The status and characteristics of each node and each link in an APPN network are propagated to all the nodes that require the information. There is no need to define any paths, since route selection is done by comparing node and link characteristics with the values specified in the requested class of service. The NNs maintain the topology database in an APPN network.

- In base APPN, sessions are not grouped in any way, so that routing and flow control are performed individually for each session. With HPR, however, sessions of the same class of service are grouped on logical connections called rapid transport protocol (RTP) connections. An RTP connection, often known as an RTP pipe, can be switched to a different path without affecting the sessions it carries. With HPR, flow control and routing are done on the basis of RTP pipes. The RTP pipes are transmitted through intermediate nodes using a simple, efficient protocol called automatic network routing (ANR).

  In APPN, the only restriction on routing is that only NNs can act as intermediate nodes (with or without HPR). There is no need for a terminal emulator session, for instance, to pass through its attached NCP before wending its way across the network. This is in contrast with subarea networking.

- In APPN there are no SSCPs or PUs, only LUs. The control points are always associated with LUs (and share their names), and network control information is carried on LU-LU sessions between these CP LUs. These control sessions use LU type 6.2 communication, and have their own special class of service. They are referred to as *CP-CP sessions*. CP-CP sessions are always between adjacent nodes, and always occur in pairs of unidirectional sessions. Each session is used for traffic in one direction only.

By contrast, SSCP-SSCP sessions are bidirectional, occur singly (only one between each VTAM pair) and need not be between directly connected VTAMs.

Like subarea, APPN networks can be subnetted. However, there are some important differences:

- APPN subnetting is usually used to isolate topology and broadcast traffic, rather than definitions. There are no definitions in APPN, but the dynamics of APPN mean that more topology updates and network searches tend to flow.
- Multiple APPN subnetworks may have the same network ID, whereas end nodes (only) can have a network ID different from that of their adjacent nodes.
- CP-CP sessions, LU-LU sessions and RTP connections can all cross an APPN border.

An APPN border is *always between* two network nodes with the appropriate border function.

1.3 VTAM Node Types

Because VTAM can participate in both APPN and subarea networks, there are five basic ways in which a VTAM system can be set up.

1.3.1 VTAM Subarea Node

A VTAM subarea node is a node that contains an SSCP, which functions in a hierarchical environment. Subarea nodes provide services for, and control over, peripheral nodes. Peripheral nodes in return require the services of a VTAM subarea node to communicate with other peripheral nodes and subarea nodes in the network. Peripheral nodes (which may be SNA type 2.1, type 2.0 or even type 1) function as distributed processors, cluster controllers or workstations. Type 2.1 nodes may well be APPN nodes, but when they are attached to a subarea network they are restricted to using subarea functions within that network.

The way type 2.1 nodes are connected to a subarea network is known as *low entry networking* (LEN). LEN is effectively APPN without CP-CP sessions, and therefore a small, low-function subset of APPN. In fact, LEN predates APPN since the very first implementations of the AS/400′s predecessors used LEN. They did not have enough processing power for full APPN.

VTAM subarea node characteristics are:

- It requires path definitions to specify all the possible routes sessions may use in and out of this node.
- It supports SSCP-SSCP, SSCP-PU, SSCP-LU, and LU-LU sessions.
- It does not support CP-CP sessions since it has no APPN capability.
- It has a unique subarea number defined.
- It can activate NCPs (in other words, establish SSCP-PU sessions with them), and can have SSCP-SSCP sessions with other VTAM nodes that understand subarea networking. These can be subarea nodes, interchange nodes and migration data hosts.. The last two are described below.

1.3.2 Interchange Node (ICN)

An interchange node resides on the border between an APPN and a subarea network. It provides protocol conversion between APPN and subarea networks to enable the integration of the two types of networks. In addition, it can provide intermediate routing and can therefore establish sessions between subarea and APPN networks. An interchange node combines the functions of a subarea node and an APPN network node. All of the characteristics described for network nodes and subarea nodes apply to interchange nodes.

Interchange node characteristics are:

- It uses subarea path definitions to determine routes within the subarea network.
- It uses the APPN topology database to determine routes within APPN networks.
- It uses both SSCP-SSCP and CP-CP sessions to communicate with other nodes.
- It supports the establishment of LU-LU sessions that pass through it, between the APPN and subarea networks.
- It has a subarea number assigned.
- It can act as a CDS.
- It can own and activate NCPs. When a VTAM ICN owns one or more NCPs, the combination of VTAM and NCP(s) comprises a single APPN node and is called a *composite network node* (CNN).

1.3.3 Migration Data Host (MDH)

A migration data host resides on the border between an APPN and a subarea network. A migration data host combines the function of an APPN end node with the functions and role of a subarea data host. An MDH is dedicated to processing application programs. Like a data host in a subarea network, it does not control network resources, except for its own local resources such as directly attached terminals. All of the characteristics described for end nodes apply to migration data hosts.

Migration data host characteristics are:

- It uses subarea network routing definitions.
- It does not perform intermediate routing, except wholly within the subarea network.
- It uses CP-CP and SSCP-SSCP sessions to communicate with other nodes.
- It requires a network node server to help it establish a session with an APPN resource.
- It cannot support sessions that pass through it between the APPN and subarea networks.
- It has a subarea number defined.
- It cannot own or activate an NCP.

1.3.4 VTAM Network Node

A VTAM NN supports its own end users as well as any end nodes it serves by providing directory and route selection services. It performs intermediate routing of data for sessions that cross it. The NN performs searches of the network to locate resources, and can calculate the best route between the nodes containing the session partner LUs. An NN can also act as a network node server for one or more ENs.

To be able to perform these network services NNs maintain a topology database and a directory services database. The directory services database contains information on resources in the network, including the network node′s local resources, and is updated dynamically as a result of searches for partner LUs. The topology database contains information on network nodes and TGs and is updated dynamically whenever there is a change to the intermediate routing network.

VTAM network node characteristics are:

- It provides intermediate network routing for APPN.
- It maintains CP-CP sessions with adjacent network nodes and with all end nodes for which it provides network services.
- It can establish SSCP-PU and SSCP-LU sessions with peripheral node resources, but no SSCP-SSCP sessions with other VTAMs.
- It participates in searches for network resources.
- It can act as a CDS.
- It does not have a subarea number defined.
- It cannot own or activate an NCP, since subarea protocols are required for this.

1.3.5 VTAM End Node

A VTAM EN is an APPN node that provides communication services for those resources defined within the node. It also provides directory and session services for its own LUs only. For directory and routing services to other nodes, an end node uses the services of its network node server. An EN can have links to multiple network nodes for connectivity and routing, but only one of these network nodes can be used as a network node server at any given time. All resources owned by an end node can be registered in the directory services database of its network node server, as well as with a CDS. This allows searches from other nodes to locate the resources on this end node, since VTAM end nodes do not allow themselves to be searched for previously unknown resources. An EN does not perform intermediate routing of sessions; it can only be the origin or destination of a session.

VTAM end node characteristics are:

- It interacts with a network node server to request and receive directory and route selection services.
- It supports CP-CP sessions between the EN and its NNS.
- It can establish SSCP-PU and SSCP-LU sessions with peripheral node resources, but no SSCP-SSCP sessions with other VTAMs.

- It submits search requests to its NNS and obtains the session route from its server when initiating LU-LU sessions.
- It dynamically registers its resources with its NNS, and optionally with a CDS selected by the NNS.
- It does not have a subarea number defined.
- It cannot activate an NCP.

1.4 VTAM Connection Types

In subarea networking, connections between VTAMs and NCPs are called *transmission groups* (TGs), and are identified in the network topology by TG numbers. Connections between subarea nodes (VTAMs or NCPs) and peripheral nodes are not called TGs, since they are not regarded as part of the backbone network.

In APPN, *all* connections between nodes are called TGs, and are identified by TG numbers in the network topology. Since VTAM is able to perform both subarea and APPN functions, there are four types of connections that must be distinguished from each other. These are described in the subsequent sections.

In an APPN network each TG has a set of *TG characteristics* associated with it, which are used in selecting the best route for a particular session. These comprise:

1. TG capacity
2. Propagation delay
3. Cost per byte
4. Cost per connect time
5. Security
6. Three parameters reserved for use by the installation

Every APPN TG is represented twice in the topology database (once in each direction), so the same link can have different characteristics in each direction.

1.4.1 Subarea TG

A subarea TG is always used between routing nodes (VTAMs and NCPs) in a subarea network, regardless of whether APPN is also being used within that network. A subarea TG is also referred to as a FID-4 TG or a FID-4 connection.

Virtual routes in a subarea network always flow over subarea TGs. Such TGs can be between:

- SSCP and SSCP
- SSCP and NCP
- NCP and NCP

1.4.2 APPN TG or Boundary Function TG

A subarea network uses FID-4 headers (format identifiers) on subarea TGs, but uses FID-2 headers on peripheral links. Peripheral nodes have no network routing or flow control responsibilities, so they do not need the full 26 bytes of a FID-4 header and content themselves with just six bytes. The translation of the full subarea protocols to the simpler peripheral protocols is done by the *boundary function* in the VTAM or NCP to which the peripheral node is attached.

In a pure APPN network (no subarea or HPR), all connections use FID-2 headers. They are similar but not identical to those used on subarea peripheral links. In APPN the routing and flow control information is held in tables in the intermediate network nodes, so there is no need to send it in the packet headers.

When a subarea network is connected to an APPN node, it sees FID-2 packets on the link and therefore treats it as a peripheral link. It recognizes the differences between the two forms of FID-2 header (which distinguish a type 2 from a type 2.1 node in subarea terms), but they are still FID-2 headers. Therefore, the boundary function must translate between the APPN and subarea protocols in a similar fashion to what it does for type 2 nodes. Such a connection is therefore referred to as a *boundary function TG* (BF-TG), or an APPN TG. To the subarea part of VTAM it is not a TG at all, but to the APPN part of VTAM it is a normal TG.

Although HPR connections do not use FID-2 packets, HPR is an extension of APPN and therefore an HPR connection is also treated as a BF-TG by VTAM.

Any connection between VTAM, NCP and other APPN nodes can be implemented as an APPN TG, *except* those between a VTAM and any NCPs it owns. This is because subarea protocols are always required. There is no ownership concept in APPN, and an NCP cannot act as an APPN node on its own.

When VTAM (or its NCP) is connected to any non-VTAM APPN node, the connection is always seen by VTAM as a BF-TG. All the other types of TGs require knowledge of subarea protocols in the attached node.

1.4.3 Interchange TG

When a VTAM acting as an interchange node connects subarea and APPN networks together, it must ensure that each network sees the other in its own terms. As described above, an APPN connection looks to the subarea network like a peripheral (type 2.1) connection. A subarea connection must also be made known to the APPN network in APPN language, and the only concept APPN understands is the TG. The ICN ensures that resources in the subarea network are viewed from APPN as being on end nodes attached to (and served by) the ICN. Each of these pseudo end nodes is perceived as being connected to the ICN by a special TG whose number is always 254. This is known as an interchange transmission group (IC-TG). The whole session path within the subarea network is seen as a single APPN TG 254, even if it crosses an SNI boundary.

1.4.4 Virtual Route-Based TG

When the same network is required to be used both for subarea and APPN protocols, it is necessary first to define it as a subarea network. Next, we must ensure that APPN protocols can be carried over it in parallel with the subarea protocols. CP-CP sessions must be established between the VTAMs that comprise this dual network. This is accomplished by using subarea (FID-4) protocols to carry the CP-CP sessions over the subarea backbone.

Next, we must ensure that the subarea network is correctly represented in the APPN topology database. Since only VTAMs (not NCPs) can have CPs (and therefore can be APPN nodes), we need a way of mapping inter-VTAM connections, and no others, to the APPN topology.

The obvious (and only) way to do this is to map the SSCP-SSCP sessions to APPN TGs. Each SSCP-SSCP session, together with all the VRs that connect them and their NCPs, is represented to APPN by a special TG always numbered 255. When a session request is received that needs to cross the dual network, the APPN network may calculate a route that uses TG 255. The subarea network then selects a VR from all the possible ones represented by TG 255. This special TG 255 is called a virtual route-based transmission group (VR-TG).

Although only one VR-TG can be defined between any two VTAM nodes, this does not limit the use of this TG to only one virtual route (VR). In fact, the VR-TG is used to represent all possible VRs between any two subareas in the domains of the two VTAMs.

Because a VR-TG requires its partners to be both APPN- and subarea-capable, a VR-TG can be established only between the following types of VTAM nodes:

- ICN to ICN
- ICN to MDH
- MDH to MDH

If subarea and APPN networks are combined without using VR-TG, they are in complete ignorance of each other′s topology and route calculation is done independently. VR-TG allows APPN and subarea networks to be overlaid in such a way that APPN route calculation takes some account of the subarea topology. This leads to more optimal routes and a better performing network.

Because a VR cannot cross an SNI boundary, neither can a VR-TG even though there may be SSCP-SSCP sessions across the boundary. In a mixed network, an SNI boundary can be represented to APPN only as part of an IC-TG.

1.4.5 High-Performance Routing Connections

With a few exceptions, HPR does not affect the nature of the links between APPN and subarea nodes. HPR is implemented by carrying LU-LU sessions on logical connections between APPN nodes, the underlying APPN network being unchanged except for the requirement to support ANR and RTP. Some implications of HPR are:

- An HPR connection cannot use an IC-TG, since an IC-TG is actually pure subarea and only pretends to be APPN.
- An HPR connection *can* use a VR-TG since a VR-TG is APPN.
- A session path can be partially HPR and partially APPN. If some of the nodes on the APPN route do not support HPR, then the RTP connection may be established on part of the route.
- A session path may even comprise HPR, APPN and subarea portions if the APPN portion ends in an IC-TG.

HPR logical connections, and therefore the underlying physical connections, carry data in network layer packets (NLPs) rather than FID-2 packets. A unique format of packet is needed because HPR routing (ANR) is very different from base APPN routing. NLPs and FID-2 packets may be carried on the same link, so that some sessions can be base APPN and some can be HPR.

1.5 APPN Route Calculation

The way a session path is chosen differs greatly in APPN and subarea networks. In each case the process is driven from a *class of service* (COS), but there the resemblance ends.

Route selection in a subarea network is essentially static, in that the installation defines the paths available between any two subarea nodes. The COS then points to a list of paths (virtual routes) in decreasing order of desirability.

In an APPN network there is no predefinition of routes; the choice of session path is dynamic and is based on:

- The TG characteristics described in 1.4, "VTAM Connection Types" on page 7
- The node characteristics which comprise:
    1. Node congestion
    2. Node additional resistance (undesirability of using this node for intermediate routing)

In APPN the session COS name points, not to a list of routes, but to a set of ranges of node and TG characteristics. This table, arranged in decreasing order of desirability, assigns *weights* to the nodes and TGs. Figure 1 illustrates the concept.

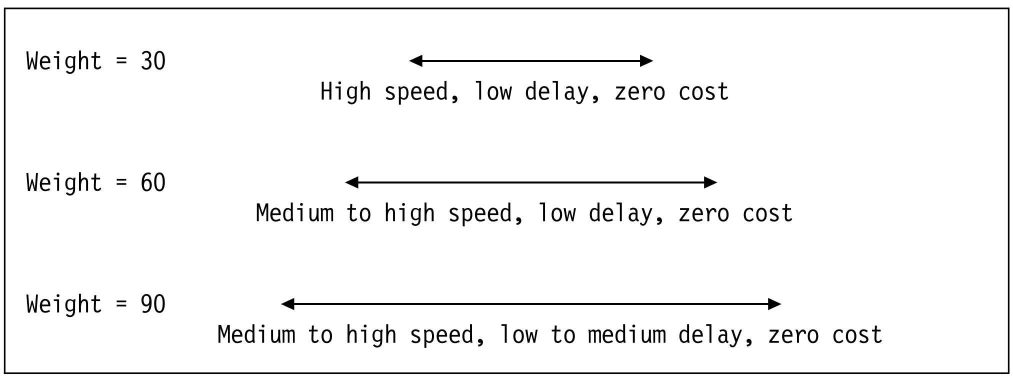


Figure 1. APPN COS Table

In APPN, route choice is the responsibility of the network node server of the primary LU. This is what the NN server does:

- It checks the COS name and looks up the appropriate entry in the COS table.
- For every possible path it matches every possible TG and node on that path against the COS table entry. It checks the actual characteristics of the TG or node against each range in the COS entry, until it finds one that fits. Since the ranges are in decreasing order of desirability (the higher weights are normally less restrictive), the first range chosen will be the closest match to the actual characteristics. The weight associated with this range is assigned to the node or TG.
- In this manner, every intermediate node and every TG on every possible path is assigned a weight.
- Next, every possible route is assigned a weight by adding all the TG and node weights on that route.

- Finally, the route with the lowest overall weight is selected. This will be the route that most closely matches the characteristics defined in the COS table entry.

In addition to the route characteristics, an APPN COS also specifies the transmission priority at which session traffic is to travel. APPN defines four levels of priority, the highest (network priority) being reserved for control traffic such as searches and topology updates.

Among the classes of service defined in APPN are:

1. #BATCH (batch transmission).

   Ideally suited for large file transfers, printing and so on. Data is transmitted at the lowest priority.

2. #INTER (interactive transmission).

   Suitable for interactive transactions that typically send and receive short messages, such as terminal emulation or client/server applications. Data is transmitted at the highest priority next to network priority.

3. #BATCHSC (batch secure)

   To be used when you want to protect the data being transmitted via your networking links. Similar to #BATCH, but with more stringent security requirements.

4. #INTERSC (interactive secure)

   Similar to #INTER, but with more stringent security requirements.

5. #CONNECT (default)

   A medium-priority class of service for general traffic, often used as a default when no more specific requirement exists.

So if you ask APPN for a batch session, it will compute a route that will give you high capacity and low cost. If you need an interactive session, APPN will favor the route with the least propagation delay. If you select secure sessions, APPN will only select secure paths.

If there are no routes available that match the minimum required service level as defined in the COS, the session will fail.

By altering certain values, you can control the routes that will be selected during the route calculation process. The three user-defined TG characteristics are provided for exactly this purpose.

Chapter 2. Introducing NetDA/2

Network Design and Analysis/2 is an OS/2 based, object-oriented, network design tool with support for multiple functions and network types. The redbook *NetDA/2 V1R5 Design Tool Guide and Tutorial* SG24-4225-03 provides a general treatment of all functions. We refer to it as the *NetDA/2 redbook*. This redbook is a supplement concentrating on NetDA/2′s use for APPN networks.

The reference manual is the *NetDA/2 User′s Guide* SC31-6440-05.

2.1 Network Input and Network Design Objects

We prepare a network definition in the form of a network input (NI) object. We then read this into a network design (ND) object. Most of the functions are accessed through the NI and ND menus. These include:

- Graphical input
- Migration and conversion input, mainly for subarea portions of networks
- Topology design
- Subarea routing
- Routing statements generation for subarea and APPN
- Frame relay PVC generation and DLCI numbering
- Simulation of networks including some TCP/IP and ATM
- Extensive report creation

Figure 2 on page 14 is taken from the NetDA/2 redbook and may be used as a reference when you need more detail on general topics. Various methods of building the NI are shown at the top of the figure, most relating to the automatic capture of data from subarea networks. For APPN networks we use the graphical editor.

The circled 8 means that chapter 8 of the NetDA/2 redbook covers the graphical editor. Similarly, chapters 21, 22 and 23 deal with design and simulation. Chapter 17 has some APPN specific information, mostly included here.

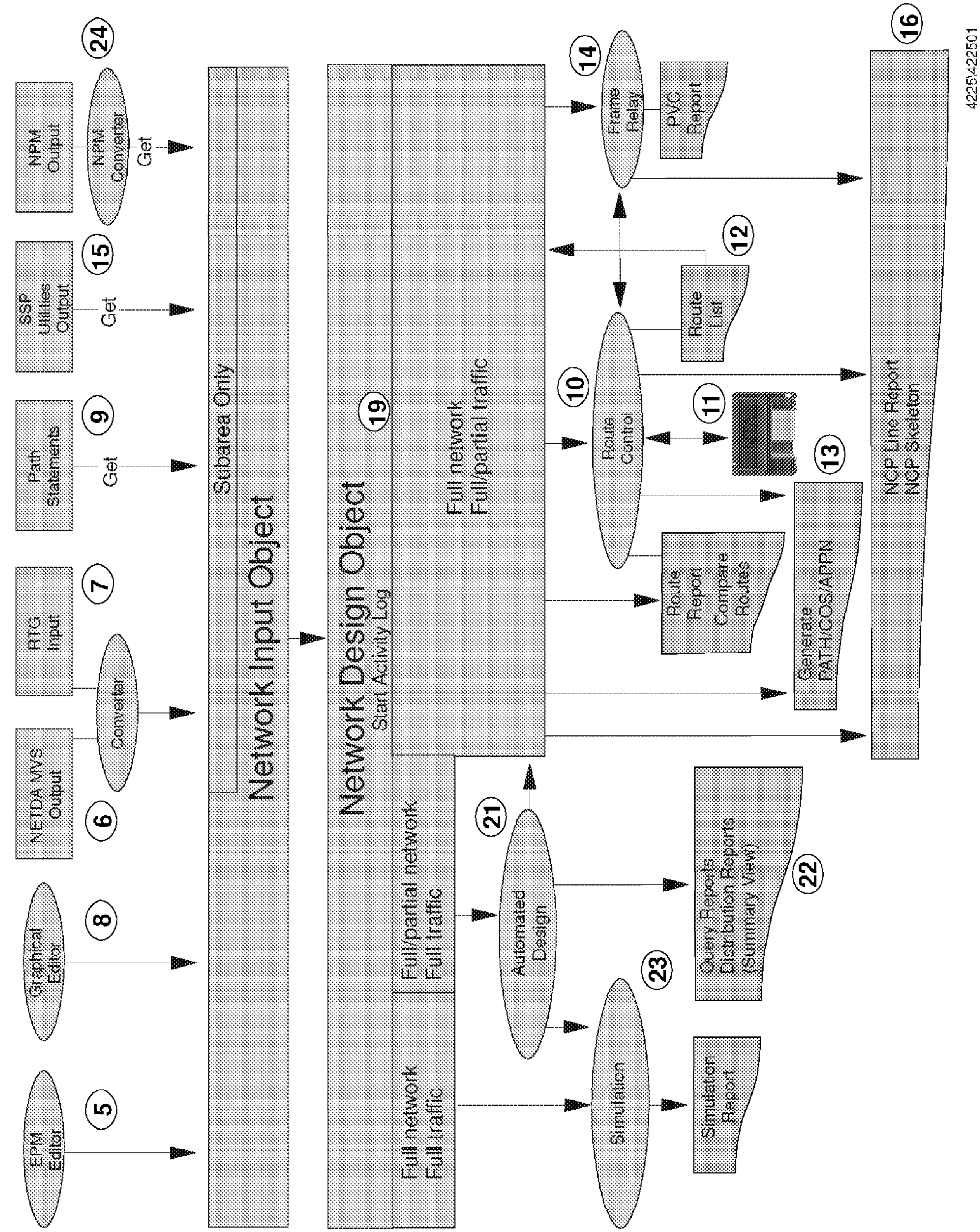


Figure 2. NetDA/2 Flow

Now back to this redbook. We first address pure APPN networks, and later consider mixed APPN and subarea networks. The functions are addressed in the following sequence:

1. Graphical input to build the NI.
2. Designing the network. Some by-products of this process are still of interest even when the topology of our network is already known.
3. Class of Service.
4. Routing reports.
5. Influencing the routes.
6. Data collection.
7. High-Performance Routing.
8. Simulation.
9. Mixed APPN and subarea networks.

Chapter 3. Graphical Input of APPN Networks

The network input and network design objects are basically flat files of tables, with some associated characteristics. That means we may create the NI using either the graphical user interface or our choice of flat file editor. The GUI is much the simpler because it removes the need for us to know the details of the tables, but there a few cases where we must use a flat file editor such as the OS/2 Enhanced Editor, EPM.

Where do we start? After installation of NetDA/2, our desktop folder will contain an icon representing the NetDA/2 folder (see Figure 3).

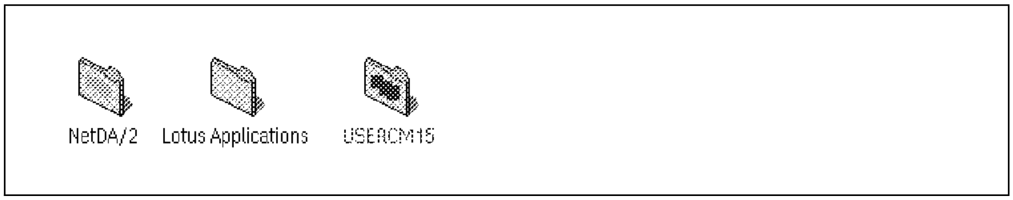


Figure 3. Part of a Desktop Folder

Double-clicking mouse button one on the **NetDA/2** icon opens that folder and displays up to six icons representing its contents (see Figure 4).

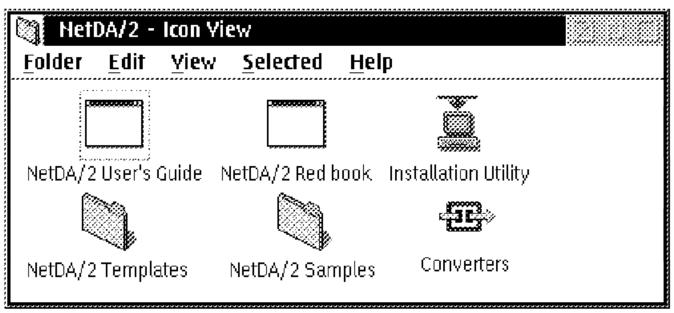


Figure 4. NetDA/2 Folder—Icon View

Two of the icons represent folders: the NetDA/2 Templates folder and the NetDA/2 Samples folder.

Double-click on the **NetDA/2 Samples** folder to reveal eight icons (see Figure 5). Six icons are samples of network input objects, one contains sample parameters for automated design, and the eighth is a network design object.

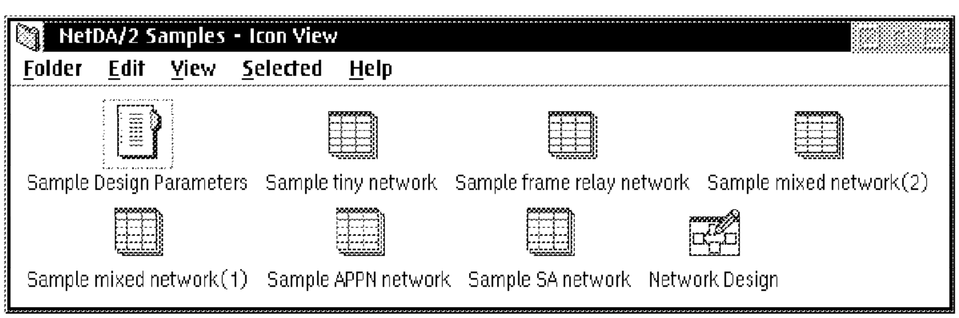


Figure 5. NetDA/2 Samples Folder—Icon View

Opening the NetDA/2 Templates folder displays seven templates (see Figure 6 on page 18).

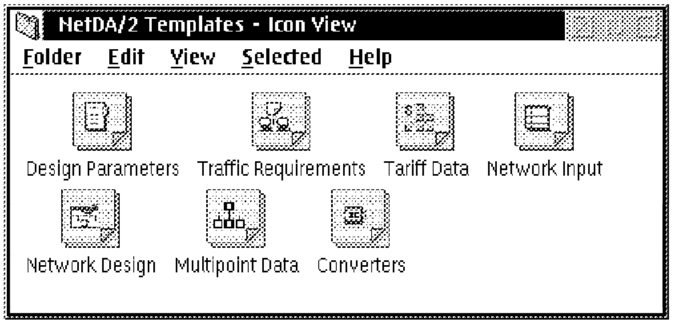


Figure 6. NetDA/2 Templates Folder—Icon View

The Templates folder contains the basic building blocks for using the tool. Many users will need only two of the templates, Network Input and Network Design.

3.1.1 Creating a Folder

The next step is to create a folder in which to place our objects. We could work in a NetDA folder or on the desktop but to keep things tidy we create a new folder. We click with the right button on any existing folder, select **Create another** and **Desktop**.

We change the title of the folder to APPN and click on **Create**. Double-click to open the folder which, of course, is empty.

3.1.2 Tidying the Desktop

So far we have opened the following folders on the desktop:

- NetDA/2
- NetDA/2 Templates
- NetDA/2 Samples
- APPN

We drag and drop an ND from the Network Design icon in the Templates folder to the APPN folder, and then copy the Sample tiny network from the Samples folder to our APPN folder. We could build our NI from a template but it will be simpler and quicker to work from the sample.

We then close the three NetDA/2 folders and are left with the APPN folder open (see Figure 7). We rename the objects to APPN1.NIO and APPN1.NDO. We will use these objects for network input and network design.

Figure 7. APPN Folder

The NI content is shown in Figure 8 on page 19. The ND is empty at this stage.

```
%TABLE QOS_DEFNS
%NAME++
 HIGH

%TABLE NETIDS
%NAME
 NETA

%TABLE SUBNETS
%NAME  TYPE
 SUBA  SA

%TABLE EQUIPMENT
%EQPNAME
 @_HOST
 @_NCP

%TABLE LOCATIONS
%NAME EQPNAME TYPE AC_EX+
 1    @_HOST  T5   919250
 2    @_HOST  T5   919250

%TABLE LOCATIONS_OUT
%NAME NET_NAME SA_NUMBER SUBNET_NAME
 1    NETA     1         SUBA
 2    NETA     2         SUBA
*
%TABLE LINETYPES
%LNTY SVTY SPEEDIN LENGTH_BREAK COST_RATE2

%TABLE MESSAGES
%MSGNAME MSGIN MSGOT QOS+++
 MSG1    1024  1024  HIGH

%TABLE REQUIREMENTS
%SOURCE DEST MSGNAME NUM_SESS MRATEOT MRATEIN
 1      2    MSG1    1        1       1
*
%TABLE GLOBALPARMS
%PARAMETER++++ VALUE
 NETID_CONN    1
```

Figure 8. Sample Tiny Network Input

3.2 Network Input Object

The network information used by NetDA/2 relates to:

- Nodes
- Connections
- Traffic
- Class of service
- Routing
- Design constraints

The amount of detail needed about each aspect is dependent on the functions we wish to use. We will always require the type of each node and its location, although an indefinite type may be specified, for example, NNEN. This allows NetDA/2 to choose whether a node should be a network node (NN) or end node (EN) when it designs the topology.

If we wish to design, then obviously we must provide details of traffic, of the PTT services and costs and equipment connectivity. If we are not going to design a new topology, then we have to provide details of all links.

The first table in Figure 8, QOS_DEFNS, is one relating to class of service. We′ll defer discussion of the class of service information and its entry using graphical input until Chapter 5, "Class of Service" on page 45.

The next five tables provide information about the nodes. EQUIPMENT must provide detailed connectivity and cost information if used for design but otherwise most values may be defaulted as shown.

LINETYPES is to connections as EQUIPMENT is to nodes. It gives details of line speeds and may also include simple, distance-dependent tariffs. Other methods of providing cost information for use with topology design are described in the NetDA/2 redbook and the *NetDA/2 User′s Guide*.

The design function will need traffic detail but other functions require only a nominal definition, as shown.

We may wish to invoke design, although the network is fully defined, so that NetDA/2 will add various values to our tables. In such a case, specifying NETID_CONN=1 in the GLOBALPARMS table will prevent addition of new TGs to our network.

Let′s get to work on our definitions.

3.3 Graphical Editing

Clicking button two on APPN1.NIO, the Network Input object, causes its pop-up menu to be displayed (Figure 9).

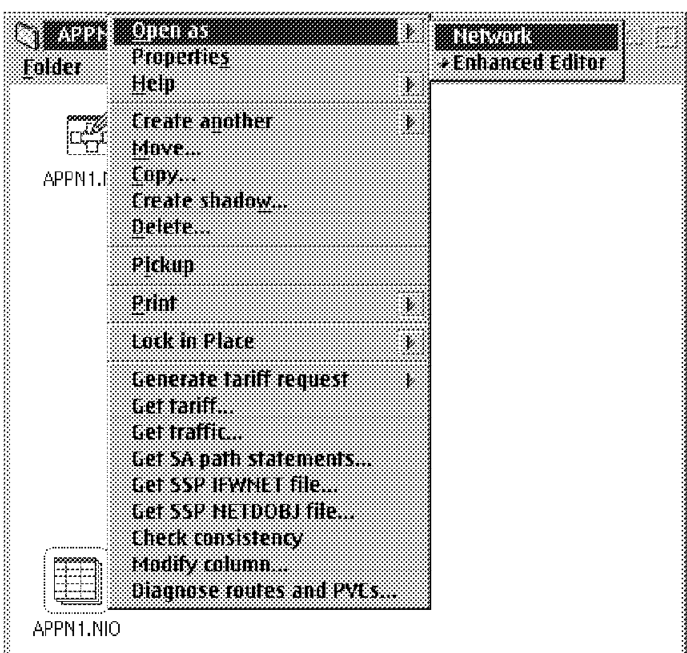


Figure 9. Pop-Up Menu of the Network Input Object

The final nine items in the pop-up menu are NetDA/2 options, some of which we discuss later. The others are OS/2 Workplace Shell items.

Clicking on the arrow to the right of **Open as** gives the cascaded menu of views (see Figure 9 on page 20). We click on **Network** and see a map with the two nodes. By changing to the layout view (click on **View** and **Layout**) we see a network diagram.

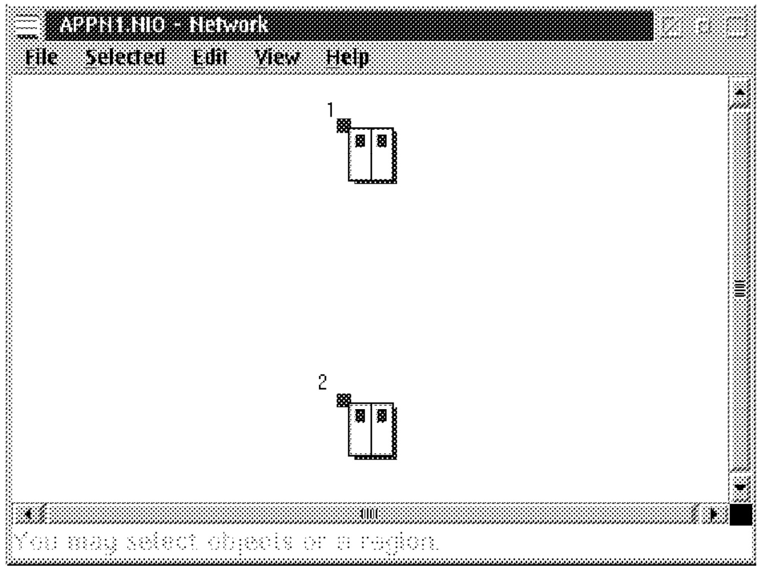


Figure 10. Network with Two Nodes

The initial sample includes definitions for two VTAM subarea nodes, subareas 1 and 2, one NETID named NETA, and one subnet SUBA. NETA should correspond with the VTAM NETID. SUBA is a name used by NetDA/2 to reference this subnet, and each subarea or APPN subnet must be given a unique name.

We will be defining NETIDs, subnets, nodes, and TGs. We suggest defining in that order because we need nodes before we can define the TGs between them. Other definitions we may want to supply include equipment and line types, particularly if we use the design facility, but we′ll ignore those for the present. We would define new equipment before node definition, and new line types before TG definition.

There are several ways of adding, modifying and deleting objects.

A summary appears in Figure 11 on page 23. The flow starts at the top and ends at the panels listed on the right. The main methods are:

- Clicking on Edit and then on one of the following:
  - Nodes
  - Connections
  - Class Of Service
  - NETIDs/Subnets
- Clicking on Edit, on Add object and then on an object type.
- Clicking to select an object and then choosing Selected on the menu bar, followed by Delete or Change.

- Double-clicking on an existing node or TG. This is normally the quickest way to make modifications. It is equivalent to using Change in the previous option and is not shown explicitly in Figure 11.

Changing a TG number requires deletion of the TG and definition of a new one. This will delete the definition of any subarea routes using the TG.

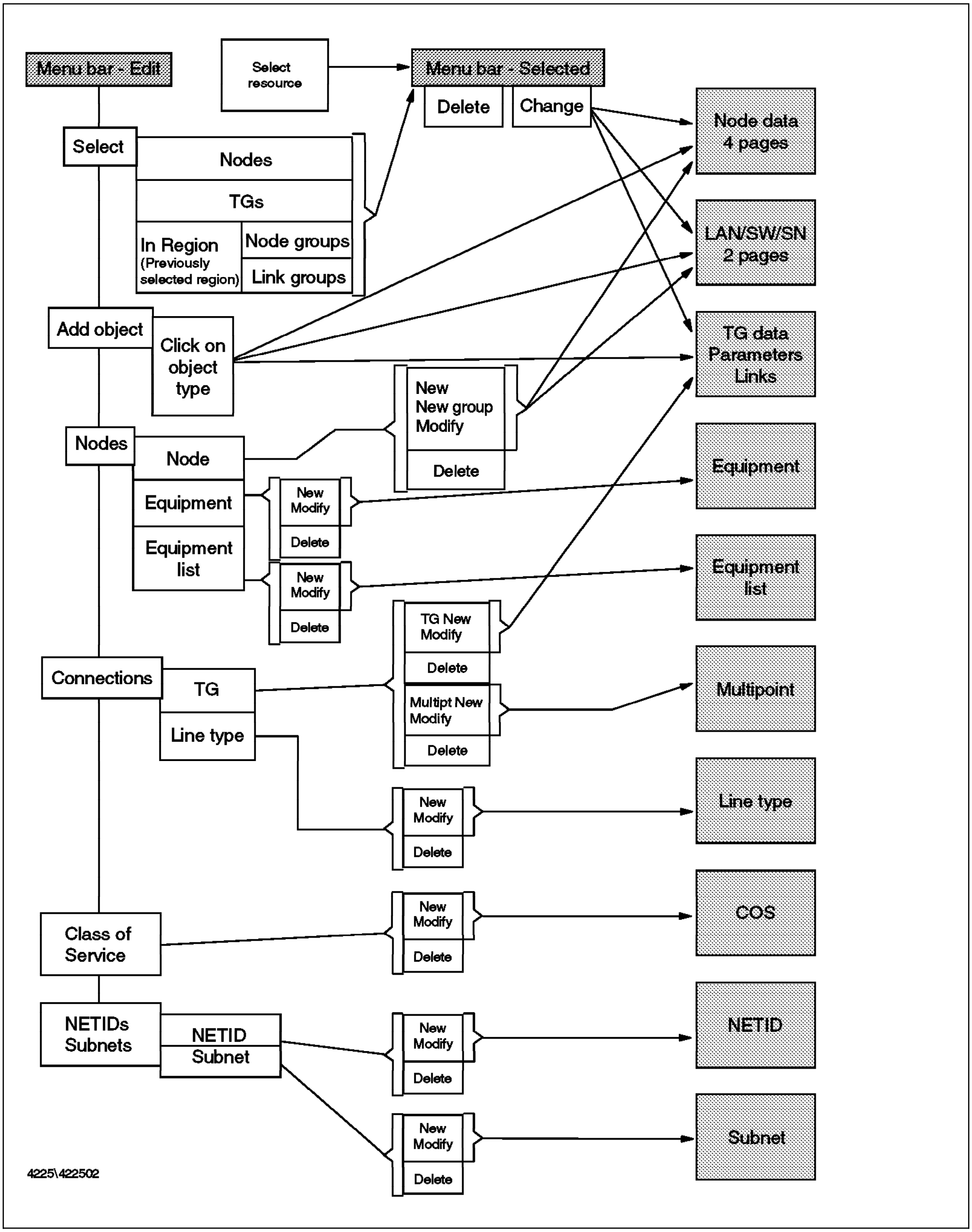


Figure 11. Summary of Graphic Edit Options

We mainly illustrate use of the last method. Other examples appear in the NetDA/2 redbook, Chapter 8, Graphical Editing.

3.3.1 Network with a Network Node and Two End Nodes

We start by defining a network node named NN1 and end nodes EN1 and EN2, each linked to the network node by a TG. The APPN network will have a NETID of NET1 and a subnet named (for NetDA/2′s use) SUB1.

Note that a subnet in NetDA/2 terms differs from SNA′s use of the term. To NetDA/2 a subnet is merely a convenient way of describing part of a network. In SNA a subnet is a distinct piece of a network that is separated from other pieces by an SNI boundary (subarea) or an APPN border. In subarea SNA there is a one-to-one correspondence between NETIDs and subnets, while in APPN there is not even that correspondence.

3.3.1.1 NETIDs and Subnets

We click on **Edit**, then on **NETIDs/Subnets** and **NETID** (see Figure 12).

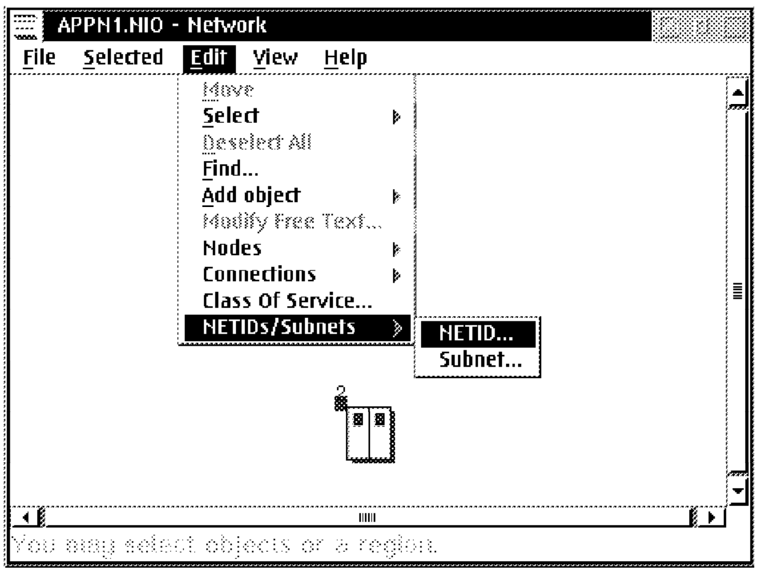


Figure 12. Network with Two Nodes

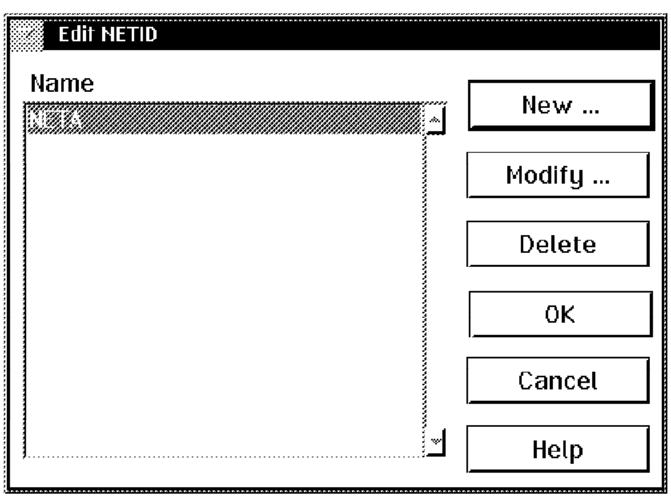


Figure 13. Modifying the NETID

We could select New to add NET1 but it will be tidier if we select **NETA** and **Modify** and change the name to NET1. We click on **OK**, and again on the Edit NETID window.

We repeat this sequence to change the subnet name to SUB1 but this time we start with Edit, NETIDs/Subnets and Subnet. We remember to change the subnet type from SA to APPN.

3.3.1.2 Nodes

Now we change node 1. Instead of the clicking sequence Edit, Nodes, Node, selecting the name 1 and Modify, we just double-click on the icon for node 1.

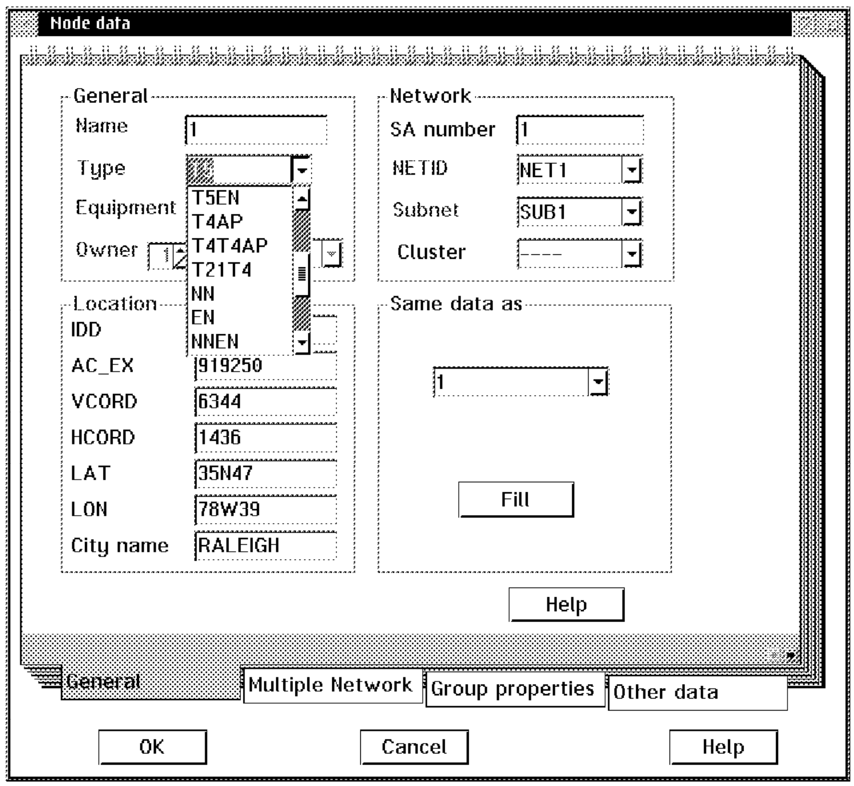


Figure 14. Modifying a Node

In Figure 14 the fields Name, Type, Equipment, Location and NETID are in red to indicate required fields.

We change the name to NN1 and blank the SA number field before selecting **NN** from the pull-down list beside the Type field. Part of the pull-down list is shown. (If we change the Type field first, we will find that the SA number box is no longer accessible.)

Among the node types are T5 for VTAM subarea only, T5NN for a combined subarea and network node (interchange node), T5EN for a migration data host, NN for a network node and EN for an end node. If NetDA/2 is to design the network, several types such as NNEN allow NetDA/2 to choose the node′s role.

If you are following this at a terminal, before selecting OK take a quick look at the other three pages of this notebook. Then click on **OK**.

We change the second node to EN1 in a similar way, but with type EN. Note that by changing the original nodes instead of deleting them and adding new ones, we have kept our nominal traffic definition. The node names in the REQUIREMENTS table will also have been changed.

Now we need a second end node. We click on **Edit**, **Nodes**, **Node** and on **New Node**. From the pull-down list beside Same data as we select **EN1** (see Figure 15).

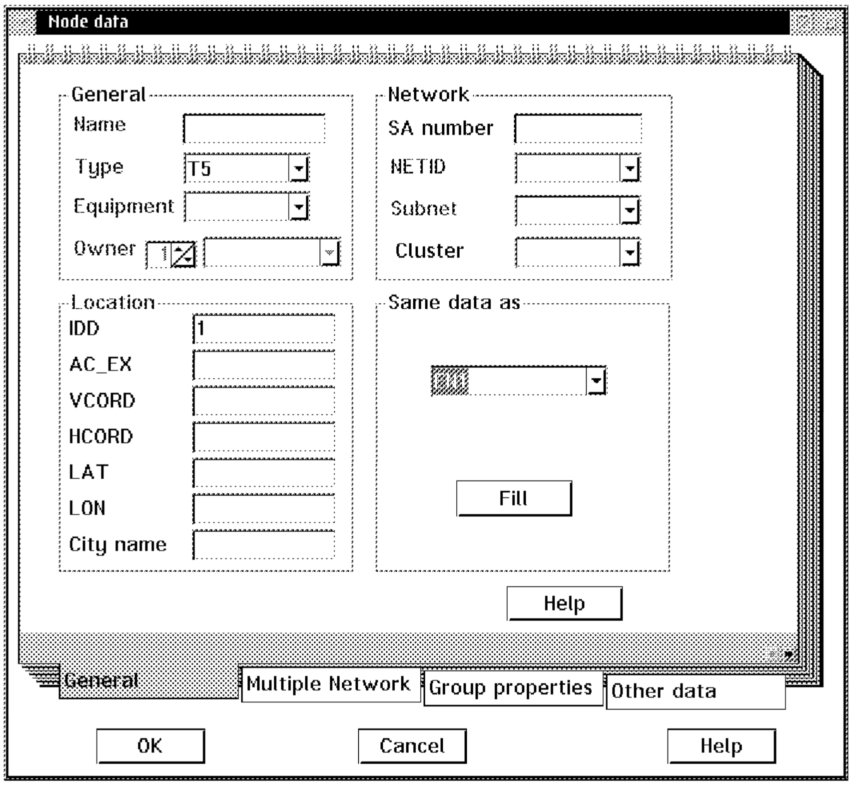


Figure 15. Using Fill

Now the Fill button can be used to fill boxes (apart from the Name field) with the data relating to node EN1. We select **Fill** and then insert the name EN1. We suggest that you enter your node definitions in the best sequence to take advantage of the fill capability.

Sometimes, as here, we need to refresh the layout view to see the change. The simplest way is to select **View** then switch to the map view and back. Occasionally it is necessary to save the NI and re-open the network view.

Next we click on **Edit**, **Connections** and **TGs**. In the New TG box we select **NN1** and **EN1** as the two ends and supply a TG number, here 21.

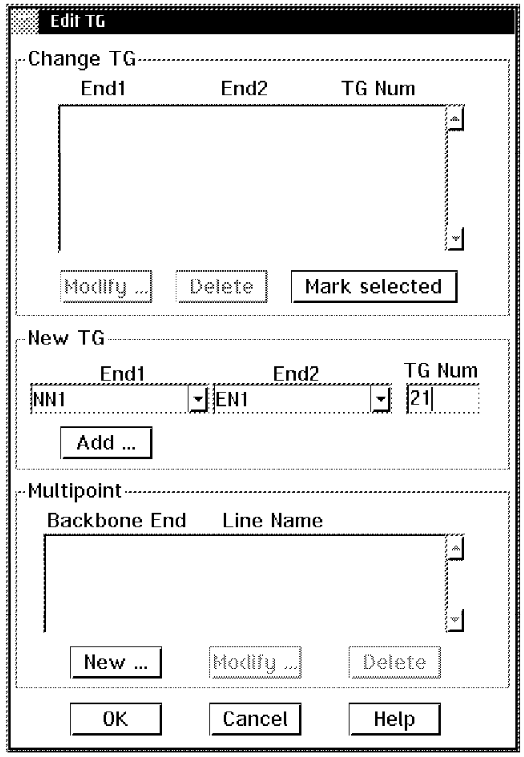


Figure 16. Defining a New TG

This last step enables the Add button. We select that and see the TG Data window. Here we select the 56000 capacity link and change the Multiplicity from 0 to 1.

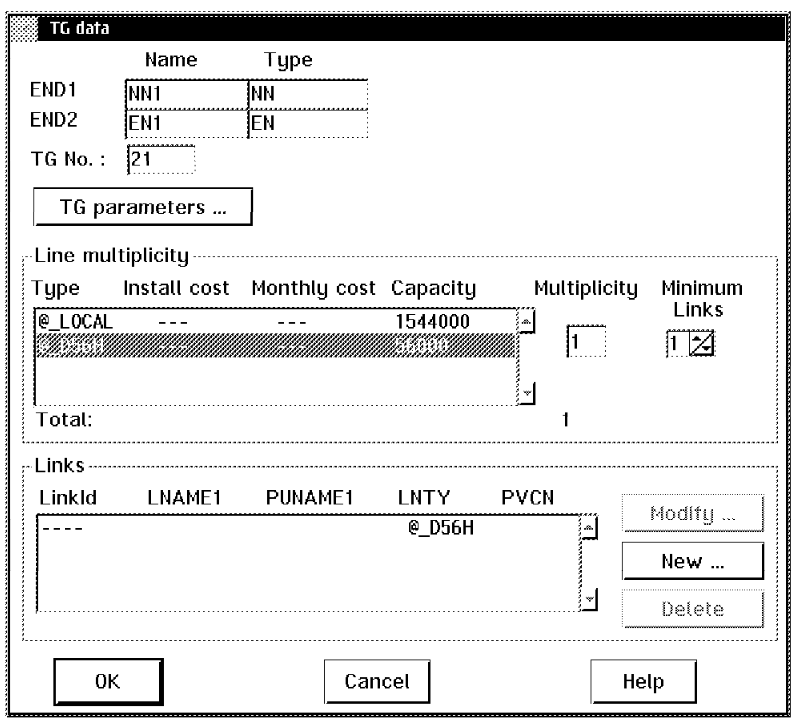


Figure 17. TG Data Window

After clicking on **OK**, on the Edit TG window we select **EN2** as End2 and **Add** to define the second TG.

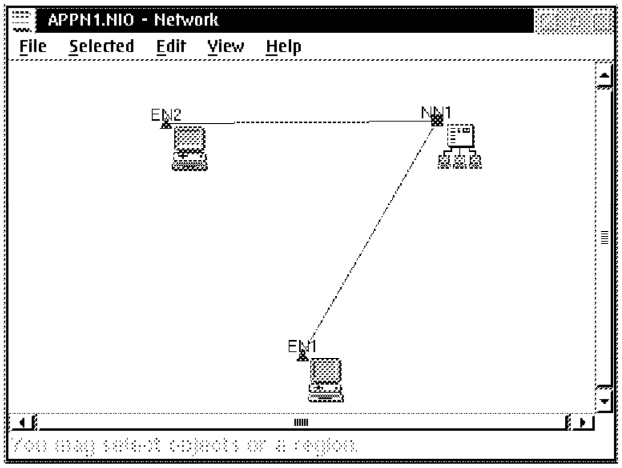


Figure 18. Modified Network

3.3.1.3 Line Types

The default types provided are a 1.5 Mb channel and the 56 kb link. To add alternatives we′d use the layout view Menu bar and select **Edit**, **Connections**, **Line type** and **New**. This shows us Figure 19. We supply a new name, the service type (from the pull-down list) and the speed.

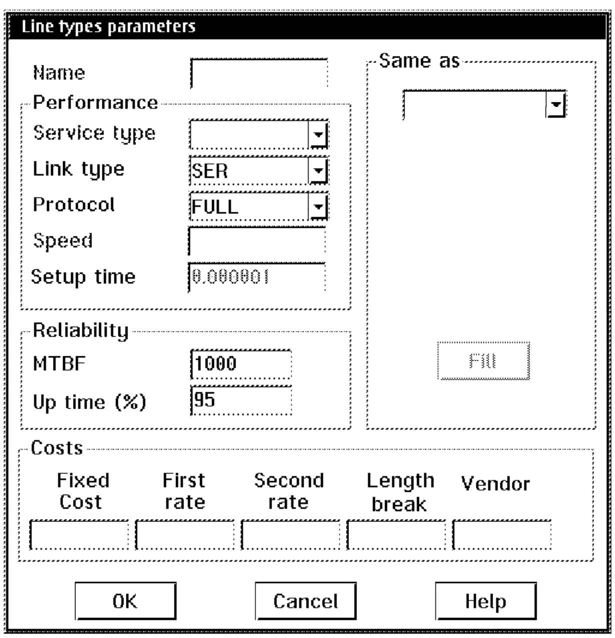


Figure 19. Entering the Line Types Parameters

If we plan to invoke the design function, we should provide some cost information. It is probably useful to enter four 1s in the Costs section even if not expecting to design.

In Figure 20 on page 29 we see one of the default Line types. Click on **Edit, Connections, Linetypes** and then modify each line type to show costs.

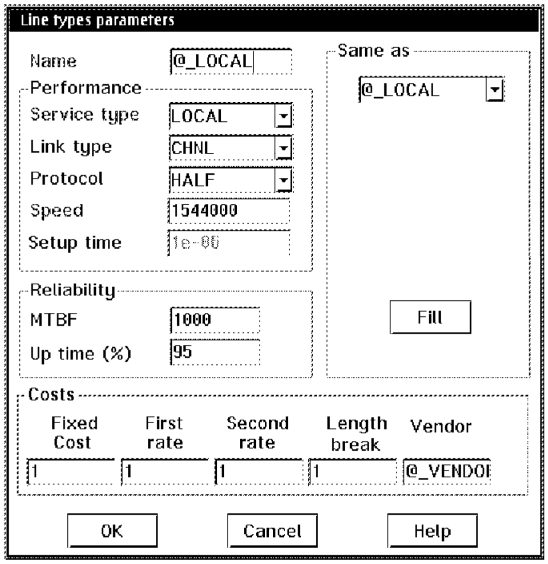


Figure 20. Adding Nominal Tariffs

3.3.1.4 Saving The Definition

Now we select **File** on the menu bar and **Save** and then close the NI object. Let′s take a quick look at the tables, which are shown in Figure 21 on page 30.

Some less important columns have been omitted from Figure 21. We have updated the NETIDS and SUBNETS tables, and taken defaults in the EQUIPMENT table.

In the LOCATIONS table, the telephone numbers have been mapped to VCORD and HCORD coordinates. We would see latitude and longitude if any part of the network were outside of North America. There is more information about the nodes, with extra columns in the LOCATIONS table and new tables LOCATIONS_OUT and LOCATIONS_OPTIONS. Here, and elsewhere, default values have been inserted as a result of using graphical edit.

Two default LINETYPES rows are supplied.

The nominal traffic is described by the unchanged MESSAGES table and the REQUIREMENTS table. In the latter, the node names have been updated to reflect the changes we made.

The two TGs we added are recorded in the last two tables.

```
%TABLE QOS_DEFNS

%NAME PRIO++ INTERACTIVE SECURE++ RELIABLE
 HIGH MEDIUM YES         NONSECUR NO

%TABLE NETIDS

%NAME
 NET1

%TABLE SUBNETS

%NAME TYPE
 SUB1 APPN

%TABLE EQUIPMENT

%EQPNAME COST LS_BRK LSP_COST HSP_COST M_LCL M_CHNL M_LSL M_HSL M_TR M_ETH M_FDDI M_DEG AP_ATT DUP TRND+ more...
 @_HOST  0    56000  0        0        500   50     1000  1000  1000 0     0      1000  NO     YES 1e-06
 @_NCP   0    56000  0        0        500   50     1000  1000  1000 0     0      1000  NO     YES 1e-06

%TABLE LOCATIONS

%NAME EQPNAME MAX_SWITCH_UTIL TYPE AC_EX+ IDD CITYNAME ST VCORD HCORD NAM IRN
 NN1  @_HOST  50              NN   919250 1   RALEIGH  NC 6344  1436  ENA YES
 EN1  @_HOST  50              EN   919250 1   RALEIGH  NC 6344  1436  ENA YES
 EN2  @_HOST  50              EN   919250 1   RALEIGH     6344  1436  ENA YES

%TABLE LOCATIONS_OUT

%NAME EQPNAME TYPE NET_NAME SUBNET_NAME DIR_SERV DLU_SERV DNS QUEUE RAR CON more...
 NN1  @_HOST  NN   NET1     SUB1        POSSIBLE POSSIBLE NO  0     LOW
 EN1  @_HOST  EN   NET1     SUB1        POSSIBLE POSSIBLE NO  0     LOW
 EN2  @_HOST  EN   NET1     SUB1        POSSIBLE POSSIBLE NO  0     LOW

%TABLE LINETYPES

%LNTY+++ SVTY+ PR++ LINKTYP SPEEDIN VENDOR++ FIXED_COST LENGTH_BREAK COST_RATE1 COST_RATE2 UPTIME MTBF SETUP_TIME
 @_LOCAL LOCAL HALF CHNL    1544000 @_VENDOR 1          1            1          1          95     1000 1e-06
 @_D56H  D56   HALF SER     56000   @_VENDOR 1          1            1          1          95     1000 1e-06

%TABLE MESSAGES
%MSGNAME MSGIN MSGOT QOS+++
 MSG1    1024  1024  HIGH

%TABLE REQUIREMENTS

%SOURCE DEST NUM_SESS MSGNAME MRATEOT MRATEIN
 NN1    EN1  1        MSG1    1       1

%TABLE GLOBALPARMS
%PARAMETER++++ VALUE
 NETID_CONN    1

%TABLE LOCATIONS_OPTIONS

%NAME CNN+++++ ER_LIM SA_LIM HPR_TYPE FR_TYPE BNDRY_CONN
 NN1  POSSIBLE 16     255    OT       NO      1
 EN1  POSSIBLE 16     255    OT       NO      1
 EN2  POSSIBLE 16     255    OT       NO      1

%TABLE TRUNKS

%END1 END2 LNTY++ LINK_ID TG_NUM FIXED_TG NUM MIN UD1_12 UD1_21 SEC_12++ SEC_21++ more
 NN1  EN1  @_D56H @0      21     NO       1   1   255    255    NONSECUR NONSECUR
 NN1  EN2  @_D56H @0      21     NO       1   1   255    255    NONSECUR NONSECUR

%TABLE TRUNKS_OUT

%END1 END2 LINK_ID TG_NUM UTIL
 NN1  EN1  @0      21     80
 NN1  EN2  @0      21     80
```

Figure 21. Modified Network Input

Chapter 4. Automated Design

In this example we define an APPN network of about thirty nodes. Figure 22 shows the map view.

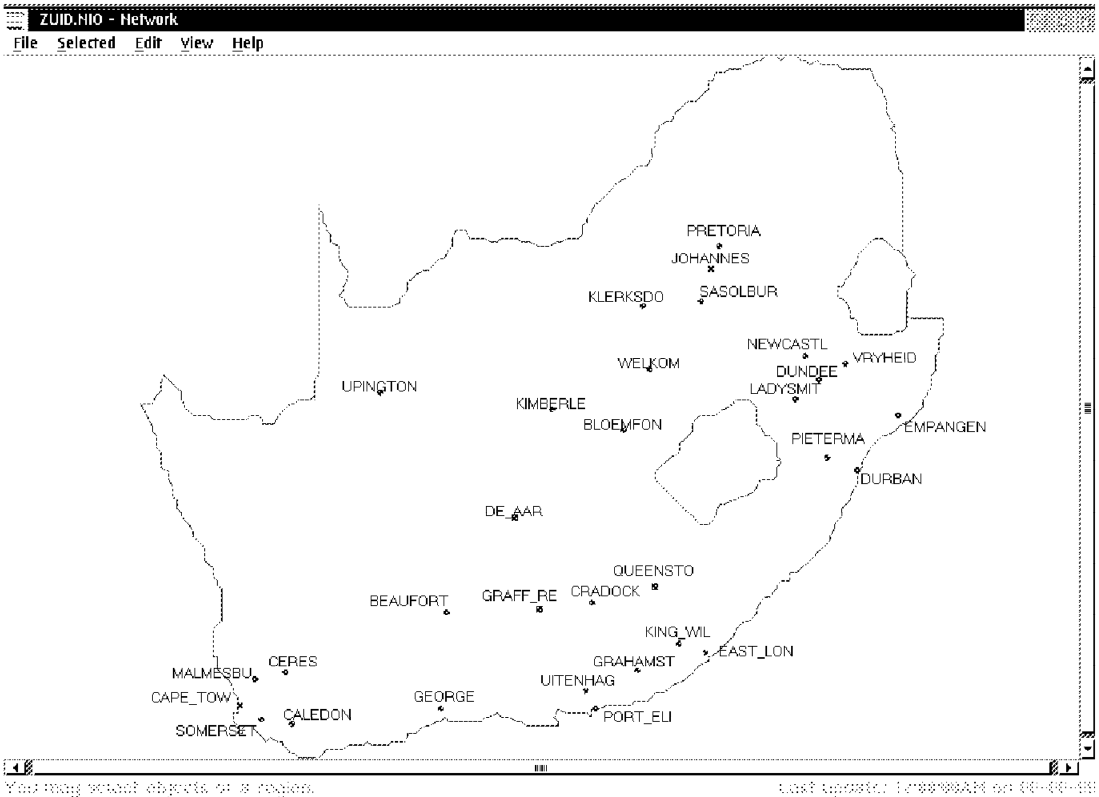


Figure 22. Locations in the Network

4.1 Defining Equipment

To define a piece of equipment, we select **Edit** from the menu bar, and then **Nodes**, **Equipment** and **New**. The default values appear in Figure 23 on page 32.

Only the Name field is shown in red as a required field with defaults being supplied for most others.

Max attachment degrees defines the maximum number of connections that the equipment can support. The Total must be as large as any of the other values but no larger than their sum. The value for Local is not included in these calculations.

We want NetDA/2 to decide on the model to use therefore we include details of attachment capability and costs.

The Costs section gives the base cost and that of each low- and high-speed port, with Low speed break defining the maximum speed of low-speed ports.

The APPN connection box is not used.

The switching speeds are used during design, and most default switching times assume a message handling capability of one million messages per second. If you wish NetDA/2 equipment selection during design to reflect processor utilization you can supply switching times here. By default, NetDA/2 will allow up to 50% of a processor's utilization to be used for switching, assuming that the remaining cycles should be reserved for local processing. The 50% value can be varied when defining each node.

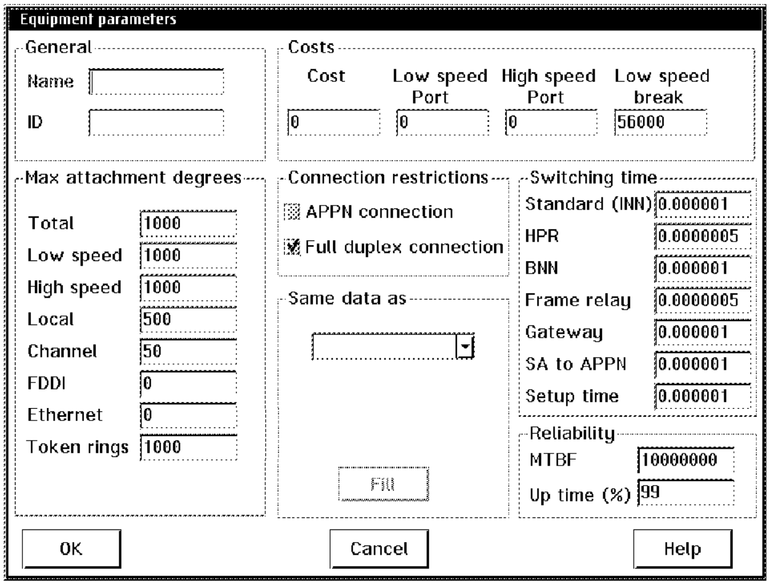


Figure 23. Equipment Parameters

We define two models in our equipment table, one with up to five low-speed and three high-speed ports, the other with up to ten low-speed and five high-speed ports. The smaller model definition is shown in Figure 24 and the larger in Figure 25 on page 33.

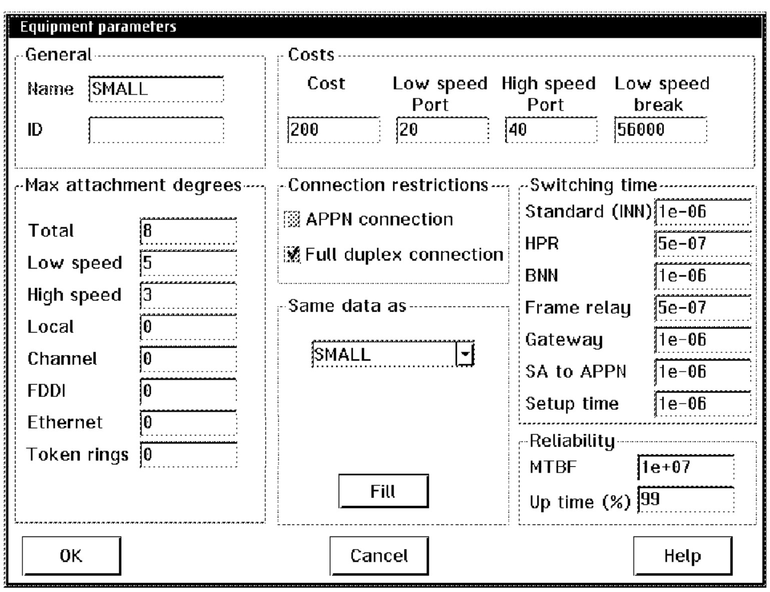


Figure 24. SMALL Equipment Parameters

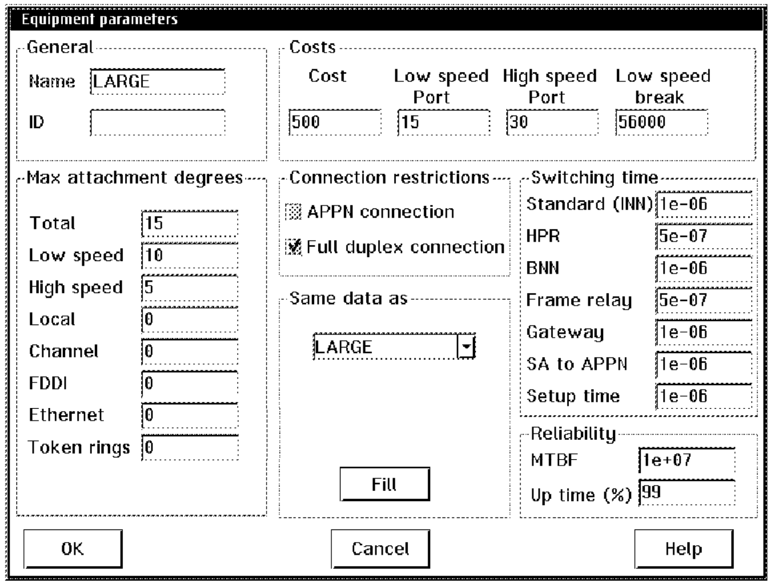


Figure 25. LARGE Equipment Parameters

We ask NetDA/2 to decide which model of equipment is necessary at each location. This will depend on connectivity requirements and costs in our case. As we have defaulted the processing capabilities for each piece of equipment, this constraint is unlikely to affect the design.

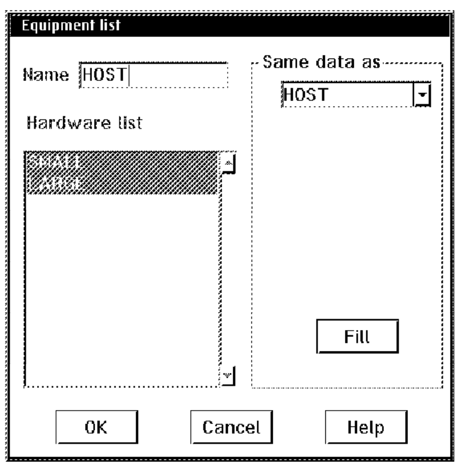


Figure 26. HOST Hardware List

For each node we specified neither LARGE nor SMALL, but HOST, the name of the hardware list, seen in Figure 26. This is how we give NetDA/2 a choice of equipment.

4.2 Entering Nodes

Two locations, DURBAN and JOHANNES, run applications and are defined as TYPE=NN. Cape Town is defined as a third network node. The definition of the Johannes node is shown in Figure 27 on page 34.

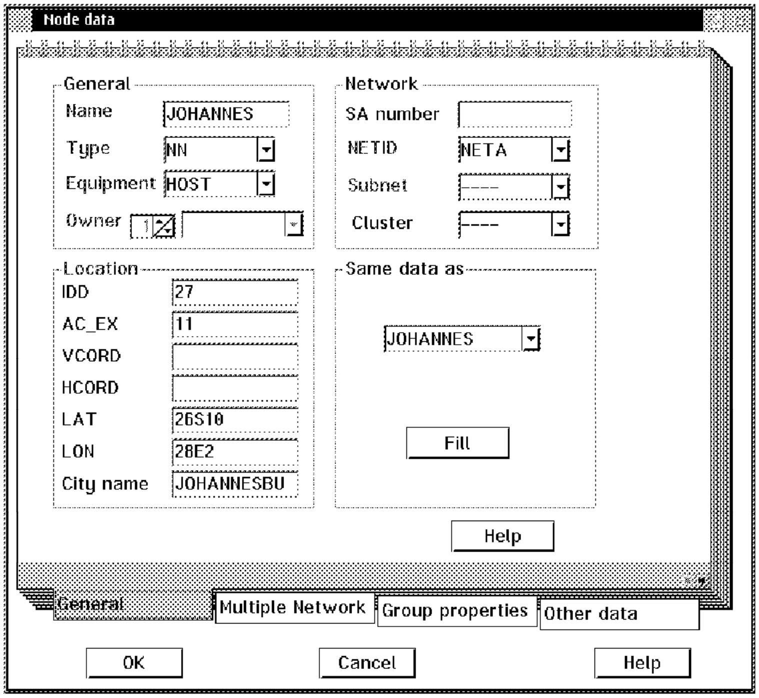


Figure 27. Input Data for Johannes

IDD is the country dialing code and AC_EX is the first part of the location's telephone number. We could supply these or the LAT and LON fields, but telephone numbers are often more readily available. We supply the IDD and AC_EX information and NetDA/2 looks up the latitude and longitude in its telephone number database. The city name is also taken from the database and enables us to check that no error was made in our input.

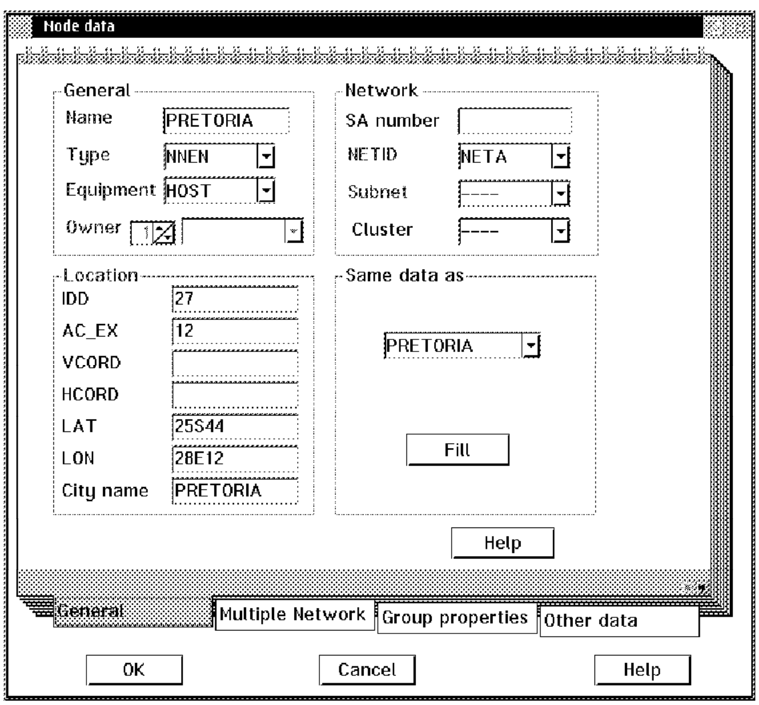


Figure 28. Input Data for Pretoria

The remaining nodes are defined as TYPE=NNEN which means that NetDA/2 can choose during the design whether they are network or end nodes.

Figure 28 on page 34 is an example of an NNEN definition.

The Other Data page includes APPN COS information, shown in Figure 29.

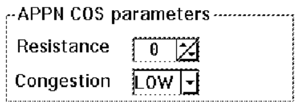


Figure 29. COS Parameters

4.3 Entering Linetypes

We provide details of three PTT services, at 19.2, 64, and 256 kbps. Input data for the first two are shown in Figure 30 and Figure 31 on page 36.

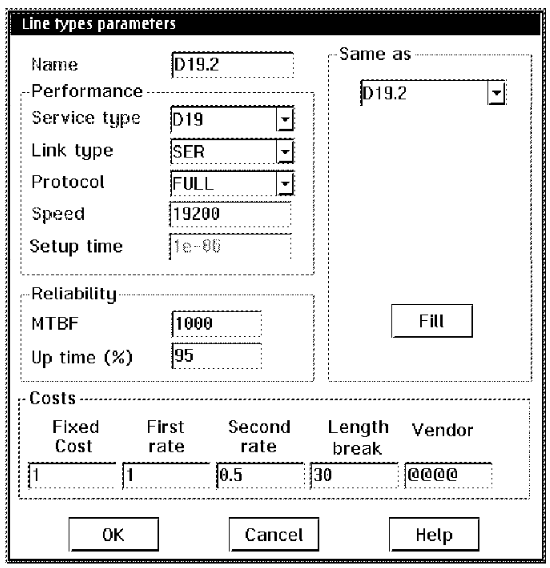


Figure 30. 19.2 kb Line Types Parameters

The Vendor field was filled by the tool and is only of interest if line tariffs will be requested from an external provider.

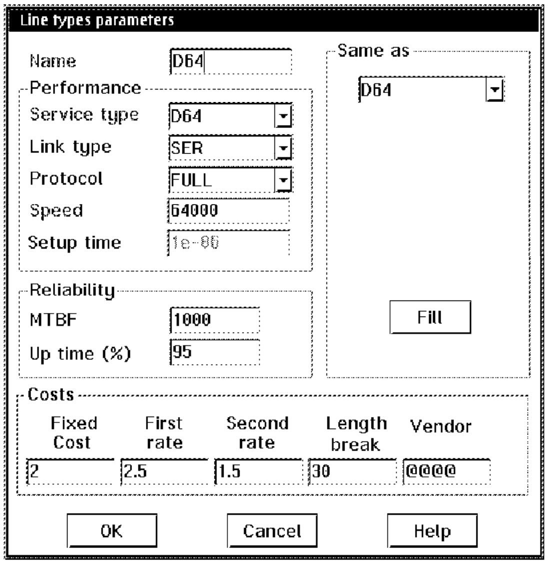


Figure 31. 64 kb Line Types Parameters

4.4 Design

We save the NI and drag and drop its icon on that of the ND. Double-clicking on the ND brings up the network view. We select **Network** on the menu bar and then **Design**. Figure 32 shows the first design.

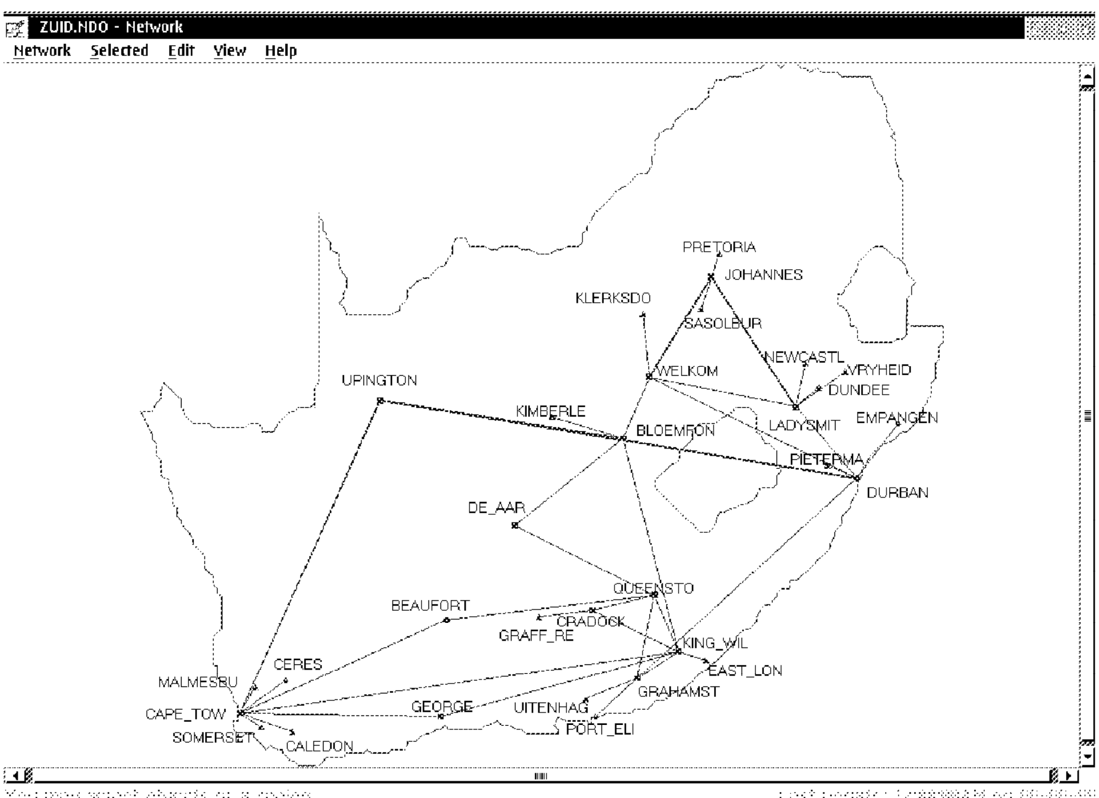


Figure 32. Topology Output

This first design has more NNs than we would wish so we decide to explore other designs and consider a topology with fewer NNs. One way of achieving this is by clicking on **Network**, **Design parameters** and changing Max total backbone nodes. We can also set a minimum value. These appear in the panel shown in Figure 33 from which we can also access the algorithm parameters. The Max total backbone nodes defaults to 1000.

The Algorithm parameters Delay/Cost tradeoff and Backbone density may be used to improve reliability and reduce message delay subject to increased overall cost due to the inclusion of additional backbone links.

The performance section of Figure 33 specifies the design constraints for different message priorities. We default to the values of one second for high priority and two for medium priority. NetDA/2 designs so that the average response time for all messages at a priority level is less than the value specified.

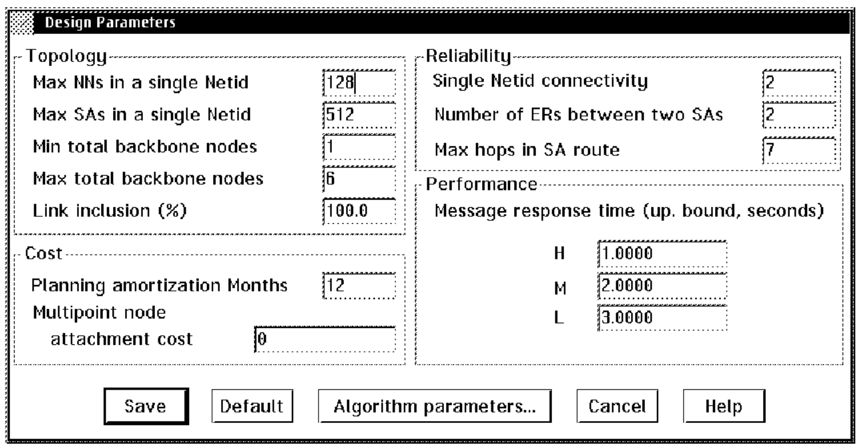


Figure 33. Design Parameters

We design again but this time with Max total backbone nodes set to 6. NetDA/2 is allowed to add up to three NNs to the three we defined. The result is an error message, shown in Figure 34.

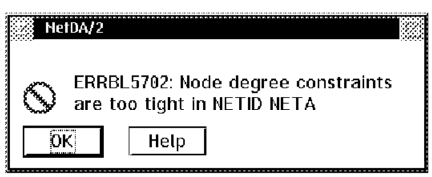


Figure 34. ERRBL5702 Message

A number of user parameters influence the design process and some combinations result in a message rather than a design. The NetDA/2 redbook suggests user actions which may be taken to obtain a design in these circumstances.

Here we simply experiment by changing three more of our nodes to be NNs and design again, without restricting the number of NNs. This time we show the layout view of the design instead of the map view. This view sometimes gives a clearer picture of the balance of the design. In Figure 35 on page 38 we see several clusters of ENs connected to an NN and some stand-alone NNs.

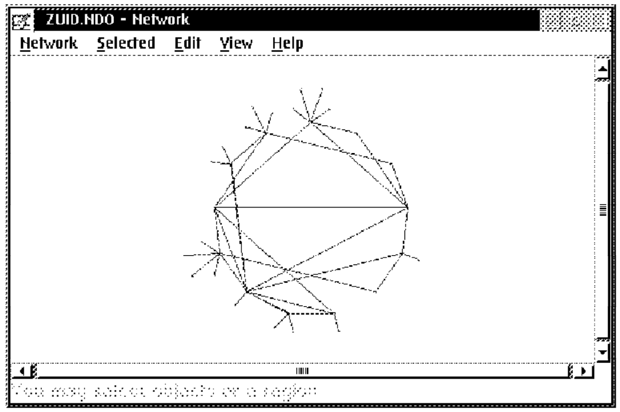


Figure 35. Layout View

As we still have more NNs than we would like, we repeat this design specifying Max backbone nodes of 6.

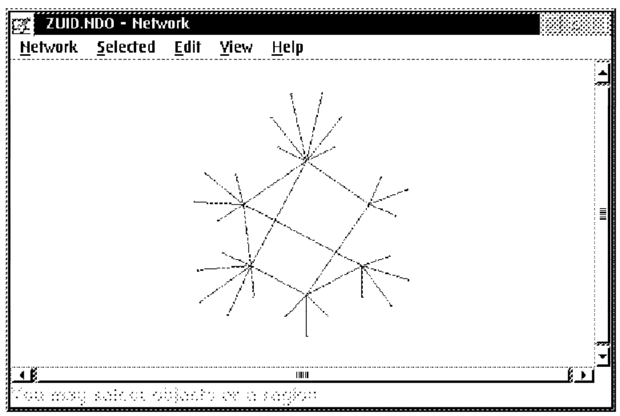


Figure 36. Layout View

This time Figure 36, the layout view, shows clearly the six clusters and the backbone.

Figure 37 shows a map view of just the backbone portion of the network.

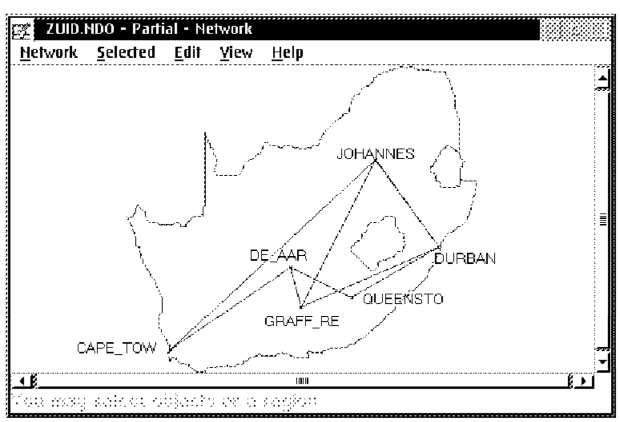


Figure 37. Network Backbone

4.4.1 Reports

We click on **View** and **Summary** to see some statistics about the design. Figure 38 has the design summary. We see the total cost of node hardware and communications costs, response times for the different priority messages and statistics about the topology.

Max NETID Backbone Diameter is 2 so there are never more than two hops between a pair of NNs, with an average of about 1.5. These are good low numbers.

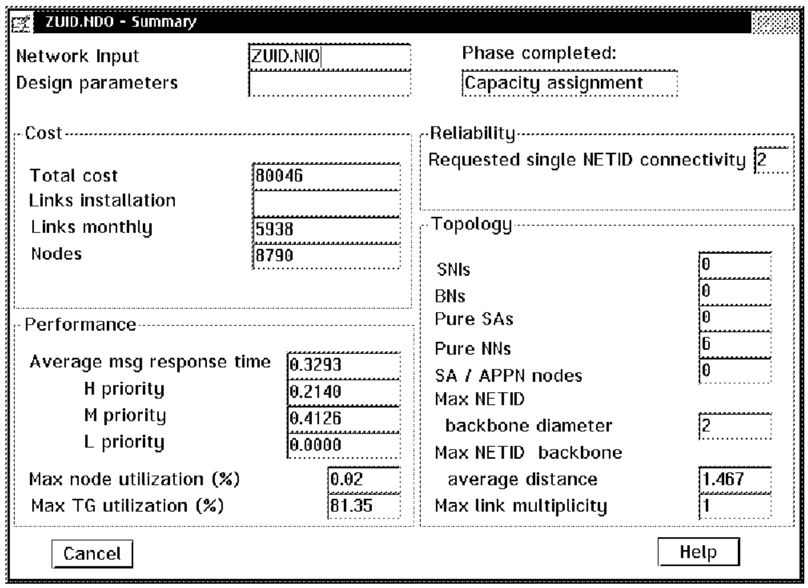


Figure 38. Summary View

Limiting the number of backbone nodes to 6 has improved the performance. The average backbone distance in the first design was about 2.

We dragged the cursor to create a region around the network and selected the nodes and TGs within the network.

After clicking on **Edit** on the menu bar and then on **Select**, we chose most of the options in turn. Next we clicked on **Selected** and **Query**, followed first by **Delay** and then by **Utilization** to obtain reports on the network. Figure 40 on page 41 and Figure 41 on page 42 have the output. The output has been condensed and the normal column headings are less terse.

The range of reports includes:

- Delay
- Utilization
- MTBF
- Uptime
- Cost
- Capacity
- Multiplicity
- Traffic load
- APPN Weights

The statistics cover:

- Nodes
- TGs
- Sessions
- Routes
- QoS
- Applications

Figure 39 is an example of the output. Here we see listed the capacity of each backbone link, followed by the Delay query report in Figure 40 on page 41.

```
Appended at 15:27:40, Wednesday, 10 September 1997.  ZUID.NDO

TG  Capacities - ( Bits/second )

End1        End2        TG      Capacity

CAPE_TOW    DE_AAR      1       19200
CAPE_TOW    JOHANNES    1       64000
DE_AAR      GRAFF_RE    1       19200
DE_AAR      QUEENSTO    1       19200
DURBAN      GRAFF_RE    1       19200
DURBAN      GRAFF_RE    2       19200
DURBAN      JOHANNES    1       64000
DURBAN      JOHANNES    2       64000
DURBAN      QUEENSTO    1       64000
GRAFF_RE    JOHANNES    1       64000
```

Figure 39. Query Capacity Report

Two TGs have been created between DURBAN and GRAFF_RE and between DURBAN and JOHANNES. All of the peripheral links (not shown) are at the lower speed.

In The Query Delay report, delays are shown in five categories. We did not select the NN-EN TGs which would add little but volume to the listings, but their contribution is reflected in the Session Delays.

The TG Delays are shown for each direction, first with the average and then for each priority.

Node delays are not meaningful here as we have defaulted the message processing capabilities.

Route Delays are shown between the selected nodes, here only NNs.

The #INTER values are averages of all session delays for MSG1 traffic, and #CONNECT for MSG2.

Session delays are perhaps the most interesting as we see the breakdown for all locations. The topology is designed to meet our response time requirements for each priority, and these are taken as an average. Inspection of the session delays may indicate further tuning is needed to ensure all locations meet the response time.

```
Listing condensed from D:\APPN\QUERY2 on 10 Sep 1997 at 11:51:24
First timestamp:   Appended at 11:41:46, Wednesday, 10 September 1997.  ZUID.NDO

TG  Delays - ( msec )
 End1     End2     TG  1->2A  1->2H  1->2M  1->2L  2->1A  2->1H  2->1M  2->1L
 CAPE_TOW DE_AAR   1   251.5  ----   251.5  ----   28.59  ----   28.59  ----
 CAPE_TOW JOHANNES 1   17.51  20.40  14.62  ----   64.08  20.40  107.7  ----
 DE_AAR   GRAFF_RE 1   28.59  ----   28.59  ----   251.5  ----   251.5  ----
 DE_AAR   QUEENSTO 1   94.19  94.19  ----   ----   94.19  94.19  ----   ----
 DURBAN   GRAFF_RE 1   103.9  103.9  ----   ----   103.9  103.9  ----   ----
 DURBAN   GRAFF_RE 2   86.16  86.16  ----   ----   86.16  86.16  ----   ----
 DURBAN   JOHANNES 1   12.72  17.97  10.98  ----   84.51  17.97  106.7  ----
 DURBAN   JOHANNES 2   16.82  20.01  14.75  ----   138.3  20.01  214.8  ----
 DURBAN   QUEENSTO 1   64.08  20.40  107.7  ----   17.51  20.40  14.62  ----
 GRAFF_RE JOHANNES 1   8.832  ----   8.832  ----   68.93  ----   68.93  ----

Node  Delays - ( msec )
 Node     Delay
 CAPE_TOW 0.002
 DE_AAR   0.002
 DURBAN   0.002
 GRAFF_RE 0.002
 JOHANNES 0.002
 QUEENSTO 0.002

Route  Delays - ( Sec )
 Source   Dest     ERN      S--->D D--->S  Hops
 DURBAN   CAPE_TOW ---      0.0768 0.1020  DURBAN(1)JOHANNES(1)CAPE_TOW
 DURBAN   CAPE_TOW ---      0.0809 0.1558  DURBAN(2)JOHANNES(1)CAPE_TOW
 DURBAN   DE_AAR   ---      0.1583 0.1117  DURBAN(1)QUEENSTO(1)DE_AAR
 DURBAN   DE_AAR   ---      0.3377 0.1148  DURBAN(2)GRAFF_RE(1)DE_AAR
 DURBAN   GRAFF_RE ---      0.0862 0.0862  DURBAN(2)GRAFF_RE
 DURBAN   GRAFF_RE ---      0.1039 0.1039  DURBAN(1)GRAFF_RE
 DURBAN   JOHANNES ---      0.0127 0.0845  DURBAN(1)JOHANNES
 DURBAN   JOHANNES ---      0.0168 0.1383  DURBAN(2)JOHANNES
 JOHANNES DURBAN   ---      0.1383 0.0168  JOHANNES(2)DURBAN
 JOHANNES DURBAN   ---      0.0845 0.0127  JOHANNES(1)DURBAN
 JOHANNES QUEENSTO ---      0.2023 0.0343  JOHANNES(2)DURBAN(1)QUEENSTO
 JOHANNES QUEENSTO ---      0.1486 0.0302  JOHANNES(1)DURBAN(1)QUEENSTO

QOS  Delays - ( Sec )
 QOS      NETID             D--->S S--->D
 #CONNECT NETA              0.3764 0.0409
 #INTER   NETA              0.1052 0.1052

Application  Delays - ( Sec )
 Node     Application       D--->S S--->D
 DURBAN   MSG1              0.1067 0.1067
 JOHANNES MSG2              0.3682 0.0403

Session  Delays - ( Sec )
 Node     Appl     Dest     D--->S S--->D
 DURBAN   MSG1     BEAUFORT 0.1725 0.1725
 DURBAN   MSG1     BLOEMFON 0.1832 0.1832
 DURBAN   MSG1     CALEDON  0.1091 0.1091
 DURBAN   MSG1     CAPE_TOW 0.0404 0.0404
 DURBAN   MSG1     CERES    0.1070 0.1070
 DURBAN   MSG1     CRADOCK  0.1548 0.1548
 DURBAN   MSG1     DE_AAR   0.1146 0.1146
 DURBAN   MSG1     DUNDEE   0.0687 0.0687
 DURBAN   MSG1     EAST_LON 0.0891 0.0891
 DURBAN   MSG1     EMPANGEN 0.0687 0.0687
 DURBAN   MSG1     GEORGE   0.1548 0.1548
 DURBAN   MSG1     GRAFF_RE 0.1039 0.1039
 DURBAN   MSG1     JOHANNES 0.0200 0.0200
 JOHANNES MSG2     DURBAN   0.1067 0.0110
 JOHANNES MSG2     GRAHAMST 0.5740 0.0580
 JOHANNES MSG2     KIMBERLE 0.5719 0.0660
 . . .
 . . .
```

Figure 40. Query Delay Reports

```
TG  Utilizations - ( % )
 End1     End2     TG 1---->2  2----->1
 CAPE_TOW DE_AAR   1  50.00    10.00
 CAPE_TOW JOHANNES 1  28.20    64.20
 DE_AAR   GRAFF_RE 1  10.00    50.00
 DE_AAR   QUEENSTO 1  42.67    42.67
 DURBAN   GRAFF_RE 1  48.00    48.00
 DURBAN   GRAFF_RE 2  37.33    37.33
 DURBAN   JOHANNES 1  19.25    64.25
 DURBAN   JOHANNES 2  30.35    81.35
 DURBAN   QUEENSTO 1  64.20    28.20
 GRAFF_RE JOHANNES 1  9.000    45.00

Node  Utilizations - ( % )
 Node     Utilization
 CAPE_TOW 0.0048
 DE_AAR   0.0032
 DURBAN   0.0168
 GRAFF_RE 0.0056
 JOHANNES 0.0176
 QUEENSTO 0.0048
```

Figure 41. Query Utilization Reports.

Figure 41 shows the TG and Node utilizations. Again, the node values result from the use of defaults. Another way of viewing utilizations is to select **View**, **Show** and **Capacity** which results in the TGs and nodes being displayed on the map or layout view in a color according to their utilization. Five colors were used, representing 20% bands.

4.4.2 Failure Reports

Although we defaulted to a design with a two-connected backbone, giving us full connectivity in the event of a single failure, the alternative connections do not necessarily provide capacity to meet the response time requirements if a failure does occur.

We clicked to select one of the two TGs between DURBAN and GRAFF_RE, then on **Selected** and **Fail**. Displaying the utilizations with this TG being unavailable caused changes in several of the TG colors, reflecting the new loads.

The new TG utilization report appears in Figure 42 on page 43.

```
Listing condensed from D:\APPN\CAP4 on 10 Sep 1997 at 12:20:33
First timestamp:   Appended at 12:18:07, Wednesday, 10 September 1997.  ZUID.NDO

TG  Utilizations - ( % )
 End1     End2     TG 1---->2  2----->1
 CAPE_TOW DE_AAR   1  37.50    7.500
 CAPE_TOW JOHANNES 1  27.45    60.45
 DE_AAR   GRAFF_RE 1  12.50    62.50
 DE_AAR   QUEENSTO 1  42.67    42.67
 DURBAN   GRAFF_RE 1  85.33    85.33
 DURBAN   JOHANNES 1  23.10    77.10
 DURBAN   JOHANNES 2  26.50    68.50
 DURBAN   QUEENSTO 1  64.20    28.20
 GRAFF_RE JOHANNES 1  9.750    48.75

Node  Utilizations - ( % )
 Node     Utilization
 CAPE_TOW 0.0046
 DE_AAR   0.0032
 DURBAN   0.0168
 GRAFF_RE 0.0058
 JOHANNES 0.0176
 QUEENSTO 0.0048
```

Figure 42. Utilization Report after Failure

Unfortunately, if the result of failure is that the network is unable to support the traffic, the request for a display or report simply receives a message such as that in Figure 43.

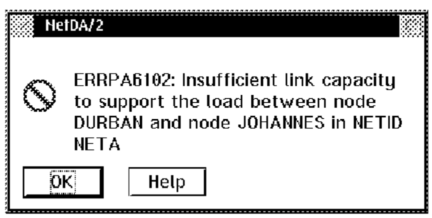


Figure 43. Insufficient Capacity

There is more about this topic in Chapter 13, “Capacity Planning” on page 149.

Chapter 5. Class of Service

Figure 44. Part of a Desktop Folder

In a pure APPN environment, with a single NETID and no subnet boundaries, computation of the route to be used between two nodes for a particular COS is straightforward. All nodes and TGs have characteristics, which, in conjunction with the COS tables, establish a weight for each component in a route.

The network node computing the route has the characteristics in its topology database. Suppose we have a pair of nodes connected by two TGs. One link runs at 2M, the other at 19.2k. Suppose also that we choose which TG to use based on the columns in Figure 45.

ROW	MAX_SPEED	MIN_SPEED	WEIGHT
1	MAXIMUM	16M	1
2	MAXIMUM	64K	20
3	MAXIMUM	10K	100

Figure 45. Determining The TG Weight

We start with the 2M TG. Its capacity satisfies the MAX_SPEED criterion of row 1 but is less than the MIN_SPEED. It satisfies both criteria in row 2, and therefore has a WEIGHT of 20.

Note: MAX_SPEED and MIN_SPEED should really be MAX_CAP and MIN_CAP. APPN does not have a speed characteristic, only capacity, although the result is the same when working with single link TGs.

The 19.2K TG doesn′t satisfy the MIN_SPEED criterion of row 1, or row 2, but satisfies MAX_SPEED and MIN_SPEED in row 3. It therefore has a weight of 100. The route using the 2M TG is chosen because it has the lower weight.

When evaluating routes with multiple hops, this process is performed to determine a weight for each intermediate node and TG in each route, and thereby the route with the lowest total weight.

Seem too simple? We cheated. There are more columns to be checked. In addition to the line capacity, there are maximum and minimum columns for propagation delay, cost per byte, cost per unit of time, security and three user values. All of these must be compared before we can decide which row applies. They are compared to a TG′s characteristics to determine the first row with criteria satisfied by a TG, and therefore the weight to be assigned.

The characteristics held in a topology database will have been supplied by the nodes at both ends of the TG, and all of the characteristics have values for both directions. If the values for each direction are identical, as is generally recommended, then the route weight will be the same regardless of which end initiates the session. NetDA/2 allows for specification of a single TG capacity, applicable to both directions.

Now that we have introduced the concept, let's take a more detailed look at choosing routes using NetDA/2. We start with how to supply the component characteristics and the COS information, then look at the process of selecting a route.

5.1.1 Node COS Parameters

In Chapter 3, "Graphical Input of APPN Networks" on page 17 we described graphic edit of node information. Now we are going to take a closer look at the APPN node characteristics.

On the third page of the notebook, we can select parameters that will affect the way APPN will route the traffic, as shown in Figure 46, under APPN COS parameters.

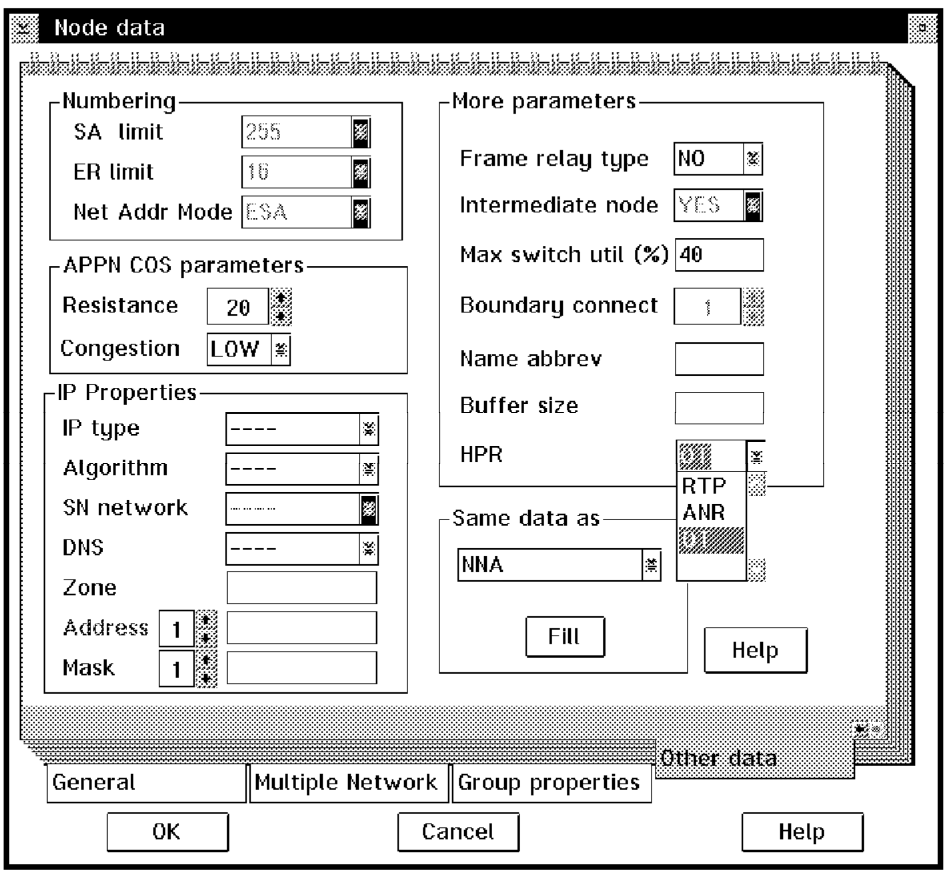


Figure 46. COS Node Definition

Resistance or Route Addition Resistance (RAR) and Congestion will be checked by NetDA/2 while determining the route to choose between two endpoints.

In real life, Congestion is dynamically set by the node, based on the node CPU and buffer utilization. In NetDA/2 it is assumed to be static; nevertheless, it is possible to manually change its value and check how it influences the route selection. Congestion has a binary value: zero or one in APPN architecture, Low or High in VTAM and NetDA/2. If Congestion is set to High, this node will be assigned a high weight when being considered for use as an intermediate routing node on a new APPN route.

RAR is equivalent to the weight of the node; the higher the value, the less likely it will be selected as part of a new route. This parameter is also dynamic (set by the node), but most of the implementations set it as a static value.

RAR and Congestion have meaning when the node acts as an intermediate node in an APPN route.

There are some other APPN node parameters that are not supported in NetDA/2, such as Quiescing and Intermediate Routing Resources Depleted, but their influence in route establishment can somehow be taken into account with the APPN node characteristics defined in NetDA/2. In any case they are not part of the COS table; for route calculation purposes they are treated as meaning that the node is inactive.

In an already installed APPN network, you can get the APPN characteristics of the nodes by issuing commands. For example, from VTAM you can issue:

```
d net,topo,id=node_name,list=all
```

If NetView is installed, you can use the TOPO CLIST to get the APPN characteristics of each node. In VTAM, RAR is known as ROUTERES.

If the network has AS/400 nodes, you can get the APPN node characteristics of them by issuing the AS/400 command DSPAPPNINF.

On a Communications Server/2 workstation, you can obtain the APPN node characteristics by invoking the **Display Active Configuration** panel from the **Administration** folder.

5.1.2 TG APPN Parameters

Figure 47 on page 48 shows the APPN parameters associated with a TG that can be set in NetDA/2.

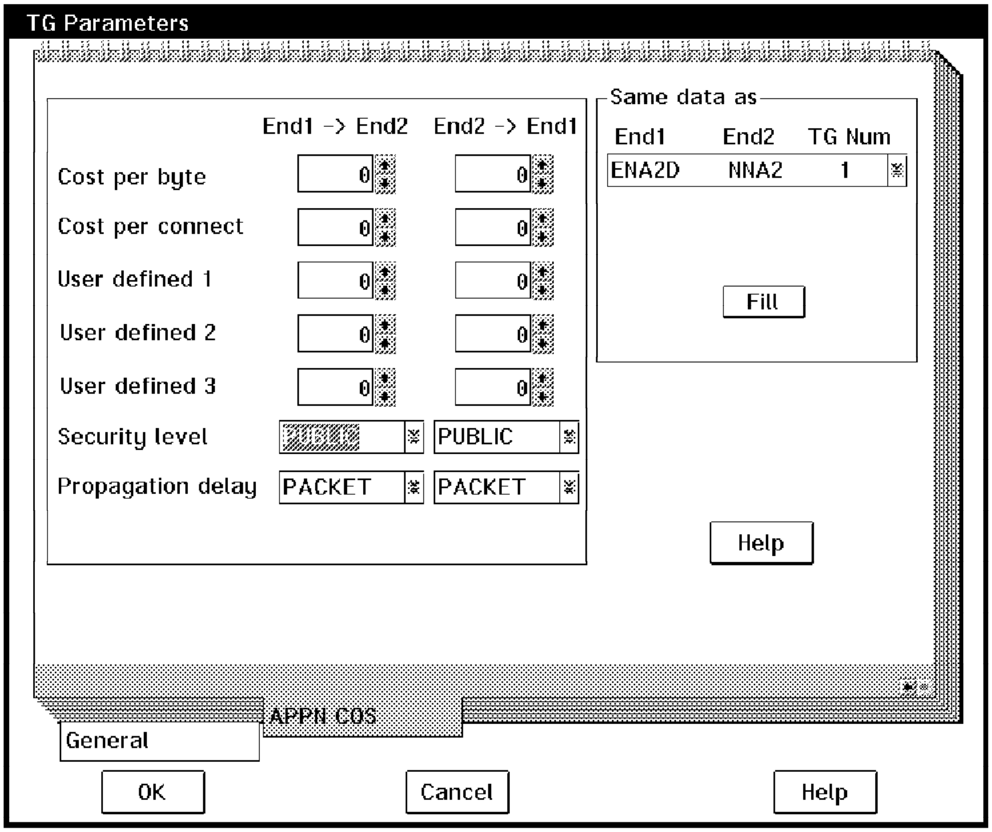


Figure 47. TG APPN Definition

These characteristics are all static, both in real life and in NetDA/2 (except for Propagation Delay, as defined in the APPN architecture).

There are a few APPN TG characteristics that are not included in this panel, such as Quiescing, Garbage Collection Indicator, etc., but they are either not related to NetDA/2 functions or can be somehow simulated using other capabilities in the product.

In an already installed APPN network, you can get the APPN characteristics of the TGs by issuing commands. For example, from VTAM you can issue the following to get the APPN characteristics of the TG between two nodes:

```
d net,topo,orig=node_name1,dest=node_name2
```

If the network has AS/400 nodes, you can get the APPN TG characteristics of the the links by issuing the AS/400 command DSPAPPNINF.

On a Communications Server/2 workstation, you can obtain the APPN TG characteristics by invoking the **Subsystem Management**, **Details**, **SNA Subsystem**, **Logical Links**, **Link** and **Details** panels from the **Administration** folder.

5.1.3 COS Definitions

Figure 48 on page 49 is a graphical representation of how the APPN COS table looks.

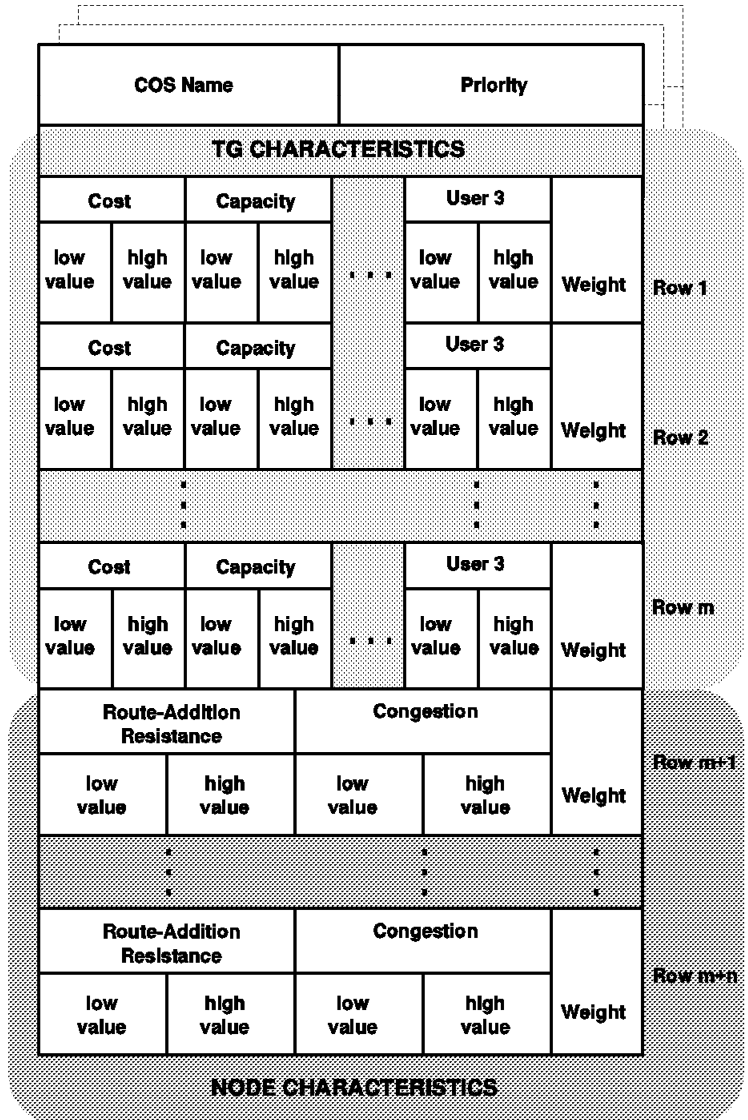


Figure 48. COS Entry with m Rows of TG Characteristics and n Rows of Node Characteristics

As you can see at the top of Figure 48, the COS Name identifies each one of the multiple entries in a COS table (DCOS for NetDA/2). Each one of these entries has two tables appended: the TG Characteristics table (TGCOS for NetDA/2) and the Node Characteristics table (NCOS for NetDA/2).

Each one of these tables has several rows. The contents of the different fields in each row of the TGCOS table are the same as those defined in 5.1.2, "TG APPN Parameters" on page 47. The contents of the different fields in each row of the NCOS table are the same as those defined in Figure 54 on page 53.

There is an extra field in each of the rows in both tables: the Weight.

Let′s see how we define in NetDA/2 the DCOS, NCOS and TGCOS tables. Later we try to put everything together; in other words, we see how APPN or NetDA/2 uses all this information to select routes.

5.1.4 APPN COS Table

Figure 49 shows the panel where we select/modify/delete the different COS entries of the COS table, or add a new one.

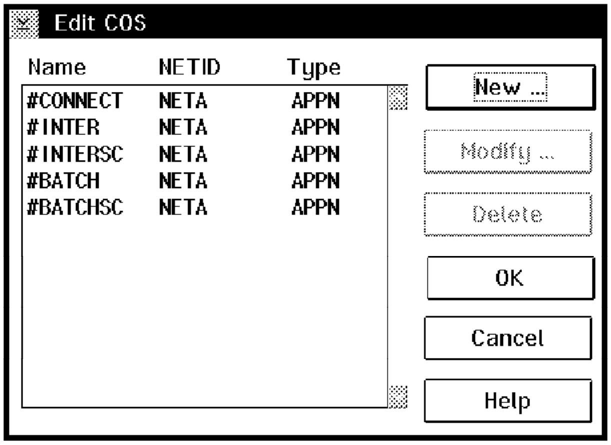


Figure 49. COS Table Menu

You can reach the menu of Figure 49 either from the network input object or from the network design object; go to **Edit** and then select **Class of Service...**.

If you select the #INTER COS entry in the menu of Figure 49 and then click on **Modify**, you will get the menu shown in Figure 50.

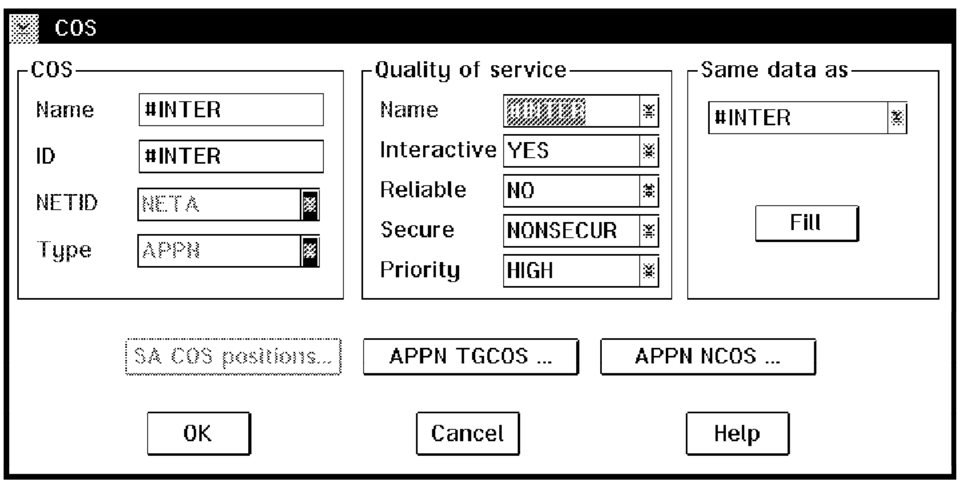


Figure 50. COS Table: Entry Menu

In this menu there are two main sections: COS and Quality of Service. Quality of Service is also known in APPN terminology as Mode. The data you enter in the area COS is equivalent to the field you see in the top area of Figure 48 on page 49, and will create/modify the NetDA/2 DCOS_DEFNS table (see **4** in Figure 57 on page 56).

The data you enter in the area Quality of Service will modify the NetDA/2 QOS_DEFNS table (see **3** in Figure 57 on page 56). You cannot create a new QOS entry from here; you will have to edit the network input object to do that.

There is a slight difference here between real life and NetDA/2. In real life the priority is stored with the COS name, rather than with the QOS (mode) entry. Nevertheless, the end result is similar.

If you click on **APPN TGCOS**, then you will be presented with Figure 51. As you can see, this menu contains information that is the same as that under TG Characteristics in Figure 48 on page 49, and also the same as the TG parameter data in Figure 47 on page 48.

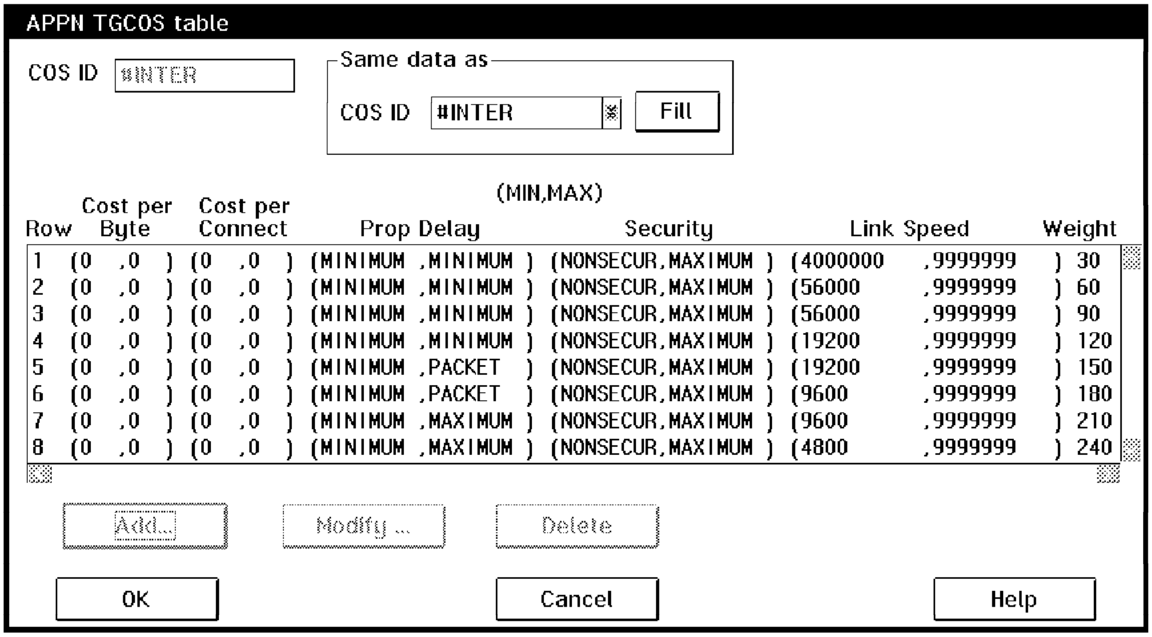


Figure 51. TGCOS Menu

In order to modify any of the values in a row, you must select the row and click on **Modify**; you will get the menu shown in Figure 52 on page 52.

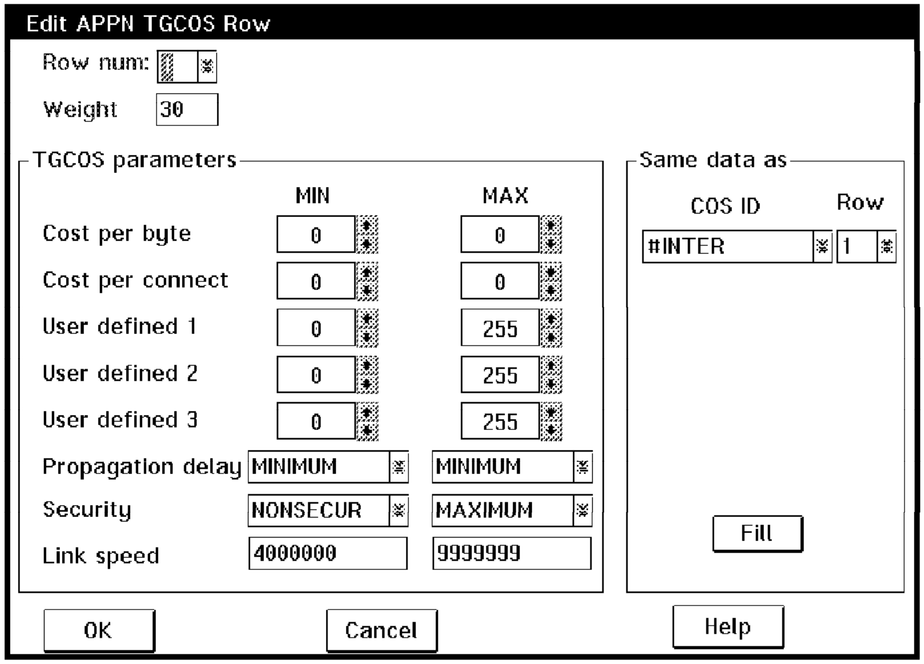


Figure 52. TGCOS Row

In this menu you can modify the values of the different fields that compose the row of the TGCOS table. NetDA/2 stores these values in the TGCOS table.

Similarly to the way of modifying the TGCOS, you can modify the values of the NCOS. If you click on **APPN NCOS** in Figure 50 on page 50, you will be presented with the menu of Figure 53.

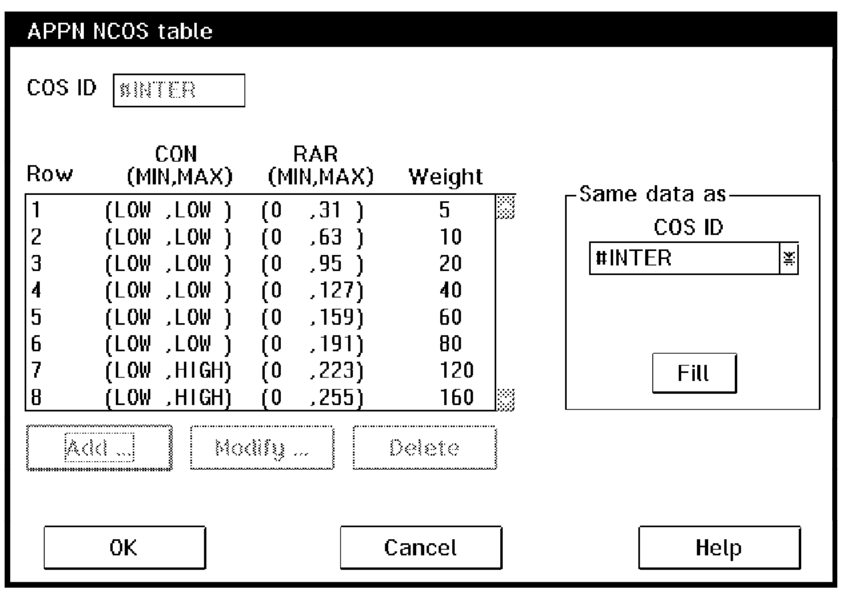


Figure 53. NCOS Menu

As you can see, this menu contains information that is the same as that under Node Characteristics in Figure 48 on page 49, and also the same as the Node data in Figure 46 on page 46, under APPN COS parameters.

In order to modify any of the values in a row, you must select the row and click on **Modify**; you will get the menu shown in Figure 54 on page 53.

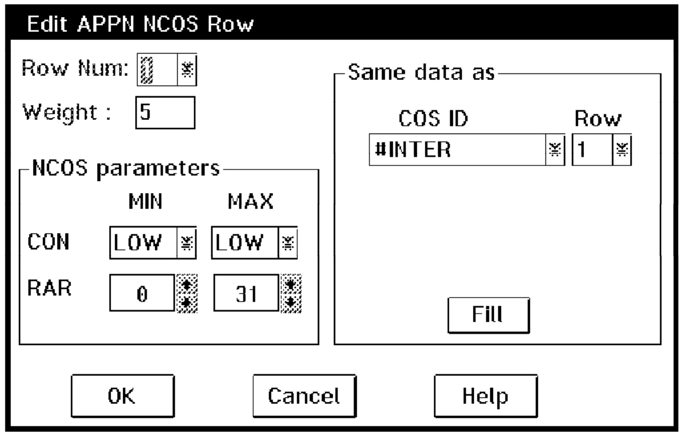


Figure 54. NCOS Row

In this menu you can modify the values of the different fields that compose the row of the NCOS table; NetDA/2 stores these values in the NCOS table.

You must click on **OK** all the way back, including the Edit COS menu, in order to save the changes you have made.

There is another way to edit the TGCOS and NCOS tables: through the Route Control view; in this notebook there is a page called APPN, as shown in Figure 55 on page 54.

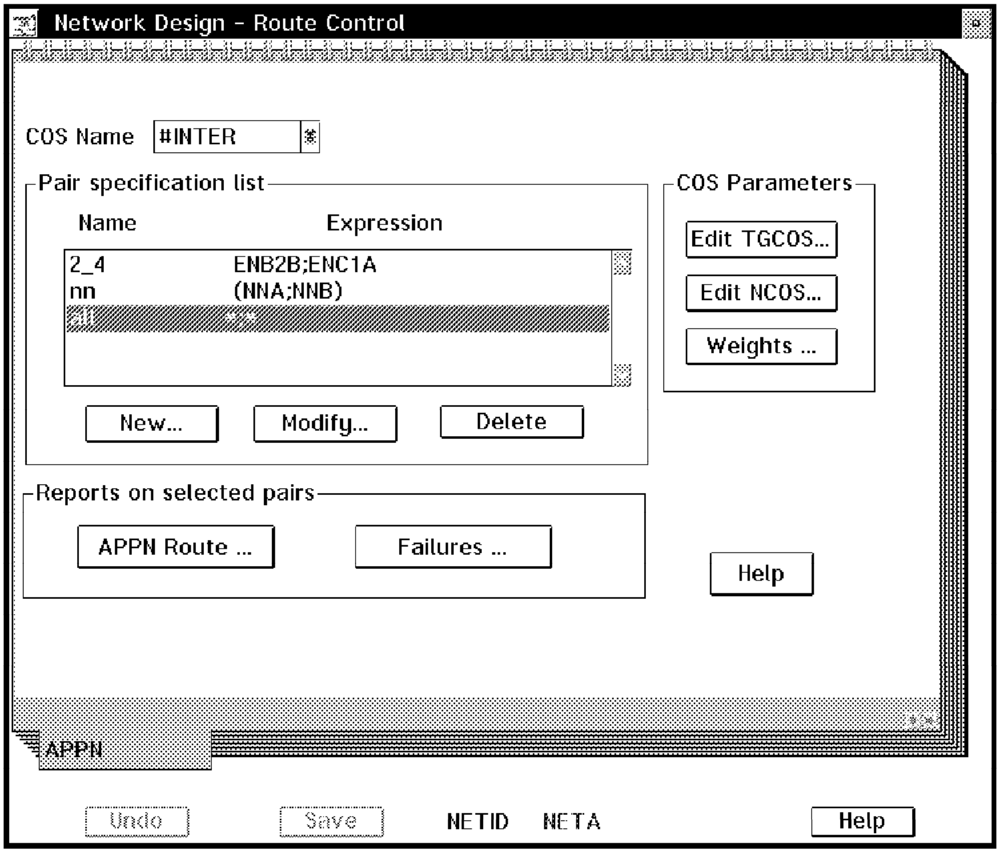


Figure 55. APPN Page in Route Control View

From this page you must first select one of the COS names available, and then click either on **Edit TGCOS** where you will get the menu of Figure 51 on page 51, or on **Edit NCOS** where you will get the menu of Figure 53 on page 52.

If you only have APPN defined in the network design object, opening the Route Control view gives you a message saying that no subarea data was found in the network design object, as shown in Figure 56.

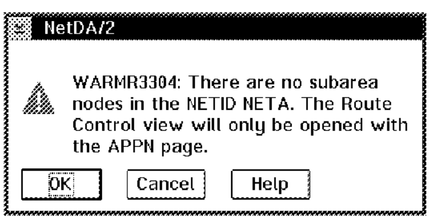


Figure 56. Route Control Warning Message When Only APPN Is Defined

5.2 Putting It All Together: How NetDA/2 Selects an APPN Route

We try to explain how all the tables described above are used by NetDA/2 to determine the route that APPN selects. This is a high-level explanation; you should read the APPN documents cited in Appendix B, "Related Publications" on page 169 (or for an even more detailed description read *SNA APPN Architecture Reference*, SC30-3422) to see how APPN works in the real world.

The numbers in the following description refer to those in Figure 57 on page 56.

1. From the REQUIREMENTS table, NetDA/2 gets the endpoints of the route and the message name **1**.
2. From the APPLICATION_REQUIREMENT table, with the message name it gets the COS_ID **2**.
3. With the COS_ID, NetDA/2 goes to the DCOS_DEFNS table and gets the Quality of Service (QOS name) that you assigned to this traffic **3**.
4. With the QOS name, NetDA/2 goes to the QOS_DEFNS table and gets the priority. QOS is better known as MODE in the APPN world **4**.

   NetDA/2 does not support the network priority.
5. At this point, NetDA/2 is able to start evaluating the links and nodes that can be used in the possible routes between both endpoints. As NetDA/2 knows the topology of the network, it can find all the possible routes between the endpoints. Now NetDA/2 can find the weight of each link and node of all these routes.
6. When evaluating the links, NetDA/2 goes to the TGCOS table and only considers those entries whose COS_ID matches the one found in the DCOS_DEFNS table **5**. As noted earlier, we have defined the APPN characteristics of each link in the network; NetDA/2 will compare the APPN characteristics of the link against those defined in the different rows in the TGCOS table. The APPN characteristics of the link are stored in the TRUNKS table **6**. Based on this comparison, a weight will be assigned to each link. NetDA/2 has to do the weight evaluations process for each direction, since the TG APPN characteristics can be different in each direction.
7. NetDA/2 also assigns a weight to each node that is an intermediate node in each of the possible routes. This is done using the NCOS table **7** and the APPN Characteristics we assigned to each node. **8**
8. Adding the weights of the links and nodes that compose each route, NetDA/2 will choose the route with the minimum weight.

Figure 57 on page 56 is a NetDA/2 network input object with just the information related to APPN route selection. The values assigned to the different APPN parameters are just for the illustration. You should not consider them as values to use when defining an APPN network to NetDA/2.

```
%TABLE QOS_DEFNS

%NAME+++ PRIO++ INTERACTIVE SECURE++ RELIABLE
 INTER1  HIGH   YES         SECURE   YES       4

%TABLE DCOS_DEFNS

%COS_ID++ NET_NAME NET_TYPE QOS_NAME COSNAME+
 INTER1AP NETA     APPN     INTER1   INTER1AP    3


%TABLE TGCOS   5

%COS_ID++ ROW_NUM SPD_MAX+ SPD_MIN CCO_MAX CCO_MIN CBY_MAX CBY_MIN SEC_MAX SEC_MIN+ PRD_MAX+ PRD_MIN UD1_MAX UD1_MIN WEIGHT
 INTER1AP 1       16777216 4194304 0       0       0       0       MAXIMUM NONSECUR TOKENRIN MINIMUM 255     0       30
 INTER1AP 2       16777216 56000   0       0       0       0       MAXIMUM NONSECUR TELEPHON MINIMUM 255     0       60
 INTER1AP 3       16777216 56000   128     0       128     0       MAXIMUM NONSECUR TELEPHON MINIMUM 255     0       90
 INTER1AP 4       16777216 19200   0       0       0       0       MAXIMUM NONSECUR TELEPHON MINIMUM 255     0       120
 INTER1AP 5       16777216 19200   128     0       128     0       MAXIMUM NONSECUR PACKET   MINIMUM 255     0       150
 INTER1AP 6       16777216 9600    0       0       0       0       MAXIMUM NONSECUR PACKET   MINIMUM 255     0       180
 INTER1AP 7       16777216 9600    196     0       196     0       MAXIMUM NONSECUR MAXIMUM  MINIMUM 255     0       210
 INTER1AP 8       16777216 600     250     0       250     0       MAXIMUM NONSECUR MAXIMUM  MINIMUM 255     0       240

%TABLE NCOS

%COS_ID++ ROW_NUM RAR_MAX RAR_MIN CON_MAX CON_MIN WEIGHT
 INTER1AP 1       31      0       LOW     LOW     5      7
 INTER1AP 2       63      0       LOW     LOW     10
 INTER1AP 3       95      0       LOW     LOW     20
 INTER1AP 4       127     0       LOW     LOW     40
 INTER1AP 5       159     0       LOW     LOW     60
 INTER1AP 6       191     0       LOW     LOW     80
 INTER1AP 7       223     0       HIGH    LOW     120
 INTER1AP 8       255     0       HIGH    LOW     160

%TABLE LOCATIONS

%NAME+ EQPNAME MAX_SWITCH_UTIL TYPE IDD CITYNAME++ LAT++ LON+++
 ENA1A AS4     40              EN   1   CHEYENNE   41N9  104W50 ESA YES
 NNA   AS4     40              NN   1   SEATTLE    47N37 122W21

%TABLE LOCATIONS_OUT   8

%NAME+ EQPNAME TYPE NET_NAME SUBNET_NAME  DIR_SERV DLU_SERV  RAR CON
 ENA1A AS4     EN   NETA     SUBNETA      NO       NO        120 LOW
 ENA1B AS4     EN   NETA     SUBNETA      NO       NO        180 LOW
 NNA   AS4     NN   NETA     SUBNETA      NO       NO        20  LOW

%TABLE TRUNKS  6

%END1+ END2 LNTY   UD1_12 UD1_21 UD2_12 UD2_21 UD3_12 UD3_21 CCO_12 CCO_21 CBY_12 CBY_21 SEC_12++ SEC_21++ PRD_12++ PRD_21++
 ENA1A NNA1 D96    1      1      1      1      1      1      1      1      1      1      NONSECUR NONSECUR MINIMUM  MINIMUM
 ENA1B NNA1 D56    1      1      1      1      1      1      1      1      1      1      NONSECUR NONSECUR MINIMUM  MINIMUM
 ENA1C NNA1 D96    1      1      1      1      1      1      1      1      1      1      NONSECUR NONSECUR MINIMUM  MINIMUM
 ENA2A NNA2 D56    1      1      1      1      1      1      1      1      1      1      SECURE   SECURE   MINIMUM  MINIMUM

%TABLE MESSAGES

%MSGNAME  QOS+++++  MSGIN+++++  MSGOT++++++
 CICS1     INTER1         800          80      2

%TABLE REQUIREMENTS    1

%SOURCE DEST+ NUM_SESS MSGNAME MRATEOT MRATEIN
 NNA    ENA1A 5        CICS1   0.003   0.003

%TABLE APPLICATION_REQUIREMENTS     2

%MSGNAME+ APPL_NODE COS_ID++
 CICS1    NNA       INTER1AP
```

Figure 57. Extracted Network Input Object File for APPN

We have produced the four tables that reflect the APPN COS table distributed with VTAM: QOS_DEFNS, DCOS_DEFNS, TGCOS and NCOS. You may retrieve these tables from the FTP site ftp.almaden.ibm.com (see "How to Access

Samples from This Book” on page xi). The name of the file containing these tables is VTAMAPPN.COS.

Let′s see a short example of how traffic between two locations can flow through different routes depending on the class of service selected.

For example, let′s suppose that APPN has to find routes for two types of traffic between end nodes EN1 and EN2 (see (A) in Figure 58). One type is interactive so it chooses the mode FAST; the other is batch and, of course, uses the mode BATCH.

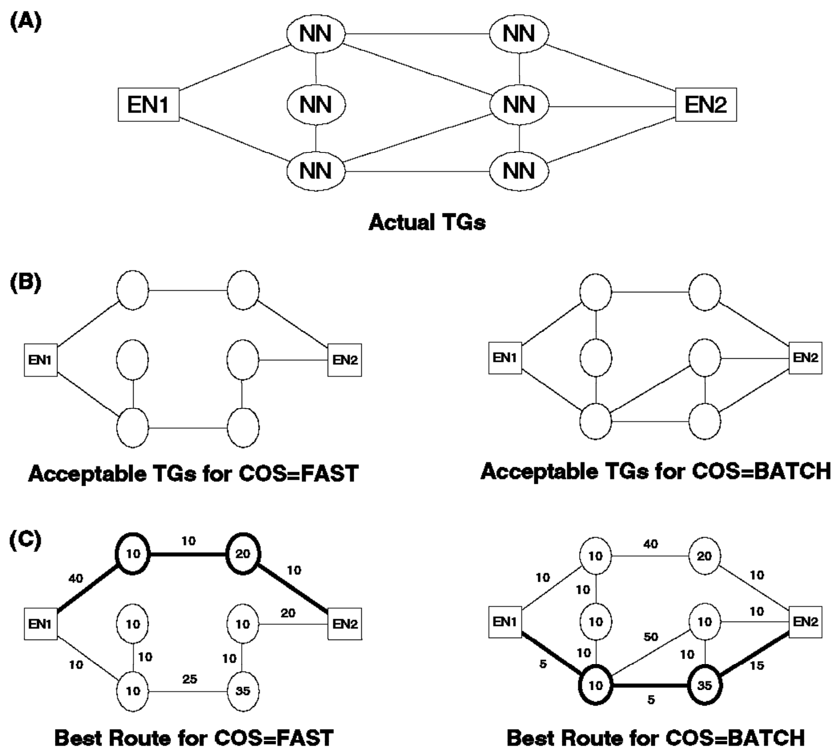


Figure 58. APPN Routes Weights

APPN (or NetDA/2) will find the COS name based on the mode (QOS) names. Based on this it will first discard the TGs that cannot be used for each type of traffic (see (B) in Figure 58).

After this initial selection, APPN will assign a weight to each of the TGs and nodes that could be used for routes between EN1 and EN2, and select the route with minimum weight (see (C) in Figure 58).

5.3 Generating Routing Statements

In Chapter 4, “Automated Design” on page 31 we designed a network. Whether we provide the detailed COS tables or require NetDA/2 to do so, we can ask it to generate APPN routing statements. Figure 59 on page 58 shows the ND menu from which we can request routing statements.

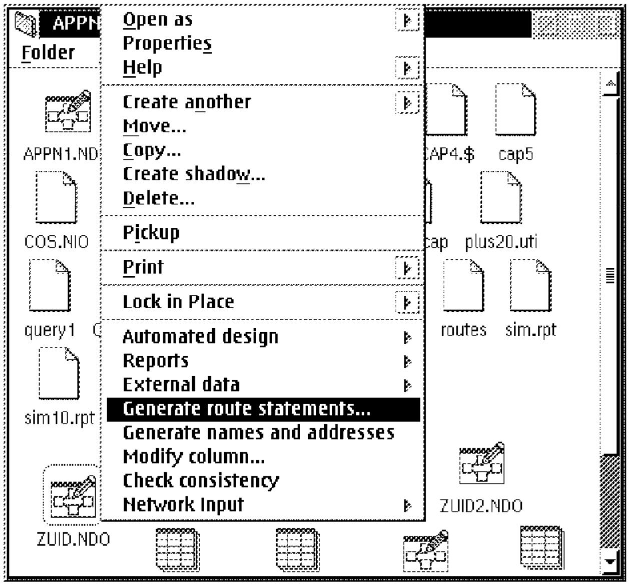


Figure 59. Requesting Routing Statements

Clicking on the Generate route statements... option causes a folder to be built within our own folder. The folder is open and the routing statements object is displayed.

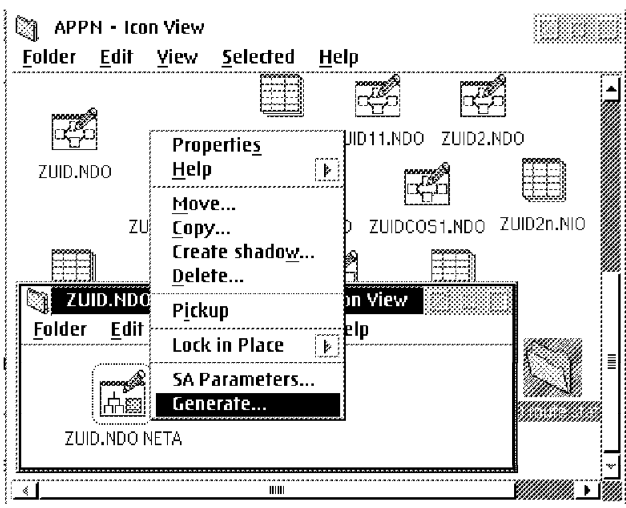


Figure 60. Routing Statements Object Menu

The pop-up menu of this object includes SA parameters, only used for subarea statement generation. We click on **Generate**.

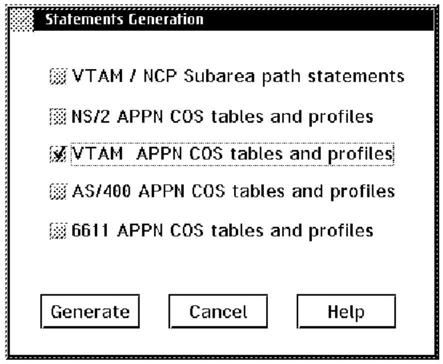


Figure 61. Which APPN Product

The statement format for the various APPN products differs. In Figure 61 we chose the VTAM format from those supported by NetDA/2.

This adds a file to the folder containing the routing statements.

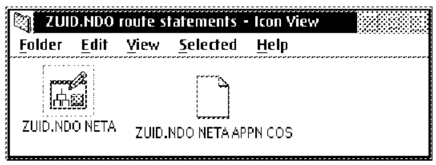


Figure 62. Route Statements Folder

In the report, part of which is in Figure 63 on page 60, for each COS there is a set of LINEROW and NODEROW statements, followed by TG Profiles for the 70 or so TGs.

In this example, the COS statements are the standard VTAM APPN values, as we supplied them in the NetDA/2 network input object. The TG profiles show variations in the user parameter 1 values. These are generated by NetDA/2 and would have been used in determining weights had NetDA/2 been providing the COS tables.

```
**************************************************
* COS Tables and TG Profiles for ACF/VTAM        *
* NETID: NETA                                    *
* Created: 14:04:49, Tuesday, 16 September 1997. *
**************************************************

* C O S   T a b l e s *


#BATCH   APPNCOS PRIORITY=LOW
         LINEROW NUMBER=1,                                              *
               WEIGHT=30,                                               *
               CAPACITY=(56000,4M),                                     *
               COSTTIME=(0,0),                                          *
               COSTBYTE=(0,0),                                          *
               SECURITY=(UNSECURE,MAXIMUM),                             *
               PDELAY=(NEGLIGIB,MAXIMUM),                               *
               UPARM1=(0,255),                                          *
               UPARM2=(0,255),                                          *
               UPARM3=(0,255)
         LINEROW NUMBER=2,                                              *
               WEIGHT=60,                                               *
 . . .
         NODEROW NUMBER=1,                                              *
               WEIGHT=5,                                                *
               ROUTERES=(0,31),                                         *
               CONGEST=(LOW,LOW)
         NODEROW NUMBER=2,                                              *
               WEIGHT=10,                                               *
 . . .
*
#INTER   APPNCOS PRIORITY=HIGH
         LINEROW NUMBER=1,                                              *
               WEIGHT=30,                                               *
               CAPACITY=(4M,4M),                                        *
               COSTTIME=(0,0),                                          *
               COSTBYTE=(0,0),                                          *
               SECURITY=(UNSECURE,MAXIMUM),                             *
 . . .

* T G   P r o f i l e s *


* TG BEAUFORT_1_GRAFF_RE *
TGP00001 TGP COSTTIME=0,                                                *
               COSTBYTE=0,                                              *
               SECURITY=UNSECURE,                                       *
               PDELAY=NEGLIGIB,                                         *
               UPARM1=255,                                              *
               UPARM2=0,                                                *
               UPARM3=0
 . . .
*
* TG CAPE_TOW_1_DE_AAR *
TGP00006 TGP COSTTIME=0,                                                *
               COSTBYTE=0,                                              *
               SECURITY=UNSECURE,                                       *
               PDELAY=NEGLIGIB,                                         *
               UPARM1=3,                                                *
               UPARM2=0,                                                *
               UPARM3=0
*
* TG CAPE_TOW_1_JOHANNES *
TGP00007 TGP COSTTIME=0,                                                *
               COSTBYTE=0,                                              *
               SECURITY=UNSECURE,                                       *
               PDELAY=NEGLIGIB,                                         *
               UPARM1=0,                                                *
               UPARM2=0,                                                *
               UPARM3=0
 . . .
```

Figure 63. Extract from Routing Statements

Chapter 6. Route Reports

We continue with the APPN network used in Chapter 4, "Automated Design" on page 31 and create the route reports. To do this we first open the ND's Route control view.

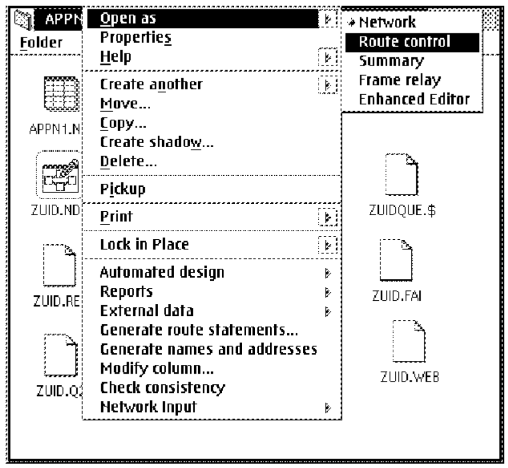


Figure 64. Opening the Route Control View

Because the network is purely APPN and has no subarea nodes, the notebook has a single page.

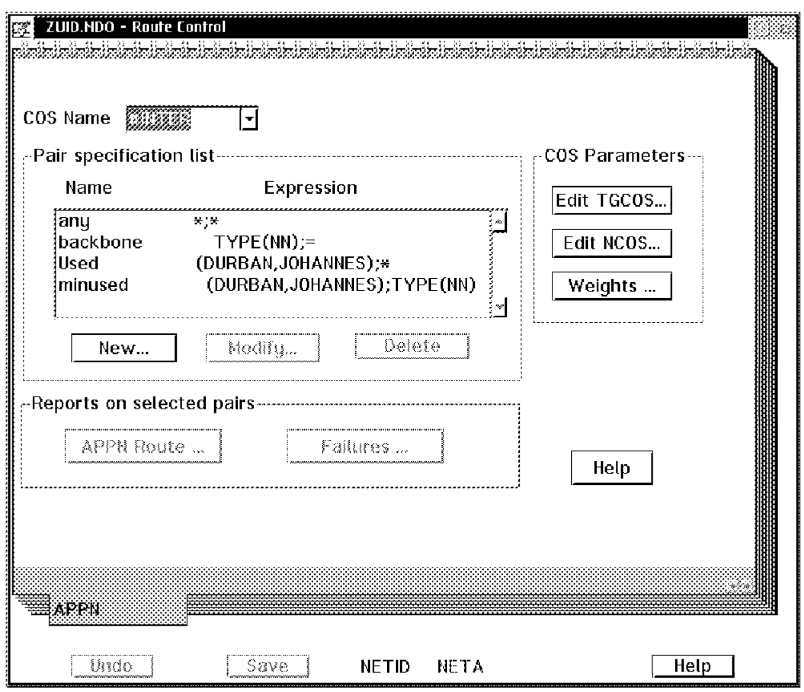


Figure 65. APPN Route Control Notebook

All of the actions use the COS selected in the COS Name box and here we have selected the COS used with the application MSG1. The COS Parameters box on the right offers us the choice of editing the COS parameters, or listing the weights for each node and TG resulting from use of the selected COS.

The Pair specification list box is initially empty. After selecting a specification we can list the least weight routes, or those routes and their alternatives in the event of component failure.

To define a pair specification, we select **New** and then specify the nodes at the ends of the routes. Four examples are shown, the last being the most restrictive.

any * means any node so this covers all node pairs and therefore all routes.

backbone The = sign means that the one definition, TYPE(NN), applies to both ends. It includes all routes between NNs; in other words, combinations of the backbone TGs.

used Here we define all node pairs where one end is either Durban or Johannesburg. This limits the set of routes to those carrying traffic.

minused Because no ENs have connections to multiple NNs, the NN_EN peripheral links do not affect the choice of route in our case. This definition further reduces the volume of output without losing any information. We see the routes from Durban and Johannesburg to the other NNs.

6.1 Least Weight Routes Report

After selecting **minused** we click on **APPN route**. A report is created and displayed using four colors. The first few lines are shown in Figure 66.

```
Appended at 20:35:57, Wednesday, 01 October 1997.  ZUID.NDO

APPN Routes


 APPN Least Weight Routes Report for Network Design Object ZUID.NDO,
 Started Wed Oct  1 1997 20:35:47.
 Pair specification: minused            (DURBAN,JOHANNES);TYPE(NN)
 APPN Weights are Defined by COS Name #INTER.


 Subnet: @_SUBNO


 Origin:  CAPE_TOW,  Destination:  DURBAN,  Least Weight: 125.
   CAPE_TOW(1)JOHANNES(1)DURBAN
   CAPE_TOW(1)JOHANNES(2)DURBAN


 Origin:  CAPE_TOW,  Destination:  JOHANNES,  Least Weight: 60.
   CAPE_TOW(1)JOHANNES


 Origin:  DE_AAR,  Destination:  DURBAN,  Least Weight: 185.
   DE_AAR(1)QUEENSTO(1)DURBAN


 Origin:  DE_AAR,  Destination:  JOHANNES,  Least Weight: 185.
   DE_AAR(1)CAPE_TOW(1)JOHANNES
   DE_AAR(1)GRAFF_RE(1)JOHANNES
  . .
```

Figure 66. Routes for COS #INTER

The least weight route is shown, with multiple routes being shown if their weights are equal. This report may be used to establish whether the node and

TG characteristics defined or defaulted, in conjunction with the COS which applies to the traffic, result in the route we expect or require.

6.2 Routes Failures Report

A second report is obtained by clicking on **Failures**. In Figure 67, in addition to the least weight routes we see the route that would be used in the event of failure of any component in a least weight route.

```
Appended at 19:29:52, Wednesday, 01 October 1997.  ZUID.NDO

APPN Routes with Failures


 APPN Least Weight Routes Report With Failures of Nodes and TGs
 for Network Design Object ZUID.NDO, Started Wed Oct  1 1997 19:28:06.
 Pair specification: minused            (DURBAN,JOHANNES);TYPE(NN)
 APPN Weights are Defined by COS Name #INTER.


 Subnet: @_SUBNO


 Origin:  CAPE_TOW,  Destination:  DURBAN,  Least Weight: 125.
   CAPE_TOW(1)JOHANNES(1)DURBAN
   CAPE_TOW(1)JOHANNES(2)DURBAN
 Failure:  JOHANNES,  Least Weight: 310.
   CAPE_TOW(1)DE_AAR(1)QUEENSTO(1)DURBAN
 Failure:  CAPE_TOW(1)JOHANNES,  Least Weight: 310.
   CAPE_TOW(1)DE_AAR(1)QUEENSTO(1)DURBAN


 Origin:  CAPE_TOW,  Destination:  JOHANNES,  Least Weight: 60.
   CAPE_TOW(1)JOHANNES
 Failure:  CAPE_TOW(1)JOHANNES,  Least Weight: 310.
   CAPE_TOW(1)DE_AAR(1)GRAFF_RE(1)JOHANNES


 Origin:  DE_AAR,  Destination:  DURBAN,  Least Weight: 185.
   DE_AAR(1)QUEENSTO(1)DURBAN
 Failure:  QUEENSTO,  Least Weight: 245.
   DE_AAR(1)GRAFF_RE(1)DURBAN
   DE_AAR(1)GRAFF_RE(2)DURBAN
 Failure:  DE_AAR(1)QUEENSTO,  Least Weight: 245.
   DE_AAR(1)GRAFF_RE(1)DURBAN
   DE_AAR(1)GRAFF_RE(2)DURBAN
 Failure:  DURBAN(1)QUEENSTO,  Least Weight: 245.
   DE_AAR(1)GRAFF_RE(1)DURBAN
   DE_AAR(1)GRAFF_RE(2)DURBAN


 Origin:  DE_AAR,  Destination:  JOHANNES,  Least Weight: 185.
   DE_AAR(1)CAPE_TOW(1)JOHANNES
   DE_AAR(1)GRAFF_RE(1)JOHANNES
 No common nodes or TGs to all the above non-failure routes.
  . . .
```

Figure 67. Routes and Failures

For the first node pair, CAPE_TOW and DURBAN, two routes are shown with equal weight. Alternative routes are listed for failures of any intermediate nodes and then any TGs. Here we see backup for the intermediate node JOHANNES, and then for the CAPE_TOW to JOHANNES TG. Two TGs exist between JOHANNES and DURBAN, so no additional route is needed to cover the failure of either TG.

6.3 Weights Report

The Weights report requires selection of a COS but not a pair specification. In Figure 68, we show part of the Weights report for COS name #INTER.

```
Appended at 19:31:18, Wednesday, 01 October 1997.  ZUID.NDO

APPN Weights


 APPN Weight Report for Network Design Object ZUID.NDO, Started Wed Oct  1
1997 19:29:56.

 Weights of All APPN Nodes in NetID NETA per COS Name #INTER.
 Name       Weight     CON   RAR
 BEAUFORT        5     LOW     0
 BLOEMFON        5     LOW     0
 CALEDON         5     LOW     0
 CAPE_TOW        5     LOW     0
 CERES           5     LOW     0
 CRADOCK         5     LOW     0
 DE_AAR          5     LOW     0
  . . .

 Weights of All APPN TGs in NetID NETA per COS Name #INTER.
 End1     End2    TG#  Wgt_12 Wgt_21   CCO_12 CCO_21 CBY_12 CBY_21 PRD_12+ PRD_21+  UD1_12 UD1_21 SPEED ..more
 BEAUFORT GRAFF_RE  1     120    120        0      0      0      0 MINIMUM MINIMUM     128    128 19200
 BLOEMFON DE_AAR    1     120    120        0      0      0      0 MINIMUM MINIMUM     128    128 19200
 CALEDON  CAPE_TOW  1     120    120        0      0      0      0 MINIMUM MINIMUM     128    128 19200
 CAPE_TOW CERES     1     120    120        0      0      0      0 MINIMUM MINIMUM     128    128 19200
 CAPE_TOW DE_AAR    1     120    120        0      0      0      0 MINIMUM MINIMUM       3      3 19200
 CAPE_TOW JOHANNES  1      60     60        0      0      0      0 MINIMUM MINIMUM       0      0 64000
 CAPE_TOW MALMESBU  1     120    120        0      0      0      0 MINIMUM MINIMUM     128    128 19200
 CAPE_TOW SOMERSET  1     120    120        0      0      0      0 MINIMUM MINIMUM     128    128 19200
 CRADOCK  GRAFF_RE  1     120    120        0      0      0      0 MINIMUM MINIMUM     128    128 19200
 DE_AAR   GRAFF_RE  1     120    120        0      0      0      0 MINIMUM MINIMUM       3      3 19200
 DE_AAR   KIMBERLE  1     120    120        0      0      0      0 MINIMUM MINIMUM     128    128 19200
 DE_AAR   QUEENSTO  1     120    120        0      0      0      0 MINIMUM MINIMUM       3      3 19200
 DE_AAR   UPINGTON  1     120    120        0      0      0      0 MINIMUM MINIMUM     128    128 19200
 DUNDEE   DURBAN    1     120    120        0      0      0      0 MINIMUM MINIMUM     128    128 19200
 DURBAN   EMPANGEN  1     120    120        0      0      0      0 MINIMUM MINIMUM     128    128 19200
 DURBAN   GRAFF_RE  1     120    120        0      0      0      0 MINIMUM MINIMUM       3      3 19200
 DURBAN   GRAFF_RE  2     120    120        0      0      0      0 MINIMUM MINIMUM       3      3 19200
 DURBAN   JOHANNES  1      60     60        0      0      0      0 MINIMUM MINIMUM       3      3 64000
 DURBAN   JOHANNES  2      60     60        0      0      0      0 MINIMUM MINIMUM       3      3 64000
  . . .
```

Figure 68. Weights Report

We are using the default VTAM APPNCOS tables in which TG weights are largely determined by the line capacity. The node and TG characteristics used to determine the weights are included in the report for reference and some such as security and additional user parameters have been omitted from the figure.

Chapter 7. Influencing The Routes

Many users will probably use the default COS and other values that are provided.

When working with subarea routes, we can associate preferred routes with COS entries by means of VR numbers. If we wish to influence the routes in APPN, we have to allocate node and TG characteristics and develop COS tables to achieve the desired results.

During design, NetDA/2 sets these values to satisfy our design criteria. We decided to explore ways of influencing the routes used. To do this there are two areas we can modify. One is the individual node and TG characteristics; the other the class of service tables.

From the ND′s menu bar we can select **Edit** and **Class of service** and then modify the COS tables, or we can return to the NI and use graphic editing to change the TG and node characteristics. Having made our changes we can generate new APPN route and failure reports illustrated in Chapter 6, "Route Reports" on page 61 to see the impact of the changes.

7.1.1 Combined Reports

Instead of experimenting with characteristics and COS tables, we thought it might be easier to directly change weights. Then, when we had seen the new reports, we could investigate what to change to force the new weights.

Some REXX code, which read the Weights report and the Failures report, gave us a listing of the main routes between pairs of nodes with a breakdown of the component weights (you can find this REXX code in the FTP site mentioned in "How to Access Samples from This Book" on page xi with the names APPN.CMD and APPN1.CMD).

```
 Subnet: @_SUBNO
Origin: CAPE_TOW, Destination: DURBAN,
 New Orig
  125 125.    CAPE_TOW(1)JOHANNES(1)DURBAN      Weights->(60)5(60)
  125 125.    CAPE_TOW(1)JOHANNES(2)DURBAN      Weights->(60)5(60)
  310 310.    CAPE_TOW(1)DE_AAR(1)QUEENSTO(1)DURBAN
Weights->(120)5(120)5(60)
Origin: CAPE_TOW, Destination: JOHANNES,
 New Orig
  60 60.    CAPE_TOW(1)JOHANNES      Weights->(60)
  310 310.    CAPE_TOW(1)DE_AAR(1)GRAFF_RE(1)JOHANNES
Weights->(120)5(120)5(60)
Origin: DE_AAR, Destination: DURBAN,
 New Orig
  185 185.    DE_AAR(1)QUEENSTO(1)DURBAN      Weights->(120)5(60)
  245 245.    DE_AAR(1)GRAFF_RE(1)DURBAN      Weights->(120)5(120)
  245 245.    DE_AAR(1)GRAFF_RE(2)DURBAN      Weights->(120)5(120)
Origin: DE_AAR, Destination: JOHANNES,
 New Orig
  185 185.    DE_AAR(1)CAPE_TOW(1)JOHANNES      Weights->(120)5(60)
  185 185.    DE_AAR(1)GRAFF_RE(1)JOHANNES      Weights->(120)5(60)
 . . .
```

Figure 69. Combined Listing - #INTER

Figure 69 lists all routes mentioned in the Failures report, with duplicates eliminated. The value in the Orig weight column comes from the Failures report and we show the individual component weights to the right.

Looking at the fourth set, we see that the COS #INTER has two routes of equal weight between locations DE_AAR and JOHANNES. By modifying a component weight, we could move the traffic on to one or the other route. Because the TG from JOHANNES to CAPE_TOW had a higher utilization in our query report than the one from JOHANNES to GRAFF_RE, we told the REXX code to reduce the weight of DE_AAR.1.GRAFF_RE from 120 to 90.

This resulted in the following output.

```
 Subnet: @_SUBNO
Origin: CAPE_TOW, Destination: DURBAN,
 New Orig
  125 125.    CAPE_TOW(1)JOHANNES(1)DURBAN      Weights->(60)5(60)
  125 125.    CAPE_TOW(1)JOHANNES(2)DURBAN      Weights->(60)5(60)
  310 310.    CAPE_TOW(1)DE_AAR(1)QUEENSTO(1)DURBAN
Weights->(120)5(120)5(60)
Origin: CAPE_TOW, Destination: JOHANNES,
 New Orig
  60 60.    CAPE_TOW(1)JOHANNES      Weights->(60)
  280 310.    CAPE_TOW(1)DE_AAR(1)GRAFF_RE(1)JOHANNES
Weights->(120)5(90)5(60)
Origin: DE_AAR, Destination: DURBAN,
 New Orig
  185 185.    DE_AAR(1)QUEENSTO(1)DURBAN      Weights->(120)5(60)
  215 245.    DE_AAR(1)GRAFF_RE(1)DURBAN      Weights->(90)5(120)
  215 245.    DE_AAR(1)GRAFF_RE(2)DURBAN      Weights->(90)5(120)
Origin: DE_AAR, Destination: JOHANNES,
 New Orig
  185 185.    DE_AAR(1)CAPE_TOW(1)JOHANNES      Weights->(120)5(60)
  155 185.    DE_AAR(1)GRAFF_RE(1)JOHANNES      Weights->(90)5(60)
 . . .
```

Figure 70. Combined Listing - #INTER

Successive requests for component weight changes cause updates to the New weight column while the Orig column shows the value before any changes. When we have finished experimenting, we can check that only the routes we wish to change are changed.

Here we see that in the fourth set the second route now has the lower weight. This will reduce traffic on the two TGs in the first route.

Before considering how to implement the weight change we should look at the combined listing for the #CONNECT COS name.

```
 Subnet: @_SUBNO
Origin: CAPE_TOW, Destination: DURBAN,
 New Orig
  125 125.    CAPE_TOW(1)JOHANNES(1)DURBAN      Weights->(60)5(60)
  125 125.    CAPE_TOW(1)JOHANNES(2)DURBAN      Weights->(60)5(60)
  250 250.    CAPE_TOW(1)DE_AAR(1)QUEENSTO(1)DURBAN
Weights->(90)5(90)5(60)
Origin: CAPE_TOW, Destination: JOHANNES,
 New Orig
  60 60.    CAPE_TOW(1)JOHANNES      Weights->(60)
  250 250.    CAPE_TOW(1)DE_AAR(1)GRAFF_RE(1)JOHANNES
Weights->(90)5(90)5(60)
Origin: DE_AAR, Destination: DURBAN,
 New Orig
  155 155.    DE_AAR(1)QUEENSTO(1)DURBAN      Weights->(90)5(60)
  185 185.    DE_AAR(1)GRAFF_RE(1)DURBAN      Weights->(90)5(90)
  185 185.    DE_AAR(1)GRAFF_RE(2)DURBAN      Weights->(90)5(90)
Origin: DE_AAR, Destination: JOHANNES,
 New Orig
  155 155.    DE_AAR(1)CAPE_TOW(1)JOHANNES      Weights->(90)5(60)
  155 155.    DE_AAR(1)GRAFF_RE(1)JOHANNES      Weights->(90)5(60)
 . . .
```

Figure 71. Combined Listing - #CONNECT

Here we see that for #CONNECT, a route using the TG DE_AAR to GRAFF_RE is second choice between DE_AAR and DURBAN, and another is equal lowest weight from DE_AAR to JOHANNES. If we wish to move #CONNECT traffic onto the DE_AAR to GRAFF_RE TG, we need to make its routes more attractive. A weight of 30 for the TG would make the routes containing it the least weight route in each case.

We show this to illustrate that in a multiple COS case, preferring routes is not elementary.

7.1.2 How Do We Adjust the Weights?

We noted earlier that we could change the TG characteristics or the COS tables.

Most of the values will be facts, leaving us just the user parameters to change. If we are using the standard APPNCOS tables supplied with VTAM, we find that the user parameters do not affect the weight allocated to a TG.

So we need either to change the supplied tables, generally undesirable, or to define new ones for new COSs.

7.1.3 NetDA/2 Defines the COS Tables & Weights

If we allow NetDA/2 to design the network and we do not supply the default COS tables, NetDA/2 will select weights and COS tables to force the routes to satisfy the design constraints.

In our first design example, we supplied full node and TG COS details. In the following extract, we have reduced the QOS_DEFNS and DCOS_DEFNS tables to just two values, deleted the NCOS and TGCOS tables, and added the APPLICATION_REQUIREMENTS table which relates an application to a COS_ID.

```
%TABLE QOS_DEFNS

%NAME+++ PRIO++
 CONNECT MEDIUM
 INTER   HIGH

%TABLE DCOS_DEFNS

%COS_ID+ NET_NAME NET_TYPE QOS_NAME COSNAME
 CONNECT NETA     APPN     CONNECT  CONNECT
 INTER   NETA     APPN     INTER    INTER

%TABLE MESSAGES
%MSGNAME MSGIN MSGOT QOS++++
 MSG1    1024  1024  INTER
 MSG2     480  2400  CONNECT

%TABLE APPLICATION_REQUIREMENTS
%APPL_NODE MSGNAME COS_ID+
 JOHANNES  MSG2    CONNECT
 DURBAN    MSG1    INTER

%TABLE REQUIREMENTS

%SOURCE++ DEST++++ NUM_SESS MSGNAME MRATEOT MRATEIN
 DURBAN   BEAUFORT 4        MSG1    1       1
 DURBAN   BLOEMFON 4        MSG1    1       1
 DURBAN   CALEDON  4        MSG1    1       1
  . . .
```

Figure 72. NI Extract

Now when we design, NetDA/2 select user parameters for the TGs and weights in the COS tables so that the traffic uses specific routes.

7.1.3.1 TG Characteristics

```
%TABLE TRUNKS

%END1++++ END2++++ LNTY+ LINK_ID TG_NUM NUM UD_FR UD1_12 UD1_21 ..more
 CAPE_TOW DE_AAR   D19.2 @0      1      1   0     3      3
 CAPE_TOW JOHANNES D64   @0      1      1   0     0      0
 DE_AAR   GRAFF_RE D64   @0      1      1   0     3      3
 DE_AAR   GRAFF_RE D64   @1      2      1   0     3      3
 DE_AAR   KIMBERLE D19.2 @0      1      1   0     128    128
 DE_AAR   QUEENSTO D19.2 @0      1      1   0     3      3
 DURBAN   GRAFF_RE D19.2 @0      1      1   0     3      3
 DURBAN   GRAFF_RE D19.2 @1      2      1   0     3      3
 DURBAN   JOHANNES D64   @0      1      1   0     3      3
 DURBAN   JOHANNES D64   @1      2      1   0     3      3
 DURBAN   QUEENSTO D64   @0      1      1   0     0      0
 GRAFF_RE JOHANNES D64   @0      1      1   0     3      3
```

Figure 73. ND Extract with User Parm

Some of the backbone links have a first user defined parameter of 0 and some are 3, and this is not directly related to speed. Some of the D64 links are 0 and some 3.

Just one of the peripheral links, DE_AAR to KIMBERLE, is shown for reference. Their weights do not affect the routes here and all have the (VTAM) default value of 128.

```
%TABLE TGCOS

%COS_ID+ ROW_NUM SPD_MAX++ SPD_MIN UD1_MAX UD1_MIN WEIGHT .. more
 CONNECT 1       999999999 0       255     255     1
 CONNECT 2       999999999 0       255     127     3
 CONNECT 3       999999999 0       255     63      6
 CONNECT 4       999999999 0       255     31      11
 CONNECT 5       999999999 0       255     15      19
 CONNECT 6       999999999 0       255     7       32
 CONNECT 7       999999999 0       255     3       53
 CONNECT 8       999999999 0       255     0       87
 INTER   1       999999999 0       255     255     1
 INTER   2       999999999 0       255     127     3
 INTER   3       999999999 0       255     63      6
 INTER   4       999999999 0       255     31      11
 INTER   5       999999999 0       255     15      19
 INTER   6       999999999 0       255     7       32
 INTER   7       999999999 0       255     3       53
 INTER   8       999999999 0       255     0       87

%TABLE NCOS

%COS_ID+ ROW_NUM RAR_MAX RAR_MIN CON_MAX CON_MIN WEIGHT
 CONNECT 1       255     0       HIGH    LOW     17
 INTER   1       255     0       HIGH    LOW     17
```

Figure 74. New COS Table

All of the other parameters are set to maximum and minimum values, and consequently the weights are determined solely by the user parameter 1.

```
Appended at 22:53:25, Wednesday, 01 October 1997.  ZUIDCOS.NDO

APPN Routes with Failures


 APPN Least Weight Routes Report With Failures of Nodes and TGs
 for Network Design Object ZUIDCOS.NDO, Started Wed Oct  1 1997 22:53:01.
 Pair specification: minused            (DURBAN,JOHANNES);TYPE(NN)
 APPN Weights are Defined by COS Name INTER.

 Subnet: @_SUBNO

 Origin:  CAPE_TOW,  Destination:  DURBAN,  Least Weight: 157.
   CAPE_TOW(1)JOHANNES(1)DURBAN
   CAPE_TOW(1)JOHANNES(2)DURBAN
 Failure:  JOHANNES,  Least Weight: 193.
   CAPE_TOW(1)DE_AAR(1)GRAFF_RE(1)DURBAN
   CAPE_TOW(1)DE_AAR(2)GRAFF_RE(1)DURBAN
   CAPE_TOW(1)DE_AAR(1)GRAFF_RE(2)DURBAN
   CAPE_TOW(1)DE_AAR(2)GRAFF_RE(2)DURBAN
 Failure:  CAPE_TOW(1)JOHANNES,  Least Weight: 193.
   CAPE_TOW(1)DE_AAR(1)GRAFF_RE(1)DURBAN
   CAPE_TOW(1)DE_AAR(2)GRAFF_RE(1)DURBAN
   CAPE_TOW(1)DE_AAR(1)GRAFF_RE(2)DURBAN
   CAPE_TOW(1)DE_AAR(2)GRAFF_RE(2)DURBAN


 Origin:  CAPE_TOW,  Destination:  JOHANNES,  Least Weight: 87.
   CAPE_TOW(1)JOHANNES
 Failure:  CAPE_TOW(1)JOHANNES,  Least Weight: 193.
   CAPE_TOW(1)DE_AAR(1)GRAFF_RE(1)JOHANNES
   CAPE_TOW(1)DE_AAR(2)GRAFF_RE(1)JOHANNES
 . . .
```

Figure 75. Failures Report for COS INTER

NetDA/2 has added a second link between DE_AAR and GRAFF_RE and the two links are included in the least weight routes between CAPE_TOW and DURBAN. In our previous example, DE_AAR and QUEENSTO were the intermediate nodes.

If we had no need to design the topology, we could define the full network but still ask NetDA/2 to develop the COS tables. Alternatively, we can define the topology and our own COS tables, and then ask NetDA/2 to perform the design so that we may check that sufficient capacity exists.

For example, in our network there are only two line speeds. We could consider a COS table with the following rows.

```
  COS_ID  ROW_NUM SPD_MAX SPD_MIN UD1_MAX UD1_MIN ... WEIGHT
  CONNECT 1       9999999 64000     255    255         30
  CONNECT 2       9999999 64000     255    127         60
  CONNECT 3       9999999 64000     255     63         90
  CONNECT 4       9999999 64000     255     0          120
  CONNECT 5       9999999 19200     255    255         60
  CONNECT 6       9999999 19200     255    127         90
  CONNECT 7       9999999 19200     255     63         120
  CONNECT 8       9999999 19200     255     0          150
```

Figure 76. User COS Table

If other parameters such as security and cost per byte are equivalent for each row, we can control the weight allocated to a TG by the value we choose for the TG′s user-defined parameter 1. This could be done to have a different result with different COS tables simply by changing the weight column.

Second and third user-defined parameters exist so an alternative would be to use similar tables for INTER and CONNECT, controlling the weight selection in one by means of user parameter 1, and in the other by user parameter 2.

Chapter 8. An Example of a Real Network

In this chapter we describe a fairly complex network, taken from real life, where APPN route calculation is an important design issue. We show how the design issues were addressed manually, and how NetDA/2 was able to confirm that the final APPN design was valid and would yield the desired traffic patterns.

8.1 Subarea Network before Migration

Figure 77 illustrates a simplified view of the subarea network before migration.

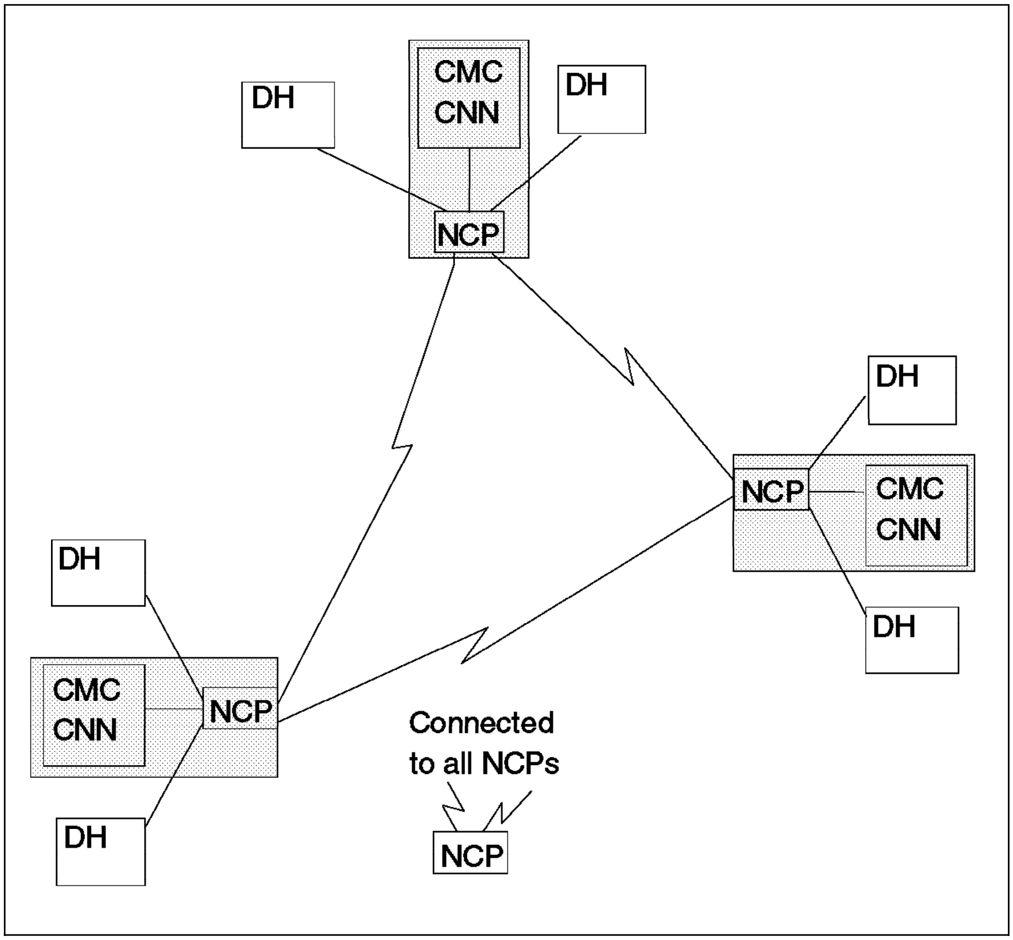


Figure 77. Subarea Network before Migration

The subarea network comprises three CMC configurations, each of which has a number of data hosts attached. All the data hosts are in fact attached via SNI gateways to their CMCs, since they all have different net IDs. This means that the connections are through NCPs and cannot utilize CTCs. The three CMCs are linked by an inter-NCP backbone of high-speed lines. Feeding into this backbone network is a second NCP-based network through which the user terminals are connected to the application hosts. The LUs in this second network are mostly dependent (3270-type) terminals, but there are also APPC workstations communicating with the application hosts using LU 6.2.

The NCP backbone is close to its capacity, and the network is so complex that installing additional NCPs (and their corresponding path tables) is almost impossible. The objectives of the exercise are:

- To provide an upgrade path by converting to APPN
- To maximize backbone capacity by converting to HPR
- To ensure that no additional loading is placed on the NCPs
- To allow direct CTC connections between hosts

8.2 Proposed APPN Network

In order to meet the conversion objectives while migrating to APPN, some basic design decisions must be made:

- To accommodate increasing traffic, a parallel backbone of 3746-950 network node processors (NNPs) is to be installed. Existing traffic will continue to use the NCP backbone, but will be migrated to use HPR over the NNP backbone.
- The secondary NCP network will be migrated from NCPs to 2216 network nodes. The APPC workstations will be connected to the 2216s, while the dependent terminals will be migrated to use the Dependent LU Requester (DLUR) function on the 2216s. DLUR allows dependent LUs to be located anywhere in the network without the old requirements that they be attached to a VTAM or NCP and that their sessions always pass through that VTAM or NCP. They still require a VTAM owner (dependent LU server, DLUS), but that owner can be reached by an APPN network path rather than a direct SSCP-LU session. Therefore, dependent LUs will be able to take full advantage of HPR over the NNP backbone.

  The 2216 nodes will have APPN connections both to the existing backbone NCPs and to the new backbone NNPs.
- Where possible, the data hosts will be converted to MDHs to use APPN. Ultimately they will become pure end nodes. The conversion from SNI-attached data host to APPN end node is particularly pleasing because an APPN border is not required in these circumstances. The data hosts can retain their net IDs so that minimal changes are required in the network. Also, sessions between VTAM hosts can be routed directly over a CTC instead of requiring to pass through a gateway NCP.

  During migration there will be a mixture of MDHs and unconverted data hosts, which fact we must take into account.
- The CMCs must become ICNs because they own NCPs, and because they will form the boundary between the APPN-incapable data hosts and the APPN network. In addition, they will become central directory servers since the network is large enough to warrant several CDSs. They will also serve as DLUS nodes for the dependent LUs whose DLURs will be the 2216s.
- In order to preserve the existing facility for NCP takeover between CMC VTAMs, subarea connectivity is required between them. Therefore, the connections between the new ICNs across the backbone will be defined as VR-TGs. This preserves their subarea nature while integrating them into the APPN network. Another advantage of using VR-TGs is that the backbone NCPs require no additional capacity.

Figure 78 on page 73 shows the outline of the network after conversion to APPN.

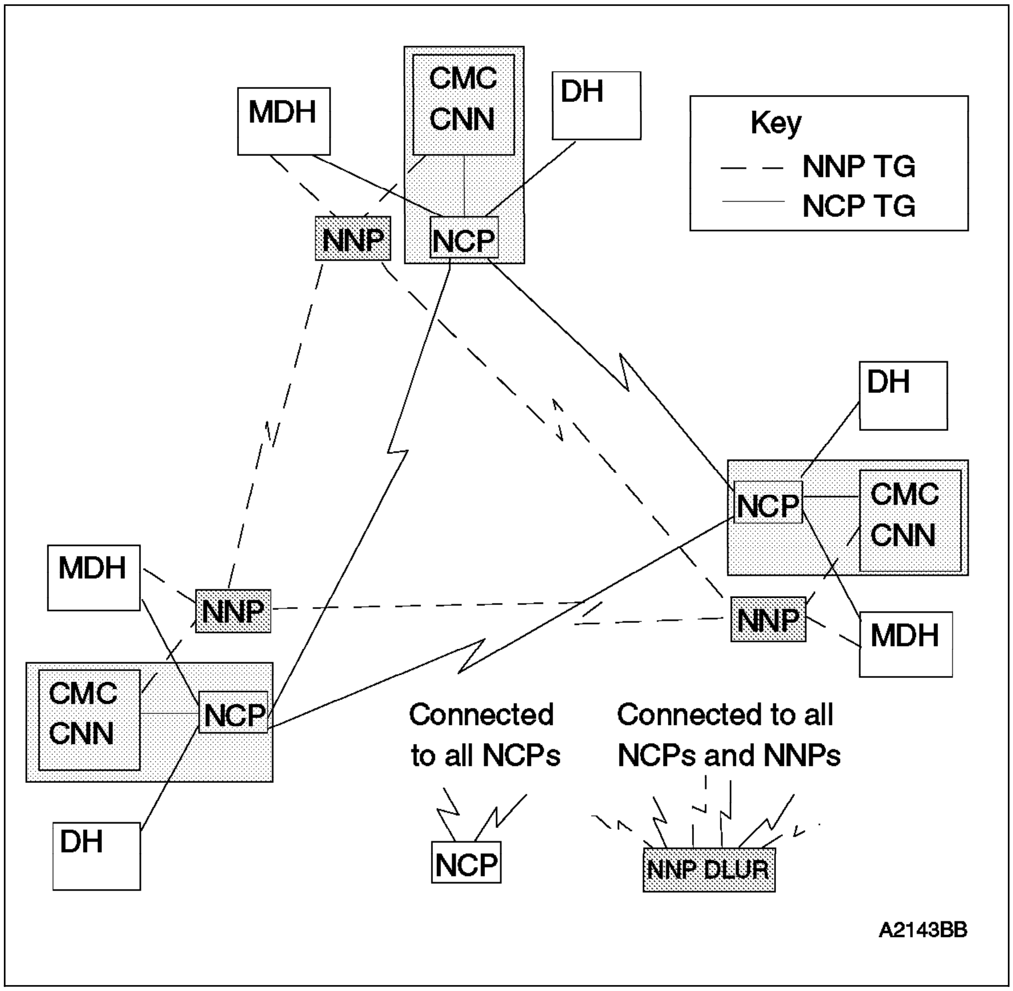


Figure 78. Proposed APPN Network

8.3 Routing Design

The essence of the design problem before us is to ensure that certain sessions take the 3746-950 NNP backbone to reach their destinations, while others take the NCP backbone. In order to solve this problem we must first analyze the network and its sessions. The design steps are:

1. Identify and classify the different types of links in the network
2. Identify and classify the different types of sessions that exist in the network
3. Identify the preferred paths for each type of session (NCP or 3746)
4. Determine the relative route and TG weights for the session paths
5. Determine the actual TG weights and characteristics
6. Plan and implement the APPN link and COS definition changes

Please note that this is a specific example, and has certain features of symmetry that make the design easier than it otherwise might be.

8.3.1 Identify the Types of Link

Figure 79 shows the classification of the link types. We are concerned primarily with the twin (NCP and NNP) backbone networks, so we ignore the "secondary" parts of the session paths and concentrate on where the sessions go after they have reached the backbone NCP or NNP.

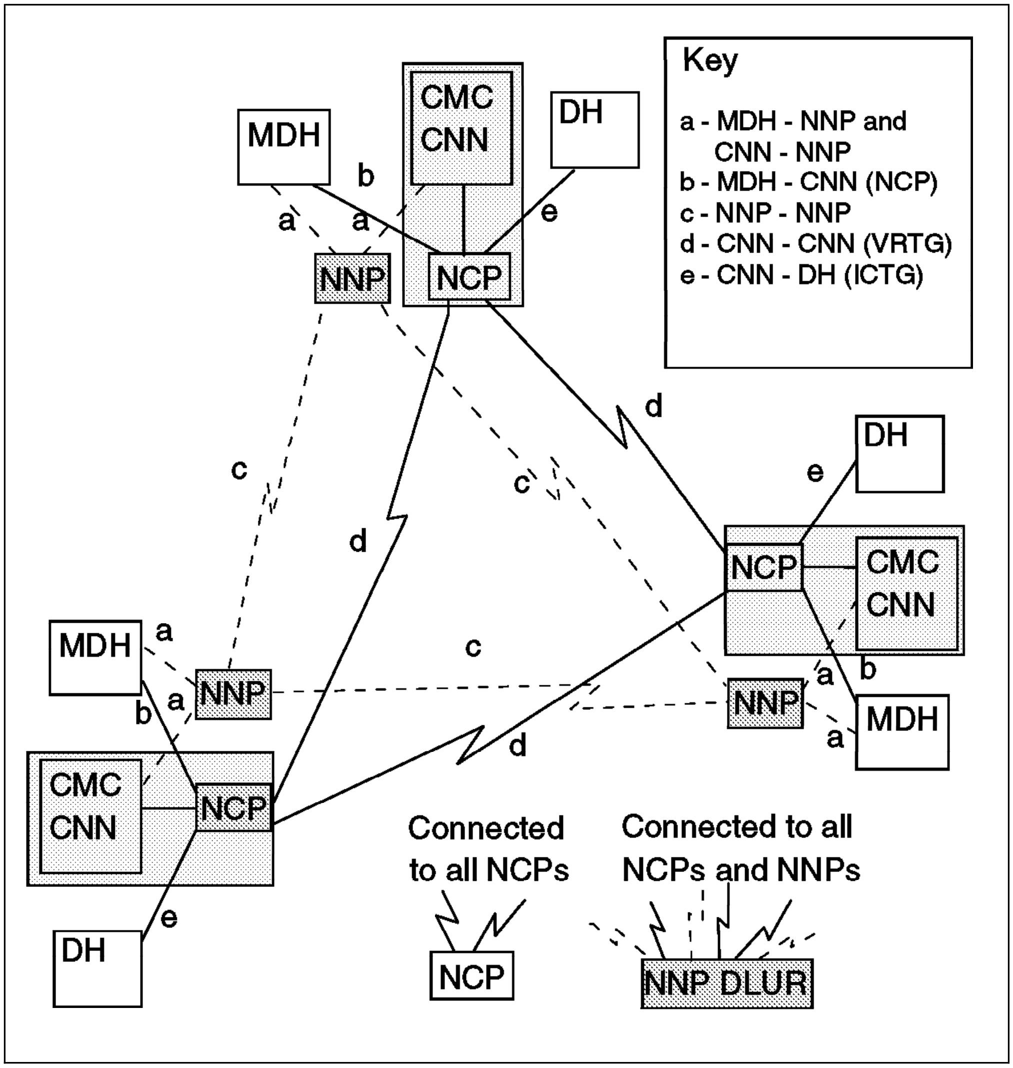


Figure 79. Link Types in Network

The classes of links we wish to distinguish between are as follows:

- **a** denotes connections between the new APPN nodes (3746, DLUR 2216s) and the existing subarea nodes (NCPs and VTAMs). These include the ESCON channel connections to the CMCs as well as the high-speed links between DLUR nodes and NCPs. They are all APPN TGs.
- **b** denotes channel connections between MDHs and their "parent" CMC hosts, via the CMC's NCP. These are also APPN TGs.
- **c** denotes the high-speed links between APPN NNs (3746s, DLUR 2216s), once again APPN TGs.
- **d** denotes the VR-TGs between each pair of CMC hosts.

- **e** denotes the subarea connections between the data hosts (which are not being converted to APPN MDHs) and their parent CMCs. These connections are represented as IC-TGs to the APPN network.

8.3.2 Identify the Types of Session

Nine different types of session are required to be analyzed:

- Sessions to the (subarea) data hosts:
  - Data host to data host (no change, continue to use subarea)
  - Dependent LU on NCP to data host
  - Dependent LU on 2216 (DLUR) to data host
  - APPC LU on 3746 to data host
- Sessions to the MDHs:
  - Data host to MDH
  - MDH to MDH
  - Dependent LU on NCP to MDH
  - Dependent LU on 2216 (DLUR) to MDH
  - APPC LU on 3746 to MDH

The DH-to DH-sessions do not concern us, as they will continue to use the same (subarea) paths as they use now. All the other types of sessions will have the opportunity to use APPN paths.

Figure 80 illustrates the first session type we analyze, that between an NCP LU and a data host.

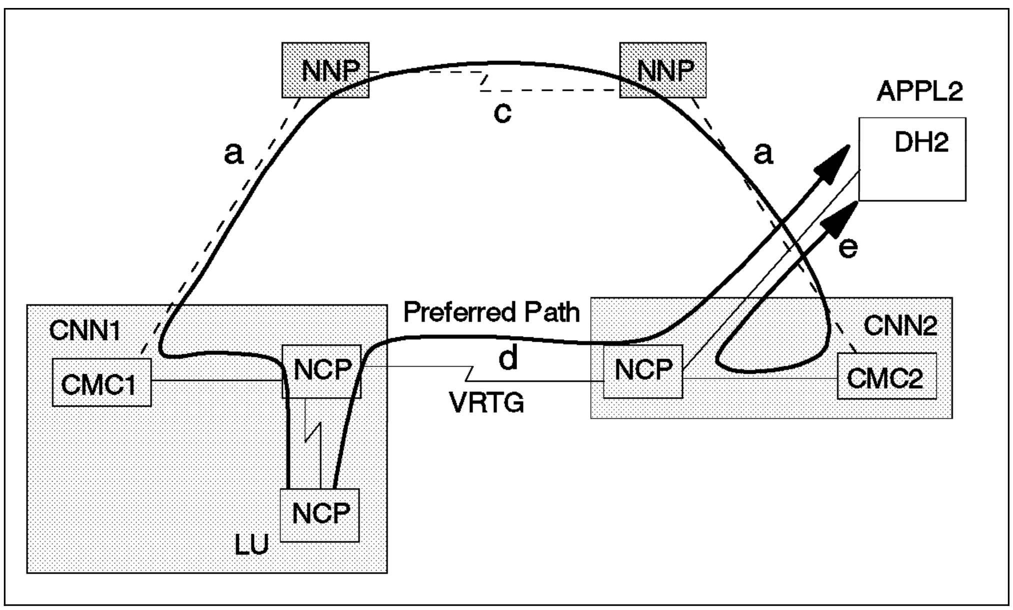


Figure 80. Sessions from NCP LU to Data Host

Here we can see the case where the data host and the NCP are on *different* CMCs. If the NCP and the data host connections are owned by the same CMC then the session path will be purely subarea and need not be considered. You will also notice that we have ignored the routes that go from NCP to DLUR 2216 to NCP. This is because:

1. They are parallel to the VR-TG routes that we prefer.

2. They are two-hop APPN routes, and therefore will not be chosen in preference to the (one-hop) VR-TG routes unless we make a real mess of our TG weights.
3. Even if these routes *are* chosen as backup routes, we are not unduly concerned. Our main concern is to ensure that the even longer routes via the CMC hosts are *not* preferred to the VR-TGs.

If the direct path via the VR-TG is to be chosen in preference to the path via two NNPs, the sum of the TG weights for the VR-TG must be less than the total weight of the other route. The formula we come up with is therefore:

$$\mathbf{d} < \mathbf{2a} + \mathbf{c}$$

Figure 81 shows the paths used by the sessions between the DLUR nodes and the data hosts. Both APPC LUs and DLUR LUs can be considered together because they have the same choices of route.

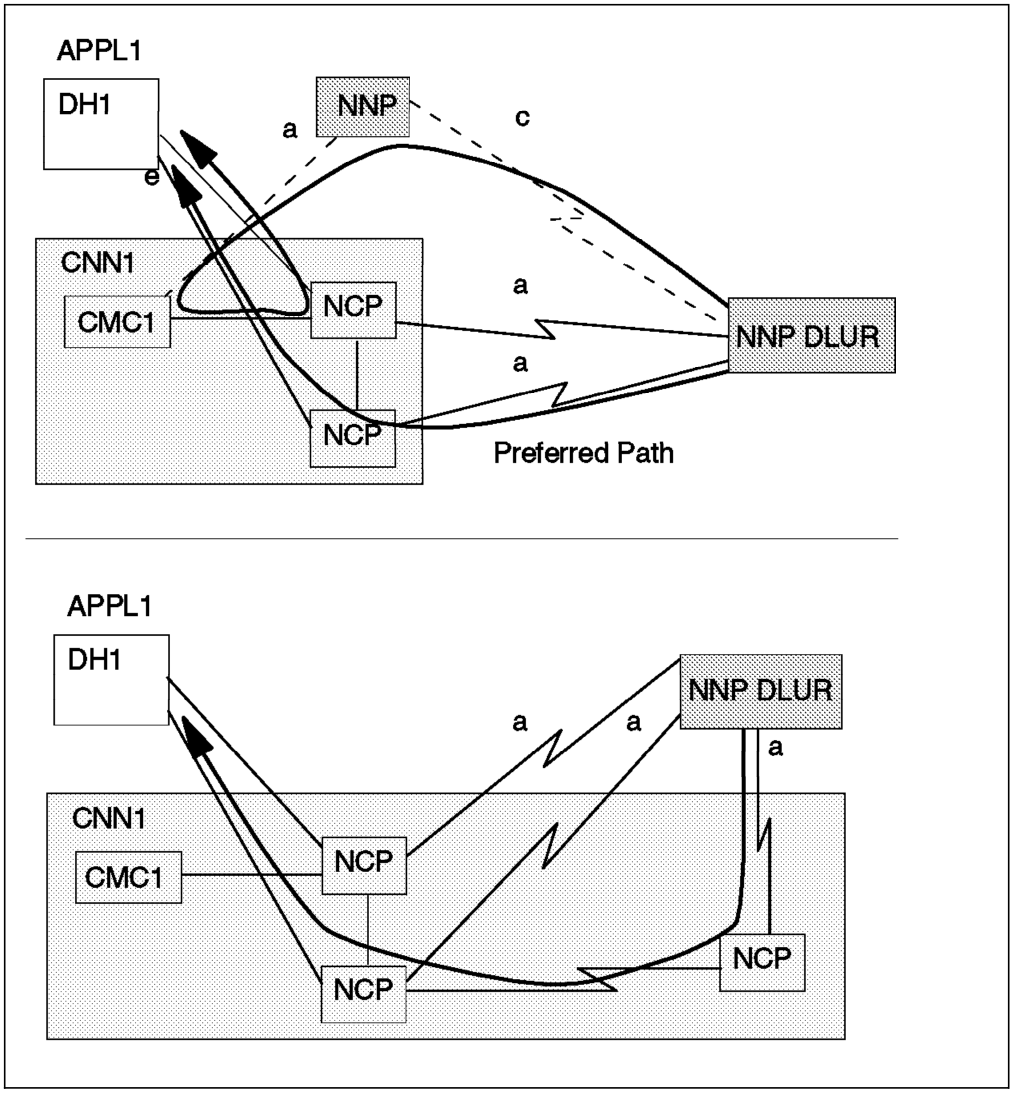


Figure 81. Sessions from DLUR to Data Host

The path through the NCPs is preferable to the path via a 3746, even if the session must pass through an extra remote NCP. Again, both possible paths use the IC-TG to reach the data host, so the formula for these sessions is:

a < **a** + **c**

This is trivial because we know that all TG weights are greater than zero.

Figure 82 shows the session routes between a data host and an MDH. Again (and subsequently where the VR-TGs figure in the diagrams) we ignore the NCP-DLUR-NCP paths.

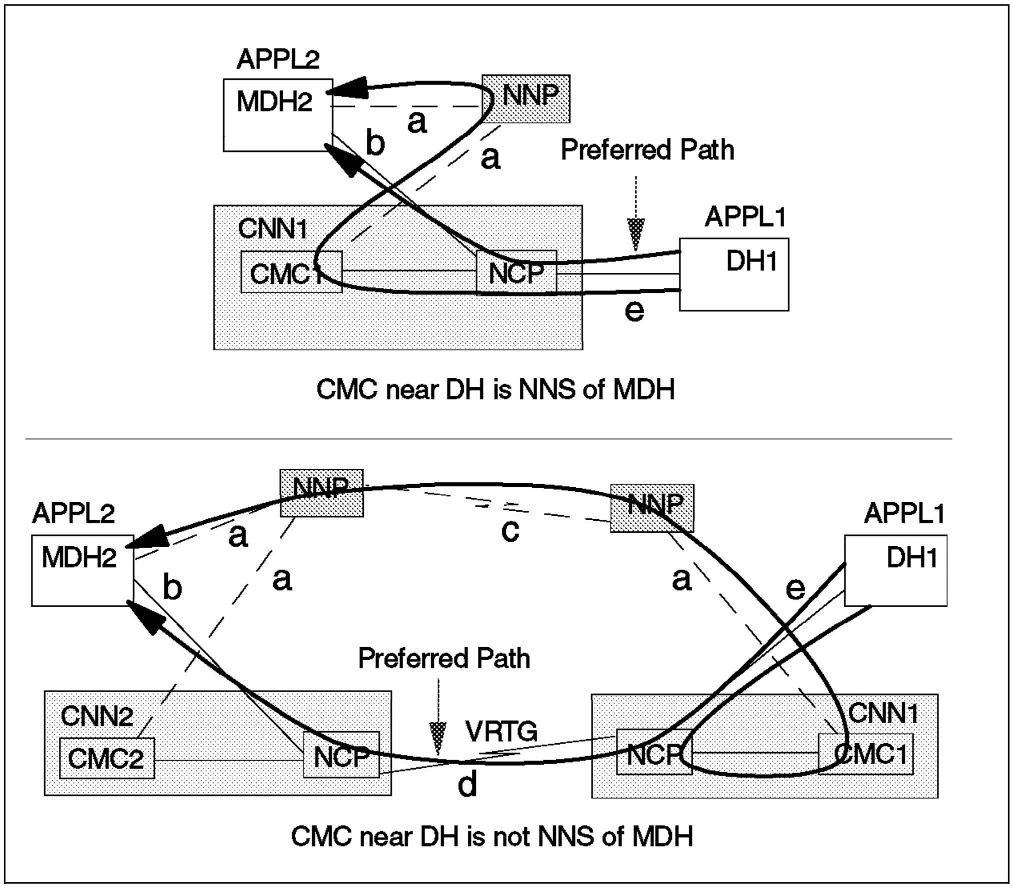


Figure 82. Sessions from MDH to Data Host

Here there are two distinct cases, depending on whether the CMC local to the data host is the network node server of the MDH. In each case, again, the NCP path is better and the IC-TG is common to both. Therefore, the two formulas for this session type are:

b + **d** < **2a** + **c**
b < **2a**

Figure 83 on page 78 illustrates the possible paths between a pair of MDHs.

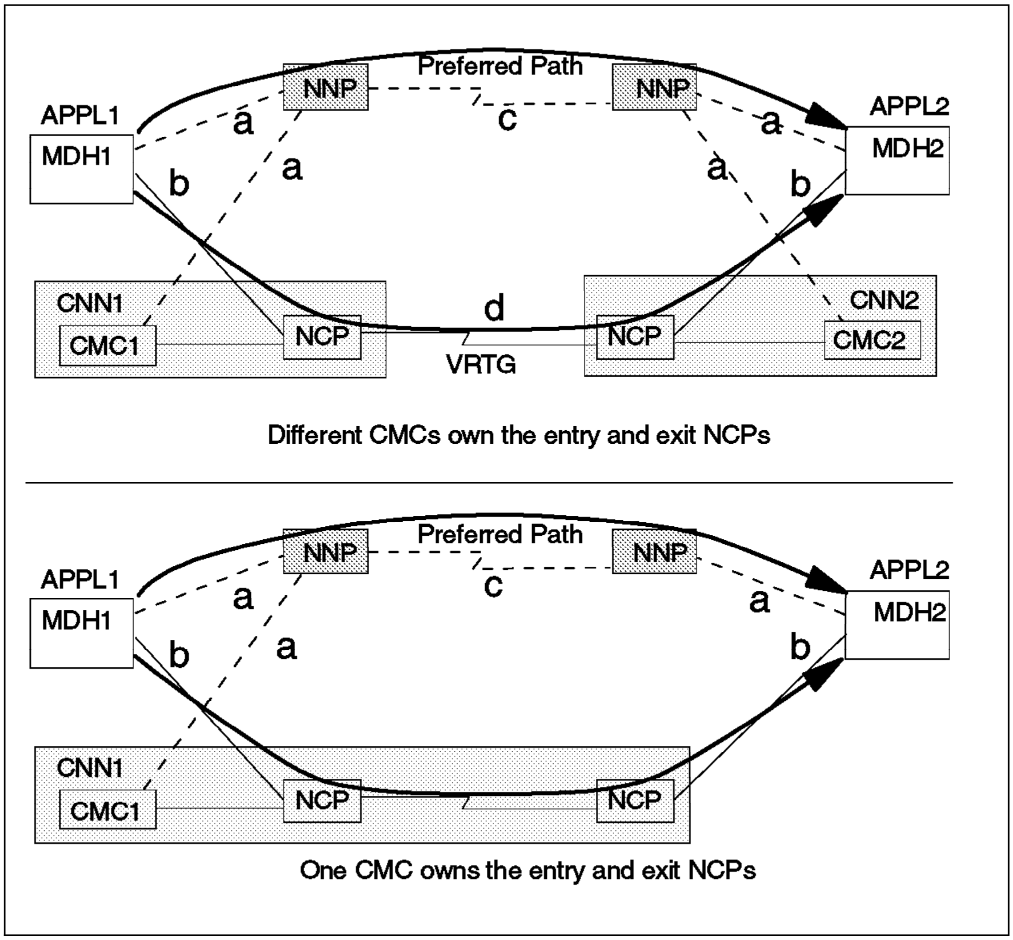


Figure 83. Sessions From MDH to MDH

Here there are also two cases, depending on whether the NCPs nearest to the MDHs are owned by the same CMC or not. There is no obvious advantage to either path, but we prefer the 3746 route because of NCP loading considerations. The formulas are therefore:

2a + c < 2b + d
2a + c < 2b

The second formula implies the first.

Figure 84 on page 79 shows the possible paths between an NCP LU and an MDH.

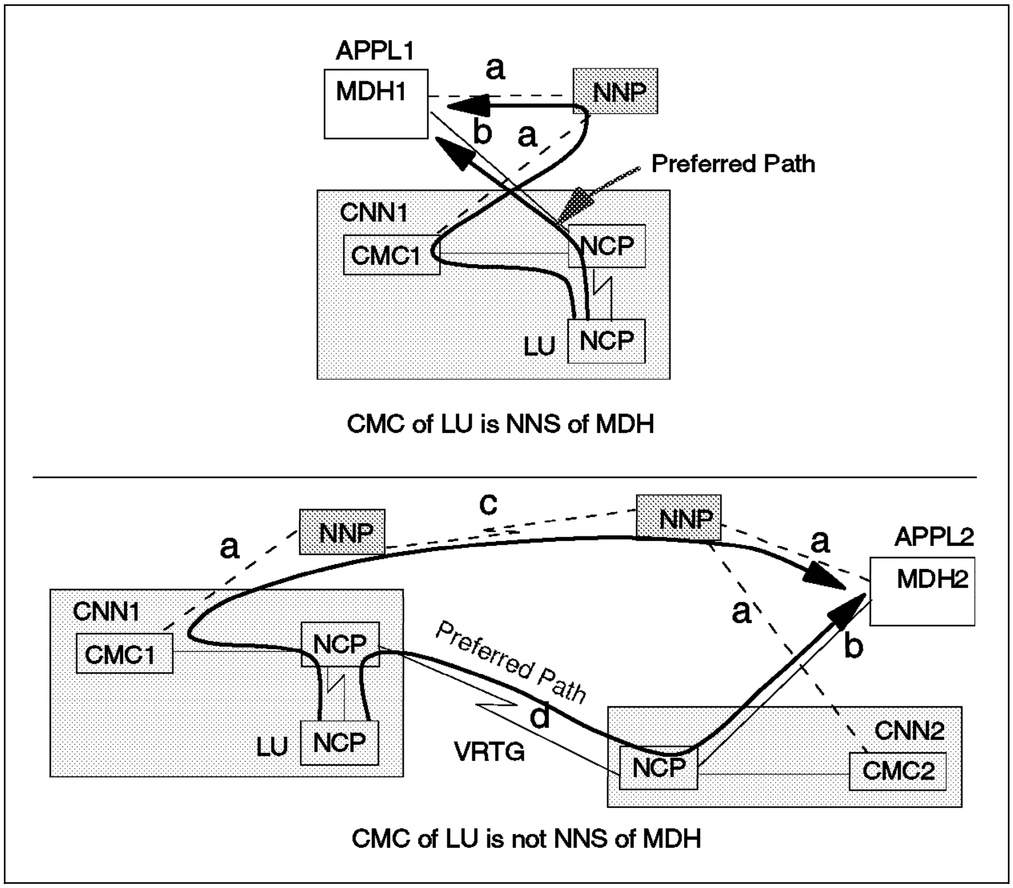


Figure 84. Sessions from NCP to MDH

The distinction between the two cases here is whether the owner of the NCP is also the network node server of the MDH. In either case the NCP route is preferable and so the formulas are:

b + d < 2a + c
b < 2a

These are identical to the formulas for the data host to MDH case.

Finally, Figure 85 shows the routes from a 2216 (APPC or DLUR LU) to an MDH.

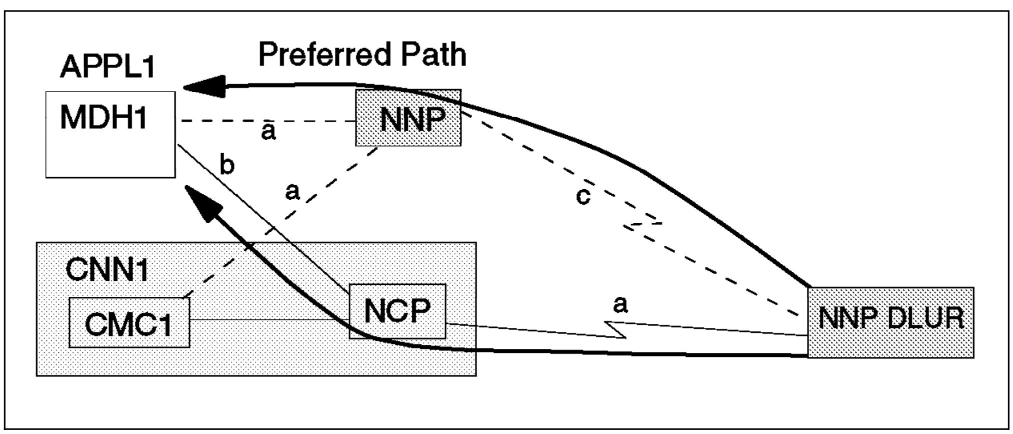


Figure 85. Sessions from DLUR to MDH

Here neither path is clearly better, but we again prefer the route through the 3746s to save NCP capacity. Therefore we have the formula:

$$\mathbf{a} + \mathbf{c} < \mathbf{a} + \mathbf{b}$$

Taking all the formulas we have obtained, we come up with a simple series of equations that when solved should tell us the relationships between the TG weights to ensure our preferred routes in each case. After we remove the trivial ones and the duplicates, the equations are now:

1. $\mathbf{d} < \mathbf{2a} + \mathbf{c}$
2. $\mathbf{b} + \mathbf{d} < \mathbf{2a} + \mathbf{c}$
3. $\mathbf{b} < \mathbf{2a}$
4. $\mathbf{2a} + \mathbf{c} < \mathbf{2b}$
5. $\mathbf{a} + \mathbf{c} < \mathbf{a} + \mathbf{b}$

(2) implies (1) and (5) means that $c < b$. There is now only one equation (2) involving d and we find that letting $d = c$ yields (3) again. So now we are left with:

- $\mathbf{c} < \mathbf{b} < \mathbf{2a}$
- $\mathbf{2a} + \mathbf{c} < \mathbf{2b}$

A flash of inspiration allows us to make $a = c$ and so we end up with $3a < 2b < 4a$. Now we can solve our problem using only two different TG weights throughout the network: one for the MDH to NCP connections (type b) and another for all the other connections (types a, c and d).

It does not matter what actual weights we choose as long as the formula $3a < 2b < 4a$ holds good. We choose $a = c = d = 90$ and $b = 150$ which satisfies the formula. Next we must plan the TG characteristics and the COS tables to ensure that the TGs have the correct weights.

8.3.3 COS and TG Definitions

To minimize the number of changes required, we decide on the following strategy:

1. All links except the MDH-to-NCP links will have TG characteristics as defined in the IBM-supplied standard TG profiles. We need at least to ensure that the link capacity values are correct because the HPR flow control algorithms depend on this.

2. The MDH to NCP links will be differentiated simply by changing the user-specified TG characteristic known as UPARM1. This is one of the three parameters provided for precisely this purpose. We code UPARM1=150 because the default (which will apply to all other links) is UPARM1=128. We must remember to code UPARM1=150 at both ends of each type b TG since TGs are bidirectional.

3. To enable APPN to distinguish between the UPARM1=150 TGs and the UPARM1=128 TGs, we devise our own COS table which takes account *only* of the UPARM1 parameter. We do not wish to influence route calculation in any other way.

Recall that not just TGs, but nodes, have a weight in APPN route calculation. Some of our preferred routes can traverse multiple nodes as well as our desired TGs. Therefore, we ensure that the weights assigned to nodes are significantly less than the weights assigned to TGs. In fact, this is exactly the same principle as in the IBM-supplied COS tables.

We define an identical (apart from transmission priority) COS table entry for each of the possible classes of service used in the network. Aside from the priority, we are only interested in distinguishing the UPARM1 values. Figure 86 shows the entry for the SNASVCMG class of service.

```
SNASVCMG APPNCOS  PRIORITY=NETWORK
         LINEROW  WEIGHT=90,                                          *
                  NUMBER=1,                                           *
                  UPARM1=(0,130),                                     *
                  UPARM2=(0,255),                                     *
                  UPARM3=(0,255),                                     *
                  CAPACITY=(MINIMUM,MAXIMUM),                         *
                  COSTTIME=(0,0),                                     *
                  COSTBYTE=(0,0),                                     *
                  PDELAY=(MINIMUM,MAXIMUM),                           *
                  SECURITY=(UNSECURE,MAXIMUM)
         NODEROW  NUMBER=1,                                           *
                  WEIGHT=5,                                           *
                  CONGEST=(LOW,LOW),                                  *
                  ROUTERES=(0,191)
         LINEROW  WEIGHT=150,                                         *
                  NUMBER=2,                                           *
                  UPARM1=(0,255),                                     *
                  UPARM2=(0,255),                                     *
                  UPARM3=(0,255),                                     *
                  CAPACITY=(MINIMUM,MAXIMUM),                         *
                  COSTTIME=(0,0),                                     *
                  COSTBYTE=(0,0),                                     *
                  PDELAY=(MINIMUM,MAXIMUM),                           *
                  SECURITY=(UNSECURE,MAXIMUM)
         NODEROW  NUMBER=2,                                           *
                  WEIGHT=10,                                          *
                  CONGEST=(LOW,HIGH),                                 *
                  ROUTERES=(0,255)
```

Figure 86. COS Table Entry

The principles are straightforward:

- There are only two TG entries (LINEROWs) as we wish to differentiate only between two types of link. The weights associated with the TGs are 90 and 150, as we have decided.
- There are also two node entries (NODEROWs) that distinguish between a congested node (weight 10) and an uncongested one (weight 5). This is not sufficient to disturb our TG scheme.
- Every TG characteristic except UPARM1 and the cost parameters has the maximum range specified in each TG entry. Thus every TG, whatever its characteristics, would match each TG entry in the COS were it not for the UPARM1. We have no switched TGs in the network that might have variable costs.
- The UPARM1 differs between the two entries. The weight 90 entry allows UPARM1 values between 0 and 130. The weight 150 entry allows all UPARM1 values between 0 and 255. Thus a TG with UPARM1 = 128 will

match the first entry and receive a weight of 90. A TG with UPARM1 = 150 will fail the first entry, but will match the second entry and receive a weight of 150.

8.4 Defining Case Study to NetDA/2

We entered the topology described earlier in this chapter to NetDA/2.

The Network Input Object network view is shown in Figure 87.

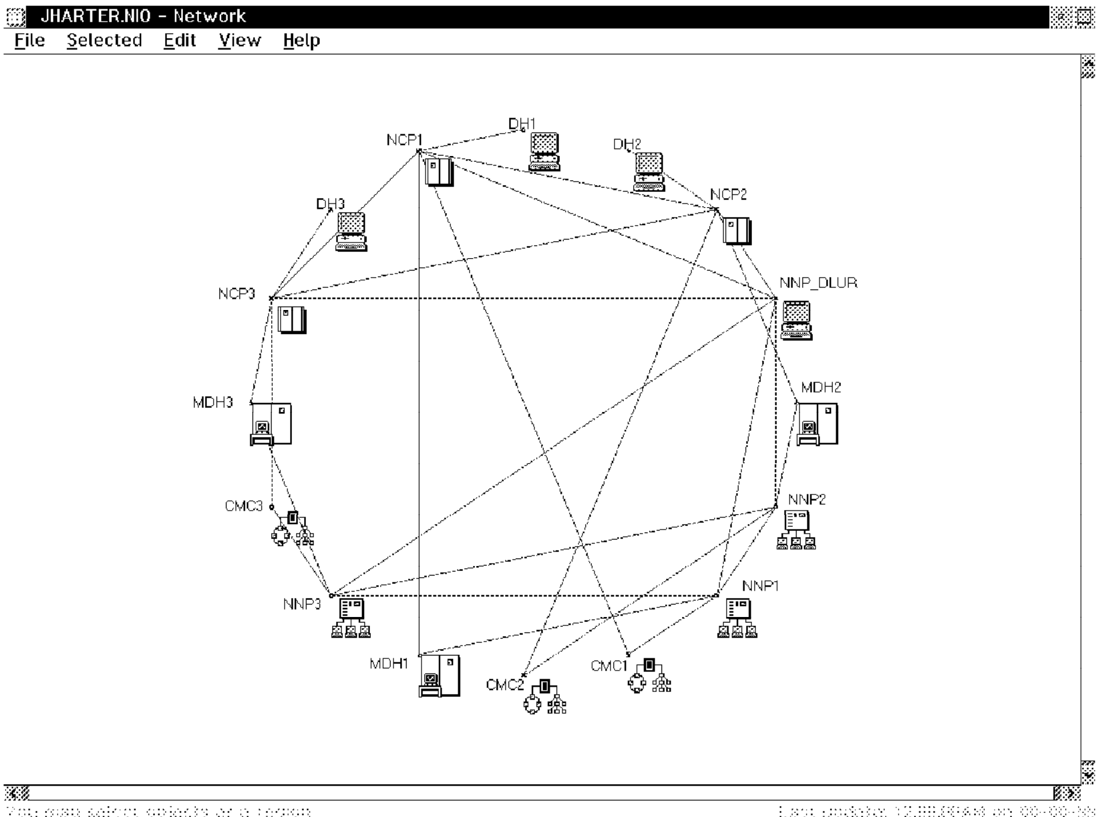


Figure 87. Network Input Object Network View

We made some simplifications to the network. We defined only one NNP DLUR node in the network and only one NCP per composite network node.

We defined the migration data hosts as type T5EN, the VTAMs in the composite network node as T5NN, the NCPs as T4AP, the data hosts as EN, the NNPs as NN and the NNP DLUR as NN.

You can get this NIO from the FTP site mentioned in “How to Access Samples from This Book” on page xi; the filename is REALCASE.NIO.

We modified the TGCOS values for line 1 of the COS entry SNASVCMG as shown in Figure 88 on page 83.

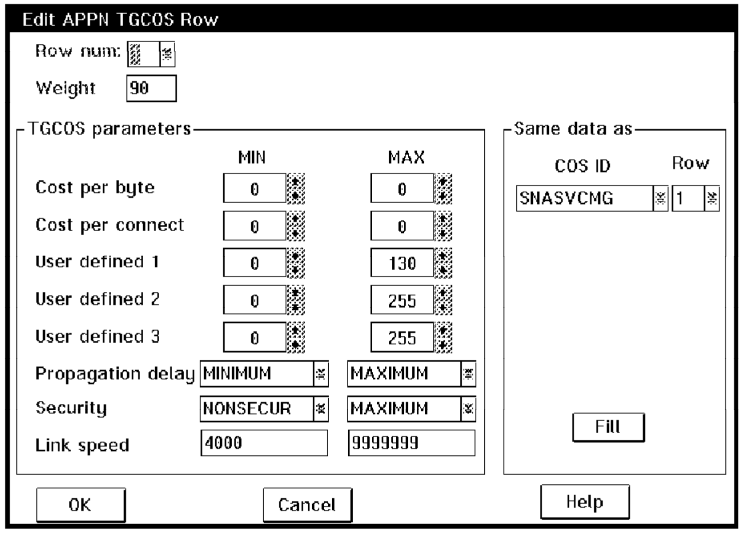


Figure 88. Line 1 TGCOS Values for SNASVCMG

The TG weight was set to 90 and the user-defined parameter 1 range to (0,130). Also the minimum link speed was reduced so that it would not be a factor in the TG line selection.

We also modified the TGCOS values of line 2 for the COS SNASVCMG as shown in Figure 89.

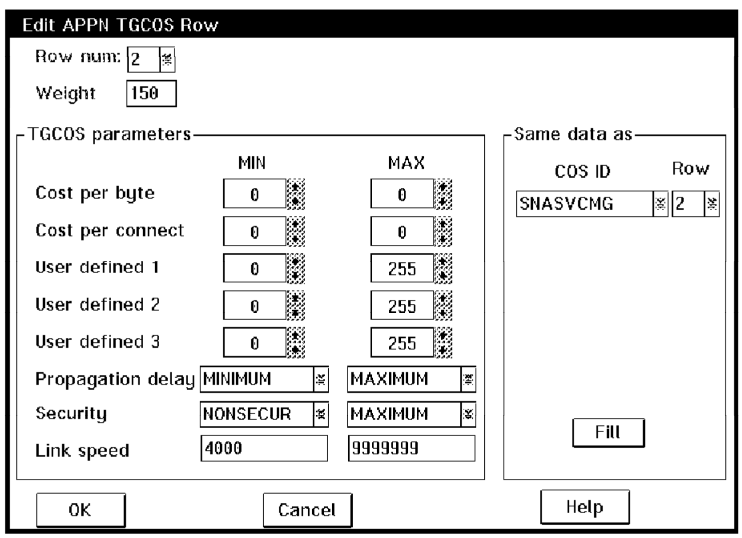


Figure 89. Line 2 TG APPN Characteristics for SNASVCMG

In this case we modified the weight to 150. Also the minimum Link Speed was reduced so that it would not be a factor in the TG line selection.

We also had to modify the APPN TG characteristics of the links between the NCPs and the migration data hosts, as shown in Figure 90 on page 84.

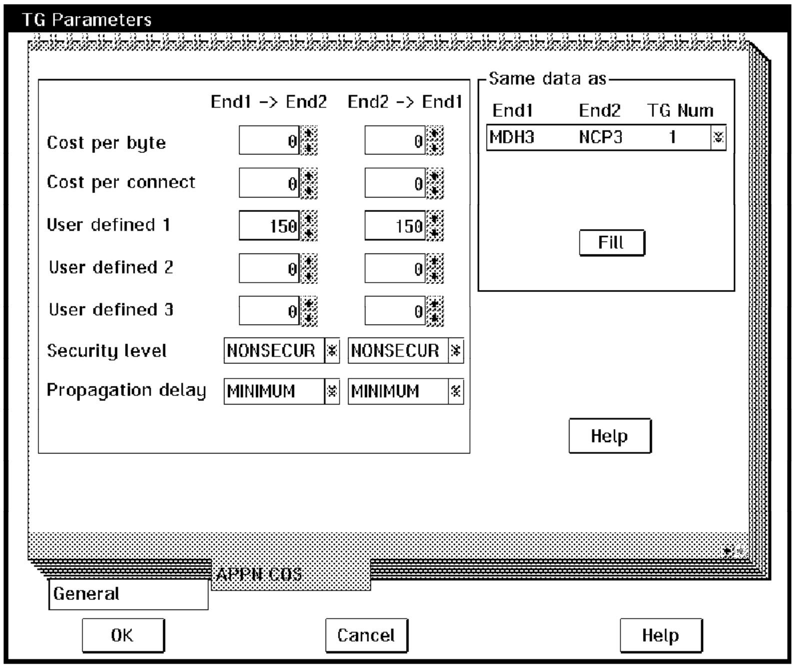


Figure 90. TG APPN Characteristics for the Link between MDH and NCP

Here we changed the User defined 1 parameter to 150; the rest of the TGs have a value of 128 for this parameter.

Once we made these definitions, the weights for the TGs were as shown in Figure 91 on page 85.

```
Appended at 14:10:21, Wednesday, 24 September 1997.  JHARTCOS.NDO

APPN Weights

 APPN Weight Report for Network Design Object JHARTCOS.NDO, Started Wed Sep 24 1997 14:09:55.

 Weights of All APPN Nodes in NetID NETA per COS Name SNASVCMG.
 Name       Weight     CON   RAR
 CMC1            5     LOW     0
 CMC2            5     LOW     0
 CMC3            5     LOW     0
 DH1             5     LOW     0
 DH2             5     LOW     0
 DH3             5     LOW     0
 MDH1            5     LOW     0
 MDH2            5     LOW     0
 MDH3            5     LOW     0
 NCP1            5     LOW     0
 NCP2            5     LOW     0
 NCP3            5     LOW     0
 NNP1            5     LOW     0
 NNP2            5     LOW     0
 NNP3            5     LOW     0
 NNP_DLUR        5     LOW     0

 Weights of All APPN TGs in NetID NETA per COS Name SNASVCMG.
 End1     End2     TG#  Wgt_12 Wgt_21 UD1_12 UD1_21    SPEED
 CMC1     NCP1       1      90     90    128    128  1544000
 CMC1     NNP1       1      90     90    128    128  1544000
 CMC2     NCP2       1      90     90    128    128  1544000
 CMC2     NNP2       1      90     90    128    128  1544000
 CMC3     NCP3       1      90     90    128    128  1544000
 CMC3     NNP3       1      90     90    128    128  1544000
 DH1      NCP1       1      90     90    128    128  1544000
 DH2      NCP2       1      90     90    128    128  1544000
 DH3      NCP3       1      90     90    128    128  1544000
 MDH1     NCP1       1     150    150    150    150  1544000
 MDH1     NNP1       1      90     90    128    128  1544000
 MDH2     NCP2       1     150    150    150    150  1544000
 MDH2     NNP2       1      90     90    128    128  1544000
 MDH3     NCP3       1     150    150    150    150  1544000
 MDH3     NNP3       1      90     90    128    128  1544000
 NCP1     NCP2       1      90     90    128    128    56000
 NCP1     NCP3       1      90     90    128    128    56000
 NCP1     NNP_DLUR 21       90     90    128    128    56000
 NCP2     NCP3       1      90     90    128    128    56000
 NCP2     NNP_DLUR 21       90     90    128    128    56000
 NCP3     NNP_DLUR 21       90     90    128    128    56000
 NNP1     NNP2       1      90     90    128    128    56000
 NNP1     NNP3       1      90     90    128    128    56000
 NNP1     NNP_DLUR 21       90     90    128    128    56000
 NNP2     NNP3       1      90     90    128    128    56000
 NNP2     NNP_DLUR 21       90     90    128    128    56000
 NNP3     NNP_DLUR 21       90     90    128    128    56000
```

Figure 91. Weights Report - Partial Listing

Figure 91 is a partial list of the Weights report generated by NetDA/2. The columns UD1_12, UD1_21 and SPEED are for information only.

Now we are ready to check if the routes chosen by NetDA/2 are the same as we expect.

In order to report the routes, we need to open the Route Control view of the network design object. We get the panel shown in Figure 92 on page 86.

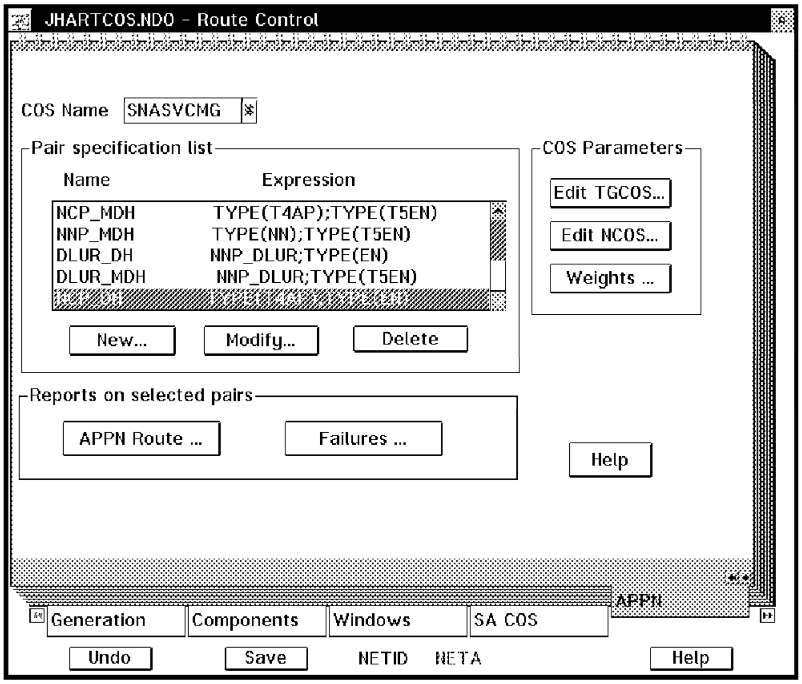


Figure 92. Route Control View - APPN Page

We must first select the COS name we will use, in our case SNASVCMG.

Now we need to build the expressions that will list the routes we are interested in. For example, in order to list the routes between the NCPs and the data hosts we could either enter:

```
(NCP1,NCP2,NCP3);(DH1,DH2,DH3)
```

or

```
TYPE(T4AP);TYPE(EN)
```

Instead of using the option Routes to list the routes, we use the option Failures. The advantage of this option is that we can also see the alternate route the traffic will take if one of the elements of the least weight route fails.

Let's see case by case.

8.4.1 NCP-to-DH Preferred Path

Using one of the pair specification expressions of above, we get the report shown in Figure 93 on page 87

```
Appended at 14:55:29, Wednesday, 24 September 1997.  JHARTCOS.NDO

APPN Routes with Failures

 APPN Least Weight Routes Report With Failures of Nodes and TGs
 for Network Design Object JHARTCOS.NDO, Started Wed Sep 24 1997 14:52:44.
 Pair specification: NCP_DH            TYPE(T4AP);TYPE(EN)
 APPN Weights are Defined by COS Name SNASVCMG.

 Subnet: APPN

 Origin:  NCP1,  Destination:  DH1,  Least Weight: 90.
   NCP1(1)DH1
 Failure:  DH1(1)NCP1,  No Route.


 Origin:  NCP1,  Destination:  DH2,  Least Weight: 185.
   NCP1(1)NCP2(1)DH2                               1
 Failure:  NCP2,  No Route.
 Failure:  DH2(1)NCP2,  No Route.
 Failure:  NCP1(1)NCP2,  Least Weight: 280.
   NCP1(1)NCP3(1)NCP2(1)DH2                        2
   NCP1(21)NNP_DLUR(21)NCP2(1)DH2                  3


 Origin:  NCP1,  Destination:  DH3,  Least Weight: 185.
   NCP1(1)NCP3(1)DH3
 Failure:  NCP3,  No Route.
 Failure:  DH3(1)NCP3,  No Route.
 Failure:  NCP1(1)NCP3,  Least Weight: 280.
   NCP1(1)NCP2(1)NCP3(1)DH3
   NCP1(21)NNP_DLUR(21)NCP3(1)DH3
```

Figure 93. NCP to DH Routes - Partial Listing

As we can see from this report, the least weight route is what we expected. The alternate route is not going through the NNPs (see Figure 80 on page 75).

For example, the main route between NCP1 and DH2 is NCP1(1)NCP2(1)DH2 **1** as we expected; the alternates are NCP1(1)NCP3(1)NCP2(1)DH2 **2** or NCP1(21)NNP_DLUR(21)NCP2(1)DH2 **3** and not going through the NNPs as shown in Figure 80 on page 75, but that′s OK. The main objective was accomplished.

The first alternate route (NCP1(1)NCP3(1)NCP2(1)DH2) **2** is only valid if the VR-TG has also a VR defined for NCP1(1)NCP3(1)NCP2.

If we want to avoid the use of the NNP_DLUR as an intermediate node, we have to increase the route resistance of this node (see 8.4.3, “DH to MDH Preferred Path” on page 88), as well as modifying the COS table to ensure that such a node is given a very high weight.

8.4.2 DLUR/APPC to DH Preferred Path

Using the expression NNP_DLUR;TYPE(EN) for the pair specification list, we get the report of Figure 94.

```
Appended at 15:09:34, Wednesday, 24 September 1997.  JHARTCOS.NDO

APPN Routes with Failures

 APPN Least Weight Routes Report With Failures of Nodes and TGs
 for Network Design Object JHARTCOS.NDO, Started Wed Sep 24 1997 14:56:00.
 Pair specification: DLUR_DH            NNP_DLUR;TYPE(EN)
 APPN Weights are Defined by COS Name SNASVCMG.

 Subnet: APPN

 Origin:  NNP_DLUR,  Destination:  DH1,  Least Weight: 185.
   NNP_DLUR(21)NCP1(1)DH1                             1
 Failure:  NCP1,  No Route.
 Failure:  DH1(1)NCP1,  No Route.
 Failure:  NCP1(21)NNP_DLUR,  Least Weight: 280.
   NNP_DLUR(21)NCP2(1)NCP1(1)DH1                      2
   NNP_DLUR(21)NCP3(1)NCP1(1)DH1                      3


 Origin:  NNP_DLUR,  Destination:  DH2,  Least Weight: 185.
   NNP_DLUR(21)NCP2(1)DH2
 Failure:  NCP2,  No Route.
 Failure:  DH2(1)NCP2,  No Route.
 Failure:  NCP2(21)NNP_DLUR,  Least Weight: 280.
   NNP_DLUR(21)NCP1(1)NCP2(1)DH2
   NNP_DLUR(21)NCP3(1)NCP2(1)DH2


 Origin:  NNP_DLUR,  Destination:  DH3,  Least Weight: 185.
   NNP_DLUR(21)NCP3(1)DH3
 Failure:  NCP3,  No Route.
 Failure:  DH3(1)NCP3,  No Route.
 Failure:  NCP3(21)NNP_DLUR,  Least Weight: 280.
   NNP_DLUR(21)NCP1(1)NCP3(1)DH3
   NNP_DLUR(21)NCP2(1)NCP3(1)DH3
```

Figure 94. DLUR/APPC to DH Routes - Partial Listing

The primary route is what we wanted (for example, NNP_DLUR(21)NCP1(1)DH1 **1**) as well as the alternates (NNP_DLUR(21)NCP2(1)NCP1(1)DH1 **2** or NNP_DLUR(21)NCP3(1)NCP1(1)DH1 **3**); see Figure 81 on page 76.

8.4.3 DH to MDH Preferred Path

Using the expression TYPE(T5EN);TYPE(EN) for the pair specification, we get the report of Figure 95 on page 89.

```
Appended at 15:10:42, Wednesday, 24 September 1997.  JHARTCOS.NDO

APPN Routes with Failures


 APPN Least Weight Routes Report With Failures of Nodes and TGs
 for Network Design Object JHARTCOS.NDO, Started Wed Sep 24 1997 15:10:09.
 Pair specification: DH_MDH             TYPE(EN);TYPE(T5EN)
 APPN Weights are Defined by COS Name SNASVCMG.

 Subnet: APPN

 Origin:  DH1,  Destination:  MDH1,  Least Weight: 245.
   DH1(1)NCP1(1)MDH1                                     1
 Failure:  NCP1,  No Route.
 Failure:  DH1(1)NCP1,  No Route.
 Failure:  MDH1(1)NCP1,  Least Weight: 375.
   DH1(1)NCP1(1)CMC1(1)NNP1(1)MDH1                       3
   DH1(1)NCP1(21)NNP_DLUR(21)NNP1(1)MDH1                 4


 Origin:  DH1,  Destination:  MDH2,  Least Weight: 340.
   DH1(1)NCP1(1)NCP2(1)MDH2                              2
 Failure:  NCP1,  No Route.
 Failure:  NCP2,  Least Weight: 375.
   DH1(1)NCP1(21)NNP_DLUR(21)NNP2(1)MDH2                 5
 Failure:  DH1(1)NCP1,  No Route.
 Failure:  MDH2(1)NCP2,  Least Weight: 375.
   DH1(1)NCP1(21)NNP_DLUR(21)NNP2(1)MDH2
 Failure:  NCP1(1)NCP2,  Least Weight: 375.
   DH1(1)NCP1(21)NNP_DLUR(21)NNP2(1)MDH2


 Origin:  DH1,  Destination:  MDH3,  Least Weight: 340.
   DH1(1)NCP1(1)NCP3(1)MDH3
 Failure:  NCP1,  No Route.
 Failure:  NCP3,  Least Weight: 375.
   DH1(1)NCP1(21)NNP_DLUR(21)NNP3(1)MDH3
 Failure:  DH1(1)NCP1,  No Route.
 Failure:  MDH3(1)NCP3,  Least Weight: 375.
   DH1(1)NCP1(21)NNP_DLUR(21)NNP3(1)MDH3
 Failure:  NCP1(1)NCP3,  Least Weight: 375.
   DH1(1)NCP1(21)NNP_DLUR(21)NNP3(1)MDH3
```

Figure 95. DH to MDH Routes - Partial Listing

The primary route is what we wanted. For example, DH1(1)NCP1(1)MDH1 **1** in the case where the CMC near the DH is the NNS of the MDH, and also DH1(1)NCP1(1)NCP2(1)MDH2 **2** in the case where the CMC near the DH is not the NNS of the MDH; see Figure 82 on page 77 for details.

One of the secondary routes for the case where the CMC near the DH is the NNS of the MDH is also what we wanted: DH1(1)NCP1(1)CMC1(1)NNP1(1)MDH1 **3** but it has the same weight as: DH1(1)NCP1(21)NNP_DLUR(21)NNP1(1)MDH1 **4**.

In the case where the CMC near the DH is not the NNS of the MDH, the backup route is not what we wanted; it is DH1(1)NCP1(21)NNP_DLUR(21)NNP2(1)MDH2 **5** instead of DH1(1)NCP1(1)CMC1(1)NNP1(1)NNP2(1)MDH2.

If we want to avoid the DLUR being used for intermediate sessions, we need to increase the route resistance of the NNP_DLUR node. Making the RAR of NNP_DLUR 255, which increases the weight of the node to 160 (see Figure 96), produces the results shown in Figure 97 on page 91.

Note that the COS table shown in Figure 86 on page 81 would result in a node weight of just 10 for a node with RAR 255. In fact we used the default NetDA/2 NCOS table which gave us a weight of 160. As we said before, in the real network it did not matter if the DLUR was used for a backup route; we merely illustrate a possible means of preventing it.

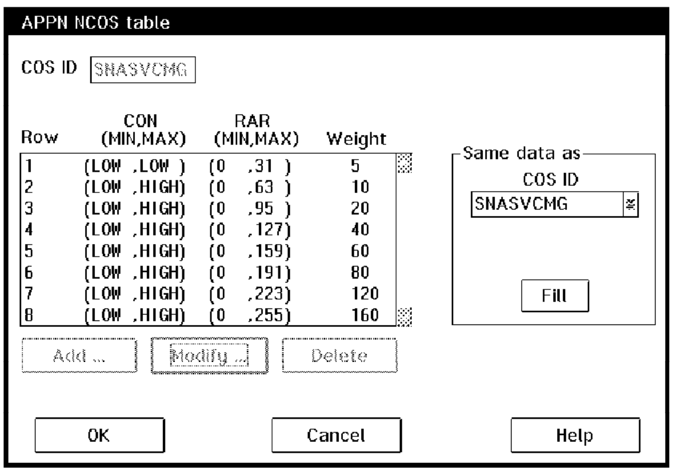


Figure 96. NetDA/2 NCOS

```
Appended at 09:20:39, Thursday, 25 September 1997.  JHARTCOS.NDO

APPN Routes with Failures

 APPN Least Weight Routes Report With Failures of Nodes and TGs
 for Network Design Object JHARTCOS.NDO, Started Thu Sep 25 1997 09:19:37.
 Pair specification: DH_MDH             TYPE(EN);TYPE(T5EN)
 APPN Weights are Defined by COS Name SNASVCMG.

 Subnet: APPN


 Origin:  DH1,  Destination:  MDH1,  Least Weight: 245.
   DH1(1)NCP1(1)MDH1
 Failure:  NCP1,  No Route.
 Failure:  DH1(1)NCP1,  No Route.
 Failure:  MDH1(1)NCP1,  Least Weight: 375.
   DH1(1)NCP1(1)CMC1(1)NNP1(1)MDH1


 Origin:  DH1,  Destination:  MDH2,  Least Weight: 340.
   DH1(1)NCP1(1)NCP2(1)MDH2
 Failure:  NCP1,  No Route.
 Failure:  NCP2,  Least Weight: 470.
   DH1(1)NCP1(1)CMC1(1)NNP1(1)NNP2(1)MDH2
 Failure:  DH1(1)NCP1,  No Route.
 Failure:  MDH2(1)NCP2,  Least Weight: 470.
   DH1(1)NCP1(1)NCP2(1)CMC2(1)NNP2(1)MDH2
   DH1(1)NCP1(1)CMC1(1)NNP1(1)NNP2(1)MDH2
 Failure:  NCP1(1)NCP2,  Least Weight: 435.
   DH1(1)NCP1(1)NCP3(1)NCP2(1)MDH2              1


 Origin:  DH1,  Destination:  MDH3,  Least Weight: 340.
   DH1(1)NCP1(1)NCP3(1)MDH3
 Failure:  NCP1,  No Route.
 Failure:  NCP3,  Least Weight: 470.
   DH1(1)NCP1(1)CMC1(1)NNP1(1)NNP3(1)MDH3
 Failure:  DH1(1)NCP1,  No Route.
 Failure:  MDH3(1)NCP3,  Least Weight: 470.
   DH1(1)NCP1(1)NCP3(1)CMC3(1)NNP3(1)MDH3
   DH1(1)NCP1(1)CMC1(1)NNP1(1)NNP3(1)MDH3
 Failure:  NCP1(1)NCP3,  Least Weight: 435.
   DH1(1)NCP1(1)NCP2(1)NCP3(1)MDH3
```

Figure 97. DH to MDH Routes - Partial Listing - RAR=255 for NNP_DLUR

The primary route is still what we wanted. The alternate route is also what we wanted in the case that the CMC near the DH is the NNS of the MDH.

In the case that the CMC near the DH is not the NNS of the MDH, then the backup route depends on the failure. It is the route we want if the failure is the NCP near the MDH or the link between the MDH and the NCP, but not if the failure is the link between both NCPs of the primary route (**1**). As before, this is true if the VR-TG between composite network nodes has a VR that links the three NCPs.

8.4.4 MDH to MDH Preferred Path

Using the expression TYPE(T5EN);= for for the pair specification list, we get the report shown in Figure 98 on page 93.

```
Appended at 15:11:05, Wednesday, 24 September 1997.  JHARTCOS.NDO

APPN Routes with Failures

 APPN Least Weight Routes Report With Failures of Nodes and TGs
 for Network Design Object JHARTCOS.NDO, Started Wed Sep 24 1997 15:10:54.
 Pair specification: MDH_MDH            TYPE(T5EN);=
 APPN Weights are Defined by COS Name SNASVCMG.

 Subnet: APPN


 Origin:  MDH1,  Destination:  MDH2,  Least Weight: 280.
   MDH1(1)NNP1(1)NNP2(1)MDH2
 Failure:  NNP1,  Least Weight: 400.
   MDH1(1)NCP1(1)NCP2(1)MDH2
 Failure:  NNP2,  Least Weight: 400.
   MDH1(1)NCP1(1)NCP2(1)MDH2
 Failure:  MDH1(1)NNP1,  Least Weight: 400.
   MDH1(1)NCP1(1)NCP2(1)MDH2
 Failure:  MDH2(1)NNP2,  Least Weight: 400.
   MDH1(1)NCP1(1)NCP2(1)MDH2
 Failure:  NNP1(1)NNP2,  Least Weight: 375.      1
   MDH1(1)NNP1(1)NNP3(1)NNP2(1)MDH2
   MDH1(1)NNP1(21)NNP_DLUR(21)NNP2(1)MDH2        2


 Origin:  MDH1,  Destination:  MDH3,  Least Weight: 280.
   MDH1(1)NNP1(1)NNP3(1)MDH3
 Failure:  NNP1,  Least Weight: 400.
   MDH1(1)NCP1(1)NCP3(1)MDH3
 Failure:  NNP3,  Least Weight: 400.
   MDH1(1)NCP1(1)NCP3(1)MDH3
 Failure:  MDH1(1)NNP1,  Least Weight: 400.
   MDH1(1)NCP1(1)NCP3(1)MDH3
 Failure:  MDH3(1)NNP3,  Least Weight: 400.
   MDH1(1)NCP1(1)NCP3(1)MDH3
 Failure:  NNP1(1)NNP3,  Least Weight: 375.
   MDH1(1)NNP1(1)NNP2(1)NNP3(1)MDH3
   MDH1(1)NNP1(21)NNP_DLUR(21)NNP3(1)MDH3


 Origin:  MDH2,  Destination:  MDH3,  Least Weight: 280.
   MDH2(1)NNP2(1)NNP3(1)MDH3
 Failure:  NNP2,  Least Weight: 400.
   MDH2(1)NCP2(1)NCP3(1)MDH3
 Failure:  NNP3,  Least Weight: 400.
   MDH2(1)NCP2(1)NCP3(1)MDH3
 Failure:  MDH2(1)NNP2,  Least Weight: 400.
   MDH2(1)NCP2(1)NCP3(1)MDH3
 Failure:  MDH3(1)NNP3,  Least Weight: 400.
   MDH2(1)NCP2(1)NCP3(1)MDH3
 Failure:  NNP2(1)NNP3,  Least Weight: 375.
   MDH2(1)NNP2(1)NNP1(1)NNP3(1)MDH3
   MDH2(1)NNP2(21)NNP_DLUR(21)NNP3(1)MDH3
```

Figure 98. MDH to MDH Routes - Partial Listing

The primary route and backup routes are what we expected; see Figure 83 on page 78. We can see that if the link between the NNPs of the primary route fails **1**, it is possible that NNP_DLUR will be used **2**. Increasing its RAR will prevent this from happening.

8.4.5 NCP to MDH Preferred Path

Using the expression TYPE(T5EN);TYPE(T4AP) for the pair specification list, we get the report of Figure 99.

```
Appended at 15:11:30, Wednesday, 24 September 1997.  JHARTCOS.NDO

APPN Routes with Failures

 APPN Least Weight Routes Report With Failures of Nodes and TGs
 for Network Design Object JHARTCOS.NDO, Started Wed Sep 24 1997 15:11:19.
 Pair specification: NCP_MDH            TYPE(T4AP);TYPE(T5EN)
 APPN Weights are Defined by COS Name SNASVCMG.

 Subnet: APPN


 Origin:  NCP1,  Destination:  MDH1,  Least Weight: 150.
   NCP1(1)MDH1
 Failure:  MDH1(1)NCP1,  Least Weight: 280.
   NCP1(1)CMC1(1)NNP1(1)MDH1
   NCP1(21)NNP_DLUR(21)NNP1(1)MDH1                              1


 Origin:  NCP1,  Destination:  MDH2,  Least Weight: 245.
   NCP1(1)NCP2(1)MDH2
 Failure:  NCP2,  Least Weight: 280.
   NCP1(21)NNP_DLUR(21)NNP2(1)MDH2                              1
 Failure:  MDH2(1)NCP2,  Least Weight: 280.
   NCP1(21)NNP_DLUR(21)NNP2(1)MDH2                              1
 Failure:  NCP1(1)NCP2,  Least Weight: 280.
   NCP1(21)NNP_DLUR(21)NNP2(1)MDH2                              1


 Origin:  NCP1,  Destination:  MDH3,  Least Weight: 245.
   NCP1(1)NCP3(1)MDH3
 Failure:  NCP3,  Least Weight: 280.
   NCP1(21)NNP_DLUR(21)NNP3(1)MDH3
 Failure:  MDH3(1)NCP3,  Least Weight: 280.
   NCP1(21)NNP_DLUR(21)NNP3(1)MDH3
 Failure:  NCP1(1)NCP3,  Least Weight: 280.
   NCP1(21)NNP_DLUR(21)NNP3(1)MDH3
```

Figure 99. NCP to MDH Routes - Partial Listing

The primary routes are what we expected (see Figure 84 on page 79).

For the backup routes, NNP_DLUR gets in the way **1**.

Increasing the resistance of NNP_DLUR to 255, we get the routes shown in Figure 100 on page 95.

```
Appended at 10:43:27, Thursday, 25 September 1997.  JHARTCOS.NDO

APPN Routes with Failures

 APPN Least Weight Routes Report With Failures of Nodes and TGs
 for Network Design Object JHARTCOS.NDO, Started Thu Sep 25 1997 10:43:01.
 Pair specification: NCP_MDH           TYPE(T4AP);TYPE(T5EN)
 APPN Weights are Defined by COS Name SNASVCMG.

 Subnet: APPN


 Origin:  NCP1,  Destination:  MDH1,  Least Weight: 150.
   NCP1(1)MDH1
 Failure:  MDH1(1)NCP1,  Least Weight: 280.
   NCP1(1)CMC1(1)NNP1(1)MDH1


 Origin:  NCP1,  Destination:  MDH2,  Least Weight: 245.
   NCP1(1)NCP2(1)MDH2
 Failure:  NCP2,  Least Weight: 375.
   NCP1(1)CMC1(1)NNP1(1)NNP2(1)MDH2
 Failure:  MDH2(1)NCP2,  Least Weight: 375.
   NCP1(1)NCP2(1)CMC2(1)NNP2(1)MDH2
   NCP1(1)CMC1(1)NNP1(1)NNP2(1)MDH2
 Failure:  NCP1(1)NCP2,  Least Weight: 340.
   NCP1(1)NCP3(1)NCP2(1)MDH2                       1


 Origin:  NCP1,  Destination:  MDH3,  Least Weight: 245.
   NCP1(1)NCP3(1)MDH3
 Failure:  NCP3,  Least Weight: 375.
   NCP1(1)CMC1(1)NNP1(1)NNP3(1)MDH3
 Failure:  MDH3(1)NCP3,  Least Weight: 375.
   NCP1(1)NCP3(1)CMC3(1)NNP3(1)MDH3
   NCP1(1)CMC1(1)NNP1(1)NNP3(1)MDH3
 Failure:  NCP1(1)NCP3,  Least Weight: 340.
   NCP1(1)NCP2(1)NCP3(1)MDH3                       1
```

Figure 100. NCP to MDH Routes - Partial Listing - RAR=255 for NNP_DLUR

The primary route and backup routes are now what we expected.

In the case of **1**, again it will depend on how the VR-TG was defined.

8.4.6 DLUR/APPC to MDH Preferred Path

Using the expression NNP_DLUR;TYPE(T5EN) for for the pair specification list, we get the report of Figure 101 on page 96.

```
Appended at 15:12:04, Wednesday, 24 September 1997.  JHARTCOS.NDO

APPN Routes with Failures

 APPN Least Weight Routes Report With Failures of Nodes and TGs
 for Network Design Object JHARTCOS.NDO, Started Wed Sep 24 1997 15:11:50.
 Pair specification: DLUR_MDH           NNP_DLUR;TYPE(T5EN)
 APPN Weights are Defined by COS Name SNASVCMG.

 Subnet: APPN


 Origin:  NNP_DLUR,  Destination:  MDH1,  Least Weight: 185.
   NNP_DLUR(21)NNP1(1)MDH1
 Failure:  NNP1,  Least Weight: 245.
   NNP_DLUR(21)NCP1(1)MDH1
 Failure:  MDH1(1)NNP1,  Least Weight: 245.
   NNP_DLUR(21)NCP1(1)MDH1
 Failure:  NNP1(21)NNP_DLUR,  Least Weight: 245.
   NNP_DLUR(21)NCP1(1)MDH1


 Origin:  NNP_DLUR,  Destination:  MDH2,  Least Weight: 185.
   NNP_DLUR(21)NNP2(1)MDH2
 Failure:  NNP2,  Least Weight: 245.
   NNP_DLUR(21)NCP2(1)MDH2
 Failure:  MDH2(1)NNP2,  Least Weight: 245.
   NNP_DLUR(21)NCP2(1)MDH2
 Failure:  NNP2(21)NNP_DLUR,  Least Weight: 245.
   NNP_DLUR(21)NCP2(1)MDH2


 Origin:  NNP_DLUR,  Destination:  MDH3,  Least Weight: 185.
   NNP_DLUR(21)NNP3(1)MDH3
 Failure:  NNP3,  Least Weight: 245.
   NNP_DLUR(21)NCP3(1)MDH3
 Failure:  MDH3(1)NNP3,  Least Weight: 245.
   NNP_DLUR(21)NCP3(1)MDH3
 Failure:  NNP3(21)NNP_DLUR,  Least Weight: 245.
   NNP_DLUR(21)NCP3(1)MDH3
```

Figure 101. DLUR/APPC to MDH Routes - Partial Listing

The primary and backup routes are what we expected (see Figure 85 on page 79).

8.5 Conclusions

The objective of this exercise was to show the convenience of using NetDA/2 to check how the APPN routes will be affected by changes in the COS tables and/or the APPN characteristics of TGs and nodes.

The user still has to make intelligent decisions on which values to change, but NetDA/2 can be used to corroborate the results of those decisions.

NetDA/2 can be especially useful in large networks, where it is not easy to detect all the possible paths between two given nodes.

Chapter 9. High-Performance Routing Support

In this section, we provide definitions of concepts in high-performance routing (HPR), discuss HPR support in NetDA/2, and then illustrate this support with an example.

9.1.1 Concepts of High-Performance Routing

High-performance routing (HPR) is an addition to APPN that enhances routing performance and session reliability; it takes advantage of fast links with low error rates. HPR replaces intermediate session routing with automatic network routing (ANR) and provides the following benefits:

- Nondisruptive session rerouting around failed nodes and links
- Improved intermediate routing performance
- Adaptive rate-based (ARB) congestion control to assure optimal link utilization

HPR operates with existing APPN nodes using existing APPN topology, control point protocols, and algorithms. LU-LU sessions can use HPR for part of the session path, without requiring end-to-end HPR.

HPR has two major components: rapid transport protocol (RTP) and automatic network routing (ANR).

Rapid transport protocol (RTP) is a connection-oriented, full-duplex transport protocol used in HPR for transporting data across HPR subnets. All data flowing over an RTP connection is carried in network layer packets (NLPs). RTP provides end-to-end recovery with selective retransmission, nondisruptive path switching, and ARB flow and congestion control procedures. As a boundary function, RTP can translate APPN to HPR protocols for sessions that have both HPR and non-HPR route segments. RTP can be implemented in end nodes and network nodes. Prior to V4R4, VTAM as an interchange node did not offer RTP capability.

Automatic network routing (ANR) is a low-level routing mechanism that minimizes cycle and storage requirements for routing packets through intermediate nodes. ANR is significantly faster than APPN routing. Unlike intermediate APPN nodes, intermediate ANR nodes are not aware of SNA sessions or RTP connections passing through the node. Network nodes can have ANR capability, but end nodes cannot. In composite network nodes, NCPs and VTAMs can provide ANR capability but only VTAM can be an RTP endpoint.

9.1.2 High-Performance Routing in NetDA/2

In the HPR_TYPE column in the LOCATIONS_OPTIONS table, you can indicate what type of HPR capability, if any, a node has:

- RTP means the node can be either an endpoint node or an intermediate node in an HPR route.
- ANR means the node can be an intermediate node in an HPR route.
- OT means the node has no HPR capability.

HPR_TYPE applies only to APPN nodes.

The TRND_HPR column in the EQUIPMENT table gives the one-way transit time in seconds for sending a message through the node when the node is an intermediate node in an HPR segment.

NetDA/2 can generate a report of backup paths for HPR route segments. The report lists primary APPN route segments between RTP nodes (that is, the segments of least weight). If a primary APPN route segment is an HPR segment, the report also lists backup paths for use when a node or TG in the primary APPN route segment fails. You can use this report to analyze HPR capability in your network. If the report includes primary APPN route segments that are not HPR segments, you might consider upgrading these segments to HPR. You might also consider upgrading backup paths to HPR. The report also reveals primary APPN route segments that do not have backup paths, if any.

Note that before you can generate an HPR backup routes report, you must have defined an APPN subnet in your design. Fill in the SUBNETS table and column SUBNET_NAME in the LOCATIONS_OUT table to define subnets. Alternately, use automated design to generate subnet definitions automatically.

9.1.3 Example of High-Performance Routing Support in NetDA/2

As an example, we use our APPN network and make five of our six NNs RTP-capable, with GRAFF_RE as the odd one out. This can be done using the Other data page of node graphical input illustrated in Figure 102 where we see the three HPR options, RTP, ANR, and OT (meaning other).

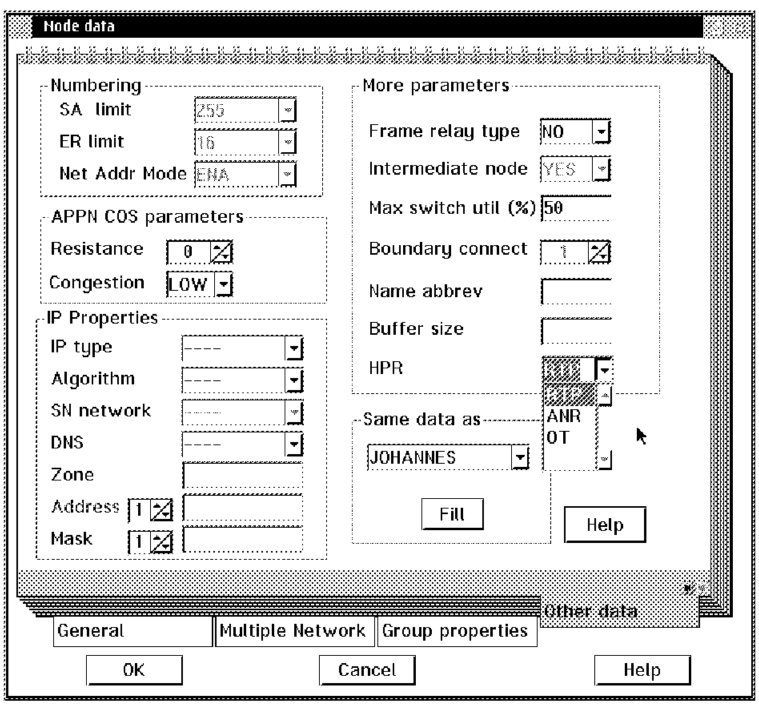


Figure 102. Specifying HPR for a Node

Alternatively, it may be quicker to edit the flat file and modify the HPR_TYPE column as shown in Figure 103 on page 101.

```
%TABLE LOCATIONS_OPTIONS

%NAME++++ CNN+++++ ER_LIM SA_LIM HPR_TYPE FR_TYPE BNDRY_CONN
 BEAUFORT POSSIBLE 16     255    OT       NO      1
 BLOEMFON POSSIBLE 16     255    OT       NO      1
 CALEDON  POSSIBLE 16     255    OT       NO      1
 CAPE_TOW POSSIBLE 16     255    RTP      NO      1
 CERES    POSSIBLE 16     255    OT       NO      1
 CRADOCK  POSSIBLE 16     255    OT       NO      1
 DE_AAR   POSSIBLE 16     255    RTP      NO      1
 . . .
```

Figure 103. HPR Definition in LOCATIONS_OPTIONS Table

The HPR Report is generated by selecting **Reports** on the ND′s pop-up menu and then **HPR Backup Routes**. The output contains a listing for each COS name, of the HPR routes and backup routes available in the event of component failure.

Figure 104 on page 102 has the output for COS #INTER and routes between the node pair CAPE_TOW and DURBAN. There are two HPR routes with backups for component failure that include some HPR and some non-HPR routes.

```
HPR backup routes report started    Fri Sep 12 1997 13:16:12

================================================================================
Netid:  NETA
Cosid:  #INTER

Number of RTP nodes in this subnet: 5
================================================================================
        RTP source node: CAPE_TOW
        RTP target node: DURBAN

        Subnet: @_SUBNO   Netid: NETA   Cosid: #INTER

        Number of HPR primary routes:  2
        Number of non-HPR primary routes:  0

                HPR primary route:       CAPE_TOW(1)JOHANNES(1)DURBAN

                        Failure                     Backup    Routes

                        JOHANNES                    Non-HPR   CAPE_TOW(1)DE_AAR(1)QUEENSTO(1)DURBAN
                        CAPE_TOW(1)JOHANNES         Non-HPR   CAPE_TOW(1)DE_AAR(1)QUEENSTO(1)DURBAN
                        JOHANNES(1)DURBAN           HPR       CAPE_TOW(1)JOHANNES(2)DURBAN

                HPR primary route:       CAPE_TOW(1)JOHANNES(2)DURBAN

                        Failure                     Backup    Routes

                        JOHANNES                    Non-HPR   CAPE_TOW(1)DE_AAR(1)QUEENSTO(1)DURBAN
                        CAPE_TOW(1)JOHANNES         Non-HPR   CAPE_TOW(1)DE_AAR(1)QUEENSTO(1)DURBAN
                        JOHANNES(2)DURBAN           HPR       CAPE_TOW(1)JOHANNES(1)DURBAN
--------------------------------------------------------------------------------
        RTP source node: CAPE_TOW
        RTP target node: GRAFF_RE

        Subnet: @_SUBNO   Netid: NETA   Cosid: #INTER

        Number of HPR primary routes:  1
        Number of non-HPR primary routes:  0

                HPR primary route:       CAPE_TOW(1)JOHANNES(1)GRAFF_RE

                        Failure                     Backup    Routes

                        JOHANNES                    Non-HPR   CAPE_TOW(1)DE_AAR(1)GRAFF_RE
                        CAPE_TOW(1)JOHANNES         Non-HPR   CAPE_TOW(1)DE_AAR(1)GRAFF_RE
                        JOHANNES(1)GRAFF_RE         Non-HPR   CAPE_TOW(1)DE_AAR(1)GRAFF_RE
 . . .
```

Figure 104. HPR Backup Routes Report

Upgrading the node DE_AAR to RTP capability would provide a full set of HPR backup routes. In fact, giving it ANR capability would give backup to these routes but would not make any of DE_AAR's routes HPR.

Chapter 10. Collecting APPN Traffic Data for NetDA/2

In a typical SNA subarea network we became used to collecting networking performance and traffic data from a focal point. Typically this data was collected on the VTAM SSCP hosts by using NetView Performance Monitor.

However, in the modern APPN environments, we have to make use of different methods to collect all data at a central point. Naturally, if an APPN session is not going through, starting or ending in a VTAM where NPM data is collected, there is no way for NPM to collect this distributed data at a focal point. One of the ways to gather at least some data is by making use of APPN Accounting Manager.

The APPN accounting agent collects APPC session and conversation accounting data from Communications Server/2 running on an OS/2 operating system. In this chapter we look at how we can collect APPC traffic data from CS/2 using APPN Accounting Manager.

10.1 APPN and Accounting Data Collection - An Overview

APPN Accounting Manager is a manager-agent type of application where the agent runs on an OS/2 Communications Server/2 platform. The manager application runs on a NetView OS/390 host. The agent collects the accounting records that in turn are forwarded to the OS/390 host where the APPN Accounting Manager, which is a subtask of the NetView subsystem, will collect these records. NetView will file these records in the MVS SMF database. The manuals refer to a user-defined external log to which you can write your records. This user log only supports a record length of up to 256 bytes. This will not be of any use to you as the SMF type 39 records will be truncated to such an extent that it will be virtually unusable. Communication support between these applications is by using LU6.2 sessions using OSI Communication Management Information Protocols (CMIP) services and SNA Multiple Domain Support.

The APPN Accounting Agent can collect the following two types of APPC data:

- APPC Conversation Data
- APPC Session Accounting Data

APPC conversation data contains counters such as:

- RU bytes sent from the source to target transaction program
- RU bytes sent from the target to source transaction program

Session accounting data will typically contain counters such as:

- PIUs sent from primary to secondary
- PIU bytes sent from primary to secondary
- PIUs sent from secondary to primary
- PIU bytes sent from secondary to primary

The SNA Topology Manager and APPN Accounting Manager functions require NetView V3R1 or later. They also require C/370 V2R1 run-time libraries on the

MVS host. The Accounting Manager application requires an external log such as the System Management Facility (SMF).

The APPN Accounting Manager runs on VTAM V4R3 and later and MVS/ESA V3R1.3 or later.

The APPN accounting agents require Communications Manager 1.1 or later and OS/2 2.0 or later.

In our test environment we were running NetView V3R1 as one manager and TME 10 NetView for OS/390 V1R1 as a second manager.

APAR OW29809 must be installed in NetView in order to get intermediate session records correctly formatted.

On the OS/2 platforms we were running Communications Server for OS/2 Warp Version 4.1 and OS/2 Version 3.

10.2 The APPN Accounting Manager

Let us take a more detailed look at the Accounting Manager. The Accounting Manager is controlled by the NetView subsystem under OS/390. The Accounting Manager essentially comprises two parts in NetView:

- A NetView command processor named FLBGMCMD
- A NetView autotask operator named FLBGMMGR

The Accounting Manager works hand in hand with the accounting agent to collect accounting records at the agent nodes. The Accounting Manager formats these retrieved records and writes them to the SMF log.

The following is a listing of some of the functions performed by the Accounting Manager:

- Operator commands to start or stop collection of session and conversation data at agent nodes
- Operator commands to display the status of collecting nodes
- Operator commands to list agents
- Operator commands to query or change current settings
- Retrieving data from agent nodes in the network
- Formatting data to SMF external log

Depending on your specific requirements, you can decide how often to collect data from the agent nodes. Some options available to you to collect data from the agent are:

- Issue a NetView operator command.
- Retrieve the data at regular intervals. (timer command)
- Retrieve the data when the agent node buffer utilization reaches a certain threshold.

10.3 The Manager and Client Interacting

Now we take a look at the sequence of events that occurs when the NetView operator issues the ACCTSNA command in NetView.

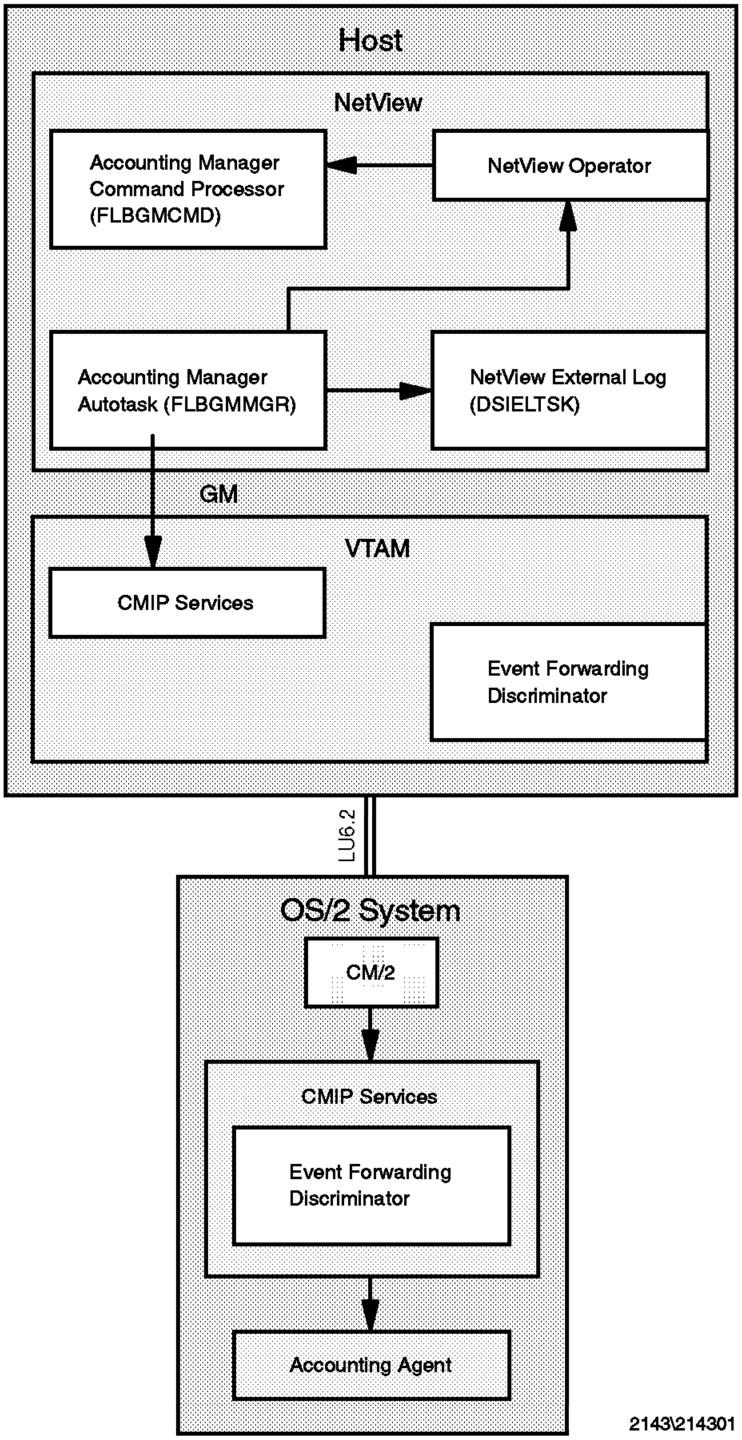


Figure 105. APPN Accounting Manager and Client interaction

1. The NetView operator issues the ACCTSNA command on NetView.

2. The NetView command processor called FLBGMCMD parses and syntax checks the ACCTSNA command and passes the command to the autotask operator ID, FLBGMMGR.
3. The autotask FLBGMMGR receives and executes the command.
4. If this ACCTSNA command needs to be forwarded to the agent node, the autotask FLBGMMGR formats a CMIP request and passes it to VTAM CMIP services.
5. VTAM formats a CMIP request and forwards it to the agent CMIP services.
6. Agent CMIP services receives the request and forwards it to the accounting agent for processing.
7. The accounting agent processes the request and sends any replies back to the Accounting Manager.
8. When accounting records are retrieved from the agent node, the Accounting Manager formats the records in SMF format and passes them to the NetView external logging task called DSIELTSK.
9. DSIELTSK sees that the record gets logged in the SMF data set or a NetView external log, depending on how NetView is customized.
10. The Event Forwarding Discriminator is there to prevent the agent nodes from causing unnecessary data flows to occur. The Event Forwarding Discriminator ensures that the agent only responds to requests from the manager, and also ensures that no unsolicited messages are forwarded to the manager application.

10.4 What Can We Do with the APPN Accounting Data?

There are several reasons for collecting APPN accounting data, which we detail in the following sections.

10.4.1 Accounting

You can use the accounting data for usage reporting and billing. A typical situation is where you are charging users for the use of APPN network resources. The APPN accounting agents collect session and conversation data such as when the session or conversation began, when and why it ended, and the number of bytes sent and received.

10.4.2 Capacity Planning

We can use the APPN Accounting feature to provide us with statistics regarding the amount of traffic data that either passed through or ended within a particular node. This is useful information when doing node sizing.

10.4.3 Tuning

This data can be of great use during your tuning exercises to determine the source and destination of data traffic in your APPN network.

10.4.4 Link Utilization

By monitoring the flow of non-FMD (control) messages, you will get an idea of whether your line was congested or not. An increase in non-FMD messages over a period of time indicates a line contention situation.

10.4.5 Charging Users for Database Usage

You can set up an accounting agent on each node where there is a DB2/2 database to capture the conversation records. Using these records you are able to bill clients for the usage of the database.

10.4.6 Load Balancing Nodes

You can set up an accounting agent at each node to collect intermediate session and accounting data. In the external log, you can use information about each session from the session information data section to determine such session characteristics as name and class of service (COS). You might discover that customers are using certain mode names more than others at a node. Or you might determine that the COS definition is not appropriate for the type of data going through the node or for the node capacity.

10.4.7 NetDA/2

In the SNA world, we can use NPM to collect data and log the records into an external file. From the external file we can create an input file for NetDA/2. NPM cannot collect session and conversation data for APPN sessions that are not flowing through the VTAM where collection is taking place. However, we can make use of the APPN accounting agent and manager to retrieve the records from the APPN agent. The manager will then create SMF records for sessions on or through the CS/2 machine.

10.5 Customizing the OS/390 Host to Run the Accounting Manager

The Accounting Manager application needs VTAM CMIP services to forward requests and receive replies from the agent nodes. You can run through the following list to ensure that your VTAM is CMIP services enabled.

10.5.1 VTAM

We describe the different steps you should follow to verify that VTAM is able to handle the CMIP services.

10.5.1.1 Check the VTAM Procedure

Check that the following DD names are there:

1. SISTCLIB
2. ISTCMIP
3. ISTASN1
4. ACYGDMO

Following is an example VTAM procedure. Please make sure that the last four DD statements as listed in the example are present in your installation′s VTAM procedure.

It is always good practice not to use the SMPE data sets. We suggest you create user data sets. In this case we created our own data sets called:

- NET.SISTCLIB using the attributes of SYS1.SISTCLIB
- NET.SISTCMIP using the attributes of SYS1.SISTCMIP
- NET.SISTASN1 using the attributes of SYS1.SISTASN1
- NET.SISTGDMO using the attributes of SYS1.SISTGDMO

```
//NET    PROC  PERF=13
//NET    EXEC  PGM=ISTINM01,REGION=0M,TIME=1440,DPRTY=(15,13),
//       PERFORM=&PERF
//STEPLIB DD DSN=SSP.V4R2.SSPLIB,DISP=SHR
//        DD DSN=NET.VTAMLIB,DISP=SHR
//VTAMLIB DD DSN=NET.VTAMLIB,DISP=SHR
//        DD DSN=SYS1.VTAMLIB,DISP=SHR
//        DD DSN=SYS1.NETVIEW.V3R1M0.SCNMLNK1,DISP=SHR
//VTAMLST DD DSN=NET.VTAMLST,DISP=SHR
//NODES1  DD DSN=NET.NODELST,DISP=OLD,AMP=AMORG
//        DD DSN=NET.NCPLOAD,DISP=SHR
//SYSABEND DD DUMMY
//SISTCLIB DD DSN=NET.SISTCLIB,DISP=SHR
//ISTCMIP DD DSN=NET.SISTCMIP,DISP=SHR
//ISTASN1 DD DSN=NET.SISTASN1,DISP=SHR
//ACYGDMO DD DSN=NET.SISTGDMO(ACYGDMO),DISP=SHR
```

Figure 106. A Sample VTAM Procedure

If the DD statements are not there then you should install the VTAM CMIP services on your VTAM system (refer to *VTAM Installation and Migration Guide*, GC31-8367). If the DD names are there, then you must check that the VTAM OSIMGMT start option is enabled.

10.5.1.2 Check OSIMGMT Startup Parameter under VTAM

Do this by issuing the following command on NetView:
`D NET,VTAMOPTS,OPTION=OSIMGMT.` Ensure that OSIMGMT is equal to YES.

```
* RAKAN    D NET,VTAMOPTS,OPTION=OSIMGMT
  RAKAN    IST097I  DISPLAY  ACCEPTED
' RAKAN
IST1188I  ACF/VTAM V4R4   STARTED AT 11:32:50 ON 09/07/97
IST1349I  COMPONENT ID IS 5695-11701-401
IST1348I  VTAM STARTED AS INTERCHANGE NODE
IST1189I  OSIMGMT  = YES
IST314I  END
```

Figure 107. Example of D NET,VTAMOPTS,OPTION=OSIMGMT Display Command

If it is NO and you do have the CMIP DD names present in the VTAM procedure, you can dynamically change the value to YES by issuing the following command: `F NET,VTAMOPTS,OSIMGMT=YES.`

```
RAKAN    15:03:26   F NET,VTAMOPTS,OSIMGMT=YES
RAKAN    15:03:26   IST097I  MODIFY   ACCEPTED
RAKAN    15:03:26   IST223I  MODIFY          COMMAND COMPLETED
RAKAN    15:03:35   IST1443I  ACYDDF LOADED = NO ACCESS AUTHORITY CHECKING
RAKAN    15:03:35   CNM493I FLBAUT   : (NO SEQ) : FLBCMIPA FLBTOPO 01:00
RAKAN    15:03:35   CNM493I FLBAUT   : (NO SEQ) : FLBCMIPA FLBGMMGR 01:00
RAKAN    15:03:35   IST1293I  CMIP SERVICES IS ACTIVE
```

Figure 108. Example of F NET,VTAMOPTS,OSIMGMT=YES Modify Command

If you are going to use more than one Accounting Manager application residing on different VTAM hosts, this exercise should be repeated on all of those systems.

Now that we know CMIP services are installed under VTAM, we can proceed with the customization of NetView to be able to run the APPN Accounting Manager Application.

10.5.2 NetView Customization for APPN Accounting

We recommend that you also read the *SNA Topology Manager and APPN Accounting Manager Implementation Guide*, SC31-8060.

NetView will make use of the Subsystem Interface (SSI) address space to communicate with the VTAM address space, therefore you must have the NetView SSI installed in your system. If you do not have the SSI installed we must refer you to the manual *NetView for MVS Installation and Administration Guide*.

During the installation of the APPN Accounting Manager you have several options to automate the startup of the Accounting Manager. One would be to automate the accounting to such an extent that it would shut down automatically during peak usage. If you are collecting accounting data in a controlled environment for processing that data as input to NetDA/2, you do not need it to be a continuous process as you would when collecting data for accounting purposes.

10.5.2.1 VTAM Definition for APPN Accounting Manager

VTAM will need an APPL definition for the APPN Accounting Manager application. Samples are included in the A01APPLS (CNMS0013) samples shipped with NetView.

```
*********************************************
* NETVIEW APPN ACCOUNTING MANAGER           *
*********************************************
ACCTMGR  APPL PRTCT=CNM01
```

Figure 109. Extract from the NetView Samples ACB Names (CNMS0013)

Make sure that you update PRTCT=CNM01 to your NetView domain ID before activating the Accounting Manager.

10.5.2.2 Update DSIOPF in NetView

Go to member DSIOPF in your DSIPARM data set and uncomment the DSIOPFT statement. Remove the asterisk so that the percentage sign starts in column one (see the example below).

```
***********************************************************************
* INCLUDE ADDITIONAL NETVIEW OPERATOR DEFINITIONS FOR A NETVIEW       *
* ENTERPRISE SYSTEM.  UNCOMMENT THESE LINES IF YOU ARE INSTALLING A   *
* NETVIEW ENTERPRISE SYSTEM.                                          *
***********************************************************************
*
%INCLUDE DSIOPFR
%INCLUDE DSIOPFT
%INCLUDE DSIOPFW
```

Figure 110. Example of DSIOPF Statements Uncommented

10.5.2.3 Update Member DSIOPFT in DSIPARM

Uncomment the FLBGMMGR and the profile name statement in DSIOPFT. This is the user ID for the Accounting Manager autotask.

```
***********************************************************************
* THE FOLLOWING STATEMENT IS TO DEFINE AN AUTOTASK OPERATOR FOR       *
* APPN ACCOUNTING MANAGER.                                            *
***********************************************************************
FLBGMMGR     OPERATOR    PASSWORD=FLBGMMGR
             PROFILEN    FLBGMMPR
```

Figure 111. Example of DSIOPFT Statements Uncommented

10.5.2.4 Update the NetView Automation Table

Go to your Automation Table member in DSIPARM and include FLBAUT (see the example).

```
***********************************************************************
*  SNA TOPOLOGY AND APPN ACCOUNTING MANAGER ENTRIES                   *
***********************************************************************
*
%INCLUDE FLBAUT
*
```

Figure 112. Example of FLBAUT Statement Uncommented

10.5.2.5 Update the VTAM MIBConnect Parameters

In order for the Accounting Manager to have authorization to access VTAM CMIP services, the VTAM MIBconnect parameters in member FLBSYSDA must match the VTAM APPL definition. The Accounting Manager will read these statements during initialization.

The application name for the Accounting Manager must match the name specified in the member FLBSYSDA in the NetView DSIPARM data set. If this member is not in a user-defined data set, we suggest that you copy it there from the default NetView DSIPARM data set. Make sure that this user-defined data set is before the default NetView DSIPARM data set concatenation in the NetView procedure.

Once you edit member FLBSYSDA you will find what is shown in Figure 113.

```
VTAM:
  APPLNAME=″ACCTMGR″
  APPLPASS=″CNM01″
```

Figure 113. Extract from the NetView FLBSYSDA DSIPARM Member

Ensure that the APPLNAME value is the same as in your VTAM APPL definition. Also update APPLPASS to reflect your NetView domain ID in the DSIDMN member in DSIPARM.

10.6 Installing the APPN Accounting Agent

First we install the APPN accounting agent on the OS/2 workstation. If your workstation is running PCOMM Version 4.1, you need to apply the latest fixes otherwise the download process will not work. In addition to that you must verify that the correct translation will take place when downloading. The default translation table supplied with PCOMM does not download the “or” symbol (|) correctly, and this must be changed. After you have upgraded your PCOMM 4.1 on your workstation to at least CSD2 you can follow these steps to ensure the correct translation will take place.

10.6.1 Changing the Default Translation Table for PCOMM

This process ensures that the “or” symbol is translated from EBCDIC X′4F′ to ASCII X′7C′. We illustrate the modification of the translation table on a US English version of PCOMM.

1. Go to the PCOMM session where you will run the downloads from and click on **Transfer.**
2. From there click on **Setup.**
3. Click on **Translation Table.**

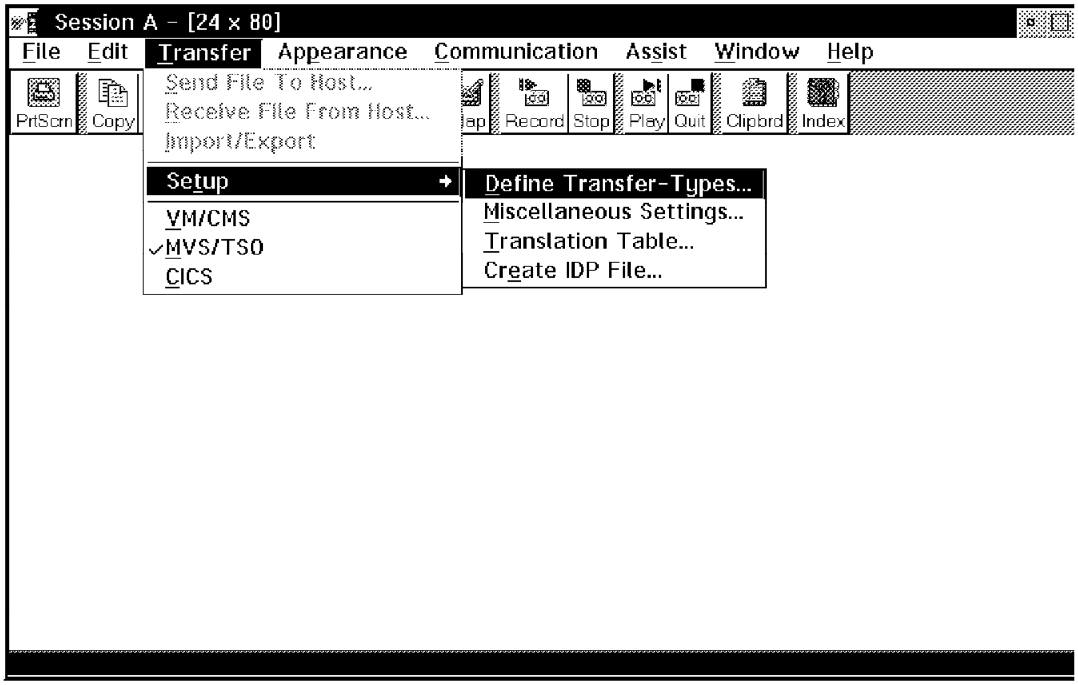


Figure 114. Transfer Menu

4. In the Download Translation Table Box:
   - Click on **User Defined**.

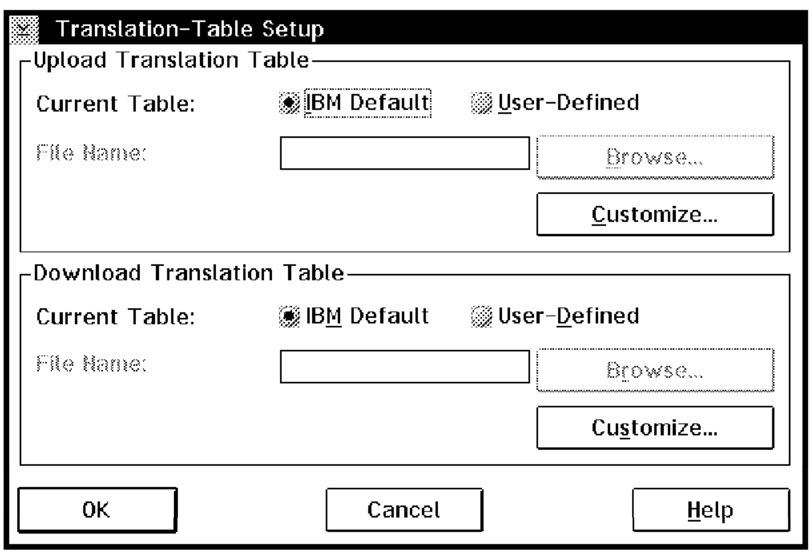
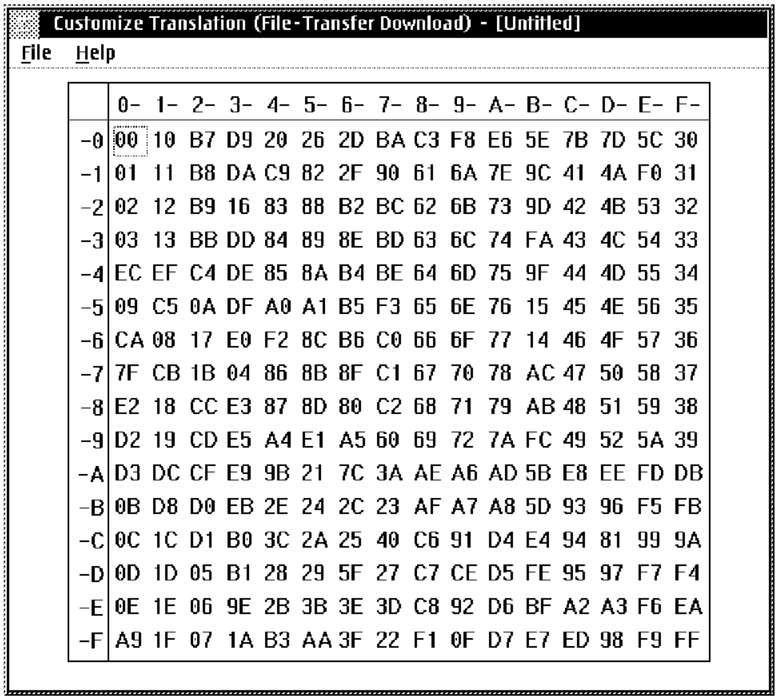


Figure 115. Translation Table Setup Menu

   - Click on **Customize**.



	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	00	10	B7	D9	20	26	2D	BA	C3	F8	E6	5E	7B	7D	5C	30
-1	01	11	B8	DA	C9	82	2F	90	61	6A	7E	9C	41	4A	F0	31
-2	02	12	B9	16	83	88	B2	BC	62	6B	73	9D	42	4B	53	32
-3	03	13	BB	DD	84	89	8E	BD	63	6C	74	FA	43	4C	54	33
-4	EC	EF	C4	DE	85	8A	B4	BE	64	6D	75	9F	44	4D	55	34
-5	09	C5	0A	DF	A0	A1	B5	F3	65	6E	76	15	45	4E	56	35
-6	CA	08	17	E0	F2	8C	B6	C0	66	6F	77	14	46	4F	57	36
-7	7F	CB	1B	04	86	8B	8F	C1	67	70	78	AC	47	50	58	37
-8	E2	18	CC	E3	87	8D	80	C2	68	71	79	AB	48	51	59	38
-9	D2	19	CD	E5	A4	E1	A5	60	69	72	7A	FC	49	52	5A	39
-A	D3	DC	CF	E9	9B	21	7C	3A	AE	A6	AD	5B	E8	EE	FD	DB
-B	0B	D8	D0	EB	2E	24	2C	23	AF	A7	A8	5D	93	96	F5	FB
-C	0C	1C	D1	B0	3C	2A	25	40	C6	91	D4	E4	94	81	99	9A
-D	0D	1D	05	B1	28	29	5F	27	C7	CE	D5	FE	95	97	F7	F4
-E	0E	1E	06	9E	2B	3B	3E	3D	C8	92	D6	BF	A2	A3	F6	EA
-F	A9	1F	07	1A	B3	AA	3F	22	F1	0F	D7	E7	ED	98	F9	FF

Figure 116. Translation Table

5. You will see a layout of the Translation Table. We need to alter the contents of entry 4F to read 7C. Go along the X-axis and find 4. Now go down along the entries in column 4 and find row F on the Y-axis. You will find it is the very last row and the fourth column from the left. In our case the value was B3.

6. Move the mouse pointer there and double-click.

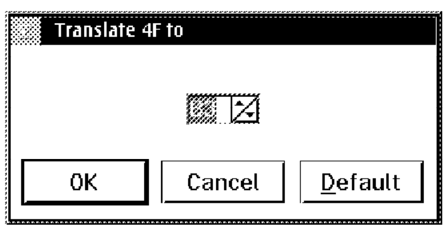


Figure 117. Translate Individual Code

7. In the Translate 4F window change the value to 7C.

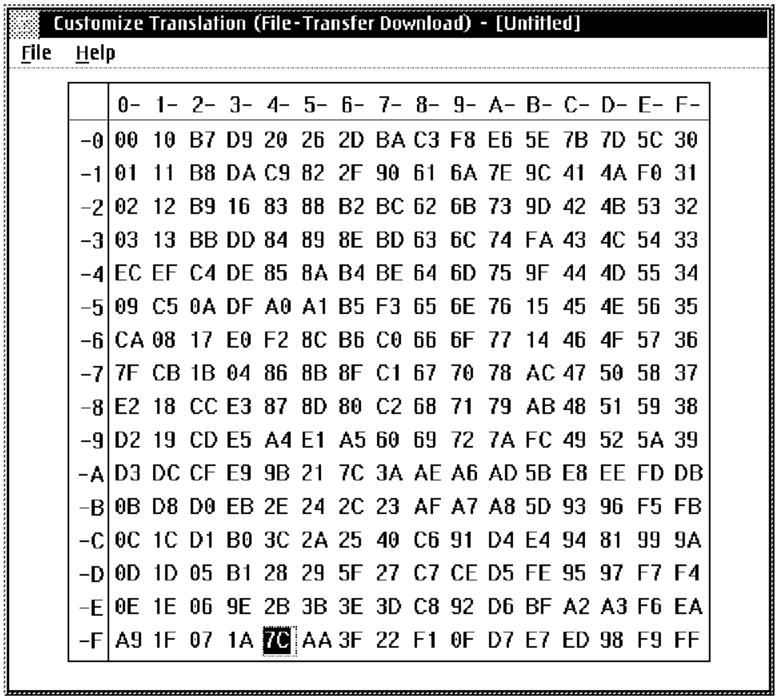
8. Click on **OK**.

9. Click on **File**.

10. Click on **Save**.

11. Enter a name in the File Name box and click on **OK**.

12. You will be presented with the changed translation table with your change highlighted.



	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0	00	10	B7	D9	20	26	2D	BA	C3	F8	E6	5E	7B	7D	5C	30
-1	01	11	B8	DA	C9	82	2F	90	61	6A	7E	9C	41	4A	F0	31
-2	02	12	B9	16	83	88	B2	BC	62	6B	73	9D	42	4B	53	32
-3	03	13	BB	DD	84	89	8E	BD	63	6C	74	FA	43	4C	54	33
-4	EC	EF	C4	DE	85	8A	B4	BE	64	6D	75	9F	44	4D	55	34
-5	09	C5	0A	DF	A0	A1	B5	F3	65	6E	76	15	45	4E	56	35
-6	CA	08	17	E0	F2	8C	B6	C0	66	6F	77	14	46	4F	57	36
-7	7F	CB	1B	04	86	8B	8F	C1	67	70	78	AC	47	50	58	37
-8	E2	18	CC	E3	87	8D	80	C2	68	71	79	AB	48	51	59	38
-9	D2	19	CD	E5	A4	E1	A5	60	69	72	7A	FC	49	52	5A	39
-A	D3	DC	CF	E9	9B	21	7C	3A	AE	A6	AD	5B	E8	EE	FD	DB
-B	0B	D8	D0	EB	2E	24	2C	23	AF	A7	A8	5D	93	96	F5	FB
-C	0C	1C	D1	B0	3C	2A	25	40	C6	91	D4	E4	94	81	99	9A
-D	0D	1D	05	B1	28	29	5F	27	C7	CE	D5	FE	95	97	F7	F4
-E	0E	1E	06	9E	2B	3B	3E	3D	C8	92	D6	BF	A2	A3	F6	EA
-F	A9	1F	07	1A	7C	AA	3F	22	F1	0F	D7	E7	ED	98	F9	FF

Figure 118. Changed Translation Table

13. Click on **File**.
14. Click on **Exit**.
15. Click on **OK**.

That will ensure that the downloaded files are translated correctly. You are now ready to start the download of the software.

10.6.2 Download the Files from the Host

You need to establish in your installation what the NetView file with the low-level qualifier of SEGVPS21 is called. We use our example where it was named TME10.V1R1.SEGVPS21.

The APPN Topology and Accounting Agent software is installed onto the workstation using the Software Installer for OS/2. You need to download and install the Software Installer for OS/2 before you can install the APPN Topology and Accounting Agent.

10.6.2.1 Installing the APPN Topology and Accounting Agent

First sign on to your TSO session. We recommend you perform the download from the raw TSO command prompt rather than option six (command prompt) of ISPF/PDF. We have experienced difficulties downloading large files using the ISPF/PDF prompt.

You will need to download the bootstrap install file called BNTHGETE. To do this make the drive onto which you want to install the Software Installer for OS/2 the current drive on the workstation. For example, if you want to install the Software Installer for OS/2 onto drive D, at the OS/2 command prompt enter:

```
D:
```

Download the BNTHGETE.CMD command file by entering the following command at the OS/2 command prompt:

```
receive d:bnthgete.cmd a:'tme10.v1r1.segvps21(bnthgete)' ascii crlf
```

The above command will download the member BNTHGETE from the TME10.V1R1.SEGVPS21 data set on host session A and translate it to ASCII. If your host TSO session is not A then you must modify the command to read b:'tme10... or whatever.

If you are using the NetView V3R1 manuals, you will notice that there the command listed for the download is incorrect. The manual shows the following command:

```
RECEIVE D:BNTHGETE.CMD A:'NETVIEW.V3R1MO.SEGVPS21(DUIHGETE)' ASCII CRLF
```

This is not correct so change your command to read as follows:

```
RECEIVE D:BNTHGETE.CMD A:'NETVIEW.V3R1MO.SEGVPS21(BNTHGETE)' ASCII CRLF
```

Wait for the message FILE TRANSFER COMPLETE to appear. This tells you that you have the Bootstrap file installed on your OS/2 system.

10.6.2.2 Run BNTHGETE to Download the Software Installer Files

Change to an active OS/2 window and change to the drive to where you have downloaded the bootstrap file. In our case it was D:.

Start the process with the following steps:

1. At the OS/2 command line, enter:

   ```
   BNTHGETE D: TME10.V1R1.SEGVPS21 A:
   ```

   This command will invoke the bootstrap process, telling it that the default drive is D:, the host data set name is TME10.V1R1.SEGVPS21 and the host session ID is A.

   The result will be a series of file downloads that will take place. Wait for the following message to appear:

   FILE TRANSFER COMPLETE.
   DUIJ043I: Would you like to continue with the installation? (Yes/No)

2. Reply Yes to this message.

   At this stage you have downloaded the software installer for OS/2.

   You will now get a window named Open Host Catalog.

3. Click on **Open**.

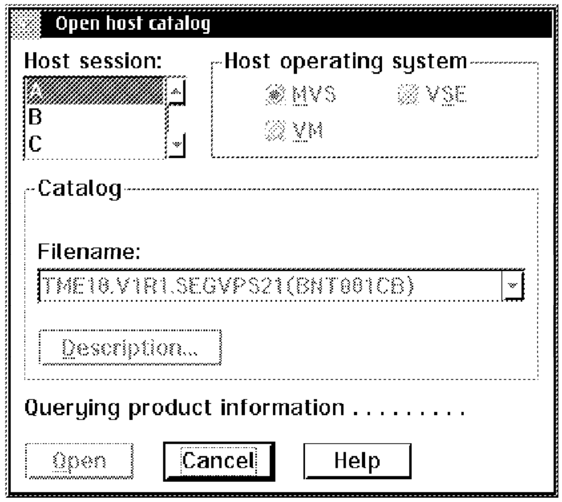


Figure 119. Open Host Catalog Window

4. Next you will get the Installation and maintenance window. Double-click on **APPN Topology and Accounting Agent**.

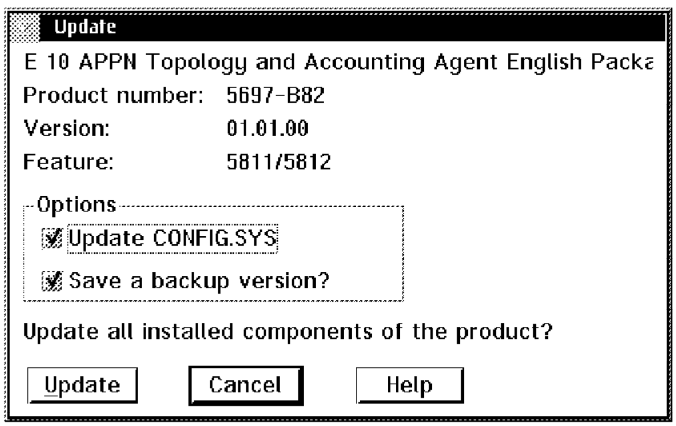


Figure 120. Update Menu

Click on the **Update CONFIG.SYS** button if you want the configuration file to be updated. You should normally allow this to happen.

5. Click on **Update**.

The installation will complete and you will get the last window confirming the installation.

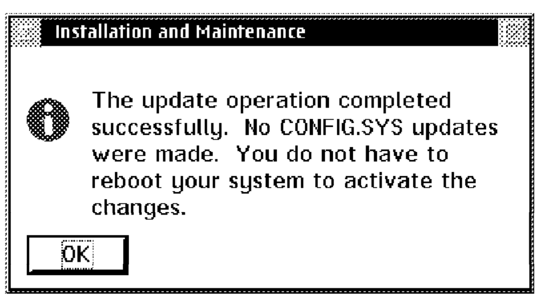


Figure 121. Completion Window

6. Click on **OK**.
7. Reboot your OS/2 system. Even though the completion window says reboot is not necessary, the manual advises you to reboot and so we chose the safe option.

10.7 Starting the APPN Accounting Agent

Once the system is up and running again after the reboot, you are ready to start the APPN accounting agent. It is important to note the the agent must be started before data collection can be started from the manager.

1. Go to a OS/2 Command prompt and change to the drive where the agent is installed.
2. Once there change to the IBMBNT directory.
3. Type APPNMGMT and press Enter.

   This will present the following window.

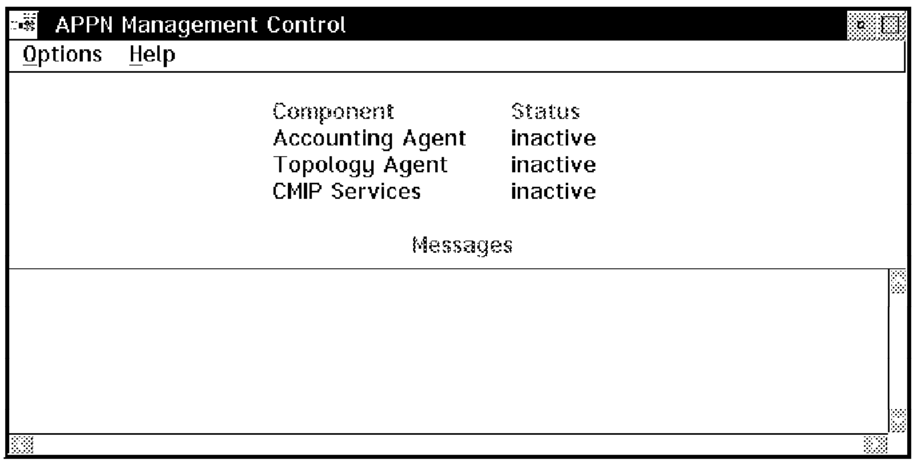


Figure 122. APPN Management Control Window

4. Click on **Options** and then **Start**.

   You will get the Start APPN Management Programs window.

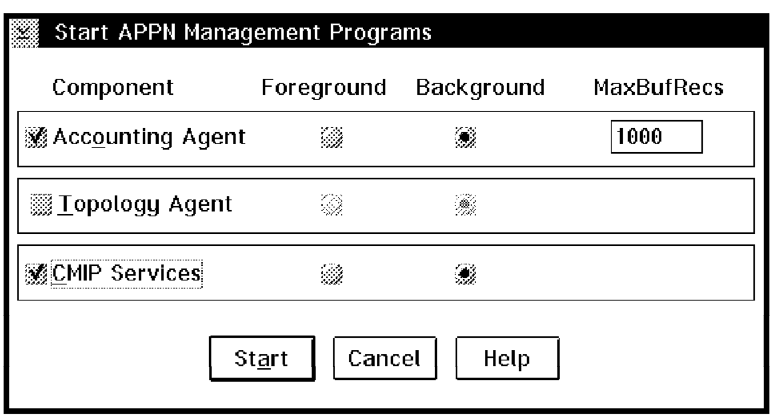


Figure 123. Start APPN Management Programs

5. Click on the **Accounting Agent** check box.

6. Click on the **CMIP Services** check box.
7. Click **Start**.

   Once started you will see the progress of the start command in the messages box on the APPN Management Control window. Make sure that the agent initialized successfully. The status under the Status heading must read ACTIVE before proceeding (see the example below).

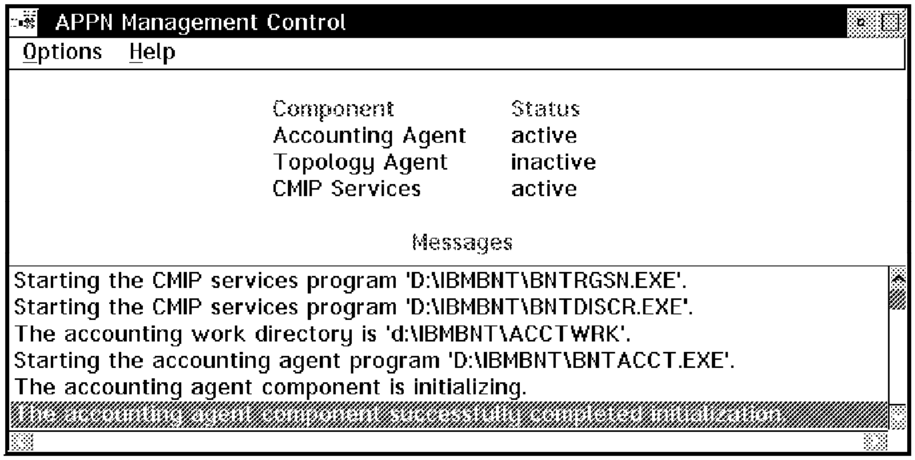


Figure 124. APPN Management Control Window - Status Active

10.8 Starting the APPN Accounting Manager

You are now ready to start the NetView APPN Accounting Manager. You can initiate the process by starting the Autotask FLBGMMGR. Please note that when you start this autotask VTAM CMIP services must be active. If you are going to automate the startup, set your automation to wait for CMIP to initialize and then issue the following command:

```
AUTOTASK OPID=FLBGMMGR
```

Below is an example of the successful startup of the autotask.

```
AUTOTASK OPID=FLBGMMGR
CNM570I STARTING AUTOMATION TASK FLBGMMGR
DSI020I OPERATOR FLBGMMGR LOGGED ON FROM TERMINAL NONE USING PROFILE (FLB
CNM493I FLBAUT   : (NO SEQ) : PURGE TIMER=FLBGMMGR
FLB230I ACCOUNTING MANAGER IS INITIALIZING
DSI205I 000 TIMER ELEMENTS PURGED  OP = 'AUTO1   '
FLB202I ACCOUNTING MANAGER INITIALIZATION IS COMPLETE
```

Figure 125. Example of AUTOTASK OPID=FLBGMMGR Command

10.9 Starting APPN Data Collection on the Manager Application

We have set up a very simple configuration in our environment.

Firstly we have a VTAM CNN named USIBMRA.RAK and a VTAM NN named USIBMRA.RAP. Secondly we have an OS/2 LAN workstation running CS/2 configured as an NN, named USIBMRA.ANTON. Lastly we have an OS/2 LAN workstation running CS/2 configured as an EN. The EN′s name is USIBMRA.ACCTENB. We have applications on the two OS/2 workstations communicating with each other. This will enable us to collect session and conversation data on both the nodes. The EN is also communicating with host applications using a route via USIBMRA.ANTON. This allows us to capture some intermediate data.

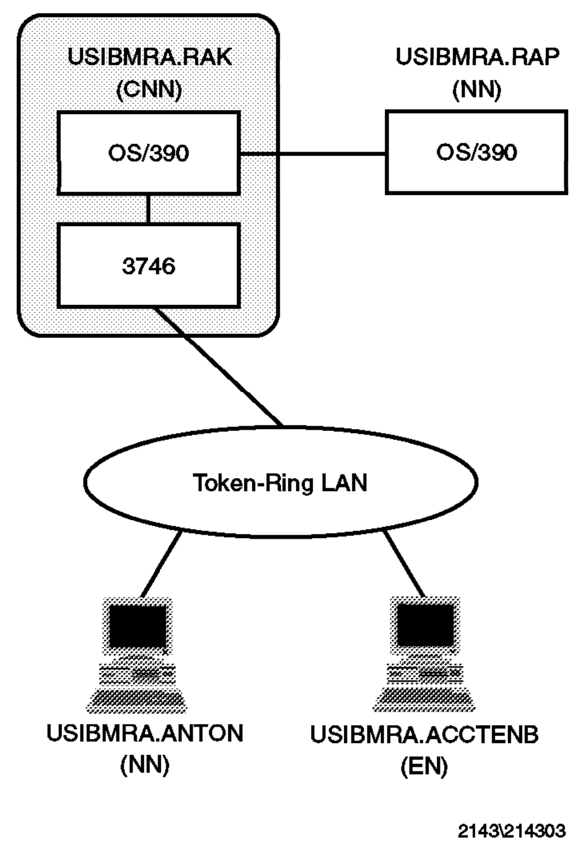


Figure 126. Sample Network Configuration

Just to summarize, we have completed the following enumerated checklist to arrive at the stage where we can start data collection:

1. Enable CMIP on VTAM.
2. Ensure OSIMGMT=YES.
3. Creat VTAM APPL definition for Accounting Manager.
4. Uncomment the %INCLUDE DSIOPFT statement in member DSIOPF.
5. Uncomment the FLBGMMGR autotask ID in member DSIOPFT.
6. Uncomment %INCLUDE FLBAUT in the Automation Table.

7. Update the VTAM MIBConnect parameters in member FLBSYSDA.
8. Install the agent.
9. Start the agent applications.
10. Start the manager application.

Now we are ready to start collection on the agent nodes. We are going to look at a simple example where we start collection on different accounting data types.

You communicate with the APPN Accounting Manager by means of the ACCTSNA command on NetView. The combination of accounting data type and agent node is known by the Accounting Manager as an accounting management control object (AMCO).

There is a comprehensive online help facility on NetView. Simply enter HELP ACCTSNA and a help panel will be displayed. From here onwards the help facility is panel driven and easy to use.

In our case we start the process, let it run for a while, and then retrieve the records from the agent nodes. This is what we do:

1. Start session data collection for node ANTON on the manager application that resides on USIBMRA.RAP:

   ACCTSNA START,NODE=USIBMRA.ANTON,ACCTDATA=SESSION

```
* RAPAN    ACCTSNA START,NODE=USIBMRA.ANTON,ACCTDATA=SESSION
- RAPAN    FLB229I ACCTSNA START COMMAND FOR SESSION DATA AND NODE
           USIBMRA.ANTON COMPLETED SUCCESSFULLY
```

Figure 127. ACCTSNA START NODE=USIBMRA.ANTON,ACCTDATA=SESSION Command

2. Start conversation data collection for node ANTON:

   ACCTSNA START,NODE=USIBMRA.ANTON,ACCTDATA=CONVERS

```
* RAPAN    ACCTSNA START,NODE=USIBMRA.ANTON,ACCTDATA=CONVERS
- RAPAN    FLB229I ACCTSNA START COMMAND FOR CONVERS DATA AND NODE
           USIBMRA.ANTON COMPLETED SUCCESSFULLY
```

Figure 128. ACCTSNA START NODE=USIBMRA.ANTON,ACCTDATA=CONVERS Command

3. Start intermediate data collection for node ANTON:

   ACCTSNA START,NODE=USIBMRA.ANTON,ACCTDATA=INTSESS

```
* RAPAN    ACCTSNA START,NODE=USIBMRA.ANTON,ACCTDATA=INTSESS
- RAPAN    FLB229I ACCTSNA START COMMAND FOR INTSESS DATA AND NODE
           USIBMRA.ANTON COMPLETED SUCCESSFULLY
```

Figure 129. ACCTSNA START NODE=USIBMRA.ANTON,ACCTDATA=INTSESS Command

4. Start session data at node ACCTENB:

   ACCTSNA START,NODE=USIBMRA.ACCTENB,ACCTDATA=SESSION

```
* RAPAN    ACCTSNA START,NODE=USIBMRA.ACCTENB,ACCTDATA=SESSION
- RAPAN    FLB229I ACCTSNA START COMMAND FOR SESSION DATA AND NODE
           USIBMRA.ACCTENB COMPLETED SUCCESSFULLY
```

Figure 130. ACCTSNA START NODE=USIBMRA.ACCTENB,ACCTDATA=SESSION Command

5. Start conversation data collection at node ACCTENB:

   ACCTSNA START,NODE=USIBMRA.ACCTENB,ACCTDATA=CONVERS

```
* RAPAN    ACCTSNA START,NODE=USIBMRA.ACCTENB,ACCTDATA=CONVERS
- RAPAN    FLB229I ACCTSNA START COMMAND FOR SESSION DATA AND NODE
           USIBMRA.ACCTENB COMPLETED SUCCESSFULLY
```

Figure 131. ACCTSNA START NODE=USIBMRA.ACCTENB,ACCTDATA=CONVERS Command

6. Node ACCTENB being an end node, it is not possible to start intermediate data collection on this node.

10.10 Managing the Data Collection Process

There are two aspects to consider in managing the APPN accounting data collection process:

- Firstly, the management process from the manager application′s point of view.
- Secondly, the management process from the agent′s point of view.

We look at both functions. Assume the NetView operator has issued commands to start collecting session and conversation data at the agent. The agent creates separate accounting management control objects (AMCOs) to manage the collection session endpoint data, intermediate session data, and conversation data. The AMCOs contain all the instructions for managing the data collection such as under what conditions to notify the manager that data is ready be retrieved.

10.10.1 Managing APPN Accounting Data Collection from the Manager

The following functions are available from the NetView manager application. We would like to draw your attention to the online help facility on NetView for the command syntax. It is a very handy place to have the help facility and is an excellent source of reference.

If your installation has installed the help facility, all you need to do is enter HELP ACCTSNA on the NetView command line otherwise you can refer to the NetView for MVS Command reference.

Following is a listing of the NetView ACCTSNA functions:

1. Display command

   Displays the settings of the current status for an AMCO at a specified agent node. If started, you can display status for:

   - Session Data

- Intermediate Session Data
- Conversation Data

2. Listnode command

   This will display a list of nodes that the accounting manager has started and the kinds of accounting data being collected at each agent node.

3. Modify command

   Can be used to temporarily change the parameter values of an AMCO while data collection is active. The changes will be lost when collection is stopped and started again.

4. Querydef command

   To query the default settings for the accounting manager.

5. Retrieve command

   Issued when you need to transfer the collected data from the agent to the manager.

6. Setdefs command

   This will control the default values for the manager application. Changing values by means of setdefs will not alter the values for current active data.

7. Start command

   Start the collection of data.

8. Stop command

   Stop the collection of data.

9. Stopmgr command

   Will cancel all outstanding commands for the accounting manager and stop the accounting manager itself.

10. Trace command

    Will start a trace that is written to the GTF data set for problem determination.

10.10.2 Managing APPN Accounting Agent

Once the agent application is started you have some options available to you from Options on the menu bar. Once you have clicked on Options you are able to do the following:

1. Start option

   With this you can start one or more components of the agent program. You can also start the components when issuing the start command from the OS/2 prompt.

2. Stop option

   You can stop the components started.

3. Trace option

   A trace facility that will normally be used in problem determination.

4. Reset accounting option.

   Will clear and reset the accounting files on the agent node.

5. Clear messages

   Will clear the messages in the messages box.

10.11 Preparing the SMF Records for Processing

Once you have completed your collection process you can extract your SMF records for processing.

In established environments you will have a batch procedure in place to extract your SMF records. If you do not have it in place, you can follow these guidelines.

Issue the D SMF command on the system console.

Check what data sets are being used for SMF recording. In our case it is SYS1.MAN1 and SYS1.MAN2.

You can use the JCL image below as an example. The DUMPIN DD name points to the SMF files. OUTDD is the output file. Do not code DCB parameters, the program IFASMFDP will do that. Update your DUMPIN DD names with your SMF data set names.

Ensure that you allocate enough space to hold your records on the OUTDD statement. In the SYSIN statements we tell the program to dump the records. We select record type 39 with subtype 9 and 10.

```
//SMFDMPJ JOB (0-111111),'SMFDMP',MSGCLASS=O,MSGLEVEL=(1,1),
//         CLASS=A
//STEP1  EXEC  PGM=IFASMFDP,REGION=3000K
//DUMPIN DD DSN=SYS1.MAN1,DISP=SHR,UNIT=SYSDA,VOL=SER=DASD1
//        DD DSN=SYS2.MAN1,DISP=SHR,UNIT=SYSDA,VOL=SER=DASD1
//OUTDD  DD DSN=SMF.DUMP,DISP=(NEW,CATLG,DELETE),
//         SPACE=(CYL,(10,1))
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
     INDD(DUMPIN,OPTIONS(DUMP))
     OUTDD(OUTDD,TYPE(39(9,10)))
/*
```

Figure 132. Sample JCL to Dump SMF Type 39 Records

Once the above job has completed you are ready to process the records.

10.12 Large Networks and Live Environments

We would like to mention some aspects to consider when you are going to collect APPN accounting data in a live environment and also when collecting data in a large network.

10.12.1 APPN Data and NetDA/2

Generally you would like to capture the traffic data for NetDA/2 under normal circumstances. You will avoid collecting data when there are networking problems and exceptionally high or low link and node utilization.

Capturing this data will result in NetDA/2 getting a false impression regarding the network utilization.

10.12.2 Network Utilization

The collection process itself can have an adverse affect on your networking performance. When you are collecting data from several resources and the process is controlled by automation or CLISTs, the possibility stands that you may issue retrieve commands simultaneously.

This can potentially overload your networking resources and add additional and unnecessary overheads.

When you are automating the collection of data we recommend that you make use of a phased approach when collecting the data from agent nodes as opposed to starting the process for all nodes at once.

An easy way of implementing a phased approach is to make use of of the SCHEDULE keyword in the START command. Use different times of the day for different agents.

You must also consider the amount of data that you are going to collect and the effect that will have on your agent nodes as well as the size of your agent nodes. If you are collecting from a node that is handling low volumes of data, you may want to increase the buffer size and retrieve the data at less frequent intervals.

If your nodes have storage constraints, you might want to retrieve data at less frequent intervals to avoid data loss. If you are collecting data from an agent node and you are making use of the NOTIFY=(,bufferpercent) feature, allow enough buffer space to remain so that the collection process can continue without the remainder of the buffers being filled before completion of the collection process.

10.12.3 Duplicate Data

When you are processing your SMF records you will notice that there might be duplicate records. The reason for having duplicate records can be because of one of the following reasons:

1. Collecting session data at several agent nodes at intermediate and endpoints of the session.

   To resolve this, use the session PCID to determine unique sessions and remove duplicate records.

2. Collecting conversation data at multiple agent nodes where the conversations have agents at each end of the conversation. Each agent reports the conversations, resulting in duplicate records. Use the conversation correlator and PCID to sort duplicate conversation records.

3. Collecting billing data on conversations and sessions at the same node. Duplicate data might be collected on the underlying sessions used by the

conversation. To resolve this, use the conversation correlator and session PCID to determine whether you have duplicate data.

10.12.4 Processing the SMF Records

The accounting manager writes records to the SMF file each time it retrieves data from the accounting agents. You can use these standard record formats to set up filtering and analysis of the accounting data. Use the assembler macros provided in the MACLIB data set to process the records. These macros are named FLBACTHD, FLBCONVG, and FLBSESSG. The macros contain DSECT definitions for each of the sections of the external log record formats. These DSECT definitions provide the assembler language programmer with the mappings required to address the accounting fields within an external log record.

NetView currently has eight subtypes of the type-39 record. The accounting function creates two new subtypes of the type-39 record:

- Subtype X′0009′ for APPC session data (endpoint and intermediate sessions)
- Subtype X′000A′ for APPC conversation endpoint data

The APPC session records are divided into the following sections:

- Log record header section:

  This is the same format used for the type-39 record, except for the addition of two new subtypes.

- Session data descriptor section
- Product section:

  This is the same format used for the type-39 record, except for addition of two new subtypes.

- Data sections:
  1. Session information data section
  2. Session counters data section
  3. Session route data section

The APPC conversation records are divided into the following sections:

1. Log record header section:

   This is the same format used for the type-39 record, except for the addition of two new subtypes.

2. Conversation data descriptor section
3. Product section:

   This is the same format used for the type-39 record, except for the addition of two new subtypes.

4. Data sections:
   - Conversation information data section
   - Conversation counters data section

10.12.5 APPC Session Accounting Records

The session accounting records contain the following fields:

1. Agent node

   The network name and the node name reporting the data.

2. Class of service name

   The COS name used in this session.

3. Fully qualified procedure correlation ID (PCID)

   This is a unique value in the network that serves as the session identifier.

4. Mode name for the session

   The mode name works in conjunction with the COS entry to determine the networking services of the session. Examples of SNA-defined mode names are:

   a. #INTER

   b. #BATCH

   c. SNASVCMG

5. Number of FMD PIUs sent by the primary LU

   This is a count of the actual user data sent. This does not include the session control and SNA command flows between the partner sessions. This field contains a count from the time the session has started or the time the records were retrieved from the agent node.

6. Number of FMD PIUs sent by the secondary LU

   This is a count of records as sent by the secondary LU.

7. Number of non-FMD PIUs sent by the primary LU

   This is a count of networking control commands that flowed between the partner LUs. This is not user data.

8. Number of non-FMD PIUs sent by the secondary LU

   The same as the primary LU with the exception that this applies to the secondary LU.

9. Number of PIU bytes sent by the primary LU

   This is a count of all data sent. It includes the user data and networking services data counts.

10. Number of PIU bytes sent by the secondary LU

    The same as above but this is a count for the secondary LU.

11. Primary LU

    The SNA network ID and name of the primary LU for the session. The primary LU is the LU that initiated the session; it is the LU that sends the session BIND request.

12. Primary LU indicator

    An indication of whether the node reporting this accounting data is the primary or secondary LU for the session. The accounting data record written to the external log is meaningful only when the node is reporting session endpoint data, as indicated in the type of data field.

13. Reason for ending session

    When this field contains zeros it means the session ended normally. If it is not zero, then it will contain the VTAM sense code.

14. Route of session

    The route that this session followed through your network. The route is specified in segments, where each segment consist of the node and the TG of every hop in the network.

15. Secondary LU

    The network ID and LU name for the secondary LU.

16. Session accounting data status

    This field will indicate any one of the following situations:

    a. Complete

       The session has completed. All fields are meaningful.

    b. Still in Use

       All fields except the end time and reason are meaningful.

    c. Incomplete

       The session ended without notifying the agent. All fields except the end time and end reason are meaningful up to the last time the agent application collected the data while the session was still in use.

    d. Accounting data loss

       The agent node and status fields are meaningful, but the remaining fields are not.

    e. Possible data loss

       The agent node and status fields are meaningful but the remaining fields are not

17. Start date and time of session

    The date and time the session was started at the node reporting the accounting data. This is the local date and time of the reporting node. The date is expressed as the year and day of year. The time is given in hundredths of seconds, with midnight as the 0-origin.

18. Stop date and time of session

    The date and time the session was ended, whether normally or abnormally, at the node from which this accounting data was reported. This is the local date and time of the reporting node. The date and time are represented in the same form as the start time. The stop time might be the same as the start time (for very short sessions, depending on the resolution of the reporting node′s timer).

19. Transmission priority

    The transmission priority for the session. The value indicates one of four possible priorities:

    a. Low
    b. Medium
    c. High

d. Network

Network is the highest priority. The transmission priority determines the order in which the PIUs of multiple sessions using a given route segment are sent. Of course, this applies only when multiple PIUs are queued for transmission on that route segment.

20. Type of data

An indication of whether the agent node is reporting endpoint or intermediate session accounting data.

21. Type of session

An indication of the SNA LU session type. LU 6.2 is the only session type defined for the APPC session accounting data.

10.12.6 APPC Conversation Accounting Records

The conversation accounting records contain the following fields:

1. Agent node

   The network ID and name of the control point (CP) of the node reporting the accounting data.

2. Conversation accounting data status:

   a. Complete

      The conversation has completed. All fields are meaningful.

   b. Still in Use

      All fields except the end time and reason are meaningful.

   c. Incomplete

      The conversation ended without notifying the agent. All fields except the end time and end reason are meaningful up to the last time the agent application collected the data while the session was still in use.

   d. Accounting data loss

      The agent node and status fields are meaningful, but the remaining fields are not.

   e. Possible data loss

      The agent node and status fields are meaningful but the remaining fields are not

3. Conversation correlator

   The conversation correlator is the identifier of the conversation. It is unique among all conversations between the two LUs. It is created at the node containing the source LU for the conversation.

4. Conversation security user ID

   The user ID associated with the APPC transaction. An APPC transaction comprises one or more conversations, each having source and target LU. The user ID is normally supplied at the node containing the source LU initiating the transaction but it can be supplied at the node of another source LU for the transaction.

5. Fully qualified procedure correlation ID

The fully qualified PCID is the identifier of the session allocated to the conversation.

6. Logical unit of work ID

   The logical unit of work ID is created at the node containing the source LU initiating the transaction. It is used with synchronization point processing, as well as for accounting purposes.

7. Number of RU bytes sent by the source LU

   The number of RU bytes sent by the source LU for the conversation. The value indicates all bytes sent, from the beginning of the conversation to (and including) the time the accounting data is generated.

8. Reason for ending the conversation

   SNA sense data indicating the reason the conversation ended. If the conversation ended with normal or confirmed deallocation, the sense data is all binary zeros.

9. Source LU

   The SNA network ID and name of the source LU for the conversation.

10. Source LU indicator

    An indication of whether the node reporting this accounting data contains the source or target LU for the conversation.

11. Start date and time of conversation

    The date and time the conversation was started at the node reporting this accounting data.

12. Stop date and time of conversation

    The date and time the conversation was ended, whether normally or abnormally, at the node from which this accounting data was reported.

13. Synchronization level of conversation

    The synchronization level requested by the program allocating conversation. The synchronization levels are:

    a. NONE: The conversation supports no APPC conversation synchronization protocols.

    b. CONFIRM: The conversation supports APPC confirm/confirmed protocols.

    c. SYNCPT: The conversation supports APPC synchronization point protocols.

14. Target LU

    The target LU is the LU at the other end of the conversation from the source LU, that receives the conversation allocation request.

15. Target transaction program name

    The transaction program name specified by the program allocating the conversation. This is the name of the transaction program as known at the target node.

16. Type of conversation

    The conversation type requested by the program allocating the conversation. The conversation types are:

a. Half-duplex basic conversation

b. Half-duplex mapped conversation

c. Full-duplex basic conversation

d. Full-duplex mapped conversation

Please consult Appendix A of the *SNA Topology and APPN Accounting Implementation Guide* for a detailed description of each and every field in the record layout.

10.13 Placing the Accounting Agent

Depending on where in the network this agent node is located, it will have the ability to collect different types of APPC data.

Putting an agent on one of the endpoint nodes of a session path enables the collection of conversation data. If your accounting agent is an intermediate node on a session path, you will be able to collect intermediate session data provided that you are not using HPR. A point to remember is that accounting manager applications cannot collect accounting data; it can only be collected at agent nodes. There is also no VTAM accounting agent. It is possible for a single agent node to forward records to multiple manager applications.

Chapter 11. Creating NetDA/2 Traffic Definitions

Now that we have collected the session and conversation APPN records, we should focus on how to create the NetDA/2 traffic definitions from them.

In our case, the SMF records were sent from the MVS where NetView was running into a VM system. The procedure and the program we used should be independent of the type of operating system used.

In order to look at the content of the APPN accounting SMF records, we could have used the Service Level Reporter (SLR) or any other program of this type. As none of them were available, we coded a REXX program to list the SMF records.

You can download this program from the FTP site mentioned in “How to Access Samples from This Book” on page xi. Its name is APPNLIST.TXT.

This program was used under VM, but it should also run under MVS with some small modifications in the I/O instructions.

11.1 APPN Accounting Records Layout

As explained earlier, we have two types of records intermixed in the SMF file:

- Session records
- Conversation records

The layout of these records can be found in *SNA Topology and APPN Accounting Implementation Guide*, SC31-8239.

All accounting data is recorded in an external log according to the structure defined in the type-39 external log record.

NetView currently has eight subtypes of the type-39 record. These are documented in ″External Log Record Formats″ in the *TME 10 NetView for OS/390 Application Programming Guide*. The accounting function creates two new subtypes of the type-39 record:

- Subtype X′0009′ for APPC session data (endpoint and intermediate sessions)
- Subtype X′000A′ for APPC conversation endpoint data

Each record is divided into sections. The APPC session record sections are (in order):

1. Log record header section

   This is the same format as used for the type-39 record.

2. Session data descriptor section
3. Product section

   This is the same format as used for the type-39 record.

4. Data sections:

   - Session information data section
   - Session counters data section

- Session route data section

The APPC conversation record sections are (in order):

1. Log record header section

   This is the same format as used for the type-39 record.

2. Conversation data descriptor section
3. Product section

   This is the same format as used for the type-39 record.

4. Data sections:
   - Conversation information data section
   - Conversation counters data section

In the data descriptor section of both types of records we can find the offsets to the other sections.

The REXX program does not list all the fields available in the records; it only lists those fields that we considered of interest in defining the traffic to NetDA/2. Figure 133 on page 133 shows an example of the listing produced by the REXX program.

```
List of APPN records in file: 970905 SMFDATA X1

Session records
AGENT    PCID             TY ST PLU_NAME SLU_NAME MOD_NAME COS_NAME PTY DATES    TIMES  DATEP    TIMEP FMD_PIU_PS NO_FMD_PIU_PS BYTE_PS FMD_PIU_SP NO_FMDPIUSP BYTE_SP ROUT_SEG
ANTON    D1DBB3E001D1055A  E  2 ACCTENB  ANTON    #BATCH   #BATCH   00 0097248F 155810 00000000 000000   101852        2    202533249     4074        2       360868         1
ANTON    D1DBB3E001D1055A  E  2 ACCTENB  ANTON    #BATCH   #BATCH   00 0097248F 155810 00000000 000000   101852        2    202533249     4074        2       360868         2
ACCTENB  D1DBB3E001D1055A  E  2 ACCTENB  ANTON    #BATCH   #BATCH   00 0097248F 155816 00000000 000000   101852        2    202533249     4074        2       360868         3
ACCTENB  D1DBB3E001D1055A  E  2 ACCTENB  ANTON    #BATCH   #BATCH   00 0097248F 155816 00000000 000000   101852        2    202533249     4074        2       360868         4
ACCTENB  D1DBB3E001D1055A  E  1 ACCTENB  ANTON    #BATCH   #BATCH   00 0097248F 155816 0097248F 165621   101853        4    202533281     4074        4       360942         5

Conversation records
AGENT    PCID             ST SRC_LU   TGT_LU TYPE TP_NAME                                                    DATES    TIMES  DATEP    TIMEP       BYTE_ST      BYTE_TS
ANTON    D1DBB3E001D1055A  1 ACCTENB  ANTON     1 NETMTP                                                     0097248F 155810 0097248F 160814    201613268       321320  6
ACCTENB  D1DBB3E001D1055A  1 ACCTENB  ANTON     1 NETMTP                                                     0097248F 155816 0097248F 160820    201613268       321319  7
```

Figure 133. Accounting Records Listing

These records were captured using the configuration shown in Figure 134.

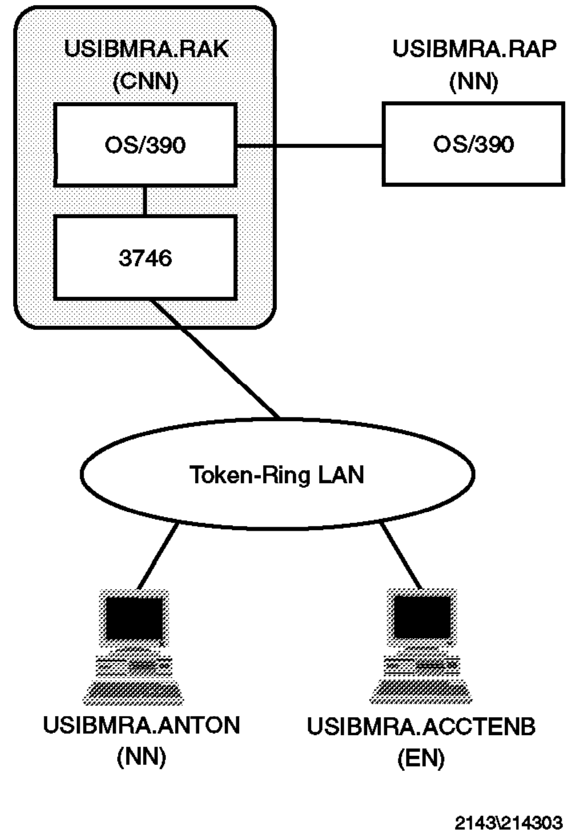


Figure 134. Sample Network Configuration

ANTON is an NN and ACCTENB is an EN. We run an application whose TP_NAME is NETMTP; it is an APPC application that transfers data between APPN nodes.

The program lists first the session accounting records and then the conversation records. The explanation of the different fields can be found in Table 1 on page 135 and Table 2 on page 136.

In Figure 133 on page 133 records **1** through **5** are session accounting records; the rest are conversation records. Records **1**, **2** and **6** were reported by the accounting agent in ANTON (NN); the rest by ACCTENB (EN).

As you can see from the records in Figure 133 on page 133, the information is not only duplicated because there are agents reporting it at both ends of the route, but also because we have retrieved information more than once from the agents using NetView. As the session between ANTON and ACCTENB does not finish when NETMTP ends, the agents will continue to report the same data every time NetView sends a command to retrieve records. We need to keep this in mind when exporting the traffic to NetDA/2.

The session records only show an end date and time when the status is complete (column ST equal to 1). Only record **5** had a stop time stamp. So we cannot count on these records to get the elapsed time during which we captured the traffic. We need this data in order to calculate the rate of messages per unit of time, as required by NetDA/2. We can get this information from the conversation records.

No routing information is present in the session records; the nodes are adjacent. We will see routing information in session records later on in this chapter.

It would appear that with the conversation records alone we should be able to collect the traffic information required by NetDA/2, but these records do not contain a counter for the PIUs involved. So we need both types of records.

We come back to this subject later on. Table 1 lists the fields of the session records listed by the REXX program. The second column refers to the name of the field in the control section that describes this particular records (see Appendix A in *SNA Topology Manager and APPN Accounting Manager Implementation Guide* SC31-8060).

Table 1. Session Records Listing

Heading in listing	**Field name**	**Description**
AGENT	LSSIAGTN	Agent CP name.
PCID	LSSIPCID	PCID session ID.
TY	LSSIDTYP	Type of data: endpoint or intermediate.
ST	LSSIDSTA	Data status: 1 Complete, 2 Still in use, 3 Acct data loss, 4 Possible data loss or 5 Incomplete.
PLU_NAME	LSSIPLNM	Primary LU name.
SLU_NAME	LSSISLNI	Secondary LU network ID.
MOD_NAME	LSSIMODN	Mode name.
COS_NAME	LSSICOSN	COS name.
PTY	LSSITPFF	Transmission priority field.
DATES	LSSISDAT	Session start date. The format is 0CYYDDDF where C is the century (0 is 19yy, 1 is 20yy) and F indicates positive sign.
TIMES	LSSISTIM	Session start time in hundredths of a second. The program converts it into hhmmss.
DATEP	LSSIEDAT	Session end date. The format is 0CYYDDDF where C is the century (0 is 19yy, 1 is 20yy) and F indicates positive sign.
TIMEP	LSSIETIM	Session end time in hundredths of a second. The program converts it into hhmmss.
FMD_PIU_PS	LSSCFMPS	FMD PIUs sent primary to secondary.
NO_FMD_PIU_PS	LSSCNFPS	Non-FMD PIUs sent primary to secondary.
BYTE_PS	LSSCPBPS	PIU bytes sent primary to secondary.
FMD_PIU_SP	LSSCFMSP	FMD PIUs sent secondary to primary.
NO_FMDPIUSP	LSSCNFSP	Non-FMD PIUs sent secondary to primary.
BYTE_SP	LSSCPBSP	PIU bytes sent secondary to primary.
ROUT_SEG	LSSRRTNM	Route segment CP name

Table 2 on page 136 list the fields of the conversation records listed by the REXX program.

Table 2. Conversation Records Listing

Heading in listing	**Field name**	**Description**
AGENT	LSCIAGTN	Agent CP name.
ST	LSCIDSTA	Data status: 1 Complete, 2 Still in use, 3 Acct data loss, 4 Possible data loss, 5 Incomplete.
PCID	LSCIPCID	PCID session ID.
SRC_LU	LSCISLNM	Source LU name.
TGT_LU	LSCITLNM	Target LU name.
TYPE	LSCICNVT	Conversation type: 1 Basic - half-duplex, 2 Mapped - half-duplex, 3 Basic - full-duplex, 4 Mapped - full-duplex.
TP_NAME	LSCITPNM	Target transaction program (TP) name.
DATES	LSCISDAT	Conversation start date: The format is 0CYYDDDF where C is the century (0 is 19yy, 1 is 20yy) and F indicates positive sign.
TIMES	LSCISTIM	Conversation start time in hundredths of a second. The program converts it into hhmmss.
DATEP	LSCIEDAT	Conversation end date. The format is 0CYYDDDF where C is the century (0 is 19yy, 1 is 20yy) and F indicates positive sign.
TIMEP	LSCIETIM	Conversation end time in hundredths of a second. The program converts it into hhmmss.
BYTE_ST	LSCCBSST	RU bytes sent source to target.
BYTE_TS	LSCCBSTS	RU bytes sent target to source.

11.2 Session Intermediate Records

In order to get intermediate session records, we established a session between ACCTENB and RAP (see Figure 134 on page 134). We used APING to create the session.

The command we entered in ACCTENB was:

```
aping usibmra.rap -i 100
```

ACCTENB had only one link defined, and it was to ANTON; for this reason, ANTON was acting as an intermediate node in the route between ACCTENB and RAP. The route is ACCTENB-ANTON-RAK-RAP.

Figure 135 on page 137 shows the records retrieved by NetView from agents ANTON and ACCTENB.

```
List of APPN records in file: RAPJF SMF X1

Session records
AGENT    PCID             TY ST PLU_NAME SLU_NAME MOD_NAME COS_NAME PTY DATES    TIMES  DATEP    TIMEP FMD_PIU_PS NO_FMD_PIU_PS BYTE_PS FMD_PIU_SP NO_FMDPIUSP BYTE_SP ROUT_SEG
ANTON    D1DBB3E00CFEF261  I  2 ACCTENB  RAP      #INTER   #INTER   02 0097275F 101225 00000000 000000       110            2        11691      117          1        11654 ANTON    RAK      RAP  1
ACCTENB  D1DBB3E00CFEF261  E  2 ACCTENB  RAP      #INTER   #INTER   02 0097275F 101219 00000000 000000       110            2        11691      117          1        11654 ANTON    RAK      RAP  2

Conversation records
AGENT    PCID             ST SRC_LU   TGT_LU TYPE TP_NAME                                                    DATES    TIMES  DATEP    TIMEP       BYTE_ST      BYTE_TS
ACCTENB  D1DBB3E00CFEF261  1 ACCTENB  RAP       3 APINGD                                                     0097275F 101219 0097275F 101243        10468        10407       3
```

Figure 135. Sample Accounting Records

Record **2** is an intermediate session record (see column TY, type, which is I).

Records **1**, **2** and **3** are reporting the same traffic. Both session records report the same byte count; the reason why **3** shows a different byte count is that the conversation records provide RU bytes, while the session records provide PIU bytes (including SNA headers).

Only the conversation accounting record shows a stop timestamp. The timestamps in the session records show the time when the link was established and the time it was taken down. This means that if you execute the APING command several times, it will produce a different conversation record for each execution, but the session records will continue to accumulate the counts.

For example, executing APING four times will produce the records shown in Figure 136 on page 139.

```
List of APPN records in file: RAPJF SMF X1

Session records
AGENT    PCID             TY ST PLU_NAME SLU_NAME MOD_NAME COS_NAME PTY DATES    TIMES  DATEP    TIMEP FMD_PIU_PS NO_FMD_PIU_PS BYTE_PS FMD_PIU_SP NO_FMDPIUSP BYTE_SP ROUT_SEG
ANTON    D1DBB3E00CFEF261  I  2 ACCTENB  RAP      #INTER   #INTER   02 0097275F 101225 00000000 000000       434          5        46101      464         1        46130 ANTON    RAK      RAP
ACCTENB  D1DBB3E00CFEF261  E  2 ACCTENB  RAP      #INTER   #INTER   02 0097275F 101219 00000000 000000       434          5        46101      464         1        46130 ANTON    RAK      RAP

Conversation records
AGENT    PCID             ST SRC_LU   TGT_LU TYPE TP_NAME                                                          DATES    TIMES  DATEP    TIMEP       BYTE_ST      BYTE_TS
ACCTENB  D1DBB3E00CFEF261  1 ACCTENB  RAP       3 APINGD                                                           0097275F 101219 0097275F 101243        10468        10407
ACCTENB  D1DBB3E00CFEF261  1 ACCTENB  RAP       3 APINGD                                                           0097275F 112408 0097275F 112430        10468        10407
ACCTENB  D1DBB3E00CFEF261  1 ACCTENB  RAP       3 APINGD                                                           0097275F 114228 0097275F 114251        10468        10407
ACCTENB  D1DBB3E00CFEF261  1 ACCTENB  RAP       3 APINGD                                                           0097275F 114256 0097275F 114319        10468        10407
```

Figure 136. Accounting Records for APING

All these transactions share the same PCID.

11.3 Creating the NetDA/2 Traffic Tables

There are different ways of defining the traffic to NetDA/2; we will show one of them.

There are two tables that have to be created using this method:

```
%TABLE MESSAGES

%MSGNAME+ MSGIN+ MSGOT QOS+++++
 M1I002T  1512   120   @_QOS1


%TABLE REQUIREMENTS

%SOURCE DEST MSGNAME+ NUM_SESS MRATEOT++++ MRATEIN++++
 3      14   M1I053I  5        0.00833333  0.00833333


%TABLE APPLICATION_REQUIREMENTS

%APPL_NODE COS_ID++ MSGNAME+
 3         M1I053I  M1I053I
```

Figure 137. NetDA/2 Traffic Tables

The content and use of these tables is explained in detail in the NetDA/2 User′s Guide and in the NetDA/2 redbook. We will briefly describe some of the columns in these tables.

- TABLE MESSAGES
  - MSGIN: is the number of bits (not bytes) sent to the application. This is the average PIU size. In NetDA/2 design, MSGIN should also include the SNA headers, while in simulation the headers are added by NetDA/2. Based on this, we should use the bytes reported in the session records for design, and the bytes reported in the conversation records for simulation.
  - MSGOT: same as MSGIN, just in the opposite direction.
  - QOS: quality of service (mode) for this application. Must match an entry under NAME in the QOS_DEFNS table.
- TABLE REQUIREMENTS
  - SOURCE: name of the node where the application resides
  - DEST: name of the node in session with the application
  - MRATEOT: message rate (per second) to the destination from the source
  - MRATEIN: same as above the other way
- TABLE APPLICATION_REQUIREMENTS
  - APPL_NODE: name of the node where the application resides
  - COS_ID: name for a unique COS for this application. Must match a name in DCOS_DEFS table.

There are many more columns in these tables than those explained above; see the NetDA/2 User′s Guide for details.

11.4 Sample

Let′s try to create the NetDA/2 traffic tables for the traffic captured in Figure 135 on page 137.

```
%TABLE MESSAGES

%MSGNAME+ MSGIN++++ MSGOT++++ QOS+++++
 APINGD   850       797       #INTER

%TABLE REQUIREMENTS

%SOURCE+ DEST MSGNAME+ MRATEOT++++ MRATEIN++++
 ACCTENB RAP  APINGD   4.58        4.88

%TABLE APPLICATION_REQUIREMENTS

%APPL_NODE COS_ID++ MSGNAME+
 RAP       #INTER   APINGD
```

Figure 138. NetDA/2 Traffic Tables for APING

Let′s explain how we get to the values entered in Figure 138.

- TABLE MESSAGES
  - MSGNAME: APINGD, we get it from TP_NAME (conversation record)
  - MSGIN: 850, from dividing 11691 (BYTE_PS) by 110 (FMD_PIU_PS), times 8 (bits). This value is good for NetDA/2 design, as it also includes the SNA headers, while for simulation, as the headers are added by NetDA/2, we should use BYTE_ST from the conversation records.
  - MSGOT: 797, from dividing 11654 (BYTE_SP) by 117 (FMD_PIU_SP), times 8 (bits). This value is good for NetDA/2 design, as it also includes the SNA headers, while for simulation, as the headers are added by NetDA/2, we should use BYTE_TS from the conversation records.
  - QOS: #INTER, from MOD_NAME
- TABLE REQUIREMENTS
  - SOURCE: ACCTENB, from SRC_LU or PLU_NAME
  - DEST: RAP, from TGT_LU or SLU_NAME
  - MRATEOT: 4.58, from dividing 110 (FMD_PIU_PS) by 24 seconds (12:12:43 - 10:12:19) from the conversation record.
  - MRATEIN: 4.88, from dividing 117 (FMD_PIU_SP) by 24 seconds (12:12:43 - 10:12:19), timestamps from the conversation record.
- TABLE APPLICATION_REQUIREMENTS
  - APPL_NODE: RAP, from TGT_LU or SLU_NAME
  - COS_ID: #INTER, from COS_NAME in the session record.

Of course we would also have to define the QOS_DEFNS and DCOS_DEFNS tables for #INTER, LOCATIONS table for ANTON and ACCTENB, etc.

11.5 Conclusions

We have briefly covered how to create NetDA/2 traffic tables from the APPN accounting records.

This process could be handled by a program, but great care should be taken to prevent some of the potential problems already mentioned such as data duplication, wrong PIU counts, wrong time intervals, etc.

It is unfortunate that the conversation records do not report on the PIU counts, since it would solve some of these problems.

Chapter 12. Simulation

NetDA/2 offers simulation at a fairly high level, to refine the performance information available from query reports after design and to permit comparison of topology options.

We take the network featured in Chapter 4, "Automated Design" on page 31 and add one parameter to the MESSAGE table, as shown in Figure 139.

```
%TABLE MESSAGES
%MSGNAME MSGIN MSGOT QOS+++++ MODEL++++
 MSG1    1024  1024  #INTER   FIXED_LEN
 MSG2     480  2400  #CONNECT FIXED_LEN
```

Figure 139. Simulation Report

There are three modelling options:

EXP Exponential arrival, geometric length

FIXED_LEN Exponential arrival, fixed length

ON_OFF Geometric transmitting and not transmitting periods

A number of parameters will be defaulted, such as the buffering capabilities of the nodes. All are described in Chapter 18 of the *NetDA/2 User's Guide*. After editing the NI to specify the fixed length model, we drop the NI on an ND and design as before. We then select from the ND pop-up menu **Reports**, and **Network Simulation**, request a simulate time of 600 seconds, specify some thresholds and run the simulation.

Figure 140 on page 144 and Figure 141 on page 146 contain extracts from the reports.

```
Report on Simulation Done on: 16:54:28, Friday, 12 September 1997, for Network Design Object: ZUID.NDO
====================================================================================================

Simulation Parameters were:
  Events Source: Automatic Generation.
  Simulated Time = 600 seconds; High Priority Threshold = 500 milliseconds;
  Medium Priority Threshold = 1000 milliseconds; Low Priority Threshold = 2000 milliseconds.
  ATM chunk size = 1000 cells.


Simulation Ended successfully.

                    Applications Information
                    ========================
                                  PIUs         Delay        Stndrd Dev.    Slow    PIUs         PIUs         Delay        Stndrd Dev.   Slow    PIUs
                      Message     Delivered    (seconds)    (seconds)      PIUs    Discarded    Delivered    (seconds)    (seconds)     PIUs    Discarded
Source     Dest.      Name        Src->Dest    Src->Dest    Src->Dest      S->D    Src->Dest    Dest->Src    Dest->Src    Dest->Src     D->S    Dest->Src
-------    -------    ---------  ------------ ------------ ------------ -------- ------------ ------------ ------------ ------------ -------- ------------
DURBAN     BEAUFORT   MSG1               2417      0.13673      0.03693        0            0         2344      0.13624      0.03839        0            0
DURBAN     BLOEMFON   MSG1               2446      0.16473      0.03881        0            0         2362      0.15520      0.04228        1            0
DURBAN     CALEDON    MSG1               2431      0.11268      0.02372        0            0         2349      0.11518      0.02340        0            0
DURBAN     CAPE_TOW   MSG1               2381      0.04883      0.01448        0            0         2456      0.05128      0.01432        0            0
DURBAN     CERES      MSG1               2431      0.11263      0.02568        0            0         2326      0.11409      0.02348        0            0
DURBAN     CRADOCK    MSG1               2451      0.13857      0.03757        0            0         2363      0.13953      0.03911        0            0
 . . .
JOHANNES   DURBAN     MSG2               2428      0.15284      0.23077       42            0         2430      0.01150      0.00727        0            0
JOHANNES   GRAHAMST   MSG2               2465      0.66625      0.34676      463            0         2385      0.05279      0.01119        0            0
JOHANNES   KIMBERLE   MSG2               2423      0.41579      0.12406        1            0         2341      0.06950      0.00940        0            0
JOHANNES   KING_WIL   MSG2               2399      0.39150      0.24816      111            0         2403      0.05216      0.01074        0            0
JOHANNES   KLERKSDO   MSG2               2375      0.19132      0.09638        0            0         2387      0.02947      0.00532        0            0
JOHANNES   LADYSMIT   MSG2               2492      0.53564      0.35340      344            0         2388      0.04091      0.00846        0            0
JOHANNES   MALMESBU   MSG2               2444      0.26929      0.12245        2            0         2367      0.04138      0.00941        0            0
 . . .
 . . .
                    Locations Information
                    =====================

                PIUs or       PIUs or                                         Standard
                Packets       Packets         Utilization      Avg Delay      Deviation        Q congest
Name            Switched      Discarded           (%)          (seconds)      (seconds)           (%)
-------         --------      ---------       -----------      ---------     -----------       ---------
BEAUFORT            4806              0           0.00080        0.00000         0.00000          0.0000
BLOEMFON            4860              0           0.00081        0.00000         0.00000          0.0000
CALEDON             4823              0           0.00080        0.00000         0.00000          0.0000
CAPE_TOW           29043              0           0.00484        0.00000         0.00000          0.0000
CERES               4801              0           0.00080        0.00000         0.00000          0.0000
CRADOCK             4854              0           0.00081        0.00000         0.00000          0.0000
```

Figure 140. Simulation Report

The traffic rate was four sessions at each location, each generating one message each way per second, giving an average of 2400 messages sent and 2400 received.

The Applications section of the output includes the delay in each direction and an indication of how many messages had delays in excess of the criteria specified, or defaulted, for the simulation run. The defaults are 100, 500, and 2000 milliseconds for high-, medium-, and low-priority traffic, respectively. As seen in Figure 141 on page 146, we overrode these values to be half of our design criteria, which were themselves defaulted to 1 second for high-priority traffic and 2 seconds for medium priority. This is not very sophisticated but the thresholds are applied where effectively independent traffic flows in each direction.

There is an option to save in a file the events that are generated. These may be used in later simulations, perhaps after making changes to the configuration, for the purpose of comparison.

TGs and Links Information

End1	End2	TG Num	Link Id	Type	PIUs Carried	PIUs Discarded	Packets Carried	Packets Discarded	Avg Queueing Delay (seconds)	Standard Deviation (seconds)	Queue congest (%)	Utili- -zation (%)
BEAUFORT	GRAFF_RE	1	@0	APPN FD	2370	0	------	------	0.00884	0.02045	0.000	21.943
GRAFF_RE	BEAUFORT	1	@0	APPN FD	2437	0	------	------	0.00387	0.01238	0.000	22.593
. . .												
CAPE_TOW	DE_AAR	1	@0	APPN FD	2465	0	------	------	0.05695	0.08100	0.000	52.053
DE_AAR	CAPE_TOW	1	@0	APPN FD	2385	0	------	------	0.00072	0.00381	0.000	10.939
CAPE_TOW	JOHANNES	1	@0	APPN FD	14323	0	------	------	0.00254	0.00616	0.000	29.809
JOHANNES	CAPE_TOW	1	@0	APPN FD	14720	0	------	------	0.02659	0.04066	0.000	67.204
. . .												
DE_AAR	GRAFF_RE	1	@0	APPN FD	2363	0	------	------	0.00068	0.00359	0.000	10.835
GRAFF_RE	DE_AAR	1	@0	APPN FD	2276	0	------	------	0.05177	0.08034	0.000	48.054
. . .												
DE_AAR	QUEENSTO	1	@0	APPN FD	4875	0	------	------	0.01986	0.03648	0.000	45.182
QUEENSTO	DE_AAR	1	@0	APPN FD	4900	0	------	------	0.02155	0.03415	0.000	45.384
. . .												
DURBAN	GRAFF_RE	1	@0	APPN FD	5476	0	------	------	0.02799	0.04112	0.000	50.699
GRAFF_RE	DURBAN	1	@0	APPN FD	5353	0	------	------	0.02270	0.03606	0.000	49.559
DURBAN	GRAFF_RE	2	@1	APPN FD	4319	0	------	------	0.01757	0.03026	0.000	40.035
GRAFF_RE	DURBAN	2	@1	APPN FD	4186	0	------	------	0.01459	0.02696	0.000	38.789
DURBAN	JOHANNES	1	@0	APPN FD	11558	0	------	------	0.00209	0.00549	0.000	22.788
JOHANNES	DURBAN	1	@0	APPN FD	11512	0	------	------	0.01746	0.02783	0.000	55.412
DURBAN	JOHANNES	2	@1	APPN FD	17699	0	------	------	0.00275	0.00666	0.000	31.174
JOHANNES	DURBAN	2	@1	APPN FD	17638	0	------	------	0.28953	0.32985	0.000	94.732
. . .												
DURBAN	QUEENSTO	1	@0	APPN FD	14659	0	------	------	0.01997	0.03011	0.000	66.608
QUEENSTO	DURBAN	1	@0	APPN FD	14700	0	------	------	0.00206	0.00495	0.000	30.546
. . .												
GRAFF_RE	JOHANNES	1	@0	APPN FD	7198	0	------	------	0.00027	0.00121	0.000	9.903
JOHANNES	GRAFF_RE	1	@0	APPN FD	7098	0	------	------	0.01611	0.02631	0.000	44.945

Figure 141. Simulation Report

The utilizations are consistent with those of Figure 41 on page 42, with one exception. The Routes report of which Figure 66 on page 62 contains an extract, showed that two APPN routes of equal weight were available for traffic between locations DE_AAR and JOHANNES. The relevant portion appears in Figure 142.

```
Origin:  DE_AAR,  Destination:  JOHANNES,  Least Weight: 195.
  DE_AAR(1)CAPE_TOW(1)JOHANNES
  DE_AAR(1)GRAFF_RE(1)JOHANNES
```

Figure 142. Simulation Report

The design assumed an equal split of traffic across the two routes but the simulation resulted in a roughly three-to-one split. This also affects utilizations of the other TGs in the routes.

Chapter 13. Capacity Planning

In Chapter 4, "Automated Design" on page 31 we designed our network to satisfy response time criteria for our traffic. As traffic requirements change, typically upwards, what can we do to monitor the situation and predict the need for any topology changes or increases in bandwidth?

Suppose that a 20% increase in traffic is predicted. By clicking on **Modify column** in the ND pop-up menu and selecting the REQUIREMENTS table, we can add 20% to the input and output message rates. Selecting **View**, **Show** and **Capacity**, again gives a picture with TGs colored to reflect the utilizations.

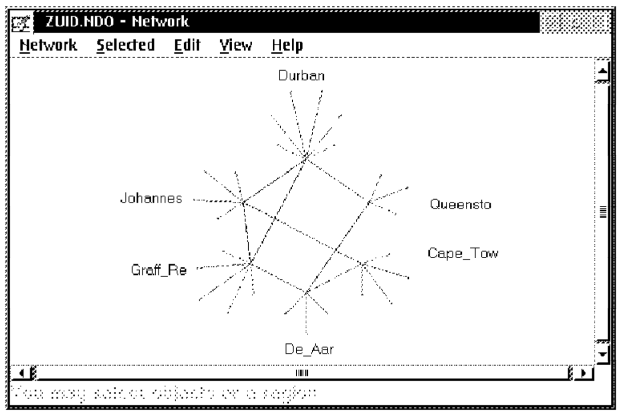


Figure 143. Utilization

The peripheral links supporting MSG1 traffic are in one color and those supporting MSG2 traffic are in another color, because the utilizations differ and the values are in different utilization bands. The backbone links appear in three different colors, with JOHANNES to DURBAN being in red.

Figure 144 shows the revised backbone TG utilizations.

Appended at 13:41:59, Monday, 15 September 1997. ZUID.NDO

TG Utilizations - (%)

End1	End2	TG	Utilization 1->2	Utilization 2->1
CAPE_TOW	DE_AAR	1	60.00	12.00
CAPE_TOW	JOHANNES	1	33.84	77.04
DE_AAR	GRAFF_RE	1	12.00	60.00
DE_AAR	QUEENSTO	1	51.20	51.20
DURBAN	GRAFF_RE	1	57.60	57.60
DURBAN	GRAFF_RE	2	44.80	44.80
DURBAN	JOHANNES	1	23.10	77.10
DURBAN	JOHANNES	2	36.42	97.62
DURBAN	QUEENSTO	1	77.04	33.84
GRAFF_RE	JOHANNES	1	10.80	54.00

Figure 144. Revised Utilizations

We see that the JOHANNES to DURBAN TGs have 77% and 97% utilizations which suggests that some reconfiguration may be necessary. Having made the changes to the traffic we may also want to re-run simulation.

Had utilization exceeded 100% on a TG we would have seen Figure 145 instead of Figure 143 on page 149. This occurred when traffic was increased by 30% instead of 20%. It is still possible, but with reduced value, to run simulation when loads exceed 100%. The results showed that more PIUs experienced delays above the thresholds specified, and the average delay on one TG between JOHANNES to DURBAN was about ten seconds. This delay would increase as the simulated time increases.

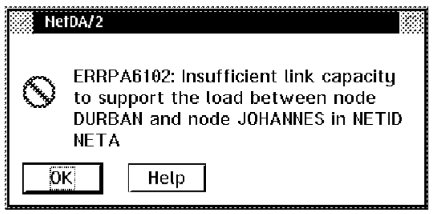


Figure 145. Utilization Exceeding 100%

13.1.1 Changing the Topology

To design again we would create a new NI from our designed ND, using the menu option shown in Figure 146. We select **Network Input** and then **Create**.

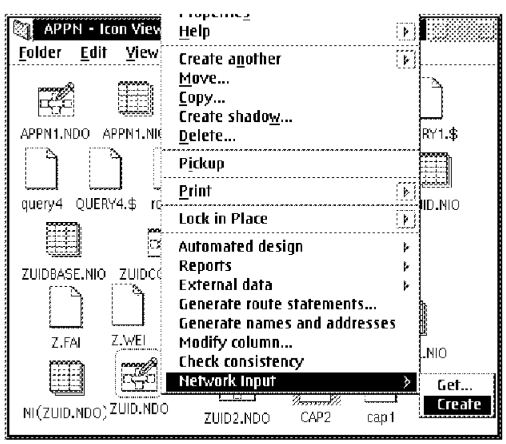


Figure 146. Create an NI

This creates an NI with the name NI(ZUID.NDO) which will include our specification of 6 for the maximum number of backbone nodes. We design again with the increased utilization of 20%. The only changes will be to the topology because all node types are now fixed. By default the design will include all of our existing links although we could vary this percentage by changing the Link inclusion parameter shown in Figure 33 on page 37.

This time the design ends with the message shown in Figure 147 on page 151.

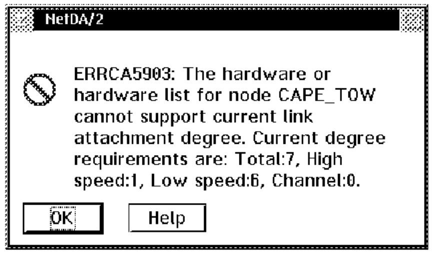


Figure 147. ERRCA5903

NetDA/2 determines how many low-speed and how many high-speed ports are needed by each node.

We see that NetDA/2 is looking for additional low-speed ports. The SMALL equipment is defined with a potential for five low speed and three high-speed ports and the tool follows the user definitions. Sometimes an approach is to define an imaginary piece of equipment with large numbers of ports and then to inspect the resulting design to see exactly what NetDA/2 was looking for.

We assumed that it was practical to use a high-speed port for a lower speed requirement and changed the SMALL equipment definition. All ports were defined as low-speed, with the top end of the low range being 100 kbps (Figure 148).

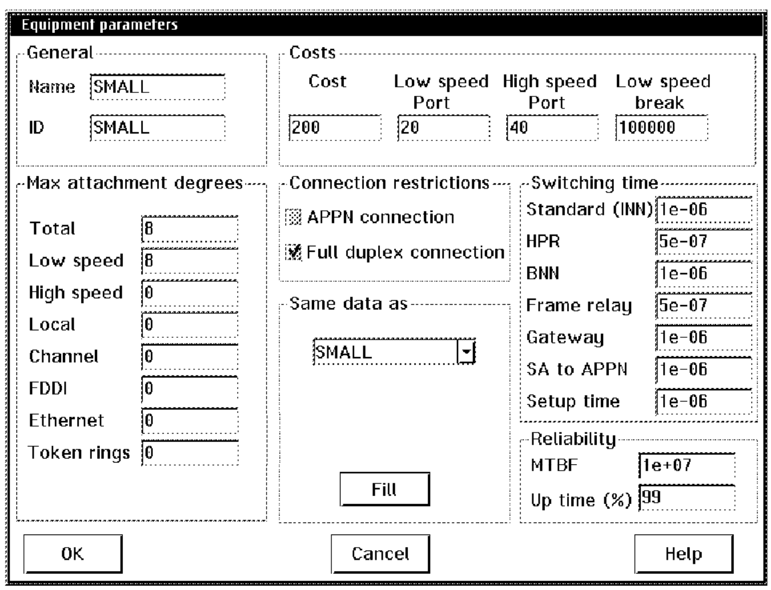


Figure 148. Revised SMALL Equipment

We repeat the design step and generate query reports.

Looking at a query report of capacity we find that an additional 19.2 kbps link has been added to the 64 kbps link between CAPE_TOW and JOHANNES. Two 19.2 kbps links have been added to the pair of 64 kbps links between DURBAN and JOHANNES.

```
Appended at 14:54:33, Monday, 15 September 1997.  ZUID.NDO

TG  Capacities - ( Bits/second )

End1        End2        TG     Capacity

CAPE_TOW    DE_AAR      1      19200
CAPE_TOW    JOHANNES    1      64000
CAPE_TOW    JOHANNES    2      19200
DE_AAR      GRAFF_RE    1      19200
DE_AAR      QUEENSTO    1      19200
DURBAN      GRAFF_RE    1      19200
DURBAN      GRAFF_RE    2      19200
DURBAN      JOHANNES    1      64000
DURBAN      JOHANNES    2      64000
DURBAN      JOHANNES    3      19200
DURBAN      JOHANNES    4      19200
DURBAN      QUEENSTO    1      64000
GRAFF_RE    JOHANNES    1      64000
```

Figure 149. Query Capacity Report

The new design summary view is shown in Figure 150 and the previous one is reproduced in Figure 151 on page 153.

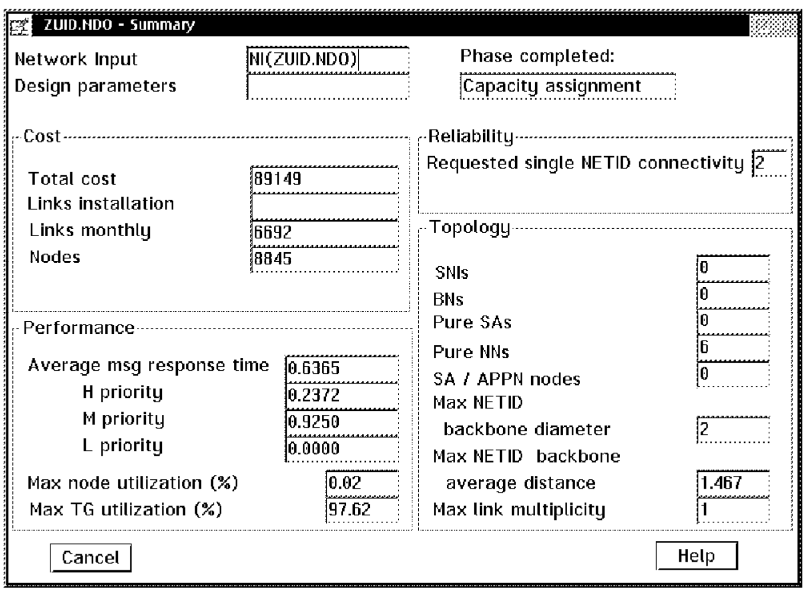


Figure 150. New Summary

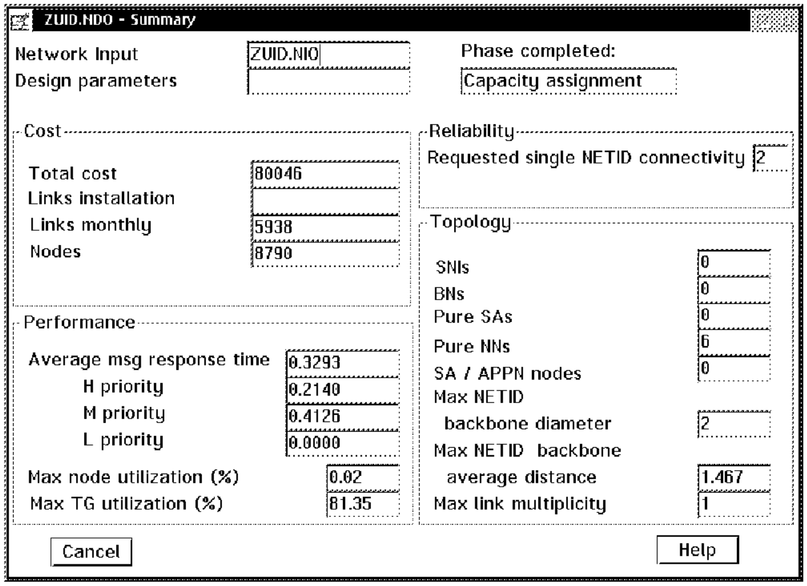


Figure 151. Old Summary

We see that the cost has increased due to the additional link rental and the cost of extra ports. Response times have increased, mainly for the lower priority MSG2. The Max node utilization is misleading because the traffic has not been spread over the new links. The value is that from Figure 144 on page 149.

Chapter 14. Not So Pure APPN

So far we have confined ourselves to pure APPN single NETID networks. Now it is time to consider adding some complexity.

14.1 Multiple NETIDs

In an APPN network with multiple NETIDs, we specify each NETID using graphical edit and also state in which NETID a node resides. There must be at least one border node and we must identify this node or nodes.

Suppose that in our ZUID network, we wish to place the group of nodes around CAPE_TOW in NETID NETB, leaving the remainder in NETA. So far we have only defined NETA, and no subnets. We would start by defining another NETID NETB, and then two subnets ASUB and BSUB, both of type APPN. This process was described in Chapter 3, “Graphical Input of APPN Networks” on page 17.

To move the nodes such as CALEDON into NETB/BSUB, we simply double-click on the icon in the NI network view, and update the General page, completing the NETID and Subnet boxes.

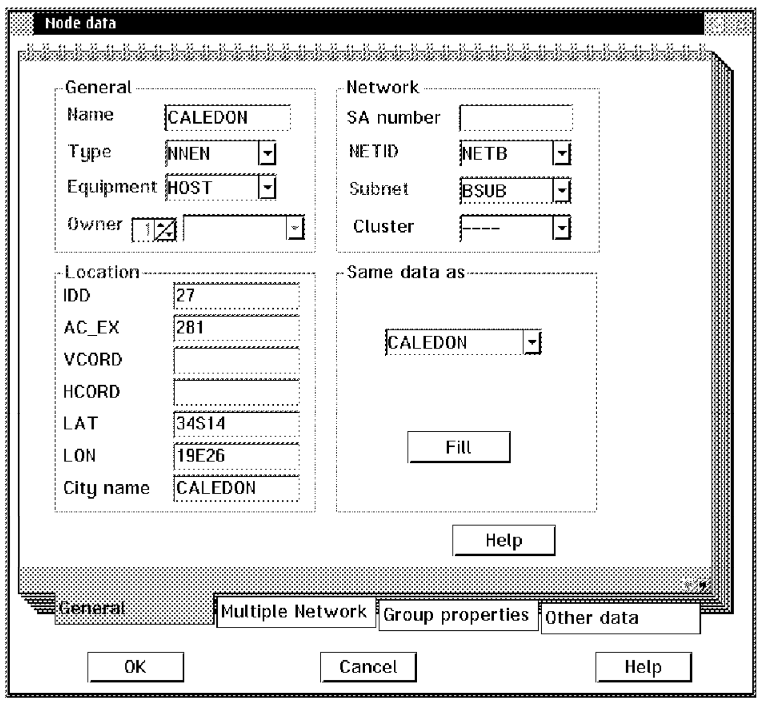


Figure 152. Moving to NETB

For CAPE_TOW we specify on the Multiple Network page that it is a BN1 gateway.

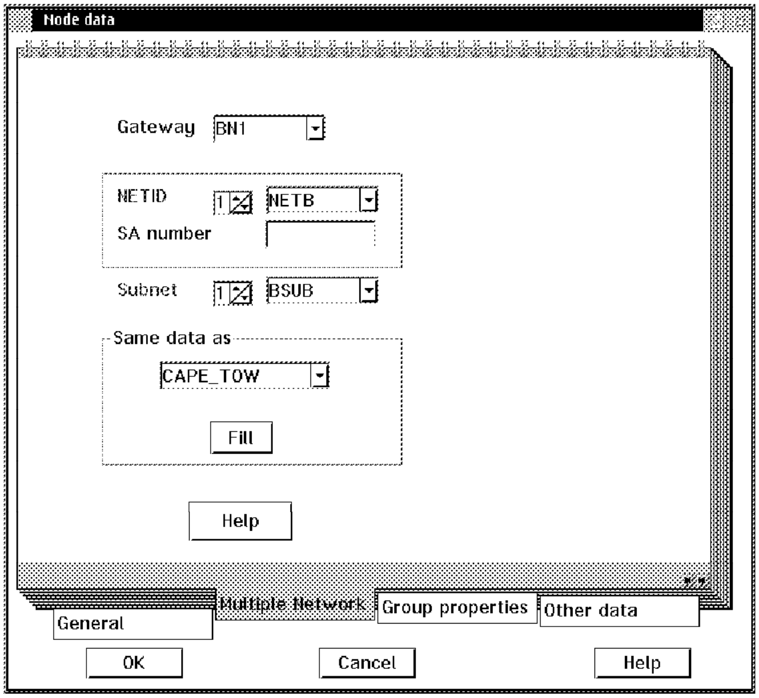


Figure 153. Moving to NETB

There is an additional requirement whenever gateways are present and this is to define which ones are used for particular sessions. This is a somewhat laborious process that requires definitions in the INTERNET_REQUIREMENTS table. For each record in the REQUIREMENTS tables that refers to a cross NETID or subnet session, in most cases we will define one record for each net or subnet involved and another for each border crossing.

We also must define the TGs between the networks. If the network is to be designed, it may be easier to design the network first as a single network and then the connecting TGs will be known and can be defined.

We used graphic edit to add two TGs to the TRUNKS table.

```
%TABLE TRUNKS

%END1++++ END2++++ LNTY+ LINK_ID TG_NUM FIXED_TG NUM MIN
 CAPE_TOW JOHANNES D64   @0      21     NO       1   1
 CAPE_TOW DE_AAR   D19.2 @0      21     NO       1   1
```

Figure 154. Connecting TGs

The INTERNET_REQUIREMENTS table must be modified with a flat file editor.

```
%TABLE INTERNET_REQUIREMENTS

%SOURCE++ DEST++++ MSGNAME CURRENT_SOURCE NEXT_NODE
 JOHANNES SOMERSET MSG1    JOHANNES       CAPE_TOW
 JOHANNES SOMERSET MSG1    CAPE_TOW       SOMERSET
 JOHANNES MALMESBU MSG1    JOHANNES       CAPE_TOW
 JOHANNES MALMESBU MSG1    CAPE_TOW       MALMESBU
 DURBAN   CALEDON  MSG2    DURBAN         DE_AAR
 DURBAN   CALEDON  MSG2    DE_AAR         CAPE_TOW
 DURBAN   CALEDON  MSG2    CAPE_TOW       CALEDON
 DURBAN   CERES    MSG2    DURBAN         DE_AAR
 DURBAN   CERES    MSG2    DE_AAR         CAPE_TOW
 DURBAN   CERES    MSG2    CAPE_TOW       CERES
 DURBAN   CAPE_TOW MSG2    DURBAN         DE_AAR
 DURBAN   CAPE_TOW MSG2    DE_AAR         CAPE_TOW
```

Figure 155. INTERNET_REQUIREMENTS Table

In our example we have five nodes in NETB, all sending traffic to DURBAN or JOHANNES in NETA. The records define the progression from the source to the edge of the net or subnet then across the next one, until the destination is reached. For example, for DURBAN to CALEDON, we have a row from source DURBAN to DE_AAR, the last point in NETA, another for the border crossing DE_AAR to CAPE_TOW and one for CAPE_TOW to the destination, CALEDON.

Note

Looks good but for some reason NetDA/2 insists on designing an extra NN in NETB and produces an expensive configuration. The Query reports show no traffic between NETA and NETB!

The *NetDA/2 User's Guide* has some examples of definitions.

14.2 Composite Network Nodes

These are a combination of a VTAM and the NCPs it owns, combining to appear as an NN to the rest of the APPN network. APPN routing uses a single weight for the CNN, regardless of which components are used in a route.

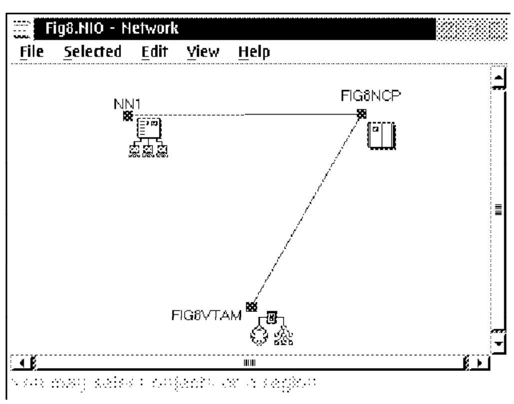


Figure 156. FIG8.NIO

In Figure 156 we show FIG8.NIO, which has a CNN with one NCP, and a separate NN. For the VTAM T5NN we must enter data on two pages, shown in Figure 157 on page 158 and Figure 158 on page 158.

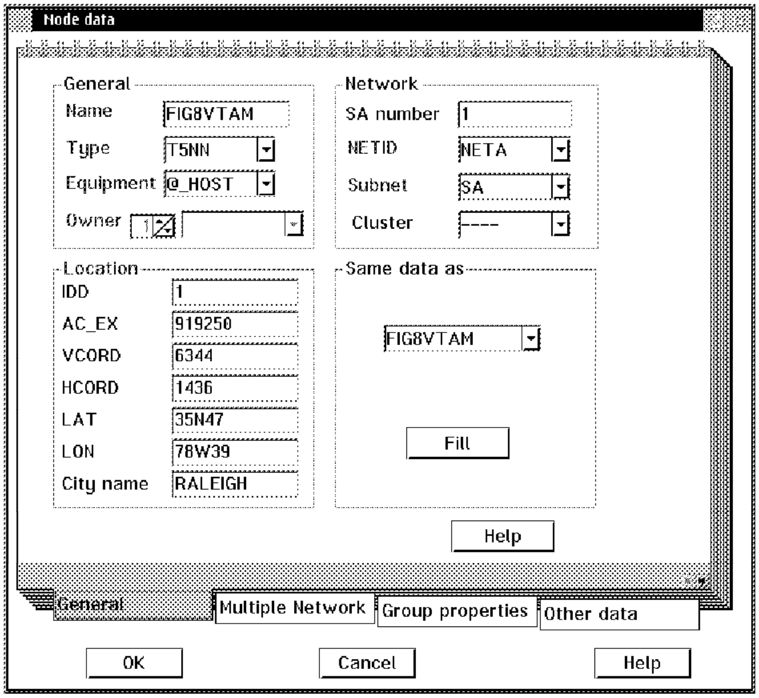


Figure 157. T5NN - General Page

The General page shows that FIG8VTAM is in NETID NETA and subnet SA. On the Multiple Network page we spin the subnet button to 2 and select APPN as the name of the second subnet to which the node belongs.

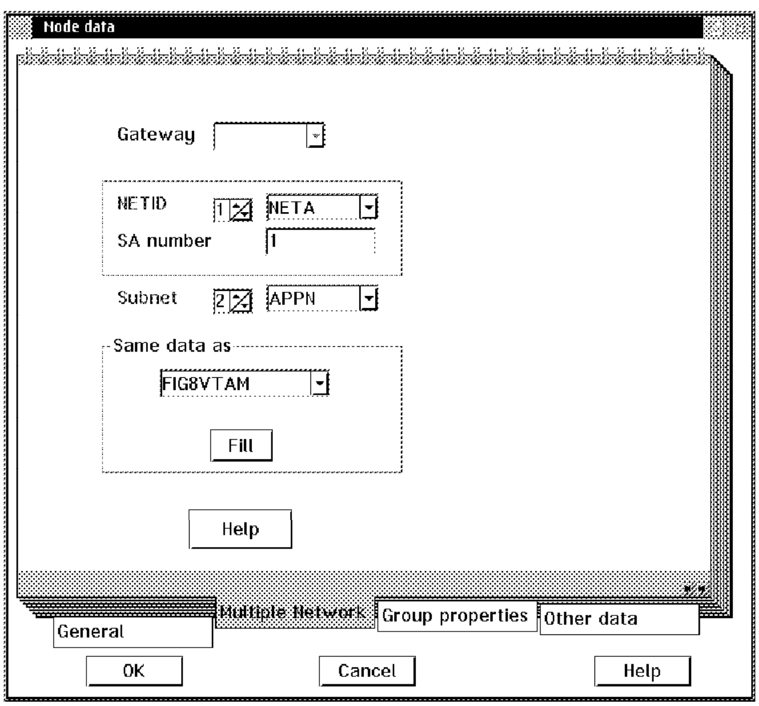


Figure 158. T5NN - Multiple Network Page

For the NCP we again use two pages. This time it is not necessary to use the Multiple Networks page to say that the NCP is part of the APPN subnet. Instead, on the Group properties page (Figure 160 on page 159) we note that it is part of the FIG8VTAM CNN.

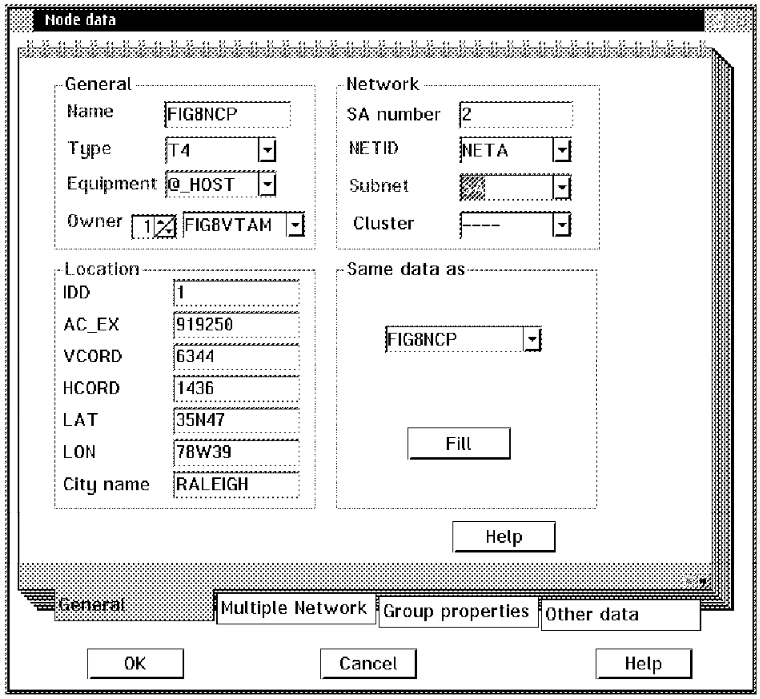


Figure 159. T4 - General Page

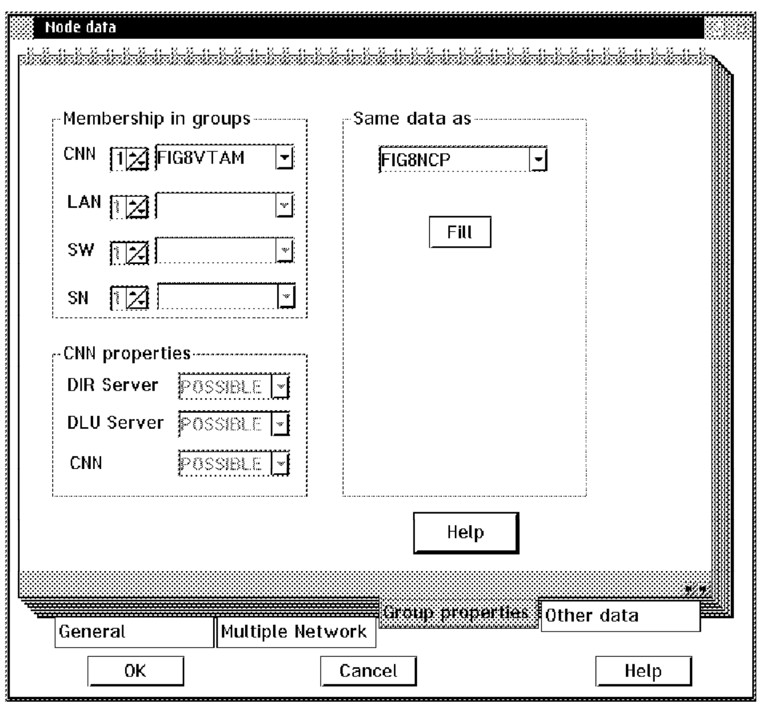


Figure 160. T4 - Group Properties Page

In NetDA/2 V1R5, the TG between the NCP and the NN may not be specified as a channel, only as a link. In a later refresh (NetDA/2 V1R5.3), it is possible to do this.

In NetDA/2 V1R5 you get the following error message if you try to define a channel connection between an NCP and a NN:

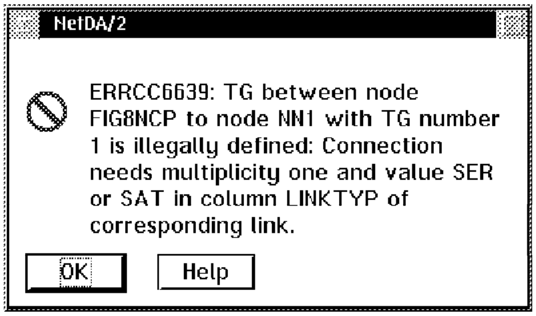


Figure 161. TG Specification

14.2.1 Routes

We may open the Route Control view, and generate the one subarea route. Using the APPN page we can specify the pair *;*, select a COS and click on **Routes**.

In reality there is only one APPN route, from NN1 to the CNN.

```
Appended at 17:22:14, Tuesday, 16 September 1997.  VAPPN.NDO

APPN Routes

 APPN Least Weight Routes Report for Network Design Object VAPPN.NDO,
 Started Tue Sep 16 1997 17:21:49.
 APPN Weights are Defined by COS Name #INTER.

 Subnet: APPN

 Origin:  FIG8NCP,  Destination:  FIG8VTAM,  Least Weight: 70.
   FIG8NCP(1)FIG8VTAM

 Origin:  FIG8NCP,  Destination:  NN1,  Least Weight: 70.
   FIG8NCP(1)NN1

 Origin:  FIG8VTAM,  Destination:  FIG8NCP,  Least Weight: 70.
   FIG8VTAM(1)FIG8NCP

 Origin:  FIG8VTAM,  Destination:  NN1,  Least Weight: 135.
   FIG8VTAM(1)FIG8NCP(1)NN1

 Origin:  NN1,  Destination:  FIG8NCP,  Least Weight: 70.
   NN1(1)FIG8NCP

 Origin:  NN1,  Destination:  FIG8VTAM,  Least Weight: 135.
   NN1(1)FIG8NCP(1)FIG8VTAM
```

Figure 162. APPN Routes

14.3 A Pair of CNNs

Figure 163 shows a configuration with two CNNs.

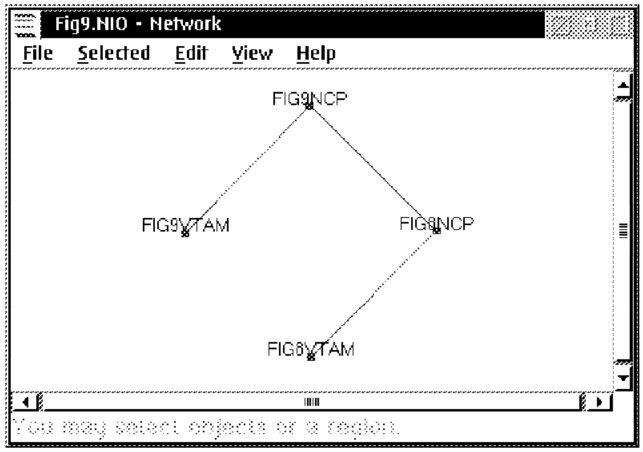


Figure 163. Pair of CNNs

We define each one using the same approach as in the previous example, each NCP being part of a different CNN. We may specify that the TG between them uses an APPN protocol by selecting **TG parameters** on the TG data page (see Figure 164) and then **APPN** from the list in Figure 165 on page 162.

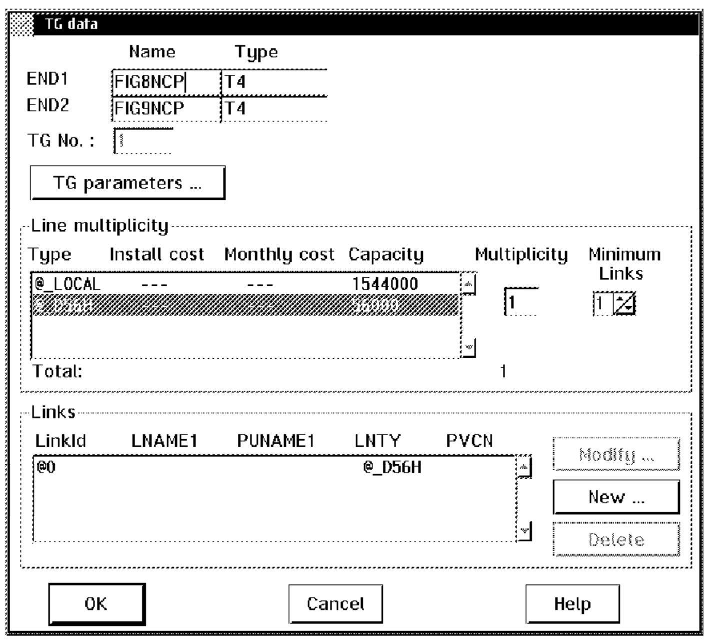


Figure 164. TG Data Page

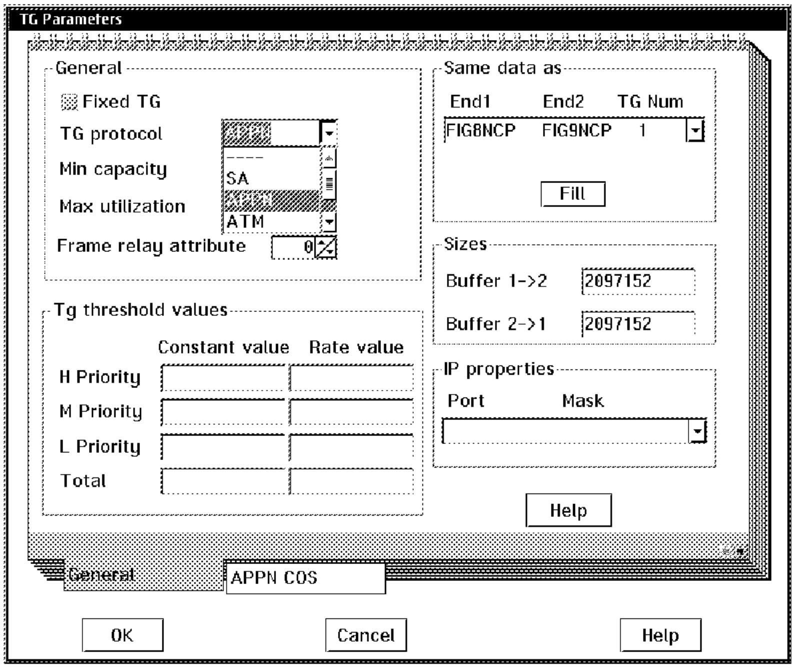


Figure 165. TG Parameters

14.3.1 Routes

Route control will generate two subarea routes. The APPN Routes report will show all six combinations although only one APPN route exists, from CNN to CNN.

14.4 Migration Data Host

To change this configuration so that FIG9VTAM is an MDH and the CNN includes both NCPs, we change:

- The TG_PR of the link between the NCPs to SA or ---.
- The TG_PR of the link between FIG9VTAM and FIG9NCP to APPN.
- The FIG9NCP's OWNER to FIG8VTAM.
- The FIG9NCP's CNN membership to FIG8VTAM
- The FIG9VTAM's TYPE to T5EN.

Route control will generate three subarea routes but again shows six combinations in the APPN Routes report.

In NetDA/2 V1R5 in some cases the MDH host is listed as an intermediate node. This bug was fixed in NetDA/2 V1R5.3.

14.5 Multiple APPN Subnetworks

A multiple APPN subnetworks configuration is illustrated in Figure 166.

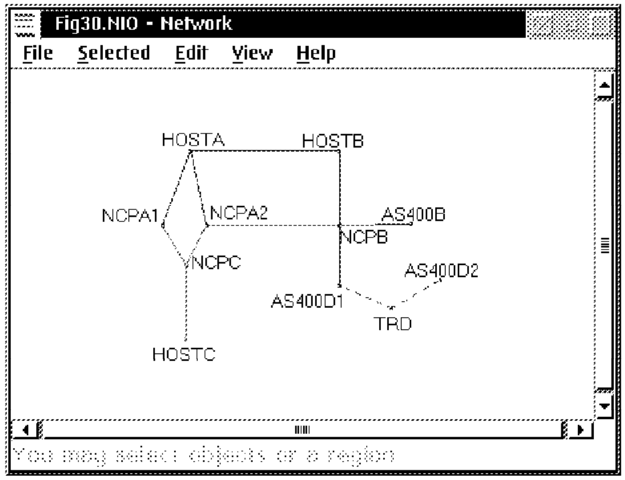


Figure 166. Multiple NETIDs and Subnets

HOSTA, NCPA1 and NCPA2, are a CNN in NETID A, APPN subnetwork A. For NetDA/2 we′ll designate the SA subnet as AASUBA and the APPN subnet as AAAPPN.

HOSTB and NCPB are a CNN in NETID A, APPN subnetwork C, together with network node AD400B. Here we use SA subnet ACSUBA and APPN subnet ACAPPN.

HOSTC and NCPC are a CNN in NETID B subnetwork B. (We did not choose this naming convention but are following the VTAM example.) Subnets are BBSUBA and BBAPPN.

Finally, AS400D1 and AS400D2 are in NETID D, subnet D. All connections between subnets use APPN protocols.

14.5.1 Routes

After dropping the NI on ND and opening the Route control view, we can generate four subarea routes, three in AASUBA and one in ACSUBA. An extract from the NETID APPN Route listing appears in Figure 167 on page 164.

This shows the same routes as are in the subarea listing plus routes from AS400B to HOSTB and to NCPB.

```
Appended at 09:38:04, Thursday, 18 September 1997.  VAPPN.NDO

APPN Routes

 APPN Least Weight Routes Report for Network Design Object VAPPN.NDO,
 Started Thu Sep 18 1997 09:36:30.
 APPN Weights are Defined by COS Name #INTER.

 Subnet: AAAPPN

 Origin:  HOSTA,  Destination:  NCPA1,  Least Weight: 70.
   HOSTA(1)NCPA1
  . . .

 Subnet: BBAPPN
 None of the selected pairs belongs to the subnet.

 Subnet: ACAPPN

 Origin:  AS400B,  Destination:  HOSTB,  Least Weight: 135.
   AS400B(1)NCPB(1)HOSTB


 Origin:  AS400B,  Destination:  NCPB,  Least Weight: 70.
   AS400B(1)NCPB
 . . .
 Origin:  NCPB,  Destination:  AS400B,  Least Weight: 70.
   NCPB(1)AS400B
 . . .
 Subnet: DDAPPN
 None of the selected pairs belongs to the subnet.
```

Figure 167. APPN Routes

14.6 Mixed Network Design

We have shown how to specify APPN nodes, and VTAM CNNs and MDHs. Entry of pure subarea nodes is described in the NetDA/2 redbook.

When NetDA/2 designs a mixed network, it can suggest which nodes should be DLU servers or directory servers, and also which VTAM/NCPs combinations should be composite network nodes. In practice, when migrating from subarea to APPN it will often already be apparent which VTAMs should perform these roles.

To ask NetDA/2 to decide, we use the Group Properties page for node data.

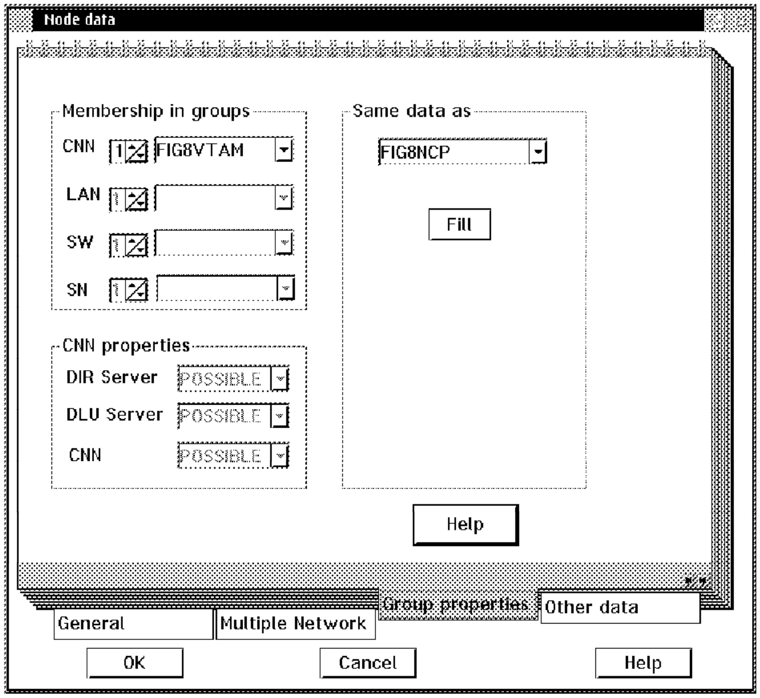


Figure 168. Group Properties Page

In Figure 168 the CNN properties box has three fields. The first two are enabled for input if the node type is T5NN, and the values are YES, NO and POSSIBLE. The third box is enabled for T5s, with POSSIBLE and NO as the choices.

Appendix A. Special Notices

This publication is intended to help network planners and systems programmers to use NetDA/2 in connection with APPN networks. The information in this publication is not intended as the specification of any programming interfaces that are provided by NetDA/2. See the PUBLICATIONS section of the IBM Programming Announcement for NetDA/2 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 use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ACF/VTAM	APPN
AS/400	AT
C/370	CICS
DB2	ESCON
IBM	MVS/ESA
NetView	OS/2
OS/390	SP
VTAM	Workplace Shell
400	

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.

B.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How to Get ITSO Redbooks" on page 171.

- *NetDA/2 V1R5 Design Tool Guide and Tutorial*, SG24-4225
- *Subarea Network to APPN Network Migration Experiences*, SG24-4656

B.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

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
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
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

B.3 Other Publications

These publications are also relevant as further information sources:

- *Network Design and Analysis/2 User's Guide*, SC31-6440.
- *TME10 NetView for OS/390 V1R1 SNATM and APPN Accounting Manager*, SC31-8224-00
- *TME 10 NetView for OS/390 V1R1 SNA Topology and APPN Accounting Implementation Guide*, SC31-8239-00
- *TME 10 NetView for OS/390 V1R1 Application Programming Guide*, SC31-8223-00

How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at http://www.redbooks.ibm.com.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks, type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO: type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Web Site on the World Wide Web**

  http://w3.itso.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

Redpieces

For information so current it is still in the process of being written, look at ″Redpieces″ on the Redbooks Web Site (http://www.redbooks.ibm.com/redpieces.htm). 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.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** — send orders to:

	IBMMAIL	Internet
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1)001-408-256-5422 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com
- **On the World Wide Web**

  Redbooks Web Site http://www.redbooks.ibm.com
  IBM Direct Publications Catalog http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet e-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an e-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (http://www.redbooks.ibm.com/redpieces.htm). 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.

IBM Redbook 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

ANR	Automatic Network Routing
APPC	Advanced Program to Program Communication
APPN	Advanced Peer to Peer Networking
ARB	Adaptive Rate-Based
CDS	Central Directory Server
CMC	Communications Management Configuration
CMC	Communications Management Configuration
COS	Class Of Service
CP	Control Point
CTC	Channel To Channel
DLUR/S	Dependent LU Requester / Server
EN	End Node
FID	Format IDentifier
FMD	Function Management Data
HPR	High-Performance Routing
IBM	International Business Machines Corporation
IC-TG	InterChange Transmission Group
ICN	InterChange Node
ITSO	International Technical Support Organization
LU	Logical Unit
MDH	Migration Data Host
NCP	Network Control Program
NLP	Network Layer Packet
NN	Network Node
NNS	Network Node Server
PCID	Procedure Correlated IDentifier
PIU	Path Information Unit
PU	Physical Unit
RTP	Rapid Transport Protocol
SSCP	System Services Control Point
SNA	Systems Network Architecture
SNI	SNA Network Interconnection
TG	Transmission Group
VR	Virtual Route
VR-TG	Virtual Route-based Transmission Group
VTAM	Virtual Telecommunications Access Method

Index

Numerics

ITSO Redbook Evaluation

Using NetDA/2 in an APPN Environment
SG24-2143-00

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.com
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@vnet.ibm.com

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!)

____

____

____

____

____

Printed in U.S.A.

SG24-2143-00