IBM

Telecommunications Systems Support - Washington

VTAM V3R2 AND NCP V4R3/V5R2 INSTALLATION CONSIDERATIONS

W. E. Clauberg
N. W. Gates
W. K. Gibbons
J. H. Weatherspoon

Telecommunications Technical Bulletin
GG66-3102-01

Washington Systems Center
Gaithersburg, MD
Technical Bulletin

VTAM V3R2 AND NCP V4R3/V5R2 INSTALLATION CONSIDERATIONS

W. E. Clauberg
N. W. Gates
W. K. Gibbons
J. H. Weatherspoon

GG66-3102-01
December, 1989

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

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 licensed program in this publication is not intended to state or imply that only IBM's program may be used. Any functionally equivalent program may be used instead.

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to: Telecommunications Systems Support. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This technical bulletin was originally written in the summer of 1988. The original bulletin was based on information about VTAM V3R2 and NCP V4R3/V5R2 prior to their general availability. Some of the examples in this bulletin are based on results using early VTAM and NCP code. Late in 1989, the bulletin was updated with information pertaining to the VTAM enhancements and NCP V4R3.1/V5R2.1. Some of the examples in this bulletin are based on results using this level of VTAM and NCP. Because of the various levels of VTAM and NCP code, it is possible that there will be some slight differences between the examples provided in this bulletin and the results experienced by installations that have different levels of VTAM and NCP. Changes between the original bulletin, GG66-3102, and the updated bulletin, GG66-3102-1, are marked with a revision code, |.

PROGRAMMING INTERFACES

The majority of this bulletin consists of general usage and guidance information. Such information should never be used as Programming Interface Information. However, this book also contains general-use Programming Interface Information.

Two functions, APPC/VTAM and Multiple Load Module Support are not discussed in this bulletin. Please refer to GG66-0283, listed below, for information on APPC/VTAM, and to GG66-0286 for information on Multiple Load Module Support.

Other information which may be of interest to the reader is contained in:

- GG66-0283 "A Technical Overview: VTAM Version 3 Release 2, NCP Version 4 Release 3, NCP Version 5 Release 2"
- GG66-0299 "VTAM Version 3 Release 2 & NCP Version 4 Release 3/Version 5 Release 2: SNA Type 2.1 Node Support Using the System/36 As An Example"
- GG66-0286 "3745 Technical Overview/3745 Model 210 Installation Planning Guide"
- GG66-3111 "3745 Model 410 Planning and Installation Topics"
- GG66-3127 "3745 Models 130/150/170 Technical Overview and Installation Topics"

We would like to thank the following people for their help in the preparation of this bulletin:

- Marge Cronin and Mary Foshee, CPD
- Dave Olson, Syd Palmer, Jim Robinson, and Si Snow, WTEC
- Karla Boyle, Kevin Cummings, Jim Lucas, Nanda Pandya, and Pat Walker, WSC

Contents

Figures

Tables

Chapter 1. Dynamic Table Replacement

1.1 Overview

1.1.1 Dynamic Table Replacement

At network initialization, a number of tables are loaded by VTAM into processor memory. These tables are used by VTAM for a variety of functions such as:

1. Translating characters keyed in at end user terminals.
2. Converting user input to "initiate self" information.
3. Displaying messages at VTAM network operator and end user terminals.
4. Selecting parameters for use in setting up sessions.
5. Selecting a route for sessions to use.

The types of tables that are used are: Interpret Tables in conjunction with function 1, Unformatted Systems Services (USS) Tables in conjunction with functions 2 and 3, Logon Mode (Logmode) Tables in conjunction with function 4, and Class of Service (COS) Tables in conjunction with function 5.

With VTAMs prior to V3R2, these tables cannot easily be changed or new tables added; in order to load a new copy of a table, VTAM must be halted and restarted. In some cases, a resource may be associated with a new table if the Major Node(s) is deactivated and reactivated. However, prior to V3R2, changing, adding or deleting a table is disruptive to some or all of the network. With VTAM V3R2 a new function called Dynamic Table Replacement (DTR) allows these tables to be added, deleted or replaced by a new command:

MODIFY TABLE

issued from the VTAM Operator console. The ability to dynamically alter the tables should improve network availability because the user does not need to restart VTAM or deactivate Major Nodes in order to cause VTAM to load a new table or new copy of a table into processor memory.

Another helpful function in Dynamic Table Replacement is the ability to name a resource or set of resources that should be associated with a particular table. For example, if all of the Logical Units within one NCP were currently using a specific USS table, the table association of one resource or group of resources could be changed to use a different USS table through use of the "MODIFY TABLE" command.

The MODIFY TABLE command can also be used in conjunction with the Session Awareness (SAW) Data Filter table, a new function in VTAM V3R2 which is described in Chapter 11 of this manual.

NOTE: It is necessary to assemble Interpret and USS tables with VTAM V3R2 libraries in order to use the MODIFY TABLE command in conjunction with these tables. See Section 1.7 for more information on table assembly with VTAM V3R2.

1.1.2 Enhanced Display ID VTAM Command

With VTAM V3R2, when the VTAM operator issues a DISPLAY command specifying a Logical Unit (LU), the resulting messages have been enhanced to show the Logon Mode, USS, and Interpret tables in use for the LU. Also displayed is the default logon mode entry (DLOGMOD) of the LU. Examples of the enhanced DISPLAY command appear later in this chapter. The first example of the enhanced DISPLAY command is in Figure 1-1 on page 1-7.

1.1.3 New Display COS VTAM Command

VTAM V3R2 provides a new operator command,

DISPLAY COS

which can be used to determine the name of the COS table(s) in use. COS tables for both native and non-native networks can be determined through use of this command. An example of the DISPLAY COS command is provided in Section 1.6.4.

1.2 How Tables Are Used in the Logon Process

To request a session with an application program, an LU sends a logon request to VTAM specifying the application program's name and, optionally, a logon mode name and user data. Programmable terminals (such as 4700s) generate INITIATE-SELF and TERMINATE-SELF requests (called field-formatted requests) to send to VTAM. Non-programmable terminals (such as 3270s) do not generate INITIATE-SELF and TERMINATE-SELF commands for VTAM processing. For non-programmable terminals, VTAM can accept logons and logoffs in character-coded (unformatted) form. In order for VTAM to accept the requests, the unformatted system services (USS) component of VTAM must have the appropriate tables to convert the character-strings into formatted requests. The logon format needs to be in the form:

LOGON APPLID(programname) LOGMODE(modename) DATA(userdata)

in order for VTAM to process the session initiation request.

The user does not actually have to key in the logon request in the above form. One or a combination of the IBM-supplied session-level USS definition table, user-written supplemental USS definition tables, or Interpret tables written to provide application program names can be used to convert the user keyed data to formatted requests.

Logon Mode Tables are used for specifying session protocols that are appropriate for different types of Logical Units. For example, differing types of user terminals may have varying display sizes, buffer sizes, and may or may not accept chaining. Different types of terminals may be in session with the same application program. Differing types of terminals can use separate Logon Mode Table entries which VTAM uses to pass session protocols to the application program.

Class of Service Tables are used to specify Virtual Routes for sessions.

1.3 Interpret Tables

1.3.1 Interpret Tables--General Description

When VTAM processes a formatted initiate or terminate request, either directly from an LU or formatted by USS from a character coded logon or logoff, it first attempts to use the Interpret Table associated with the LU to translate the character string into an application program name. Interpret Tables can be used in conjunction with USS Tables. The Network Program Products Planning manual and the VTAM Customization manual contain descriptions of how USS tables and Interpret tables are used in the logon request conversion process.

1.3.2 Using Dynamic Table Replacement with Interpret Tables

The Interpret Table that VTAM uses for a specific logical unit (LU) is pointed to from the definition of that LU in VTAM or NCP through use of the LOGTAB operand. Below is an example of coding in our lab's NCP for a 3290 (the four LUs listed) attached to a 3174 (the PU, P040A). These LUs are defined to use the Interpret table called "LOGONTAB":

```
G3174      GROUP  PUTYPE=2,LOGTAB=LOGONTAB
L040       LINE   ADDRESS=(040,FULL),ISTATUS=ACTIVE
           SERVICE ORDER=(P040A)
P040A      PU  ADDR=C1
X040A02    LU  LOCADDR=02,DLOGMOD=M2SDLCNQ
X040A03    LU  LOCADDR=03,DLOGMOD=M2SDLCNQ
X040A04    LU  LOCADDR=04,DLOGMOD=M2SDLCNQ
X040A05    LU  LOCADDR=05,DLOGMOD=M2SDLCNQ
```

This particular table, LOGONTAB, allows a user to key in a variety of different character strings in order to be connected with various applications. For example, the string DSX connects the user with DSX, NVAS with NetView/AS, HCF with HCF, etc.. We wanted the user to have the ability to key in "DYN" to be connected with NetView. To do this we used Dynamic Table Replacement to change the association of the PU (and LUs under it) to a new table.

These are the steps we followed:

1. We copied the existing Interpret table and added the statement:

   ```
   LOGCHAR  APPLID=(APPLCID,N2P1),SEQNCE='DYN'
   ```

2. We changed the name of the Interpret table to "INTDYN" by coding :

   ```
   INTDYN   INTAB
   ```

   as the first statement in the table.

3. We assembled and link-edited the table into SYS1.VTAMLIB, calling it INTDYN.

4. To change the Interpret table used by the PU, P040A, and LUs defined on that PU we issued the command:

   ```
   F CSSVTAM,TABLE,ID=P040A,TYPE=INTTAB,
   OLDTAB=LOGONTAB,NEWTAB=INTDYN,OPTION=ASSOCIATE
   ```

The effect of this command is to load the table, INTDYN, into memory and change the associations of the four LUs listed above to now use INTDYN as an Interpret table. INTDYN would be used upon the next logon of LU X040A03, for example.

Figure 1-1 on page 1-7 shows our console listing when we used Dynamic Table Replacement to replace the Interpret Table, LOGONTAB, with a new table, INTDYN, for PU P040A and its LUs:

- First, LU X040A03 is displayed and the resulting messages show that LOGONTAB is the associated Interpret table. This is an example of the enhanced DISPLAY ID command in VTAM V3R2.
- Next, the MODIFY TABLE command is issued to change the Interpret table for the PU, P040A, and its LUs to the table named INTDYN.
- Next, a display of the LU, X040A03, shows that the Interpret table is now INTDYN.
- Next, a display of an LU on a different PU, LU X000C02, shows that the Interpret table for that LU, LOGONTAB, has not changed.

```
D NET,ID=X040A03
IST097I DISPLAY ACCEPTED
IST075I NAME = X040A03, TYPE = LOGICAL UNIT
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST861I MODETAB=AMODETAB USSTAB=USS3270A LOGTAB=LOGONTAB
IST934I DLOGMOD=M2SDLCNQ
IST597I CAPABILITY-PLU INHIBITED,SLU ENABLED  ,SESSION LIMIT 00000001
IST081I LINE NAME = L040, LINE GROUP = G3174, MAJNOD = FRACKX2
IST135I PHYSICAL UNIT = P040A
IST082I DEVTYPE = LU
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000000, SESSION REQUESTS = 0000000000
IST314I END
F CSSVTAM,TABLE,NEWTAB=INTDYN,TYPE=INTTAB,OPTION=ASSOCIATE,ID=P040A,
OLDTAB=LOGONTAB
IST097I MODIFY ACCEPTED
IST865I MODIFY TABLE COMMAND COMPLETE- 4 ASSOCIATION(S) CHANGED
IST864I NEWTAB=INTDYN, OLDTAB=LOGONTAB, OPT=ASSOCIATE, TYPE=LOGTAB
IST935I ORIGIN=***NA***, NETID=***NA***, ID=P040A
D NET,ID=X040A03
IST097I DISPLAY ACCEPTED
IST075I NAME = X040A03, TYPE = LOGICAL UNIT
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST861I MODETAB=AMODETAB USSTAB=USS3270A LOGTAB=INTDYN
IST934I DLOGMOD=M2SDLCNQ
IST597I CAPABILITY-PLU INHIBITED,SLU ENABLED  ,SESSION LIMIT 00000001
IST081I LINE NAME = L040, LINE GROUP = G3174, MAJNOD = FRACKX2
IST135I PHYSICAL UNIT = P040A
IST082I DEVTYPE = LU
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000000, SESSION REQUESTS = 0000000000
IST314I END
D NET,ID=X000C02
IST097I DISPLAY ACCEPTED
IST075I NAME = X000C02, TYPE = LOGICAL UNIT
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST861I MODETAB=AMODETAB USSTAB=USS3270A LOGTAB=LOGONTAB
IST934I DLOGMOD=M3276
IST597I CAPABILITY-PLU INHIBITED,SLU ENABLED  ,SESSION LIMIT 00000001
IST081I LINE NAME = L000, LINE GROUP = G3276, MAJNOD = FRACKX2
IST135I PHYSICAL UNIT = P000C
IST082I DEVTYPE = LU
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000000, SESSION REQUESTS = 0000000000
IST314I END
```

Figure 1-1. Console Listing--Dynamic Change of Interpret Table for a PU

The above example illustrates loading a new table and associating a set of resources (all LUs under PU P040A) with the new table.

It is also possible to load a new table using Dynamic Table Replacement and have ALL resources that were using the old table automatically associated with the new table. In order to accomplish this, the MODIFY TABLE command is issued in the form:

```
F CSSVTAM,TABLE,NEWTAB=INTDYN,OLDTAB=LOGONTAB,OPTION=LOAD
```

Figure 1-2 shows loading a new table, INTDYN, and automatically having the resources formerly associated with LOGONTAB now associated with INTDYN. A display of the LU X000C02 after the command is issued illustrates that the LU is now associated with INTDYN.

```
F CSSVTAM,TABLE,NEWTAB=INTDYN,OLDTAB=LOGONTAB,OPTION=LOAD
IST097I MODIFY ACCEPTED
IST865I MODIFY TABLE COMMAND COMPLETE-TABLE INTDYN LOADED
IST864I NEWTAB=INTDYN, OLDTAB=LOGONTAB, OPT=LOAD, TYPE=**NA**
D NET,ID=X000C02
IST097I DISPLAY ACCEPTED
IST075I NAME = X000C02, TYPE = LOGICAL UNIT
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST861I MODETAB=AMODETAB USSTAB=USS3270A LOGTAB=INTDYN
IST934I DLOGMOD=M3276
IST597I CAPABILITY-PLU INHIBITED,SLU ENABLED  ,SESSION LIMIT 00000001
IST081I LINE NAME = L000, LINE GROUP = G3276, MAJNOD = FRACKX2
IST135I PHYSICAL UNIT = P000C
IST082I DEVTYPE = LU
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000000, SESSION REQUESTS = 0000000000
IST314I END
```

Figure 1-2. Console Listing--Associating all resources with A New Table

The MODIFY TABLE command can also be used to replace an Interpret table with a refreshed copy of the same table. The command to replace a table is:

```
F CSSVTAM,TABLE,NEWTAB=LOGONTAB,OLDTAB=LOGONTAB,OPTION=LOAD
```

Figure 1-3 shows the console listing when the Interpret table, LOGONTAB, was replaced with a new copy.

```
F CSSVTAM,TABLE,NEWTAB=LOGONTAB,OLDTAB=LOGONTAB,OPTION=LOAD
IST097I MODIFY ACCEPTED
IST865I MODIFY TABLE COMMAND COMPLETE-TABLE LOGONTAB LOADED
IST864I NEWTAB=LOGONTAB, OLDTAB=LOGONTAB, OPT=LOAD, TYPE=**NA**
```

Figure 1-3. Console Listing--Replacing an Interpret Table with A New Copy

It is also possible to delete the association between an Interpret table and a resource. In the following example we deleted the association between LU X040A03 and the Interpret table, INTDYN. The resulting display shows that X040A03 is no longer using an Interpret table. In our lab exercise, the end user could no longer key in the character strings, such as "DYN" and "NVAS," to be connected with an application. The LU was still associated with a USS table, USS3270A, which allowed the user to key in some different character strings to be connected with some different applications.

```
F CSSVTAM,TABLE,OLDTAB=INTDYN,TYPE=INTTAB,OPTION=DELETE,ID=X040A03
IST097I MODIFY ACCEPTED
IST865I MODIFY TABLE COMMAND COMPLETE- 1 ASSOCIATION(S) DELETED
IST864I NEWTAB=***NA***, OLDTAB=INTDYN, OPT=DELETE, TYPE=LOGTAB
D NET,ID=X040A03
IST097I DISPLAY ACCEPTED
IST075I NAME = X040A03, TYPE = LOGICAL UNIT
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST861I MODETAB=AMODETAB USSTAB=USS3270A LOGTAB=***NA***
IST934I DLOGMOD=M2SDLCNQ
IST597I CAPABILITY-PLU INHIBITED,SLU ENABLED  ,SESSION LIMIT 00000001
IST081I LINE NAME = L040, LINE GROUP = G3174, MAJNOD = FRACKX2
IST135I PHYSICAL UNIT = P040A
IST082I DEVTYPE = LU
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000000, SESSION REQUESTS = 0000000000
IST314I END
```

Figure 1-4. Console Listing--Deleting an Interpret Table

1.4 USS Tables

1.4.1 USS Tables--General Description

VTAM uses two types of Unformatted System Services (USS) tables--session-level USS tables and the operation-level USS table. The commands and messages that are defined in these USS tables are called USS commands and USS messages. Whenever VTAM receives a USS command, it uses one of these tables to process the command. Similarly, whenever VTAM is to send a USS message, it uses one of these tables to determine the message text and other characteristics of the message.

The session-level USS table handles commands and messages for logical units such as end-user terminals. IBM supplies a default table for this purpose called **ISTINCDT**. Included as part of this table is an IBM-supplied translation table named STDTRANS. Installations generally customize additional USS tables to allow end-users to key in different character strings in order to logon or logoff or to allow different messages to be displayed at the end-user terminal.

The operation-level USS table handles USS commands that can be received from the VTAM operator and messages that are sent by VTAM to the VTAM operator. IBM supplies an operation-level USS table named **ISTINCNO**. Installations can also customize an operation-level USS table and specify the customized one to be used through the VTAM start option, USSTAB=.

1.4.2 Using Dynamic Table Replacement with USS Tables

The session-level USS tables and operation-level USS table can be added, replaced, or deleted through use of VTAM V3R2's Dynamic Table Replacement function. There is an additional IBM supplied operation-level USS table named ISTCFCMM. **ISTCFCMM cannot be added, replaced or deleted using Dynamic Table Replacement.** Because VTAM may perform functions depending on the issuing of messages contained in ISTCFCMM, this table cannot be altered via Dynamic Table Replacement in order to avoid problems.

The session-level USS table used by VTAM for a specific logical unit (LU) is pointed to from the definition of that LU in VTAM or NCP through use of the USSTAB operand. Below is an example of coding in our lab's NCP for a 3290 (the four LUs listed) attached to a 3174 (the PU, P040A). These LUs are defined to use the USS table called "USS3270A":

```
G3174      GROUP  PUTYPE=2,USSTAB=USS3270A,LOGTAB=LOGONTAB
L040       LINE   ADDRESS=(040,FULL),ISTATUS=ACTIVE
           SERVICE ORDER=(P040A)
P040A      PU  ADDR=C1
X040A02    LU  LOCADDR=02,DLOGMOD=M2SDLCNQ
X040A03    LU  LOCADDR=03,DLOGMOD=M2SDLCNQ
X040A04    LU  LOCADDR=04,DLOGMOD=M2SDLCNQ
X040A05    LU  LOCADDR=05,DLOGMOD=M2SDLCNQ
```

This particular table, USS3270A, allows a user to key in a variety of different character strings in order to be connected with various applications with the correct logmode. For example, L allows the user to be logged onto TSO. In addition, MSG0 and MSG10 have been customized for use on our system. MSG0 is displayed when a user is logging on to notify the user that the logon is being executed. MSG10 is the display the appears on an end-user terminal once the terminal is active to VTAM.

We created a new USS table, USSDYN, with a different text for MSG0 and MSG10. Subsequently, we used Dynamic Table Replacement to allow LU X040A03 to use the new USS table. These are the steps we followed:

1. We copied the existing USS table and changed the text for MSG0 and MSG10 thus:

   ```
   MESSAGES USSMSG  MSG0,TEXT='% PATIENCE, PLEASE'
       ...
            USSMSG  MSG10,TEXT='(79C'*',X'15',CL19'*',CL59'HELLO @@LUNAME-
                   YOU ARE ONLINE TO MVSNM1,C'*',X'15',79C'*'),OPT=NOBLKS-
                  UP
   ```

2. We changed the name of the USS table to "USSDYN" by coding:

   ```
   USSDYN    USSTAB  TABLE=UPCASE
   ```

   as the first statement in the table.

3. We assembled and link-edited the table into SYS1.VTAMLIB, calling it USSDYN.

4. To change the USS table used by the LU, X040A03, to the new table, USSDYN, we issued the command:

   ```
   F CSSVTAM,TABLE,ID=X040A03,TYPE=USSTAB,
   OLDTAB=USS3270A,NEWTAB=USSDYN,OPTION=ASSOCIATE
   ```

The effect of this command is to load the table, USSDYN, into memory and change the table association of LU X040A03 to now use USSDYN as its session-level USS table. Upon the next logon, the new table is used. After the end user keys in an acceptable sequence of characters to logon to a specific application, the new MSG0 is displayed:

```
PATIENCE, PLEASE
```

The new MSG10 is not automatically displayed when the table is changed, but is displayed when the terminal is reactivated or after a logoff. The MSG10 that is displayed after the change to USSDYN is:

```
************************************************************************
*                 HELLO X040A03 YOU ARE ONLINE TO MVSNM1              *
************************************************************************
```

Figure 1-5 illustrates dynamically changing the session-level USS table for LU X040A03 from USS3270A to USSDYN and the messages received as a result.

```
F CSSVTAM,TABLE,NEWTAB=USSDYN,TYPE=USSTAB,OPTION=ASSOCIATE,ID=X040A03,
OLDTAB=USS3270A
IST097I MODIFY ACCEPTED
IST865I MODIFY TABLE COMMAND COMPLETE- 1 ASSOCIATION(S) CHANGED
IST864I NEWTAB=USSDYN, OLDTAB=USS3270A, OPT=ASSOCIATE, TYPE=USSTAB
```

Figure 1-5. Console Listing--Dynamic Change of USS Table for an LU

In the case above, we issued the MODIFY TABLE command so that only one LU would use the new USS table, USSDYN. Figure 1-6 on page 1-12 shows an example of loading a new USS table and having **ALL** resources which were using the old table, USS3270A, use the new table, USSDYN. LU X000C02 is one of the resources using the new table after the command is executed.

```
F CSSVTAM,TABLE,NEWTAB=USSDYN,OPTION=LOAD,OLDTAB=USS3270A
IST097I MODIFY ACCEPTED
IST865I MODIFY TABLE COMMAND COMPLETE-TABLE USSDYN LOADED
IST864I NEWTAB=USSDYN, OLDTAB=USS3270A, OPT=LOAD, TYPE=**NA**
D NET,ID=X000C02
IST097I DISPLAY ACCEPTED
IST075I NAME = X000C02, TYPE = LOGICAL UNIT
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST861I MODETAB=AMODETAB USSTAB=USSDYN LOGTAB=LOGONTAB
IST934I DLOGMOD=M3276
IST597I CAPABILITY-PLU INHIBITED,SLU ENABLED  ,SESSION LIMIT 00000001
IST081I LINE NAME = L000, LINE GROUP = G3276, MAJNOD = FRACKX2
IST135I PHYSICAL UNIT = P000C
IST082I DEVTYPE = LU
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000000, SESSION REQUESTS = 0000000000
```

Figure 1-6. Console Listing--Replacing a USS Table with A New Table

To replace a USS table with a refreshed copy of the same table, the command in the following form should be issued:

F CSSVTAM,TABLE,NEWTAB=USS3270A,OLDTAB=USS3270A,OPTION=LOAD

Figure 1-7 is an example of replacing the USS table, USS3270A, with a refreshed copy:

```
F CSSVTAM,TABLE,NEWTAB=USS3270A,OLDTAB=USS3270A,OPTION=LOAD
IST097I MODIFY ACCEPTED
IST865I MODIFY TABLE COMMAND COMPLETE-TABLE USS3270A LOADED
IST864I NEWTAB=USS3270A, OLDTAB=USS3270A, OPT=LOAD, TYPE=**NA**
```

Figure 1-7. Console Listing--Replacing a USS Table with A New Copy

Figure 1-8 on page 1-13 is an example of deleting the association between the PU, P040A (and its 4 LUs), and the USS Table, USSDYN. Following this command, in order to logon from one of the LUs the SYSREQ key needed to be used and the logon had to be keyed in long form. Additionally, no USS messages were displayed on the screens.

```
F CSSVTAM,TABLE,TYPE=USSTAB,OPTION=DELETE,ID=P040A,OLDTAB=USSDYN
IST097I MODIFY ACCEPTED
IST865I MODIFY TABLE COMMAND COMPLETE- 4 ASSOCIATION(S) DELETED
IST864I NEWTAB=***NA***, OLDTAB=USSDYN, OPT=DELETE, TYPE=USSTAB
D NET,ID=X040A03
IST097I DISPLAY ACCEPTED
IST075I NAME = X040A03, TYPE = LOGICAL UNIT
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST861I MODETAB=AMODETAB USSTAB=***NA*** LOGTAB=LOGONTAB
IST934I DLOGMOD=M2SDLCNQ
IST597I CAPABILITY-PLU INHIBITED,SLU ENABLED  ,SESSION LIMIT 00000001
IST081I LINE NAME = L040, LINE GROUP = G3174, MAJNOD = FRACKX2
IST135I PHYSICAL UNIT = P040A
IST082I DEVTYPE = LU
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000000, SESSION REQUESTS = 0000000000
IST314I END
```

Figure 1-8. Console Listing--Deleting a USS Table

1.5 Logon Mode Tables

1.5.1 LOGMODE Tables--General Description

When a logical unit requests a session with an application program, it uses a symbolic logon mode name, either directly or by default, to suggest the session protocols. Session protocols are expressed as a string of parameters, usually specified in the logon mode table. These parameters specify things such as whether the session will use chaining, what RU size to use, etc.. The session protocols are sent from the VTAM of the secondary logical unit (SLU) to the primary logical unit (PLU) during the session initiation. The PLU can choose to use these parameters, override some or all of them, or not to BIND the session.

IBM supplies a Logon Mode table, ISTINCLM, that contains a set of generally accepted session parameters for most IBM device types. However, installations ordinarily customize unique Logon Mode tables and supply them as supplementary tables. Logical units are associated with specific Logon Mode tables through use of MODETAB operands in VTAM or NCP definition statements for the resources.

1.5.2 Using Dynamic Table Replacement with Logon Mode Tables

Logon Mode tables can be added, replaced, or deleted through use of VTAM V3R2's Dynamic Table Replacement function. When VTAM is started the resources are associated with Logon Mode tables through the coding of the MODETAB operand in the VTAM or NCP definition statements. Below is an example of the coding in our lab's NCP for a 3290 (the four LUs listed) attached to a 3174 (the PU, P040A). These LUs are defined to use the logon mode table called "AMODETAB":

```
G3174      GROUP  PUTYPE=2,MODETAB=AMODETAB,USSTAB=USS3270A,LOGTAB=LOGONTAB
L040       LINE   ADDRESS=(040,FULL),ISTATUS=ACTIVE
           SERVICE ORDER=(P040A)
P040A      PU  ADDR=C1
X040A02    LU  LOCADDR=02,DLOGMOD=M2SDLCNQ
X040A03    LU  LOCADDR=03,DLOGMOD=M2SDLCNQ
X040A04    LU  LOCADDR=04,DLOGMOD=M2SDLCNQ
X040A05    LU  LOCADDR=05,DLOGMOD=M2SDLCNQ
```

This particular table, AMODETAB, contains a number of different logmode entries. The LUs on the 3290 are using a default logmode entry of M2SDLCNQ. (Specified by the DLOGMOD= operand on the LU statements.)

We created a new Logon Mode table in order to specify a new COS entry. The new COS entry was also added to the COS table which was dynamically replaced (see the following section on COS Tables). After creating the new Logon Mode table, we used Dynamic Table Replacement.

These are the steps we followed:

1. We copied the existing logon mode table, AMODETAB, and added the COS name thus:

```
M2SDLCNQ MODEENT LOGMODE=M2SDLCNQ,FMPROF=X'03',TSPROF=X'03',          X
               PRIPROT=X'B1',SECPROT=X'90',COMPROT=X'3080',           X
               RUSIZES=X'8587',PSERVIC=X'020000000000185000007E00',   X
               COS=DYN
```

2. We changed the name of the logon mode table to "MODEDYN" by coding:

   ```
   MODEDYN   MODETAB
   ```

   as the first statement in the table.

3. We assembled and link-edited the table into SYS1.VTAMLIB, calling it MODEDYN.

4. To change the logon mode table used by the LU, X040A03, to the new table, MODEDYN, we issued the command:

   ```
   F CSSVTAM,TABLE,ID=X040A03,TYPE=MODETAB,
   OLDTAB=AMODETAB,NEWTAB=MODEDYN,OPTION=ASSOCIATE
   ```

The effect of this command is to load the table, MODEDYN, into memory and change the association of the LU, X040A03, to now use MODEDYN as its Logon Mode table. At the next logon LU X040A03 will use the default logmode entry, M2SDLCNQ (unless overridden by the end user as part of a USS logon, or by the USS table default logmode entry), from the new table, MODEDYN. Part of the parameters specified in this entry are the new COS name, DYN.

NOTE: In a cross-domain or cross-network environment the COS name as specified in the Logon Mode table entry of the SLU is sent to the VTAM of the PLU. The COS name must be present in the COS table at the VTAM that owns the PLU.

Below is a listing of the console showing the command entered and messages received as a result.

```
F CSSVTAM,TABLE,TYPE=MODETAB,NEWTAB=MODEDYN,OLDTAB=AMODETAB,OPTION=ASS
OCIATE,ID=X040A03
IST097I MODIFY ACCEPTED
IST865I MODIFY TABLE COMMAND COMPLETE- 1 ASSOCIATION(S) CHANGED
IST864I NEWTAB=MODEDYN, OLDTAB=AMODETAB, OPT=ASSOCIATE, TYPE=MODETAB
D NET,ID=X040A03
IST097I DISPLAY ACCEPTED
IST075I NAME = X040A03, TYPE = LOGICAL UNIT
IST486I CURRENT STATE = ACT/S, DESIRED STATE = ACTIV
IST861I MODETAB=MODEDYN USSTAB=USSDYN LOGTAB=***NA***
IST934I DLOGMOD=M2SDLCNQ
IST597I CAPABILITY-PLU INHIBITED,SLU ENABLED  ,SESSION LIMIT 00000001
IST081I LINE NAME = L040, LINE GROUP = G3174, MAJNOD = FRACKX2
IST135I PHYSICAL UNIT = P040A
IST082I DEVTYPE = LU
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000001, SESSION REQUESTS = 0000000000
IST314I END
```

Figure 1-9. Console Listing--Dynamic Change of Logon Mode Table for an LU

In the case above, we issued the MODIFY TABLE command so that only one LU would use the new Logon Mode table, MODEDYN. In order to load a new Logon Mode table and have **ALL** resources which were using the old table, AMODETAB, use the new table, MODEDYN, we entered the commands as shown in Figure 1-10.

```
F CSSVTAM,TABLE,NEWTAB=MODEDYN,OPTION=LOAD,OLDTAB=AMODETAB
IST097I MODIFY ACCEPTED
IST865I MODIFY TABLE COMMAND COMPLETE-TABLE MODEDYN LOADED
IST864I NEWTAB=MODEDYN, OLDTAB=AMODETAB, OPT=LOAD, TYPE=**NA**
D NET,ID=X000C02
IST097I DISPLAY ACCEPTED
IST075I NAME = X000C02, TYPE = LOGICAL UNIT
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST861I MODETAB=MODEDYN USSTAB=USSDYN LOGTAB=INTDYN
IST934I DLOGMOD=M3276
IST597I CAPABILITY-PLU INHIBITED,SLU ENABLED  ,SESSION LIMIT 00000001
IST081I LINE NAME = L000, LINE GROUP = G3276, MAJNOD = FRACKX2
IST135I PHYSICAL UNIT = P000C
IST082I DEVTYPE = LU
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000000, SESSION REQUESTS = 0000000000
```

Figure 1-10. Console Listing--Replacing a Logon Mode Table with A New Table

It is also possible to replace AMODETAB with a refreshed copy by issuing the command in the form:

F CSSVTAM,TABLE,NEWTAB=AMODETAB,OLDTAB=AMODETAB,OPTION=LOAD

```
F CSSVTAM,TABLE,NEWTAB=AMODETAB,OLDTAB=AMODETAB,OPTION=LOAD
IST097I MODIFY ACCEPTED
IST865I MODIFY TABLE COMMAND COMPLETE-TABLE AMODETAB LOADED
IST864I NEWTAB=AMODETAB, OLDTAB=AMODETAB, OPT=LOAD, TYPE=**NA**
```

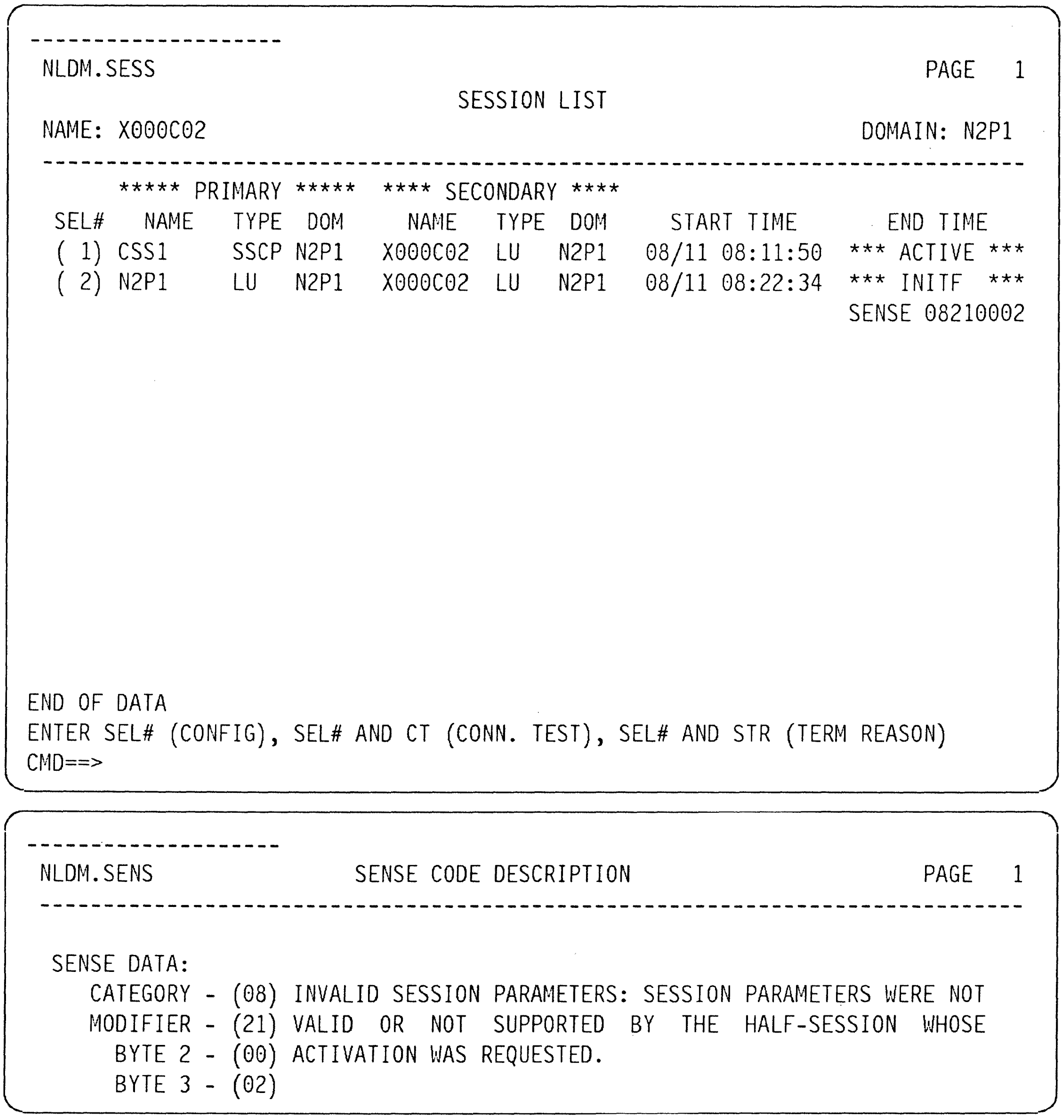
Figure 1-11. Console Listing--Replacing a Logon Mode Table with A New Copy

It is also possible to delete the association between a resource and a logon mode table. Figure 1-12 on page 1-16 demonstrates displaying the LU X000C02 which is associated with the Logon Mode table, AMODETAB. The first attempt to delete the association of LU X000C02 with AMODETAB fails because the required parameter, OLDTAB, is omitted. After the successful Logon Mode table deletion, a subsequent display of the LU shows that there is no Logon Mode table associated with it.

```
D NET,ID=X000C02
IST097I DISPLAY ACCEPTED
IST075I NAME = X000C02, TYPE = LOGICAL UNIT
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST861I MODETAB=AMODETAB USSTAB=USS3270A LOGTAB=LOGONTAB
IST934I DLOGMOD=M3276
IST597I CAPABILITY-PLU INHIBITED,SLU ENABLED  ,SESSION LIMIT 00000001
IST081I LINE NAME = L000, LINE GROUP = G3276, MAJNOD = FRACKX2
IST135I PHYSICAL UNIT = P000C
IST082I DEVTYPE = LU
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000000, SESSION REQUESTS = 0000000000
IST314I END
F CSSVTAM,TABLE,ID=X000C02,TYPE=MODETAB,OPTION=DELETE
IST456I OLDTAB REQUIRED PARAMETER OMITTED
F CSSVTAM,TABLE,ID=X000C02,TYPE=MODETAB,OPTION=DELETE,OLDTAB=AMODETAB
IST097I MODIFY ACCEPTED
IST865I MODIFY TABLE COMMAND COMPLETE- 1 ASSOCIATION(S) DELETED
IST864I NEWTAB=***NA***, OLDTAB=AMODETAB, OPT=DELETE, TYPE=MODETAB
D NET,ID=X000C02
IST097I DISPLAY ACCEPTED
IST075I NAME = X000C02, TYPE = LOGICAL UNIT
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST861I MODETAB=***NA*** USSTAB=USS3270A LOGTAB=LOGONTAB
IST934I DLOGMOD=M3276
IST597I CAPABILITY-PLU INHIBITED,SLU ENABLED  ,SESSION LIMIT 00000001
IST081I LINE NAME = L000, LINE GROUP = G3276, MAJNOD = FRACKX2
IST135I PHYSICAL UNIT = P000C
IST082I DEVTYPE = LU
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST171I ACTIVE SESSIONS = 0000000000, SESSION REQUESTS = 0000000000
IST314I END
```

Figure 1-12. Console Listing--Deleting a Logon Mode Table

After we deleted the association between X000C02 and AMODETAB, we were unable to logon because the terminal's default logmode entry, M2SDLCNQ, does not exist in the VTAM default Logon Mode table. We received the message at the terminal "LOGMODE PARAMETER INVALID." Figure 1-13 on page 1-17 shows the NetView Session List screen, indicating the session initiation failure with sense 08210002, and the NetView Sense Code Description screen, indicating sense code 08210002 is issued when session parameters are invalid.



```
--------------------
NLDM.SESS                                                               PAGE   1
                                   SESSION LIST
NAME: X000C02                                                         DOMAIN: N2P1
--------------------------------------------------------------------------------
      ***** PRIMARY *****   **** SECONDARY ****
 SEL#   NAME    TYPE  DOM     NAME    TYPE  DOM      START TIME        END TIME
 ( 1) CSS1      SSCP N2P1   X000C02  LU    N2P1   08/11 08:11:50  *** ACTIVE ***
 ( 2) N2P1      LU   N2P1   X000C02  LU    N2P1   08/11 08:22:34  *** INITF  ***
                                                                  SENSE 08210002










END OF DATA
ENTER SEL# (CONFIG), SEL# AND CT (CONN. TEST), SEL# AND STR (TERM REASON)
CMD==>
```

```
--------------------
NLDM.SENS                  SENSE CODE DESCRIPTION                       PAGE   1
--------------------------------------------------------------------------------

 SENSE DATA:
    CATEGORY - (08) INVALID SESSION PARAMETERS: SESSION PARAMETERS WERE NOT
    MODIFIER - (21) VALID  OR  NOT  SUPPORTED  BY  THE  HALF-SESSION  WHOSE
      BYTE 2 - (00) ACTIVATION WAS REQUESTED.
      BYTE 3 - (02)
```

Figure 1-13. NetView Screens--Effect of Logon Mode Table Deletion

1.6 Class of Service Tables

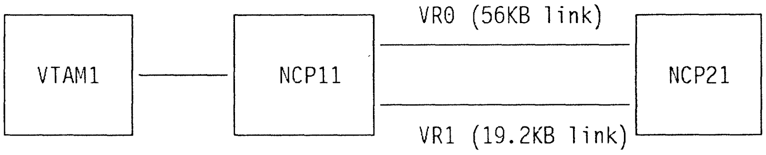
1.6.1 COS Tables--General Description

Classes of service are defined in a table called the Class of Service (COS) table. A class of service is selected for an LU-to-LU session through the COS name specified in the logon mode table entry associated with the secondary logical unit (SLU). The COS name is resolved to a virtual route list in the domain of the primary logical unit (PLU). Having multiple routes between subareas allows installations to associate the level of service required by a session with the appropriate virtual route.

In order to create multiple routes, a Class of Service table can be established to represent each of the levels of service needed for sessions in the network.

Installations can customize COS tables. ISTSDCOS is the name of the COS table which VTAM uses when establishing routes. If no ISTSDCOS is included in SYS1.VTAMLIB, then a default COS selection algorithm is used. The Network Program Products Planning manual contains more information about the default algorithm. In environments in which NCP, rather than VTAM, activates routes (such as SNI), more than one COS table may exist. See Section 1.6.5 of this Chapter for more information.

In the example below, ISTSDCOS, the required name for VTAM1's COS table, has four entries. The first entry, INTERACT, specifies that for any LU-to-LU session established using a COS name of INTERACT, that virtual route #0 with a transmission priority of 2 (the highest) is to be used. If virtual route #0 is not available, then virtual route #1 with a transmission priority of 2 is to be used. The second entry, BATCH, specifies that for any LU-to-LU session established using a COS name of BATCH, virtual route #1 with a transmission priority of 0 (the lowest) is to be used. If virtual route #1 isn't available, then virtual route #0 with a transmission priority of 0 is to be used. The ISTVTCOS entry is used for SSCP session requests and the blank entry used for LU-to-LU session requests specifying no COS name (or for SSCP session requests if there is no ISTVTCOS entry).



COS Table in VTAM1's SYS1.VTAMLIB:
ISTSDCOS

```
ISTSDCOS  COSTAB
INTERACT  COS      VR=(0,2),(1,2)
BATCH     COS      VR=(1,0),(0,0)
ISTVTCOS  COS      VR=(0,2),(1,2)
          COS      VR=(1,0),(0,0)
          COSEND
```

Figure 1-14. COS Table Example

Through coding of PATH statements, (see the Dynamic Path Update section of this manual for a description of PATH statements), VR#0 has been defined as a route which uses the 56KB link and VR#1 as a route using the 19.2KB link. When an interactive session (using a COS name of INTERACT as specified in its logon mode table entry) is established between resources located at NCP21 and VTAM1, it will use the higher speed link, if available. If the higher speed link is not available, the interactive session will use the lower speed link, but at a higher transmission priority than the batch traffic. NCP puts the higher transmission priority PIUs ahead of the lower transmission priority PIUs on the link between NCPs. Batch sessions, using a COS name of BATCH, will use the lower speed link unless it is not available. If the higher speed link is used, the batch traffic will have a lower transmission priority than the interactive traffic.

1.6.2 Using Dynamic Table Replacement with COS Tables

COS tables can be added, replaced, or deleted through use of VTAM V3R2's Dynamic Table Replacement function. When VTAM is started, if ISTSDCOS has been provided in VTAMLIB, ISTSDCOS is the table used for session establishment in VTAM's own network.

For an example, we created a new COS entry named "DYN" and added it to our existing COS table, ISTSDCOS. We then replaced ISTSDCOS using Dynamic Table Replacement. Recall that DYN is a COS name that is pointed to from a logon mode table entry that we added in the previous example of using Dynamic Table Replacement with Logon Mode Tables. In our example, we coded the COS entry for DYN so that it would specify virtual route number 7.

These are the steps we followed:

1. We copied the existing COS table, ISTSDCOS, and added the COS entry thus:

   ```
   DYN       COS    VR=(7,0)
   ```

2. We assembled and link-edited the table into SYS1.VTAMLIB

3. To have the COS table with the new entry, DYN, loaded and used we issued the command:

   ```
   F CSSVTAM,TABLE,NEWTAB=ISTSDCOS,OPTION=LOAD
   ```

The effect of this command is to load the table, ISTSDCOS, into memory. Any sessions subsequently established would use the entries as coded in the new copy of ISTSDCOS.

Below is a listing of the console showing the command entered and messages received as a result.

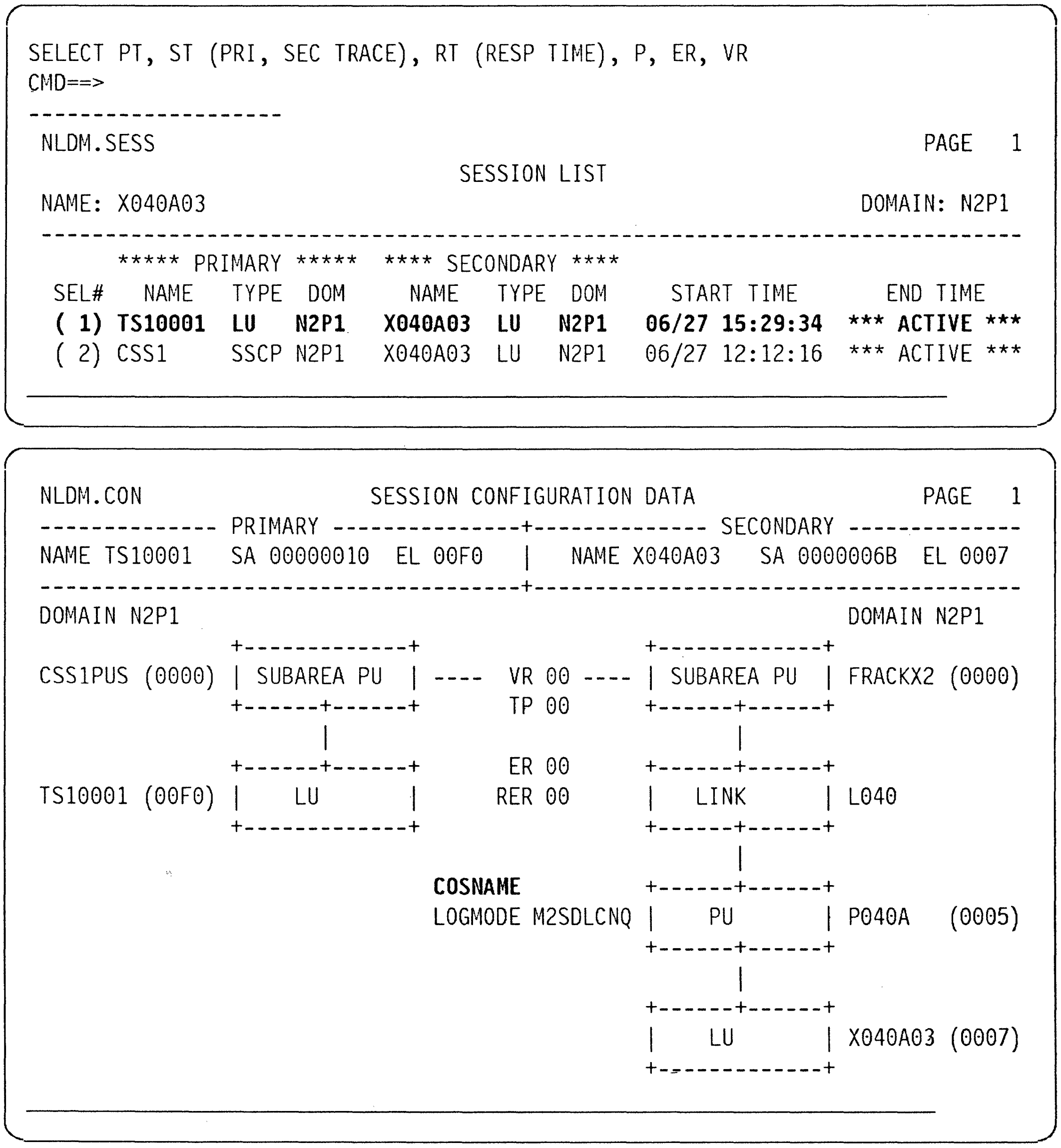
```
F CSSVTAM,TABLE,NEWTAB=ISTSDCOS,OPTION=LOAD
IST097I MODIFY ACCEPTED
IST865I MODIFY TABLE COMMAND COMPLETE- TABLE ISTSDCOS LOADED
IST864I NEWTAB=ISTSDCOS, OLDTAB=ISTSDCOS, OPT=LOAD, TYPE=**NA**
```

Figure 1-15. Console Listing--Dynamic Change of a COS Table

In our exercise, we now had a COS entry for sessions established using LU X040A03 (remember in the Logon Mode Table section that we had dynamically changed the logon mode table for this LU so that now a COS entry of DYN would be used). However, on our system, we had no route specified as VR#7. We used Dynamic Path Update to add this route to VTAM and NCP--see the Dynamic Path Update section of this manual for information.

1.6.3 Using NetView for Table Information

We attempted to logon to TSO from the terminal X040A03 before and after replacing the Logon Mode table for the terminal and the COS table, ISTSDCOS. The NetView screens shown in Figure 1-16 on page 1-21 illustrate the successful session between X040A03 and TSO before the tables were changed. Note that the terminal is using the blank COS entry (shown by the blank next to COSNAME on the Session Configuration Data screen).



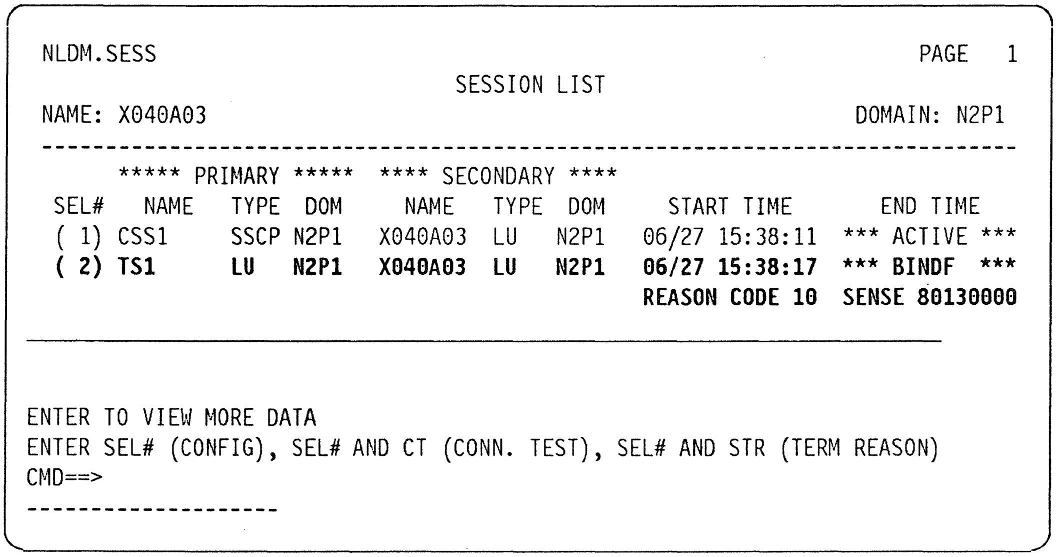
```
SELECT PT, ST (PRI, SEC TRACE), RT (RESP TIME), P, ER, VR
CMD==>
--------------------
 NLDM.SESS                                                          PAGE   1
                                       SESSION LIST
 NAME: X040A03                                                  DOMAIN: N2P1
 ----------------------------------------------------------------------------
        ***** PRIMARY *****  **** SECONDARY ****
  SEL#   NAME   TYPE  DOM     NAME   TYPE  DOM     START TIME        END TIME
  ( 1) TS10001  LU    N2P1   X040A03  LU   N2P1   06/27 15:29:34  *** ACTIVE ***
  ( 2) CSS1     SSCP  N2P1   X040A03  LU   N2P1   06/27 12:12:16  *** ACTIVE ***
```

```
 NLDM.CON                   SESSION CONFIGURATION DATA                  PAGE   1
 -------------- PRIMARY ---------------+------------- SECONDARY -------------
 NAME TS10001  SA 00000010  EL 00F0   |   NAME X040A03   SA 0000006B  EL 0007
 --------------------------------------+--------------------------------------
 DOMAIN N2P1                                                       DOMAIN N2P1
               +-------------+                    +-------------+
 CSS1PUS (0000) | SUBAREA PU  | ----  VR 00 ---- | SUBAREA PU  | FRACKX2 (0000)
               +------+------+        TP 00       +------+------+
                      |                                  |
               +------+------+        ER 00       +------+------+
 TS10001 (00F0) |    LU       |       RER 00       |    LINK     | L040
               +-------------+                    +------+------+
                                                         |
                               COSNAME            +------+------+
                               LOGMODE M2SDLCNQ   |     PU      | P040A   (0005)
                                                  +------+------+
                                                         |
                                                  +------+------+
                                                  |     LU      | X040A03 (0007)
                                                  +-------------+
```

Figure 1-16. NetView's Session Information Before Change in Logon Mode & COS Tables

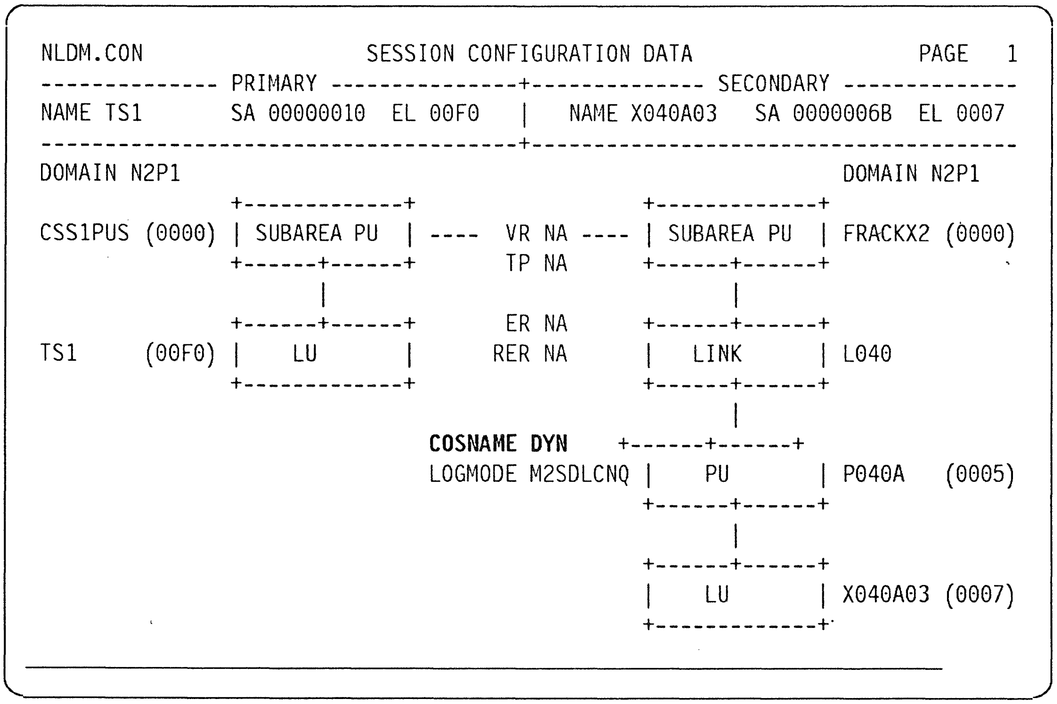
Figure 1-17 on page 1-22 is an example of NetView's Session Information after an attempt was made to logon to TSO from the terminal, X040A03, after we had changed the Logon Mode table and COS table. Although the COS entry, DYN, had been added, it pointed to a virtual route, VR#7, which had not been defined. We subsequently used Dynamic Path Update to add the route. The NLDM Sense

Code information shown in Figure 1-17 on page 1-22 details why the logon attempt failed.

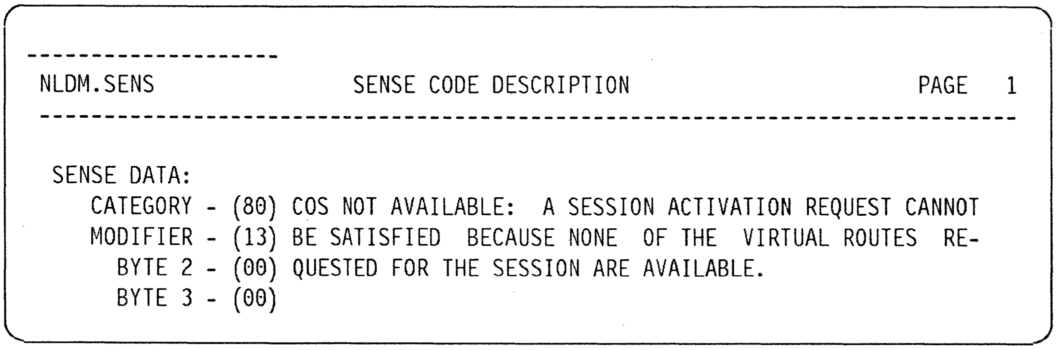


```
NLDM.SESS                                                       PAGE    1
                                SESSION LIST
NAME: X040A03                                                DOMAIN: N2P1
----------------------------------------------------------------------------
      ***** PRIMARY *****  **** SECONDARY ****
 SEL#   NAME   TYPE  DOM     NAME   TYPE  DOM     START TIME      END TIME
 ( 1) CSS1     SSCP N2P1   X040A03  LU    N2P1   06/27 15:38:11  *** ACTIVE ***
 ( 2) TS1      LU   N2P1   X040A03  LU    N2P1   06/27 15:38:17  *** BINDF  ***
                                                 REASON CODE 10  SENSE 80130000


ENTER TO VIEW MORE DATA
ENTER SEL# (CONFIG), SEL# AND CT (CONN. TEST), SEL# AND STR (TERM REASON)
CMD==>
--------------------
```



```
NLDM.CON                 SESSION CONFIGURATION DATA                  PAGE    1
-------------- PRIMARY ---------------+-------------- SECONDARY --------------
NAME TS1       SA 00000010  EL 00F0   |   NAME X040A03   SA 0000006B  EL 0007
--------------------------------------+---------------------------------------
DOMAIN N2P1                                                     DOMAIN N2P1
               +-------------+                  +-------------+
CSS1PUS (0000) | SUBAREA PU  | ----  VR NA ---- | SUBAREA PU  | FRACKX2 (0000)
               +------+------+       TP NA      +------+------+
                      |                                |
               +------+------+       ER NA      +------+------+
TS1     (00F0) |    LU       |      RER NA      |   LINK      | L040
               +-------------+                  +------+------+
                                                       |
                               COSNAME DYN   +------+------+
                               LOGMODE M2SDLCNQ |    PU       | P040A   (0005)
                                                +------+------+
                                                       |
                                                +------+------+
                                                |     LU      | X040A03 (0007)
                                                +-------------+
```



```
--------------------
NLDM.SENS               SENSE CODE DESCRIPTION                     PAGE    1
---------------------------------------------------------------------------

 SENSE DATA:
    CATEGORY - (80) COS NOT AVAILABLE:  A SESSION ACTIVATION REQUEST CANNOT
    MODIFIER - (13) BE SATISFIED  BECAUSE NONE  OF THE  VIRTUAL ROUTES  RE-
      BYTE 2 - (00) QUESTED FOR THE SESSION ARE AVAILABLE.
      BYTE 3 - (00)
```

Figure 1-17. NetView's Session Information After Change in Logon Mode & COS Tables

1.6.4 Using the New Display COS Command

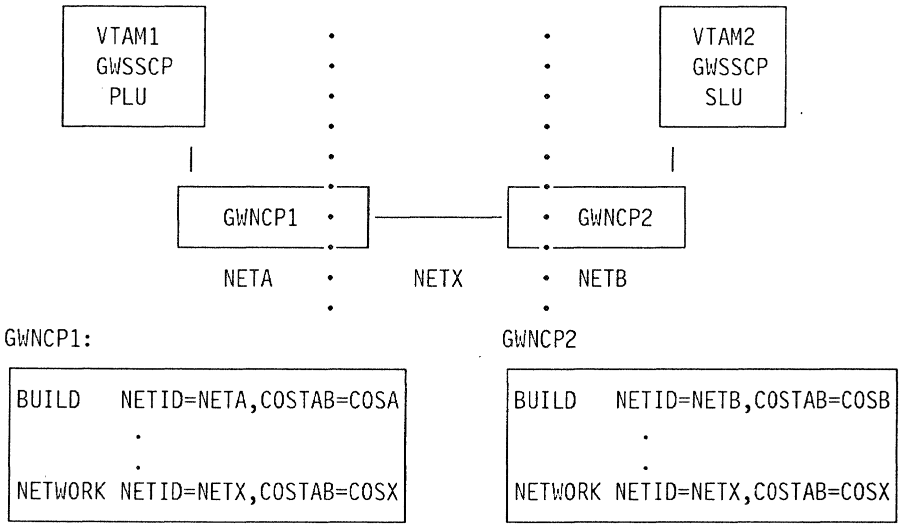
With VTAM V3R2, a new command, DISPLAY COS is provided. The command shows the name of the COS table currently in use from a particular origin VTAM or NCP. Below is an example of the DISPLAY COS command on our Lab system showing the COS table in use for both VTAM (CSS1PUS) and NCP (FRACKX2). The COS table shown is ISTSDCOS.

```
D NET,COS,ID=CSS1PUS,NETID=CSSNET
IST097I DISPLAY ACCEPTED
IST354I PU T4/5 MAJOR NODE = CSS1PUS
IST887I NO COS TABLE FOR CSSNET - ISTSDCOS MAY BE USED
IST314I END
D NET,COS,ID=FRACKX2,NETID=CSSNET
IST097I DISPLAY ACCEPTED
IST354I PU T4/5 MAJOR NODE = FRACKX2
IST887I NO COS TABLE FOR CSSNET - ISTSDCOS MAY BE USED
IST314I END
```

Figure 1-18. Console Listing--Example--DISPLAY COS Command

1.6.5 Using Dynamic Table Replacement with COS Tables in an SNI environment

In an SNI environment, more than one COS table can exist and be used within a single VTAM when it is also a gateway SSCP. ISTSDCOS is used by VTAM for routes originating in it. In other words, when VTAM is the owner of the primary logical unit, ISTSDCOS is used. ISTSDCOS can also be used in conjunction with gateway NCPs or different COS tables as named on NCP definition statements can be used.



GWNCP1:

```
BUILD   NETID=NETA,COSTAB=COSA
          .
          .
NETWORK NETID=NETX,COSTAB=COSX
```

GWNCP2

```
BUILD   NETID=NETB,COSTAB=COSB
          .
          .
NETWORK NETID=NETX,COSTAB=COSX
```

Figure 1-19. COS Tables in an SNI Environment

In the above example, VTAM1 has three COS tables defined in SYS1.VTAMLIB, ISTSDCOS, COSA, and COSX. VTAM2 also has three tables, ISTSDCOS, COSB, and COSX. For a session in which the primary logical unit is located at VTAM1, VTAM1 uses ISTSDCOS to select the list of virtual routes for use in NETA. If the secondary logical unit is located at VTAM2, then VTAM1, as the gateway SSCP controlling GWNCP1, uses the table COSX for selecting a list of virtual routes in NETX. VTAM2, the gateway SSCP controlling GWNCP2 uses the table COSB to select a list of virtual routes for use in NETB.

If the Primary Logical Unit were located at VTAM2, then VTAM2 would use ISTSDCOS to select the list of virtual routes for use in NETB, would use the table COSX for selecting a list of virtual routes in NETX, and VTAM1 would use the table COSA to select a list of virtual routes for use in NETA.

Dynamic Table Replacement can be used to change the COS tables in an SNI environment. ISTSDCOS can be changed by loading a new copy of the table as detailed in Section 1.6.2. The table, COSA, could be refreshed with a new table of the same name by entering the following command at VTAM1:

```
F CSSVTAM,TABLE,NEWTAB=COSA,OLDTAB=COSA,OPTION=LOAD
```

The table, COSA, could be replaced by a new table with a different name and the association of GWNCP1 changed to using the new table in place of COSA by issuing the following command at VTAM1:

```
F CSSVTAM,TABLE,NEWTAB=COSNEWA,OLDTAB=COSA,TYPE=COSTAB,
OPTION=ASSOCIATE,ORIGIN=GWNCP1,NETID=NETA
```

The table, COSX, could be refreshed with a new table of the same name by entering the following command at VTAM1:

```
F CSSVTAM,TABLE,NEWTAB=COSX,OLDTAB=COSX,OPTION=LOAD
```

The table, COSX,could be replaced by a new table with a different name and the association of GWNCP1 changed to using the new table in place of COSX by issuing the following command at VTAM1:

```
F CSSVTAM,TABLE,NEWTAB=COSNEWX,OLDTAB=COSX,TYPE=COSTAB,
OPTION=ASSOCIATE,ORIGIN=GWNCP1,NETID=NETX
```

1.6.6 Using the DISPLAY COS Command in an SNI Environment

The DISPLAY COS command could be used to display the COS tables in effect for the GWNCPs shown in Figure 1-19 on page 1-23. To display the COS table in effect with GWNCP1 as the origin in NETA, the VTAM1 operator would enter the command:

```
DISPLAY NET,COS,ID=GWNCP1,NETID=NETA
```

The resulting messages should indicate that COSA is the table name. (After the Dynamic Table Replacement of COSA with COSNEWA as explained in the previous section, the resulting messages would indicate that COSNEWA is the table name.) To display the COS table in effect with GWNCP1 as the origin in NETX, the VTAM1 operator would enter the command:

```
DISPLAY NET,COS,ID=GWNCP1,NETID=NETX
```

The resulting messages should indicate that COSX is the table name. (After the Dynamic Table Replacement of COSX with COSNEWX as explained in the previous section, the resulting messages would indicate that COSNEWX is the table name.)

1.7 Migration Considerations

Customers who plan to use Dynamic Table Replacement need to ensure that SYS1.VTAMLIB is allocated with enough primary space to allow for new members to be added to the library without its going into secondary extents. One problem we experienced was that if new tables were link-edited into SYS1.VTAMLIB and the new members were placed in secondary extents of the dataset that were unused at VTAM startup, the new tables could not be used with Dynamic Table Replacement. We received a message indicating a FETCH error when we tried to use these tables. The solution is to compress the dataset. We were able to compress the dataset without bringing VTAM down; however, in other than a lab environment, dataset compression with VTAM running is not a viable solution.

With VTAM V3R2, some changes have been made to VTAM tables in order to use them with the new function, Dynamic Table Replacement. **USS and INTERPRET tables need to be reassembled using VTAM V3R2 libraries before they can be dynamically added, replaced or deleted.** In some cases, once tables are assembled using VTAM V3R2 libraries, maintenance is required if the tables are then used on pre-V3R2 VTAM systems. Specifically, if LOGMODE or INTERPRET tables are assembled using VTAM V3R2 libraries, maintenance is required on VTAM V2R1, V2R2, V3R1, or V3R1.1 systems if the tables are distributed to these systems for use. With USS tables, VTAM V3R2 provides a FORMAT=V3R2|OLD parameter on the USSTAB Macro instruction. If OLD is specified, the USS table can be used on a pre-VTAM V3R2 system after assembly with V3R2 libraries; however, USS tables assembled with VTAM V3R2 libraries using FORMAT=OLD cannot be used with Dynamic Table Replacement on a VTAM V3R2 system. COS tables that are assembled with V3R2 libraries can be distributed to pre-V3R2 VTAM systems without any prerequisite maintenance.

The chart below summarizes the VTAM table considerations:

Table 1-1. VTAM V3R2 Tables

	USS	INTER-PRET	LOGMODE	COS
Table must be reassembled to use DTR	YES FORMAT = V3R2	YES	NO	NO
If DTR not used, pre-V3R2 assembled table will work with V3R2	YES	YES	YES	YES
V3R2 assembled table will work on pre-V3R2	YES FORMAT = OLD	PTF	PTF	YES

1.8 Backup and Recovery Considerations

Backup and Recovery needs to be considered when using Dynamic Table Replacement. If VTAM is restarted or if NCPs are lost and resources reactivated, the network will come back up using the original tables (not the dynamically changed ones) unless techniques are developed to ensure that the changes are made permanent or that the changes are reapplied.

One technique that could be used is:

1. Copy existing tables and rename them.
2. Make changes to these tables.
3. Test the tables using the new names with Dynamic Table Replacement.
4. Rename the new tables using the original table names.
5. Use Dynamic Table Replacement to associate the resources with the original named tables.
6. If VTAM should restart at this point the originally named tables will be used but the table information has been changed.

Another technique that could be used is:

1. Code new tables.
2. Test the tables using with Dynamic Table Replacement with a few resources.
3. Associate all system resources with appropriate new tables using Dynamic Table Replacement.
4. Change all USSTAB=, LOGTAB=, and MODETAB= parameters in all Major Node definitions to name the appropriate new tables.

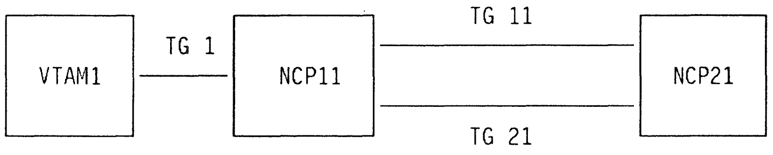
5. Change COSTAB= parameters on BUILD and NETWORK definition statements to name the appropriate new tables.

6. Since the above parameters are VTAM only parameters, the new tables will be used upon VTAM restart even if they are in NCP statements.

Chapter 2. Dynamic Path Update

2.1 VTAM and NCP Path Statements and Class of Service

In order for sessions to be established between resources that are located at different subareas (different VTAMs or NCPs), installations define explicit routes between the subareas and define virtual routes that are used to map session traffic to particular explicit routes. These routes are defined via PATH statements in VTAM and NCP. Sessions are mapped to a specific virtual route via a Class of Service table.



VTAM1's PATHs:

```
PATH  DESTSA=11,
      ER0=(11,1),
      VR0=0
PATH  DESTSA=21,
      ER0=(11,1),
      ER1=(11,1),
      VR0=0,
      VR1=1
```

NCP11's PATHs:

```
PATH  DESTSA=21,
      ER0=(21,11),
      ER1=(21,21)
PATH  DESTSA=1,
      ER0=(1,1),
      ER1=(1,1)
```

NCP21's PATHs:

```
PATH  DESTSA=1,
      ER0=(11,11),
      ER1=(11,21)
```

COS Table in VTAM1's SYS1.VTAMLIB:
ISTSDCOS

```
ISTSDCOS  COSTAB
INTERACT  COS       VR=(0,1),(1,1)
BATCH     COS       VR=(1,0),(0,0)
ISTVTCOS  COS       VR=(0,2),(1,2)
          COS       VR=(1,0),(0,0)
          COSEND
```

Figure 2-1. Example of Path Statements and COS Table

In Figure 2-1, all three subareas contain PATH definitions to the other subareas. VTAM1's PATH definitions define one explicit route to destination subarea 11 (DESTSA=11). ER0 is defined indicating to use adjacent subarea 11 (**11**,1) across TG1 (11,**1**) to reach destination subarea 11. Additionally, VTAM1 defines that for any session on virtual route 0, use explicit route 0 (VR0=0). VTAM1's PATH definitions to reach destination subarea 21 define two routes. To reach destination subarea 21, adjacent subarea 11 across the channel, TG1, should be used for both ER0 and ER1. Also virtual route 0 maps to explicit route 0 and virtual route 1 maps to explicit route 1.

NCP11's PATH statements define explicit route 0 to destination subarea 21 using TG11. TG11 in this example is a high speed link. NCP11's PATHs define explicit route 1 to destination subarea 21 using TG21, a lower speed link. NCP11 also defines explicit routes 0 and 1 to destination subarea 1 across the channel, TG1, to VTAM1.

NCP21's PATH statements define explicit route 0 to destination subarea 1 using TG11, the high speed link. NCP21's PATHs define explicit route 1 to destination subarea 1 using TG21, the lower speed link.

Note that the routes that are defined are reversible; that is, explicit route 0 between subarea 1 and 21 uses first TG1 and then TG11. There is a reverse explicit route, ER0 which traverses TG11 and TG1 from subarea 21 back to subarea 1. It is not necessary that the forward and reverse route use the same ER number, as in this example, but it is necessary that there be a reverse explicit route traversing exactly the same transmission groups in order for a session to be setup using that route.

Classes of service are defined in a table called the Class of Service (COS) table. A class of service is selected for an LU-to-LU session through the COS name specified in the Logon Mode Table entry associated with the session. The COS name is resolved to a virtual route list in the domain of the primary logical unit (PLU). Having multiple routes between subareas allows installations to associate the level of service required by a session with the appropriate virtual route. In order to create multiple routes, a class of service table can be established to represent each of the levels of service needed for sessions in the network.

In Figure 2-1 on page 2-3, ISTSDCOS, the required name for VTAM1's COS table, has four entries. The first entry, INTERACT, specifies that for any LU-to-LU session established using a COS name of INTERACT, that virtual route 0 with a transmission priority of 1 (medium) is to be used. If virtual route 0 is not available, then virtual route 1 with a transmission priority of 1 is to be used. The second entry specifies that for any LU-to-LU session established using a COS name of BATCH, virtual route 1 with a transmission priority of 0 (the lowest) is to be used. If virtual route 1 is not available, then virtual route 0 with a transmission priority of 0 is to be used. The ISTVTCOS entry is used for SSCP session requests (using transmission priority 2, the highest), and the blank entry used for LU-to-LU session requests specifying no COS name. It would also be used for SSCP session requests if there were no ISTVTCOS entry.

In Figure 2-1 on page 2-3, an interactive session involving a terminal attached to NCP21 and an application at VTAM1 could be setup using the higher speed link, TG11, by using the INTERACT COS table entry in VTAM1's COS table. Likewise, batch sessions could be separated from the interactive traffic and setup to use the lower speed link by using the BATCH COS entry. Installations can specify a COS table entry name to be used in the Logon Mode table entry used by a particular logical unit.

The Chapter on Dynamic Table Replacement in this manual describes how to change a Class of Service Table dynamically.

2.2 Dynamic Path Update

2.2.1 Overview

Prior to VTAM V3R2, VTAM PATHs can be added or changed by altering the PATH statements within an already active path definition member of the VTAM Definition Library (or new path definition member), and reactivating (or activating) that member. This can be accomplished as long as the PATHs are undefined or inoperative. With VTAM V3R2, in addition to the prior method, VTAM PATHs can be added, changed, or deleted through use of the new function, Dynamic Path Update. The PATHs are altered through activating a member of the VTAM Definition Library containing statements in the form:

```
VTAM1  VPATH   NETID=NETA
        PATH   DESTSA=21,
               ER2=(11,1),
               VR2=2
```

PATHs can be deleted by activating a member of the VTAM Definition Library containing statements in the form:

```
VTAM1  VPATH   NETID=NETA
        PATH   DESTSA=21,
               DELETER=ER1
```

When explicit routes are deleted, any virtual routes mapped to that explicit route are deleted also. In order to add, change or delete PATHs, the PATHs must be **undefined** or **inoperative**. If the ERs are not inoperative, they cannot be altered and an error message is issued. If a VR is inactive, it can be mapped to a different ER; however, this does not work if the VR is active.

With NCPs prior to NCP V4R3 or V5R2, changes in PATH statements within an NCP require a regeneration and reloading of that NCP into the communication controller. With NCP V4R3 and NCP V5R2, PATHs can be added, changed, or deleted dynamically through use of a new function, Dynamic Path Update. The PATHs are added or changed from the NCP's owning VTAM through activating a member of the VTAM Definition Library containing the new statements in the form:

```
NCP21  NCPPATH   NETID=NETA
          PATH   DESTSA=1,
                 ER2=(11,11)
```

PATHs can be deleted by activating a member of the VTAM Definition Library containing NCPPATH statements in the form:

```
NCP21  NCPPATH   NETID=NETA
          PATH   DESTSA=1,
                 DELETER=ER1
```

Virtual route definitions and virtual route pacing window sizes can be coded on the VPATH and NCPPATH PATH statements and transmission group flow-control thresholds can be coded on the NCPPATH PATH statements, if desired.

Upon activation of library members with NCPPATH statements, if VTAM owns the named NCP it sends the PATH updates to the NCP on the SSCP-to-NCP session. If VTAM does not have an SSCP-to-NCP session with the named NCP, the PATH statements are ignored.

It is necessary for the PATHs to be **undefined** or **inoperative** in order for them to be altered. Additionally, new parameters have been added to the NCP generation process to support Dynamic Path Update. These parameters are TGBEXTRA, PATHEXT, and a second operand on VRPOOL. These values must be considered during the NCP generation process to plan for using Dynamic Path Update. The parameters are discussed in Section 2.2.2.

The above Dynamic Path Update statements, if added to the example illustrated in Figure 2-1 on page 2-3, would add a new explicit route, ER2, and virtual route, VR2, between subareas 1 and 21. ER1/VR1 would not have been deleted because their status would not have been inoperative. Since the channel, TG1, is used for the SSCP-to-NCP session, it is operative and any VRs/ERs defined using that same TG would be operative, even if not in use for any session.

2.2.2 New/Changed NCP Parameters

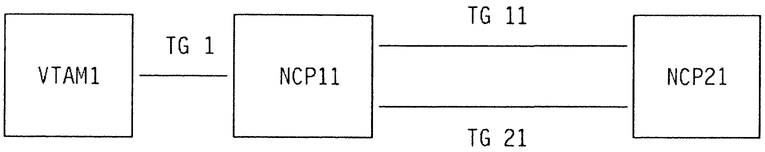
In order to support Dynamic Path Update, there are three new keywords on the BUILD definition statement that can be coded, **TGBXTRA, PATHEXT,** and **a second suboperand** on VRPOOL.

TGBXTRA specifies the number of **extra** transmission group control blocks (TGBs) that are needed within this NCP for PATHs added via Dynamic Path Update. One TGB is automatically generated for every unique adjacent subarea/transmission group number pair that is defined on NCP PATH definition statements. Additional TGBs are needed for PATHs that specify **new adjacent subarea/transmission group number pairs**.

PATHEXT specifies the number of **extra** transit routing table (TRT) rows that are required for PATHS added via Dynamic Path Update. A TRT row is used by NCP to index into the Subarea Vector Table to locate the address of the proper transmission group block for a particular PIU. A TRT row is generated for every destination subarea that is named on a PATH statement in NCP. An extra TRT row is needed for each PATH added via Dynamic Path Update that names a **new destination subarea.**

The **second suboperand on VRPOOL** specifies the number of extra flow control table (FCT) rows to generate. An FCT row holds the minimum and maximum virtual route pacing window sizes for a particular VR. An extra FCT row is needed for each VR that is added via Dynamic Path Update which has virtual route pacing window sizes coded.

NOTE: The defaults for these values may not be appropriate. TGBXTRA results in almost 600 bytes of storage per each extra TGB specified. The default is equal to the total number of subarea links and channels generated within this NCP. In some cases, taking the default value could result in a significant waste of NCP storage. Each PATHEXT coded results in a control block of 16 bytes--the default is a TRT with 254 rows if PATHEXT is not coded. Each second suboperand on VRPOOL results in a control block of 5 bytes--the default is equal to the first suboperand of VRPOOL if not coded.



VTAM1's PATHs:

```
PATH  DESTSA=11,
      ER0=(11,1),
      ER1=(11,1),
      VR0=0,
      VR1=1
PATH  DESTSA=21,
      ER0=(11,1),
      ER1=(11,1)
      VR0=0,
      VR1=1
```

NCP11's PATHs:

```
PATH  DESTSA=21,
      ER0=(21,11),
      ER1=(21,21)
PATH  DESTSA=1,
      ER0=(1,1),
      ER1=(1,1)
```

NCP21's PATHs:

```
PATH  DESTSA=11,
      ER0=(11,11),
      ER1=(11,21),
PATH  DESTSA=1,
      ER0=(11,11),
      ER1=(11,21)
```

VTAM1: Dynamic Path Update Member:

```
VTAM1   VPATH   NETID=NETA
        PATH    DESTSA=21,
                ER2=(11,1),
                VR2=2
NCP11   NCPPATH NETID=NETA
        PATH    DESTSA=2,
                ER0=(2,1)
        PATH    DESTSA=21,
                ER2=(21,21)
NCP21   NCPPATH NETID=NETA
        PATH    DESTSA=2,
                ER0=(11,21)
```

Figure 2-2. Dynamic Path Update Definitions

In Figure 2-2, a Dynamic Path Update member filed in VTAM1's Definition Library is shown. It contains these statements for VTAM1:

- A new ER, ER2, to reach NCP21.
- A new VR, VR2, mapped to ER2 to reach NCP21.

It contains these statements for NCP11:

- A new ER, ER0, to reach a new subarea, subarea 2. Subarea 2 is a test VTAM to be brought up later in place of VTAM1.
- A new ER, ER2, to reach NCP21.

It contains these statements for NCP21:

- A new ER, ER0, to reach a new subarea, subarea 2. Subarea 2 is a test VTAM to be brought up later in place of VTAM1.

After activation of this Dynamic Path Update member, sessions between resources at VTAM1 and NCP21 are able to use VR2/ER2. Additionally, when the test

VTAM, subarea 2, is brought up later, NCP11 and NCP21 will have a path, ER0, that could be used for sessions with that VTAM.

NCP11 Definition Requirements to Add PATHS:

In order for the PATH statement:

PATH DESTSA=2,ER0=(2,1)

to be added to NCP11, at least one TGBXTRA and one PATHEXT must have been included as part of NCP11's generation. Although the transmission group number, TG1, the channel, has already been defined in a generated PATH statement, the combination of the new adjacent subarea, 2, with this TG number requires that there be an extra TGB. At least one PATHEXT must have been generated to allow for an extra TRT row for the new destination subarea, 2.

An extra TGB or TRT row is **NOT** required in order for the PATH statement:

PATH DESTSA=21,ER2=(21,21)

to be added to NCP11. The TGBXTRA is not required for this PATH statement to take effect because the combination of TG21 and adjacent subarea 21 is not new--it is defined on one of the PATH statements that are part of the NCP11 generation. PATHEXT is not required because destination subarea 21 is already named on one of the PATH statements in NCP11's generation.

NCP21 Definition Requirements to Add PATHs:

NCP21 would not have had to be generated with any TGBXTRAs in order for the PATH statement:

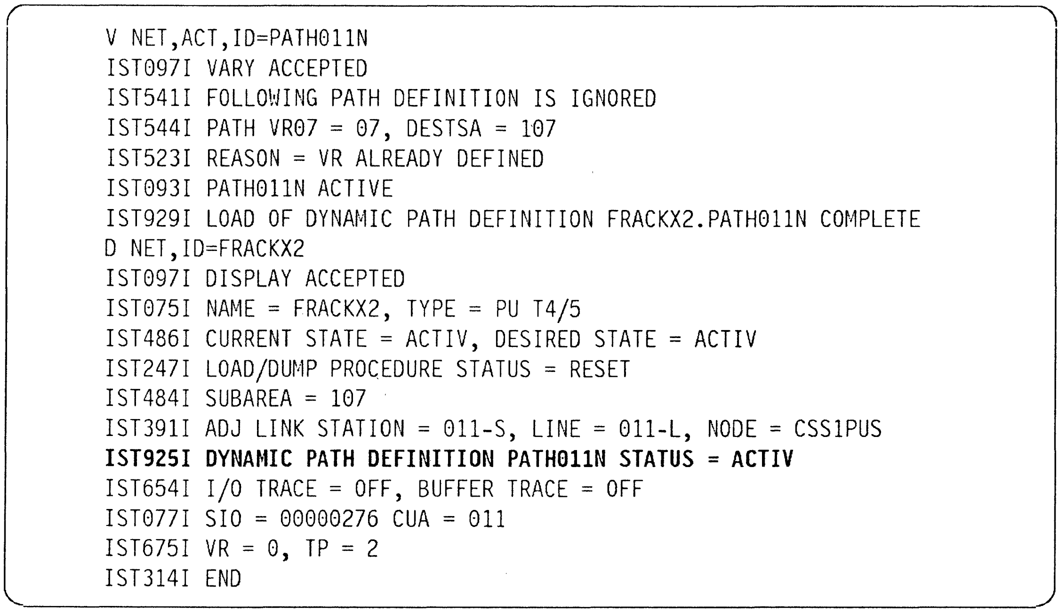
PATH DESTSA=2,ER2=(11,21)

to be added to NCP21. This is because the combination of adjacent subarea 11 and transmission group 21 exists on a PATH statement included as part of the NCP21 generation. Since destination subarea 2 is a new subarea, at lease one PATHEXT would have had to be generated for NCP21. A new virtual route, VR2, is being added with NCP21 as an endpoint. The first suboperand on VRPOOL must have been coded large enough to accomodate all virtual routes, including dynamically added ones, that will be concurrently active with this NCP as an endpoint.

The second suboperand on VRPOOL is not required for any of the PATHs shown in this example. An FCT row is required for any VR that is added to this NCP that has virtual route pacing window sizes coded. NCP PATH statements need to include VRs when the NCP is responsible for activating a virtual route for a session. This occurs in SNI environments, when the NCP activates VRs into another network or the same network, in LEN environments, where the two endpoints are not in a VTAM subarea, and in other environments such as XI.

2.2.3 Enhanced DISPLAY Command

The VTAM DISPLAY command is enhanced to show the name(s) of the Dynamic Path Update Member(s) that have been activated when either VTAM or NCP are DISPLAYed. Figure 2-3 illustrates activating the Dynamic Path Update Member PATH011N which contains PATH statements for the NCP named FRACKX2. A subsequent DISPLAY of FRACKX2 shows that the Dynamic Path Update Member, PATH011N, has been activated.

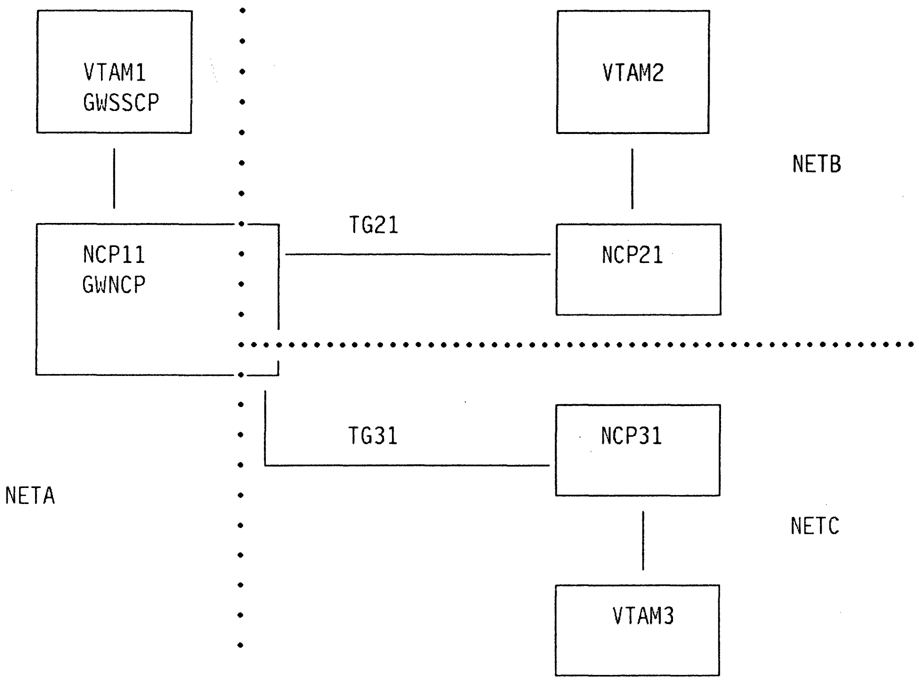


```
V NET,ACT,ID=PATH011N
IST097I VARY ACCEPTED
IST541I FOLLOWING PATH DEFINITION IS IGNORED
IST544I PATH VR07 = 07, DESTSA = 107
IST523I REASON = VR ALREADY DEFINED
IST093I PATH011N ACTIVE
IST929I LOAD OF DYNAMIC PATH DEFINITION FRACKX2.PATH011N COMPLETE
D NET,ID=FRACKX2
IST097I DISPLAY ACCEPTED
IST075I NAME = FRACKX2, TYPE = PU T4/5
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST247I LOAD/DUMP PROCEDURE STATUS = RESET
IST484I SUBAREA = 107
IST391I ADJ LINK STATION = 011-S, LINE = 011-L, NODE = CSS1PUS
IST925I DYNAMIC PATH DEFINITION PATH011N STATUS = ACTIV
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST077I SIO = 00000276 CUA = 011
IST675I VR = 0, TP = 2
IST314I END
```

Figure 2-3. Enhanced DISPLAY NCP Command Showing Dynamic Path Update Member

2.2.4 Dynamic Path Update in SNI Environments

In order to perform Dynamic Path Update, the VTAM from which the Dynamic Path Update member is activated must be an owner of the NCP(s) to which the NCPPATH statements are to be applied. If VTAM encounters NCPPATH statements for an NCP it doesn't own, a message is issued and those particular NCPPATH statements are ignored. If VTAM is an owner of a gateway-NCP, it is possible to dynamically apply NCPPATH statements for both the native and non-native networks.



VTAM1's PATHs:

```
PATH  DESTSA=11,
      ER0=(11,1),
      ER1=(11,1),
      VR0=0,
      VR1=1
```

NCP11's PATHs:

```
PATH  DESTSA=1,
      ER0=(1,1),
      ER1=(1,1)
        .
        .
NETWORK NETID=NETB
        .
PATH  DESTSA=21,
      ER0=(21,21),
      VR0=0
        .
        .
NETWORK NETID=NETC
        .
PATH  DESTSA=31,
      ER0=(31,31),
      VR0=0
```

VTAM1 Dynamic Path Update Member:

```
NCP11  NCPPATH  NETID=NETB
       PATH     DESTSA=21,
                ER1=(21,21),
                VR1=1
NCP11  NCPPATH  NETID=NETC
       PATH     DESTSA=31,
                ER1=(31,31),
                VR1=1
```

Figure 2-4. Example of Dynamic Path Update in an SNI Environment

Using Figure 2-4 as an example, VTAM1 is an owner of the gateway-NCP, NCP11. NCP11 has been generated with PATH statements to reach subarea 1, VTAM1 in the native network, NETA, as well as with PATH statements for ER0/VR0 to reach subarea 21 (NCP21) in NETB, and PATH statements for ER0/VR0 to reach subarea 31 (NCP31) in NETC. Dynamic Path Update can also be used from VTAM1 in order to add, replace, or delete NCP PATH statements for PATHs in the non-native network. Figure 2-4 on page 2-10 shows the Dynamic Path Update member that could be activated from VTAM1 for NCP11 in order to add ER1/VR1 to reach subarea 21 in NETB, and ER1/VR1 to reach subarea 31 in NETC.

NCP11 would not have needed to be generated with any extra TGBs (TGBXTRA) or TRT rows (PATHEXT) in order to have the PATHs shown in Figure 2-4 on page 2-10 applied, since neither PATH statement defines an additional adjacent subarea/transmission group pair to what has been included in NCP11's generation. **VRPOOL** needs to have been generated large enough to accomodate the number of virtual routes having endpoints in NCP11 (the first suboperand on VRPOOL) and enough Flow Control Table (FCT) rows to contain any virtual route pacing window sizes that have been coded on NCP VRs (the second suboperand on VRPOOL).

2.3 Four Ways to Activate Dynamic Path Update Members

Once Dynamic Path Update members have been coded and filed in the VTAM Definition Library, there are four different techniques that can be used to activate the members and have the PATHs applied to the specified subareas.

Technique 1--Vary Activate the Dynamic Path Update Member

A VARY **NET,ACT,ID = membername** command can be issued by the VTAM operator where membername equals the name of the VTAM Definition Library Dynamic Path Update member.

Technique 2--Vary Activate the NCP Specifying NEWPATH =

A **VARY NET,ACT,ID = ncpname,NEWPATH = membername** can be issued by the VTAM operator where membername equals the name of the VTAM Definition Library Dynamic Path Update member. Up to three membernames can be specified. The PATHS coded in the members are dynamically applied to the NCP as part of NCP activation.

Technique 3--Specify Member in VTAM's CONFIG list =

As part of VTAM startup, a CONFIG list can be used which specifies the names of major nodes to be activated. The name(s) of a Dynamic Path Update member(s) can be included in the CONFIG list and the specified PATHs will be dynamically applied during VTAM startup. Care should be exercised if the member(s) contains PATHs to be applied to NCP(s) to ensure that the NCP(s) are activated prior to the Dynamic Path Update members. If the NCP(s) is not activated prior to the Dynamic Update Member in the CONFIG list, an error message is issued at the VTAM operator console and the PATHs are not applied.

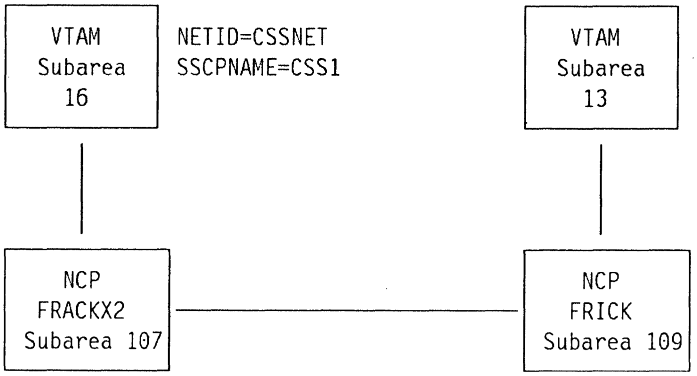
Technique 4--Specify NEWPATH = membername in NCP PCCU Statement

Up to three Dynamic Path Update members can be specified on the NEWPATH= operand of the NCP PCCU statement. The PATHs specified in these members pertaining to NCP's subarea are automatically applied during NCP activation. VPATH or NCPPATH statements pertaining to subareas other than NCP's are ignored.

2.4 Dynamic Path Update Example

In our lab, we used Dynamic Path Update to add PATHs to both VTAM and NCP. We let TGBXTRA, PATHEXT, and the second suboperand of VRPOOL default in our NCP generation for our first set of tests with Dynamic Path Update. The default values were large enough to allow us to add the PATHs that follow.

The configuration we used is shown in Figure 2-5 on page 2-13. In our lab's configuration, the VTAM, CSS1, is started with the PATH statements shown, allowing it to communicate with subareas 107, 109, and 13 using VR0/ER0. Likewise, the NCP, FRACKX2, is genned with the PATH statements shown, allowing it to communicate with subareas 109, 16, and 13 using ER0.



CSS1's PATH Statements:

```
PATH013  PATH  DESTSA=13,
               ER0=(107,1),
               VR0=0,
               VRPWS00=(10,255)
PATH107  PATH  DESTSA=107,
               ER0=(107,1),
               VR0=0,
               VRPWS00=(10,255)
PATH109  PATH  DESTSA=109,
               ER0=(107,1),
               VR0=0,
               VRPWS00=(10,255)
```

FRACKX2's PATH Statements:

```
PATHCSS1 PATH DESTSA=16,ER0=(16,1)
PATHCSS2 PATH DESTSA=13,ER0=(109,1)
PATH109  PATH DESTSA=109,ER0=(109,1)
```

CSS1 Dynamic Path Update Member PATH011D:

```
CSS1     VPATH NETID=CSSNET
         PATH  DESTSA=107,
               ER7=(107,1),
               VR7=7,
               VRPWS70=(10,255)
FRACKX2  NCPPATH NETID=CSSNET
         PATH  DESTSA=16,
               ER6=(16,1),
               ER7=(16,1)
```

CSS1 Dynamic Path Update Member PATH011N:

```
CSS1     VPATH NETID=CSSNET
         PATH  DESTSA=107,
               ER7=(107,1),
               VR7=7,
               VRPWS70=(10,255)
FRACKX2  NCPPATH NETID=CSSNET
         PATH  DESTSA=1,
         ER6=(1,1),
         ER7=1,1)
         PATH  DESTSA=2,
         ER6=(2,1),
         ER7=(2,1)
```

Figure 2-5. Lab Configuration for Dynamic Path Update

Figure 2-6 on page 2-14 shows the results of a DISPLAY ROUTE command illustrating that ER0 is defined from VTAM subarea 16 to NCP subarea 107.

VR0/transmission priority 2 is active. VR0/TP2 is used for the SSCP-to-NCP session between CSS1 and FRACKX2. ERs 1-7 are not currently defined.

NOTE: The results of the DISPLAY ROUTE command have been enhanced with VTAM V3R2. Message IST537I now displays the virtual route pacing window sizes for active routes.

```
D NET,ROUTE,DESTSUB=107
IST097I ROUTE ACCEPTED
IST535I ROUTE DISPLAY 1 FROM SA 16 TO SA 107
IST808I ORIGIN PU = CSS1PUS DEST PU = FRACKX2 NETID = CSSNET
IST536I VR  TP    STATUS   ER       ADJSUB  STATUS      CUR  MIN  MAX
IST537I  0   0    ACTIV     0          107  ACTIV3      11   10  255
IST537I  0   1    INACT     0          107  ACTIV3
IST537I  0   2    ACTIV     0          107  ACTIV3      10    1  223
IST537I                     1          107  PDEFO
IST537I                     2          107  PDEFO
IST537I                     3          107  PDEFO
IST537I                     4          107  PDEFO
IST537I                     5          107  PDEFO
IST537I                     6          107  PDEFO
IST537I                     7          107  PDEFO
IST314I END
```

Figure 2-6. DISPLAY ROUTE Subarea 16 to Subarea 107

Figure 2-7 on page 2-15 illustrates a DISPLAY ROUTE command issued at CSS1 showing the routes from FRACKX2 (Subarea 107) to CSS1 (Subarea 16). The results show that VR0/TP2 is active (it is being used for the SSCP-NCP session). ERs 1-7 are not defined (UNDEF).

NOTE: The new results of the DISPLAY ROUTE command do not show virtual route pacing window sizes for other subareas than the host at which the command is issued. In the following example, the window sizes are shown as 0, because the subarea displayed is the NCP rather than VTAM. In order to show the virtual route pacing window sizes for other subareas, new Request Units, which have not been architected at this time, would have been needed.

```
D NET,ROUTE,DESTSUB=16,ORIGIN=FRACKX2
IST097I ROUTE ACCEPTED
IST535I ROUTE DISPLAY 2 FROM SA 107 TO SA 16
IST808I ORIGIN PU = FRACKX2 DEST PU = CSS1PUS NETID = CSSNET
IST536I VR  TP    STATUS   ER       ADJSUB  STATUS      CUR  MIN  MAX
IST537I  0   0    ACTIV     0           16  ACTIV3        0    0    0
IST537I  0   1    INACT     0           16  ACTIV3
IST537I  0   2    ACTIV     0           16  ACTIV3        0    0    0
IST537I                     7               UNDEF
IST537I                     6               UNDEF
IST537I                     5               UNDEF
IST537I                     4               UNDEF
IST537I                     3               UNDEF
IST537I                     2               UNDEF
IST537I                     1               UNDEF
IST314I END
```

Figure 2-7. DISPLAY ROUTE Subarea 107 to Subarea 16

Next, the Dynamic Path Update member, PATH011D, shown in Figure 2-5 on page 2-13 is varied active with one change. The first time it was tried, FRACKX2 was mistakenly coded as FRACKX3. Note that VTAM, CSS1, ignores the statements pertaining to FRACKX3 in that it has no SSCP-to-NCP session with FRACKX3 but VTAM goes ahead and updates CSS1's PATHs. Following the activation of PATH011D, a DISPLAY ROUTE command for PATHs between subarea 16 and subarea 107 show that VR7 and ER7 have been added to CSS1's PATHs.

```
V NET,ACT,ID=PATH011D
IST097I VARY ACCEPTED
IST926I PATH FOR FRACKX3 IGNORED - NODE FRACKX3 NOT FOUND/INVALID
IST093I PATH011D ACTIVE
D NET,ROUTE,DESTSUB=107
IST097I ROUTE ACCEPTED
IST535I ROUTE DISPLAY 5 FROM SA 16 TO SA 107
IST808I ORIGIN PU = CSS1PUS DEST PU = FRACKX2 NETID = CSSNET
IST536I VR  TP    STATUS   ER       ADJSUB  STATUS      CUR  MIN  MAX
IST537I  0   0    ACTIV     0          107  ACTIV3       11   10  255
IST537I  0   1    INACT     0          107  ACTIV3
IST537I  0   2    ACTIV     0          107  ACTIV3       10    1  223
IST537I                     1          107  PDEFO
IST537I                     2          107  PDEFO
IST537I                     3          107  PDEFO
IST537I                     4          107  PDEFO
IST537I                     5          107  PDEFO
IST537I                     6          107  PDEFO
IST537I  7   0    INACT     7          107  INACT
IST537I  7   1    INACT     7          107  INACT
IST537I  7   2    INACT     7          107  INACT
IST314I  END
```

Figure 2-8. Activate PATH011D (error) and DISPLAY ROUTE Subarea 16 to Subarea 107

Next, the coding of FRACKX2 was corrected in the PATH011D Dynamic Path Update member and PATH011D was varied active again. Note that VTAM ignores the PATH statements for CSS1 because the PATHs have already been added and are in an INACT state. The messages indicate that PATHs are changed in FRACKX2. The DISPLAY ROUTE command for subarea 16 to subarea 107 shows that the PATHs are unchanged. The DISPLAY ROUTE command for subarea 107 to subarea 16 shows that ER6 and ER7 have been added to FRACKX2.

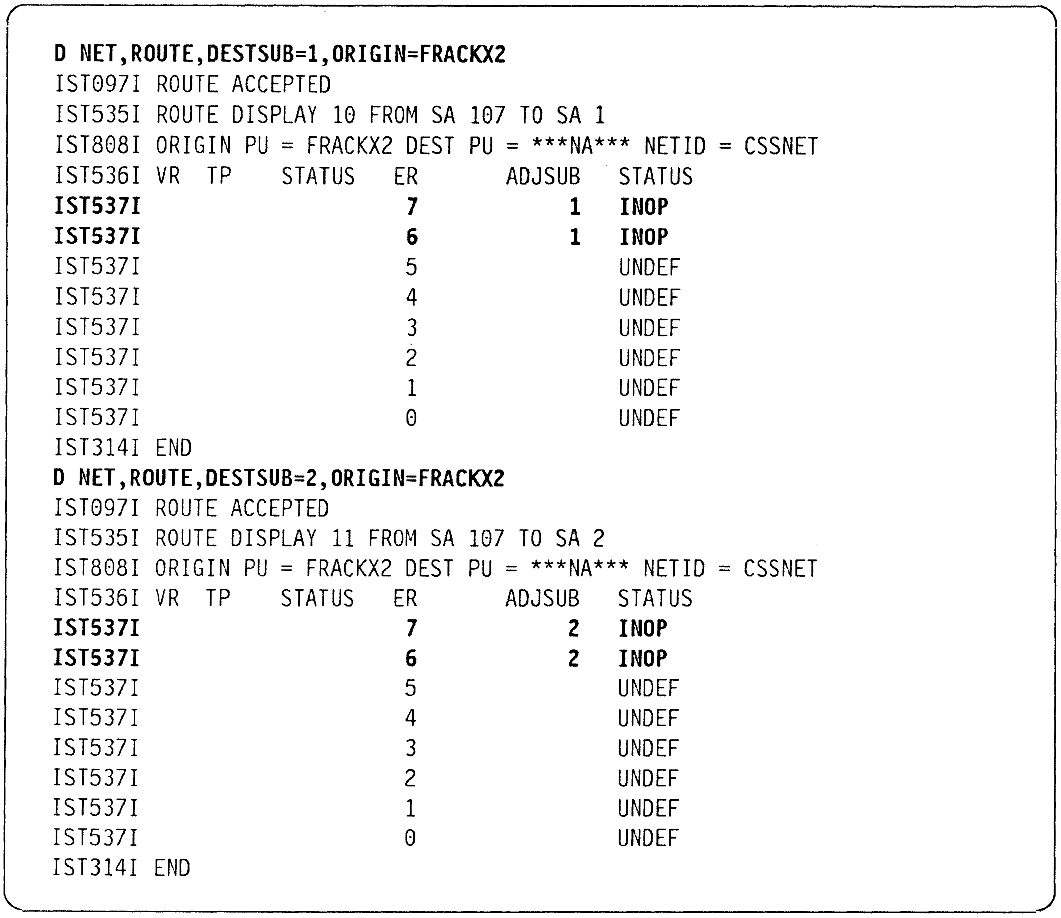
```
V NET,ACT,ID=PATH011D
IST097I VARY ACCEPTED
IST541I FOLLOWING PATH DEFINITION IS IGNORED
IST544I PATH VR07 = 07, DESTSA = 107
IST523I REASON = VR ALREADY DEFINED
IST093I PATH011D ACTIVE
IST929I LOAD OF DYNAMIC PATH DEFINITION FRACKX2.PATH011D COMPLETE
D NET,ROUTE,DESTSUB=107
IST097I ROUTE ACCEPTED
IST535I ROUTE DISPLAY 6 FROM SA 16 TO SA 107
IST808I ORIGIN PU = CSS1PUS DEST PU = FRACKX2 NETID = CSSNET
IST536I VR  TP    STATUS   ER       ADJSUB  STATUS      CUR  MIN  MAX
IST537I  0   0    ACTIV    0           107  ACTIV3      11   10  255
IST537I  0   1    INACT    0           107  ACTIV3
IST537I  0   2    ACTIV    0           107  ACTIV3      10    1  223
IST537I                    1           107  PDEFO
IST537I                    2           107  PDEFO
IST537I                    3           107  PDEFO
IST537I                    4           107  PDEFO
IST537I                    5           107  PDEFO
IST537I                    6           107  PDEFO
IST537I  7   0    INACT    7           107  INACT
IST537I  7   1    INACT    7           107  INACT
IST537I  7   2    INACT    7           107  INACT
IST314I END
D NET,ROUTE,DESTSUB=16,ORIGIN=FRACKX2
IST097I ROUTE ACCEPTED
IST535I ROUTE DISPLAY 7 FROM SA 107 TO SA 16
IST808I ORIGIN PU = FRACKX2 DEST PU = CSS1PUS NETID = CSSNET
IST536I VR  TP    STATUS   ER       ADJSUB  STATUS      CUR  MIN  MAX
IST537I  0   0    ACTIV    0            16  ACTIV3       0    0    0
IST537I  0   1    INACT    0            16  ACTIV3
IST537I  0   2    ACTIV    0            16  ACTIV3       0    0    0
IST537I                    7            16  INACT
IST537I                    6            16  INACT
IST537I                    5                UNDEF
IST537I                    4                UNDEF
IST537I                    3                UNDEF
IST537I                    2                UNDEF
IST537I                    1                UNDEF
IST314I END
```

Figure 2-9. Activate PATH011D (corrected), DISPLAY ROUTEs between SA 16 and SA 107

Next, the Dynamic Path Update member, PATH011N, was varied active. Note that it contains PATH statements for FRACKX2 to allow it to communicate with two new subareas, 1 and 2, using ERs 6 and 7. Note also that member PATH011N contains identical PATH statements for CSS1 to the ones previously activated in PATH011D. The console messages indicate that VTAM ignores these. Also note that member PATH011N contains identical PATH statements for FRACKX2 to communicate with DESTSA 16 to the ones previously activated in PATH011D. Although there is no console message indicating that the PATHs already exist in FRACKX2, the DISPLAY ROUTE command for subarea 107 to subarea 16 show that the PATHs have not been altered. The DISPLAY ROUTE commands for subarea 107 to subarea 1 and subarea 107 to subarea 2 (Figure 2-11 on page 2-19) show that the PATH statements in member PATH011N have been added to FRACKX2.

```
    V NET,ACT,ID=PATH011N
    IST097I VARY ACCEPTED
    IST541I FOLLOWING PATH DEFINITION IS IGNORED
    IST544I PATH VR07 = 07, DESTSA = 107
    IST523I REASON = VR ALREADY DEFINED
    IST093I PATH011N ACTIVE
    IST929I LOAD OF DYNAMIC PATH DEFINITION FRACKX2.PATH011N COMPLETE
    D NET,ROUTE,DESTSUB=107
    IST097I ROUTE ACCEPTED
    IST535I ROUTE DISPLAY 8 FROM SA 16 TO SA 107
    IST808I ORIGIN PU = CSS1PUS DEST PU = FRACKX2 NETID = CSSNET
|   IST536I VR  TP    STATUS   ER       ADJSUB   STATUS      CUR  MIN  MAX
|   IST537I  0   0    ACTIV     0          107   ACTIV3       11   10  255
|   IST537I  0   1    INACT     0          107   ACTIV3
|   IST537I  0   2    ACTIV     0          107   ACTIV3       10    1  223
    IST537I                     1          107   PDEFO
    IST537I                     2          107   PDEFO
    IST537I                     3          107   PDEFO
    IST537I                     4          107   PDEFO
    IST537I                     5          107   PDEFO
    IST537I                     6          107   PDEFO
    IST537I  7   0    INACT     7          107   INACT
    IST537I  7   1    INACT     7          107   INACT
    IST537I  7   2    INACT     7          107   INACT
    IST314I END
    D NET,ROUTE,DESTSUB=16,ORIGIN=FRACKX2
    IST097I ROUTE ACCEPTED
    IST535I ROUTE DISPLAY 9 FROM SA 107 TO SA 16
    IST808I ORIGIN PU = FRACKX2 DEST PU = CSS1PUS NETID = CSSNET
|   IST536I VR  TP    STATUS   ER       ADJSUB   STATUS      CUR  MIN  MAX
|   IST537I  0   0    ACTIV     0           16   ACTIV3        0    0    0
|   IST537I  0   1    INACT     0           16   ACTIV3
|   IST537I  0   2    ACTIV     0           16   ACTIV3        0    0    0
    IST537I                     7           16   INACT
    IST537I                     6           16   INACT
    IST537I                     5                UNDEF
    IST537I                     4                UNDEF
    IST537I                     3                UNDEF
    IST537I                     2                UNDEF
    IST537I                     1                UNDEF
    IST314I END
```

Figure 2-10. Activate PATH011N and DISPLAY ROUTEs between SA 16 and SA 107

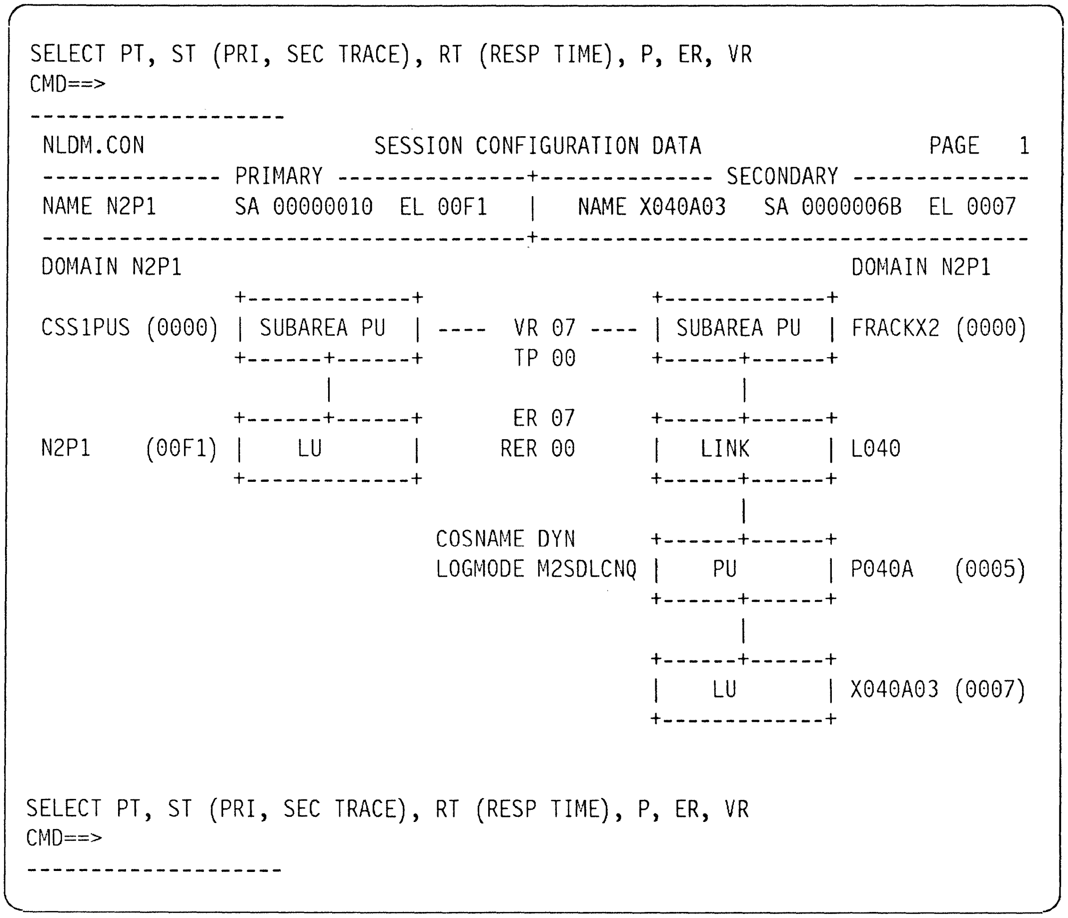


```
D NET,ROUTE,DESTSUB=1,ORIGIN=FRACKX2
IST097I ROUTE ACCEPTED
IST535I ROUTE DISPLAY 10 FROM SA 107 TO SA 1
IST808I ORIGIN PU = FRACKX2 DEST PU = ***NA*** NETID = CSSNET
IST536I VR  TP    STATUS   ER       ADJSUB  STATUS
IST537I                     7            1  INOP
IST537I                     6            1  INOP
IST537I                     5               UNDEF
IST537I                     4               UNDEF
IST537I                     3               UNDEF
IST537I                     2               UNDEF
IST537I                     1               UNDEF
IST537I                     0               UNDEF
IST314I END
D NET,ROUTE,DESTSUB=2,ORIGIN=FRACKX2
IST097I ROUTE ACCEPTED
IST535I ROUTE DISPLAY 11 FROM SA 107 TO SA 2
IST808I ORIGIN PU = FRACKX2 DEST PU = ***NA*** NETID = CSSNET
IST536I VR  TP    STATUS   ER       ADJSUB  STATUS
IST537I                     7            2  INOP
IST537I                     6            2  INOP
IST537I                     5               UNDEF
IST537I                     4               UNDEF
IST537I                     3               UNDEF
IST537I                     2               UNDEF
IST537I                     1               UNDEF
IST537I                     0               UNDEF
IST314I END
```

Figure 2-11. DISPLAY ROUTEs between SA 107 and SA 1 and SA 2

After the VR7/ER7 between Subarea 16 and Subarea 107 were added to CSS1 and a reverse explicit route existed between Subarea 107 and Subarea 16, we were able to establish the LU-to-LU session between a terminal attached to FRACKX2, X040A03, and an application on CSS1, TSO. In the Dynamic Table Replacement Chapter of this manual is described a procedure we went through where we dynamically replaced the Interpret Table, USS Table, and Logon Mode Table for the terminal, X040A03. When we replaced the Logon Mode Table for X040A03, we pointed to a new COS name, DYN. We then used Dynamic Table Replacement to refresh the COS Table for CSS1, ISTSDCOS, with a new copy having an entry of DYN. The DYN entry indicated that VR7 was to be used for sessions. At that point we were unable to logon from X040A03 to TSO because there was no VR7 on our system.

After adding the routes as explained above, we established the session between X040A03 and TSO. Figure 2-12 on page 2-20 shows the NetView Session screen showing the session partners, "DYN" COS entry name, and routes (VR7, ER7, RER0) used for the session.

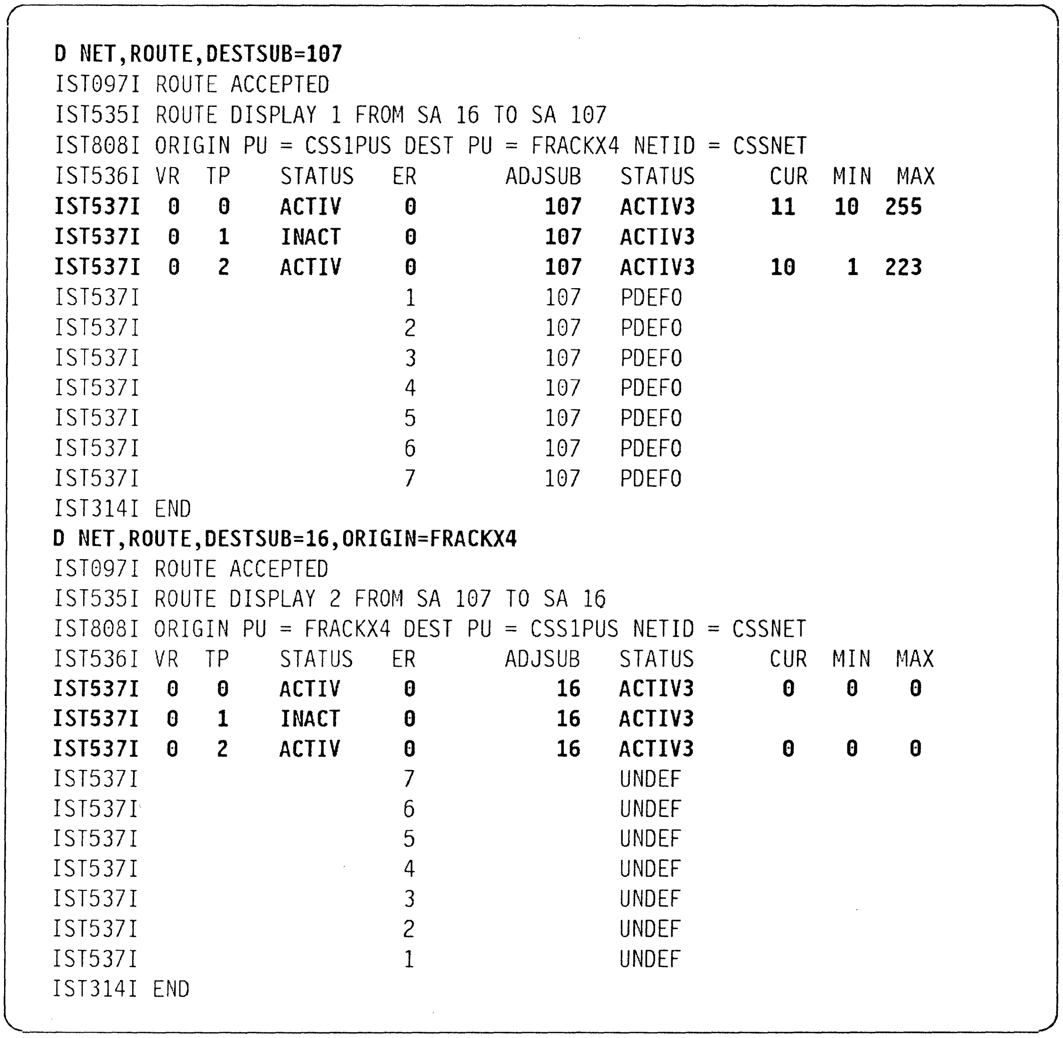


```
SELECT PT, ST (PRI, SEC TRACE), RT (RESP TIME), P, ER, VR
CMD==>
--------------------
 NLDM.CON                   SESSION CONFIGURATION DATA                PAGE   1
 -------------- PRIMARY ---------------+-------------- SECONDARY --------------
 NAME N2P1      SA 00000010  EL 00F1   |   NAME X040A03   SA 0000006B  EL 0007
 --------------------------------------+---------------------------------------
 DOMAIN N2P1                                                  DOMAIN N2P1
                +-------------+                   +-------------+
 CSS1PUS (0000) | SUBAREA PU  | ----  VR 07  ---- | SUBAREA PU  | FRACKX2 (0000)
                +------+------+       TP 00       +------+------+
                       |                                 |
                +------+------+       ER 07       +------+------+
 N2P1    (00F1) |     LU      |      RER 00       |    LINK     | L040
                +-------------+                   +------+------+
                                                         |
                                COSNAME DYN       +------+------+
                                LOGMODE M2SDLCNQ  |     PU      | P040A   (0005)
                                                  +------+------+
                                                         |
                                                  +------+------+
                                                  |     LU      | X040A03 (0007)
                                                  +-------------+

SELECT PT, ST (PRI, SEC TRACE), RT (RESP TIME), P, ER, VR
CMD==>
--------------------
```

Figure 2-12. NetView Screen -- Session Using Dynamically Added Route

We also tried activating the Dynamic Path Update member, PATH011N, shown in Figure 2-5 on page 2-13 when our NCP gen contained 0 values for TGBXTRA, and PATHEXT. For this exercise we loaded a new NCP, FRACKX4, which had been genned with values of 0 for TGBXTRA and PATHEXT. Figure 2-13 on page 2-21 illustrates that only ER0/VR0 are defined between CSS1 (subarea 16) and FRACKX4 (subarea 107) before PATH011N is activated.



```
D NET,ROUTE,DESTSUB=107
IST097I ROUTE ACCEPTED
IST535I ROUTE DISPLAY 1 FROM SA 16 TO SA 107
IST808I ORIGIN PU = CSS1PUS DEST PU = FRACKX4 NETID = CSSNET
IST536I VR  TP    STATUS   ER       ADJSUB  STATUS      CUR  MIN  MAX
IST537I  0   0    ACTIV     0          107  ACTIV3       11   10  255
IST537I  0   1    INACT     0          107  ACTIV3
IST537I  0   2    ACTIV     0          107  ACTIV3       10    1  223
IST537I                     1          107  PDEFO
IST537I                     2          107  PDEFO
IST537I                     3          107  PDEFO
IST537I                     4          107  PDEFO
IST537I                     5          107  PDEFO
IST537I                     6          107  PDEFO
IST537I                     7          107  PDEFO
IST314I END
D NET,ROUTE,DESTSUB=16,ORIGIN=FRACKX4
IST097I ROUTE ACCEPTED
IST535I ROUTE DISPLAY 2 FROM SA 107 TO SA 16
IST808I ORIGIN PU = FRACKX4 DEST PU = CSS1PUS NETID = CSSNET
IST536I VR  TP    STATUS   ER       ADJSUB  STATUS      CUR  MIN  MAX
IST537I  0   0    ACTIV     0           16  ACTIV3        0    0    0
IST537I  0   1    INACT     0           16  ACTIV3
IST537I  0   2    ACTIV     0           16  ACTIV3        0    0    0
IST537I                     7               UNDEF
IST537I                     6               UNDEF
IST537I                     5               UNDEF
IST537I                     4               UNDEF
IST537I                     3               UNDEF
IST537I                     2               UNDEF
IST537I                     1               UNDEF
IST314I END
```

Figure 2-13. Defined Routes between CSS1 and FRACKX4

Figure 2-14 on page 2-22 shows that message IST927I is received when trying to activate PATH011N with NCPPATH statements for FRACKX4 to add routes for destination subarea 1 and destination subarea 2. TGBXTRA of at least 2 and PATHEXT of at least 2 needed to be generated in FRACKX4 in order to add these PATHs. Note, however, that the ER7/VR7 route from subarea 16 to subarea 107 was added to CSS1's PATHs and that the ER6 and ER7 routes from subarea 107 to subarea 16 were added to FRACKX4's PATHs. No extra TGBs were required for ER6 and ER7 because the adjacent subarea/transmission group pair had already been generated in FRACKX4's PATH statements for ER0. No PATHEXT was needed to add ER6 and ER7 because the destination subarea, 16, had already been generated in FRACKX4's PATH statements for ER0.

```
V NET,ACT,ID=PATH011N
IST097I VARY ACCEPTED
IST093I PATH011N ACTIVE
IST927I ERROR FOR FRACKX4.PATH011N DSA 1 NETID CSSNET CODE 2
IST927I ERROR FOR FRACKX4.PATH011N DSA 2 NETID CSSNET CODE 2
IST929I LOAD OF DYNAMIC PATH DEFINITION FRACKX4.PATH011N COMPLETE
D NET,ROUTE,DESTSUB=107
IST097I ROUTE ACCEPTED
IST535I ROUTE DISPLAY 3 FROM SA 16 TO SA 107
IST808I ORIGIN PU = CSS1PUS DEST PU = FRACKX4 NETID = CSSNET
IST536I VR  TP    STATUS   ER      ADJSUB  STATUS     CUR  MIN  MAX
IST537I  0   0    ACTIV     0         107  ACTIV3     11   10  255
IST537I  0   1    INACT     0         107  ACTIV3
IST537I  0   2    ACTIV     0         107  ACTIV3     10    1  223
IST537I                     1         107  PDEFO
IST537I                     2         107  PDEFO
IST537I                     3         107  PDEFO
IST537I                     4         107  PDEFO
IST537I                     5         107  PDEFO
IST537I                     6         107  PDEFO
IST537I  7   0    INACT     7         107  INACT
IST537I  7   1    INACT     7         107  INACT
IST537I  7   2    INACT     7         107  INACT
IST314I END
D NET,ROUTE,DESTSUB=16,ORIGIN=FRACKX4
IST097I ROUTE ACCEPTED
IST535I ROUTE DISPLAY 4 FROM SA 107 TO SA 16
IST808I ORIGIN PU = FRACKX4 DEST PU = CSS1PUS NETID = CSSNET
IST536I VR  TP    STATUS   ER      ADJSUB  STATUS     CUR  MIN  MAX
IST537I  0   0    ACTIV     0          16  ACTIV3      0    0    0
IST537I  0   1    INACT     0          16  ACTIV3
IST537I  0   2    ACTIV     0          16  ACTIV3      0    0    0
IST537I                     7          16  INACT
IST537I                     6          16  INACT
IST537I                     5              UNDEF
IST537I                     4              UNDEF
IST537I                     3              UNDEF
IST537I                     2              UNDEF
IST537I                     1              UNDEF
IST314I END
```

Figure 2-14. Using Dynamic Path Update with Insufficient Values for TGBXTRA & PATHEXT

2.5 Dynamic Path Update Migration Considerations

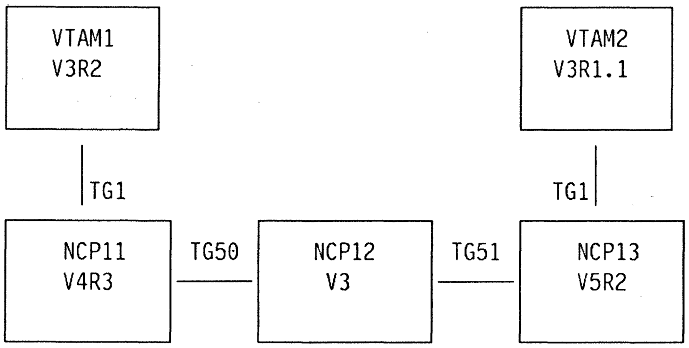
In order to activate a Dynamic Path Update member with the new VPATH and NCPPATH statements, VTAM must be at a V3R2 level. VTAMs previous to V3R2 can add PATHs for VTAM as long as the PATHs are undefined or inoperative by activating VTAM Definition Library members with VTAM PATH statements in them; however, pre-V3R2 cannot process members with the new VPATH and NCPPATH statements.

In order to be able to process changes in PATHs, NCPs must be at a V4R3 or V5R2 level and be generated properly with sufficient values for TGBXTRA, PATHEXT, and the second suboperand of VRPOOL. There must be an SSCP-to-NCP session between the VTAM activating the Dynamic Path Update member and NCP in order for the changes in PATHs to be sent to the NCP. This means that all NCPs must be generated with at least one route for the SSCP-to-NCP session. **The route for the SSCP-to-NCP session cannot be defined or changed via Dynamic Path Update because it will be an active route before the SSCP-to-NCP session can be established.**

In the example shown in Figure 2-15 on page 2-24, assuming that VTAM1 owns the NCPs and that ER1/VR1 is undefined at this point, VTAM1 could activate the Dynamic Path Update member shown. ER1/VR1 would be added as a PATH for VTAM1 to use for each subarea shown. ER1 to each subarea shown would be added to both NCP11's and NCP13's PATHs. Note that NCP12 does not support addition of dynamic paths because it is NCP V3; however, since VTAM1 has an SSCP-to-NCP session with NCP13, PATHs can be dynamically changed in NCP13, even with a backlevel NCP on the route between VTAM1 and NCP13.

Note also that VTAM cannot dynamically change PATHs in another VTAM. In Figure 2-14 on page 2-22, VTAM1 cannot dynamically change VTAM2's PATHs, even if VTAM2 were V3R2. VTAM2 can add ER1/VR1 to its PATHs by activating the PATH statements shown. (If VTAM2 were at a V3R2 level it could optionally use VPATH as well).

NCP12, since it is V3, would need to be generated with the proper PATH statements for ER1 to each subarea shown in order to have a complete ER1 route between all subareas in both directions.



VTAM1 Dynamic Path Update Member:

```
VTAM1 VPATH NETID=NETA
      PATH DESTSA=2,
           ER1=(11,1),
           VR1=1
      PATH DESTSA=13,
           ER1=(11,1),
           VR1=1
      PATH DESTSA=12,
           ER1=(11,1),
           VR1=1
      PATH DESTSA=11,
           ER1=(11,1),
           VR1=1
NCP11 NCPPATH NETID=NETA
      PATH DESTSA=2,
           ER1=(12,50)
      PATH DESTSA=13,
           ER1=(12,50)
      PATH DESTSA=12,
           ER1=(12,50)
      PATH DESTSA=1,
           ER1=(1,1)
NCP13 NCPPATH NETID=NETA
      PATH DESTSA=2,
           ER1=(2,1)
      PATH DESTSA=12,
           ER1=(12,51)
      PATH DESTSA=11,
           ER1=(12,51)
      PATH DESTSA=1,
           ER1=(12,51)
```

VTAM2 PATH Member:

```
VTAM2  PATH DESTSA=13,
            ER1=(13,1),
            VR1=1
       PATH DESTSA=12,
            ER1=(13,1),
            VR1=1
       PATH DESTSA=11,
            ER1=(13,1),
            VR1=1
       PATH DESTSA=1,
            ER1=(13,1),
            VR1=1
```

Figure 2-15. Migration Considerations for Dynamic Path Update

2.6 Backup/Recovery Considerations

Using the same configuration shown in Figure 2-15 on page 2-24, if VTAM1, the owning VTAM, went down, the PATHs changed in NCP11 and NCP13 would still remain. VTAM2 could take over ownership of the NCPs and ER1/VR1 would remain a viable route between subareas 2, 11, 12, and 13.

When VTAM1 comes back up, VTAM1 would need to again add the PATH statements needed for itself. It could again activate the Dynamic Path Update member shown, even though the PATHs had already been applied to NCP11 and NCP13. This would not cause problems within NCP11 and NCP13--the PATHs that had already been altered would just be ignored.

The PATHs altered in NCP11 and NCP13 via Dynamic Path Update would remain until the NCPs are reloaded or until changed by subsequent Dynamic Path Update operations.

Chapter 3. Dynamic Reconfiguration Enhancements

3.1 Dynamic Reconfiguration Prior to VTAM V3R2

Dynamic Reconfiguration in pre-VTAM V3R2/NCP V4R3-V5R2 environments allows the configuration of physical units and logical units to be changed in an NCP while VTAM and NCP are operating. Type 1 or Type 2 Physical Units (PUs) can be added or deleted. Logical Units can also be added or deleted. Dynamic Reconfiguration is accomplished by coding ADD and/or DELETE statements within a VTAM Dynamic Reconfiguration Definition dataset (also known as a DRDS deck). A DRDS deck is a member in the VTAM Definition Library and starts with a VBUILD TYPE=DR statement. Dynamic additions or deletions take place when a "VARY DRDS" VTAM operator command is issued naming a DRDS deck.

A PU can be "moved" from one line to another by deleting it from one line and adding it to another. Figure 3-1 shows an example of a DRDS deck that contains the statements needed to move PU P040A and its LUs (starting with X040A02) from Line L040 to Line L000.

```
DR1      VBUILD  TYPE=DR
         DELETE  FROM=L040
P040A    PU
         ADD   TO=L000
P040A    PU    ADDR=C1,ISTATUS=INACTIVE,MAXDATA=265,MAXOUT=7,
               PASSLIM=7,PUTYPE=2,MODETAB=AMODETAB,USSTAB=USS3270A
X040A02  LU    LOCADDR=02,DLOGMOD=M3SDLCNQ
X040A03  LU    LOCADDR=03,DLOGMOD=M2SDLCNQ
X040A04  LU    LOCADDR=04,DLOGMOD=M2SDLCNQ
X040A05  LU    LOCADDR=05,DLOGMOD=M2SDLCNQ
```

Figure 3-1. Example of A Dynamic Reconfiguration Definition Dataset

It is necessary to code desired operands on DR (Dynamically Reconfigured) ADDED PUs and/or LUs as there is no "sift down" effect from GROUP or LINE statements. The following parameters can be coded on PU statements to be dynamically added:

- ADDR
- PUTYPE
- BNNSUP
- ANS
- MAXOUT
- PASSLIM
- IRETRY
- MAXDATA

The following parameters can be coded on LU statements to be dynamically added:

- LOCADDR
- PACING

- BATCH

VTAM-only parameters, such as ISTATUS, can also be coded in a DR Definition dataset. For any NCP parameters not listed above, NCP default values are used. MAXDATA and LOCADDR must be coded in the definition dataset because there are no default values for these. Also ADDR must be coded as there is no default value.

The NCP must be generated correctly to allow for Dynamic Reconfiguation to take place. Following is a list of the NCP definition statements pertaining to Dynamic Reconfiguration:

Table 3-1. NCP V4R2 and Earlier Dynamic Reconfiguration Statements

DEFINITION STATEMENT	OPERAND	EXPLANATION
BUILD	DR3270	Specify YES if SDLC 3270 PU T1s have not been generated but may be DR added
BUILD	RESOEXT	Number of extra Network Addresses for generated devices that will be DR deleted & added
PUDRPOOL	NUMBER	Number of PU control blocks needed for dynamic addition of PUs
PUDRPOOL	MAXLU	Maximum number of LUs allowed per PU that is DR added
LUDRPOOL	NUMTYP1	Number of PU T1 LU control blocks for DR addition and SDLC switched
LUDRPOOL	NUMTYP2	Number of PU T2 LU control blocks for DR addition and SDLC switched
LINE	MAXPU	Number of generated PUs on this line plus number to be dynamically added
SERVICE	MAXLIST	Number of entries in Service Order Table--needs to include number for DR added PUs
PU	PUDR	Code YES if this PU may be deleted dynamically
PU	MAXLU	Highest LOCADDR of any generated or dynamically added LU on this PU
LU	LUDR	Code YES if this LU may be deleted dynamically

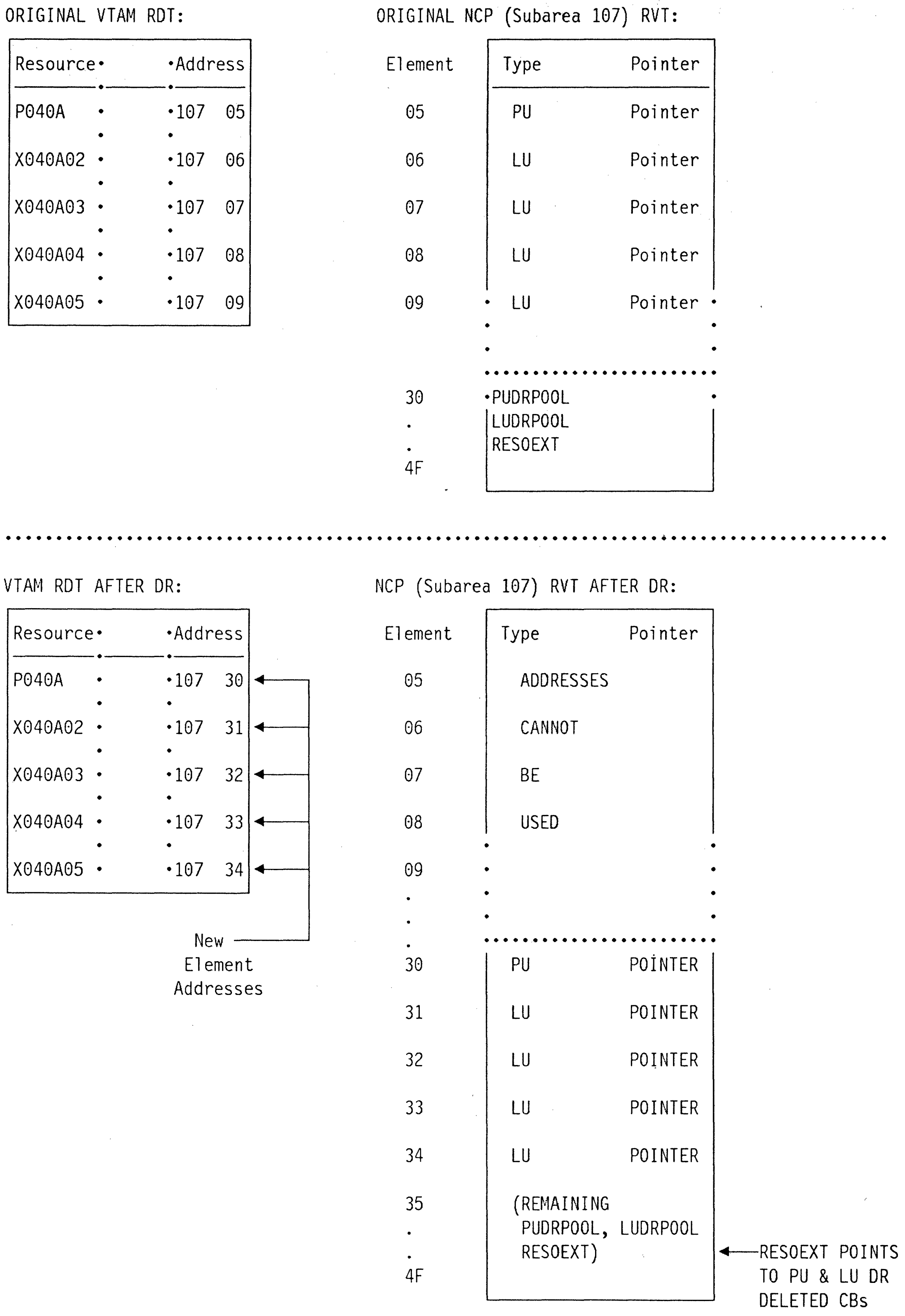
With VTAMs prior to V3R2 and NCPs prior to V4R3/V5, new network addresses are required when resources are dynamically reconfigured. That is, when a generated resource is DR DELETED, its NCP Element Address cannot be reused. However, the control blocks (PU control blocks or LU control blocks, depending on the type of resource) are put into the pool of control blocks that can be used for DR ADDED resources, if there are any available network addresses created by RESOEXT. The PUDRPOOL and LUDRPOOL statements in the NCP generate extra PU and/or LU control blocks (with valid element addresses) for use with DR ADDED resources. The RESOEXT statement in the NCP generates extra Element Addresses to be used with PU and/or LU control blocks that have been put into the pool as a result of DR DELETION.

The LUDRPOOL addresses and control blocks are used for switched line resources as well as for DR resources.

Figure 3-2 on page 3-6 shows an example of the effect on VTAM's RDT (Resource Definition Table) and NCP's RVT (Resource Vector Table) when a generated PU and its LUs are DR DELETED from one line and DR ADDED to another line.

Using Figure 3-2 on page 3-6 as an example, after VTAM activates NCP Subarea 107, VTAM's RDT, shown on the top left, reflects that PU P040A and its four LUs have Network Addresses of Subarea 107, Elements 05 through 09. The NCP RVT (Resource Vector Table), on the top right, shows how the NCP generation resulted in PU P040A being the fifth element generated, followed by the LUs. Following all the generated resources, the NCP RVT contains extra addresses (extra RVT entries) to be used for dynamic reconfiguration and switched line resources (the PUDRPOOL, LUDRPOOL, and RESOEXT). The PUDRPOOL and LUDRPOOL entries point to extra PU or LU control blocks to be used for DR ADDED or switched line resources. The RESOEXT entries to start with do not contain pointers to PU and/or LU control blocks. RESOEXT entries are used after a DR DELETE to point to PU or LU control blocks originally used for generated resources. When generated resources are DR DELETED, the RESOEXT entries are updated to point to the control blocks so that the control blocks can later be reused for DR ADDED resources.

After the Dynamic Reconfiguration Definition Dataset shown in Figure 3-1 on page 3-3 is activated, VTAM first notifies NCP to DELETE the PU and LUs. NCP invalidates Element Addresses 05 through 09 and pointers to the freed PU and LU control blocks are placed in RVT RESOEXT entries. When VTAM next notifies NCP to ADD the PU and LUs, it Requests Network Address Assignments for these resources. NCP uses RVT entries resulting from PUDRPOOL, LUDRPOOL, and/or RESOEXT (in our example, the Element Addresses used are 30 through 34). The VTAM RVT shown on the lower left, after DR, illustrates that the network addresses used for P040A and its LUs are now Subarea 107 Elements 30 through 34.



ORIGINAL VTAM RDT:

Resource	Address
P040A	107 05
X040A02	107 06
X040A03	107 07
X040A04	107 08
X040A05	107 09

ORIGINAL NCP (Subarea 107) RVT:

Element	Type	Pointer
05	PU	Pointer
06	LU	Pointer
07	LU	Pointer
08	LU	Pointer
09	LU	Pointer
30 . . 4F	PUDRPOOL LUDRPOOL RESOEXT	

VTAM RDT AFTER DR:

Resource	Address
P040A	107 30
X040A02	107 31
X040A03	107 32
X040A04	107 33
X040A05	107 34

New Element Addresses

NCP (Subarea 107) RVT AFTER DR:

Element	Type	Pointer
05	ADDRESSES	
06	CANNOT	
07	BE	
08	USED	
09		
30	PU	POINTER
31	LU	POINTER
32	LU	POINTER
33	LU	POINTER
34	LU	POINTER
35 . . 4F	(REMAINING PUDRPOOL, LUDRPOOL RESOEXT)	

RESOEXT POINTS TO PU & LU DR DELETED CBs

Figure 3-2. Example of Pre-VTAM V3R2 Dynamic Reconfiguration

3.2 Dynamic Reconfiguration Enhancements

With VTAM V3R2 and NCP V4R3 and/or NCP V5R2, Dynamic Reconfiguration has been enhanced. There are three new ways in which network configurations can be changed while VTAM and NCP are operating or without regenerating the NCP:

1. A new operator command can be used to move or delete resources.
2. A new statement, MOVE, can be used in a DR definition dataset.
3. New resource definitions can be added to NCP source and activated.

Examples of the three new ways to perform Dynamic Reconfiguration are in the following three Sections of this Chapter.

Other Dynamic Reconfiguration enhancements in VTAM V3R2 and NCP V4R3/NCP V5R2 are the reuse of dynamically deleted network addresses, the ability to code the DATMODE= parameter in a DRDS deck, and the ability to dynamically reconfigure T2.1 Nodes.

3.2.1 New Operator Command

A new operator command is available with VTAM V3R2 called MODIFY DR. The command can be used to move a PU and its LUs from one line to another, to delete a PU and its LUs, or to delete an LU. The MODIFY DR command allows a PU and its LUs to be moved from one line to another using the command in the form:

```
MODIFY DR,TYPE=MOVE,ID=puname,FROM=linename,TO=linename
```

The MODIFY DR command allows a PU to be deleted from a line using the command in the form:

```
MODIFY DR,TYPE=DELETE,ID=puname,FROM=linename
```

The MODIFY DR command allows an LU to be deleted from a PU using the command in the form:

```
MODIFY DR,TYPE=DELETE,ID=luname,FROM=puname
```

The MODIFY DR command allows a PU and its LUs to be moved and the PU's address changed using the command in the form:

```
MODIFY DR,TYPE=MOVE,ID=puname,FROM=linename,TO=linename,ADDR=address
```

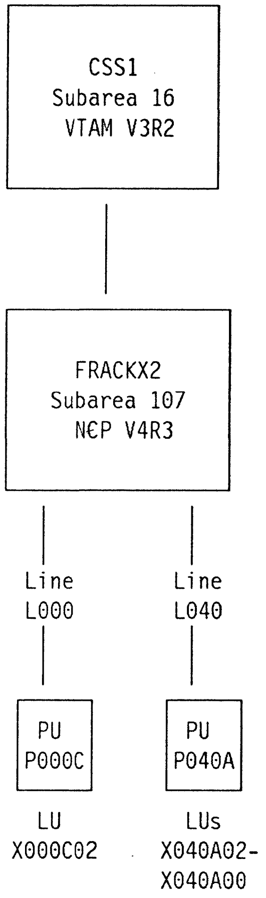
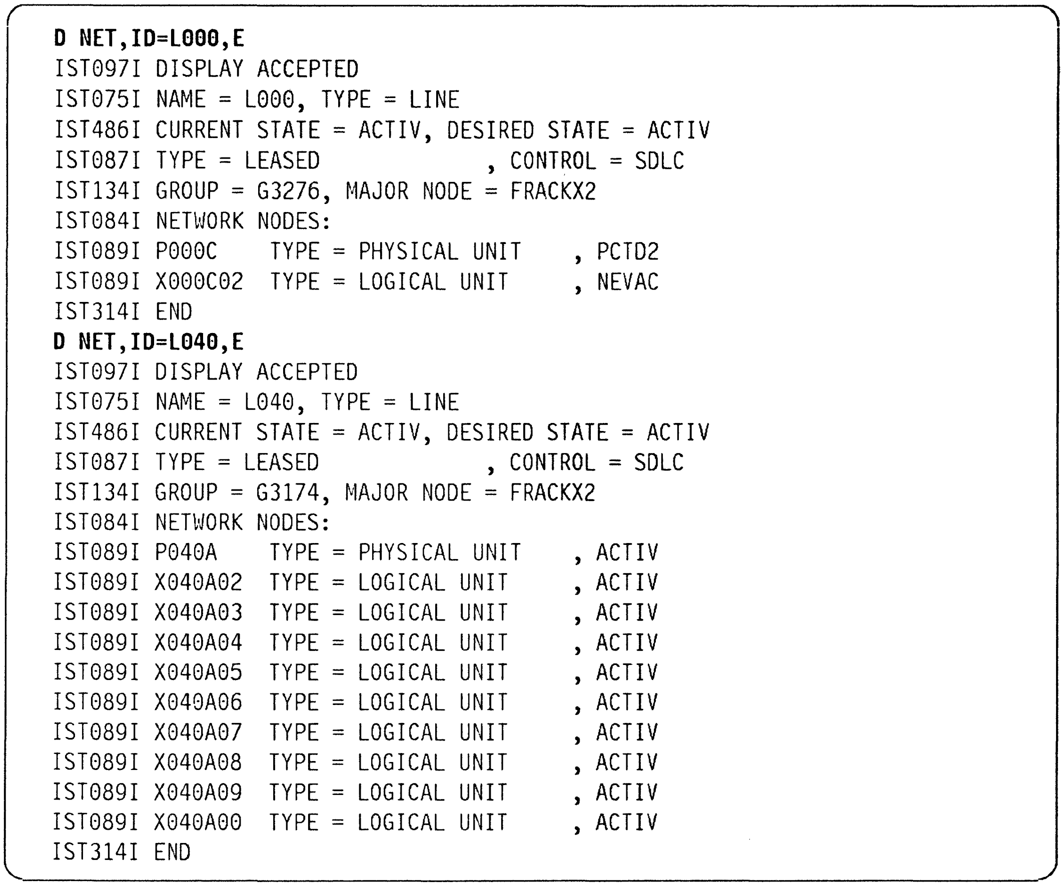


Figure 3-3. Configuration of WSC LAB

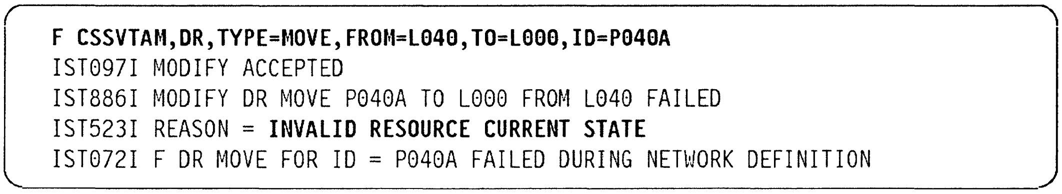
At the Washington Systems Center using the configuration illustrated in Figure 3-3, we used the new MODIFY DR command to move a PU, to delete a PU, and to delete an LU. Figure 3-4 on page 3-9 shows the console listing when PU P040A was moved from Line L040 to Line L000. First, we displayed Line L000 showing PU P000C and LU X000C02 currently configured and Line L040 showing PU P040A and LUs starting with X040A02 currently configured.



```
D NET,ID=L000,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L000, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED            , CONTROL = SDLC
IST134I GROUP = G3276, MAJOR NODE = FRACKX2
IST084I NETWORK NODES:
IST089I P000C    TYPE = PHYSICAL UNIT    , PCTD2
IST089I X000C02  TYPE = LOGICAL UNIT     , NEVAC
IST314I END
D NET,ID=L040,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L040, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED            , CONTROL = SDLC
IST134I GROUP = G3174, MAJOR NODE = FRACKX2
IST084I NETWORK NODES:
IST089I P040A    TYPE = PHYSICAL UNIT    , ACTIV
IST089I X040A02  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A03  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A04  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A05  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A06  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A07  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A08  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A09  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A00  TYPE = LOGICAL UNIT     , ACTIV
IST314I END
```

Figure 3-4. Display of Lines L000 and L040

Figure 3-5 shows our first attempt to use the MODIFY DR TYPE=MOVE command to move PU P040A to Line L000. Note that we were unsuccessful because the PU was ACTIVE. In order to use the MODIFY DR command, the PUs and/or LUs to be dynamically reconfigured must be INACTIVE.



```
F CSSVTAM,DR,TYPE=MOVE,FROM=L040,TO=L000,ID=P040A
IST097I MODIFY ACCEPTED
IST886I MODIFY DR MOVE P040A TO L000 FROM L040 FAILED
IST523I REASON = INVALID RESOURCE CURRENT STATE
IST072I F DR MOVE FOR ID = P040A FAILED DURING NETWORK DEFINITION
```

Figure 3-5. MODIFY DR Command Failure Because Resource Active

We next varied inactive PU P040A and the MODIFY DR TYPE=MOVE command was subsequently successful. A display of Line L000 shows that P040A is now configured on this line along with P000C. Note that the status shown for P040A and its LUs is "INACT----D." The D denotes that the resources are dynamically reconfigured.

```
V NET,INACT,ID=P040A
IST097I VARY ACCEPTED
IST105I P040A NODE NOW INACTIVE
F CSSVTAM,DR,TYPE=MOVE,FROM=L040,TO=L000,ID=P040A
IST097I MODIFY ACCEPTED
IST241I F DR MOVE COMMAND COMPLETE FOR P040A
D NET,ID=L000,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L000, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED            , CONTROL = SDLC
IST134I GROUP = G3276, MAJOR NODE = FRACKX2
IST084I NETWORK NODES:
IST089I P040A    TYPE = PHYSICAL UNIT    , INACT----D
IST089I X040A02  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A03  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A04  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A05  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A06  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A07  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A08  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A09  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A00  TYPE = LOGICAL UNIT     , INACT----D
IST089I P000C    TYPE = PHYSICAL UNIT    , PCTD2
IST089I X000C02  TYPE = LOGICAL UNIT     , NEVAC
IST314I END
```

Figure 3-6. Moving a PU to a new Line Using Modify DR Command

Another LAB exercise shown in Figure 3-7 illustrates using the MODIFY DR command to delete PU P040A from Line L040. The display of line L040 illustrates that P040A and its LUs are active. We then varied inactive the PU and executed the MODIFY DR TYPE=DELETE command. A subsequent display of line L040 shows that P040A has been deleted.

```
D NET,ID=L040,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L040, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED            , CONTROL = SDLC
IST134I GROUP = G3174, MAJOR NODE = FRACKX2
IST084I NETWORK NODES:
IST089I P040A    TYPE = PHYSICAL UNIT    , ACTIV
IST089I X040A02  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A03  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A04  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A05  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A06  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A07  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A08  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A09  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A00  TYPE = LOGICAL UNIT     , ACTIV
IST314I END
V NET,INACT,ID=P040A
IST097I VARY ACCEPTED
IST105I P040A NODE NOW INACTIVE
F CSSVTAM,DR,TYPE=DELETE,ID=P040A,FROM=L040
IST097I MODIFY ACCEPTED
IST241I F DR DEL COMMAND COMPLETE FOR P040A
D NET,ID=L040,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L040, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED            , CONTROL = SDLC
IST134I GROUP = G3174, MAJOR NODE = FRACKX2
IST172I NO NETWORK NODES EXIST
IST314I END
```

Figure 3-7. Deleting a PU Using Modify DR Command

Another LAB exercise shown in Figure 3-8 illustrates using the MODIFY DR command to delete LU X040A03 from PU P040A. The display of line L040 shows that P040A and its LUs, including X040A03, are active. We then varied inactive the LU, X040A03, and executed the MODIFY DR TYPE=DELETE command. A subsequent display of PU P040A shows that LU X040A03 no longer exists.

```
D NET,ID=L040,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L040, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED             , CONTROL = SDLC
IST134I GROUP = G3174, MAJOR NODE = FRACKX2
IST084I NETWORK NODES:
IST089I P040A    TYPE = PHYSICAL UNIT    , ACTIV
IST089I X040A02  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A03  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A04  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A05  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A06  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A07  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A08  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A09  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A00  TYPE = LOGICAL UNIT     , ACTIV
IST314I END
V NET,INACT,ID=X040A03
IST097I VARY ACCEPTED
IST105I X040A03 NODE NOW INACTIVE
F CSSVTAM,DR,TYPE=DELETE,ID=X040A03,FROM=P040A
IST097I MODIFY ACCEPTED
IST241I F DR DEL COMMAND COMPLETE FOR X040A03
D NET,ID=P040A,E
IST097I DISPLAY ACCEPTED
IST075I NAME = P040A, TYPE = PU_T2
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST081I LINE NAME = L040, LINE GROUP = G3174, MAJNOD = FRACKX2
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST355I LOGICAL UNITS:
IST080I X040A02  ACTIV     X040A04  ACTIV     X040A05  ACTIV
IST080I X040A06  ACTIV     X040A07  ACTIV     X040A08  ACTIV
IST080I X040A09  ACTIV     X040A00  ACTIV
IST314I END
```

Figure 3-8. Deleting an LU Using Modify DR Command

3.2.2 New MOVE Statement in DR Definition Dataset

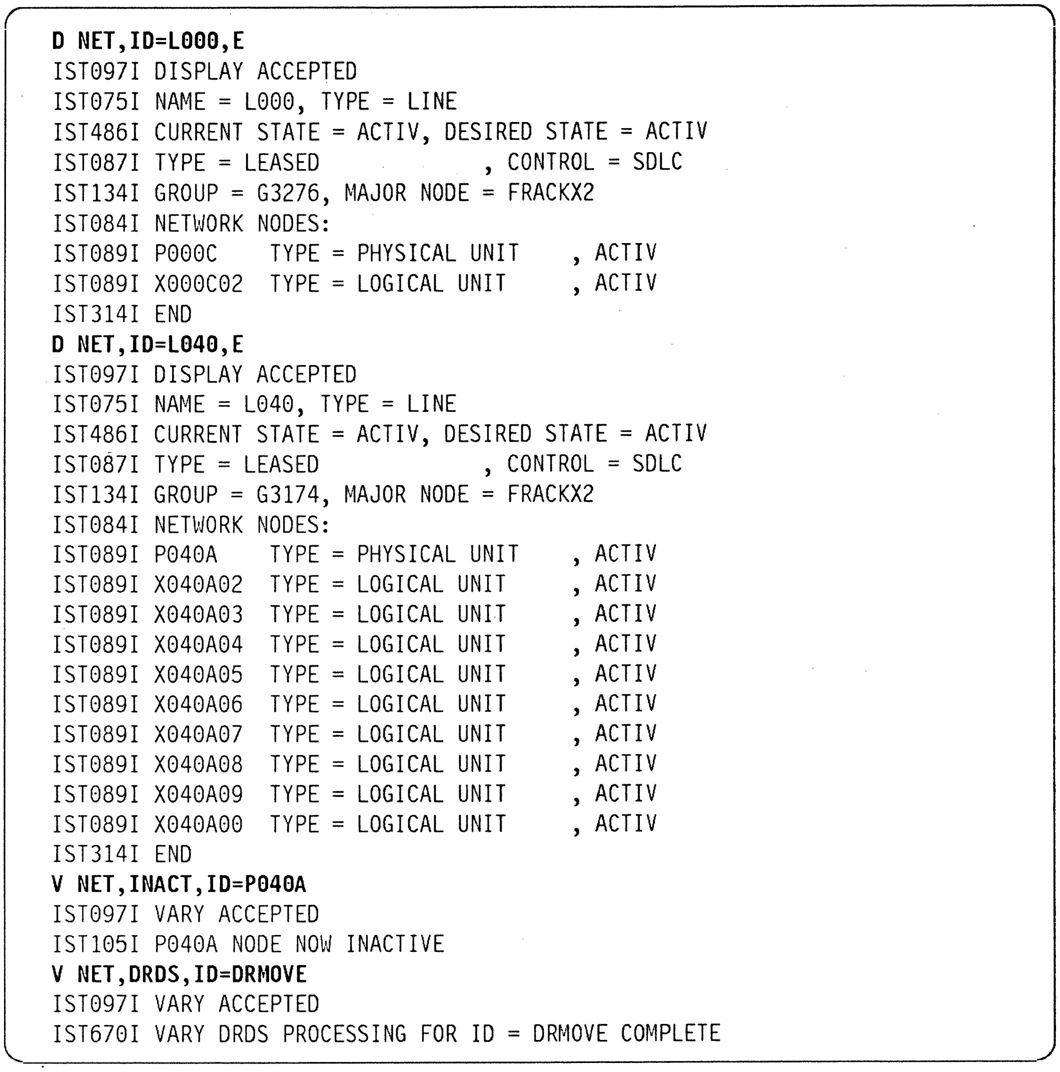
Dynamic Reconfiguration can also be performed with VTAM V3R2 and NCP V4R3 or NCP V5R2 by using the "VARY DRDS" VTAM command in conjunction with a Dynamic Reconfiguration Definition Dataset, as is true for earlier releases of VTAM and NCP. An enhancement with VTAM V3R2 and NCP V4R3/V5R2 is that the MOVE statement can be coded in a DRDS deck, simplifying the coding required. Unlike the DR example shown in Figure 3-1 on page 3-3, it is not necessary to code DELETE and ADD statements with a number of parameters to MOVE a PU. A PU and its LUs can be moved by issuing a "VARY DRDS" command in conjunction with a DRDS deck similar to Figure 3-9.

VTAM Definition Library--Member Name DRMOVE

```
        VBUILD TYPE=DR
        MOVE FROM=L040,TO=L000
 P040A  PU
```

Figure 3-9. Example of MOVE statement in DR Definition Dataset

Figure 3-10 on page 3-14 shows the console listing when we issued a "VARY DRDS" command naming the above member, DRMOVE. First a display of Line L000 shows that it has one PU and one LU and a display of Line L040 shows that it has one PU and nine LUs. After inactivating the PU, P040A, on Line L040, the VARY DRDS command is entered.



```
D NET,ID=L000,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L000, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED            , CONTROL = SDLC
IST134I GROUP = G3276, MAJOR NODE = FRACKX2
IST084I NETWORK NODES:
IST089I P000C    TYPE = PHYSICAL UNIT    , ACTIV
IST089I X000C02  TYPE = LOGICAL UNIT     , ACTIV
IST314I END
D NET,ID=L040,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L040, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED            , CONTROL = SDLC
IST134I GROUP = G3174, MAJOR NODE = FRACKX2
IST084I NETWORK NODES:
IST089I P040A    TYPE = PHYSICAL UNIT    , ACTIV
IST089I X040A02  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A03  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A04  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A05  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A06  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A07  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A08  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A09  TYPE = LOGICAL UNIT     , ACTIV
IST089I X040A00  TYPE = LOGICAL UNIT     , ACTIV
IST314I END
V NET,INACT,ID=P040A
IST097I VARY ACCEPTED
IST105I P040A NODE NOW INACTIVE
V NET,DRDS,ID=DRMOVE
IST097I VARY ACCEPTED
IST670I VARY DRDS PROCESSING FOR ID = DRMOVE COMPLETE
```

Figure 3-10. Example of Console Listing with MOVE DRDS Deck

Figure 3-11 on page 3-15 illustrates a subsequent display of line L000 showing that PU, P040A, and its LUs have been added to Line L000. The original PU, P000C, and its LU are also shown. A display of Line L040 shows that no network nodes exist after the MOVE.

```
D NET,ID=L000,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L000, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED            , CONTROL = SDLC
IST134I GROUP = G3276, MAJOR NODE = FRACKX2
IST084I NETWORK NODES:
IST089I P040A    TYPE = PHYSICAL UNIT    , INACT----D
IST089I X040A02  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A03  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A04  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A05  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A06  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A07  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A08  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A09  TYPE = LOGICAL UNIT     , INACT----D
IST089I X040A00  TYPE = LOGICAL UNIT     , INACT----D
IST089I P000C    TYPE = PHYSICAL UNIT    , ACTIV
IST089I X000C02  TYPE = LOGICAL UNIT     , ACTIV
IST314I END
D NET,ID=L040,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L040, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED            , CONTROL = SDLC
IST134I GROUP = G3174, MAJOR NODE = FRACKX2
IST172I NO NETWORK NODES EXIST
```

Figure 3-11. Console Listing Showing Display of Lines after MOVE DRDS

Figure 3-12 on page 3-16 shows the VTAM Definition Library member, DRBACK, which contains the necessary statements to MOVE PU P040A and its LUs back to Line L040 and to activate them at the same time. The console listing showing the activation of this DRDS deck is also shown.

```
VTAM Definition Library--Member DRBACK

          VBUILD TYPE=DR
          MOVE FROM=L000,TO=L040
 P040A    PU  ACTIVATE=YES
```

```
V NET,DRDS,ID=DRBACK
IST097I VARY ACCEPTED
IST487I VARY ACT FOR ID = P040A SCHEDULED BY VARY DRDS
IST670I VARY DRDS PROCESSING FOR ID = DRBACK COMPLETE
IST872I DR MOVE MISMATCH DETECTED FOR P040A
IST523I REASON = RESOURCE WAS MOVED FROM L040, NOT L000
IST093I P040A ACTIVE
D NET,ID=L000,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L000, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED            , CONTROL = SDLC
IST134I GROUP = G3276, MAJOR NODE = FRACKX2
IST084I NETWORK NODES:
IST089I P000C     TYPE = PHYSICAL UNIT    , ACTIV
IST089I X000C02   TYPE = LOGICAL UNIT     , ACTIV
IST314I END
D NET,ID=L040,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L040, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED            , CONTROL = SDLC
IST134I GROUP = G3174, MAJOR NODE = FRACKX2
IST084I NETWORK NODES:
IST089I P040A     TYPE = PHYSICAL UNIT    , ACTIV----D
IST089I X040A02   TYPE = LOGICAL UNIT     , ACTIV----D
IST089I X040A03   TYPE = LOGICAL UNIT     , ACTIV----D
IST089I X040A04   TYPE = LOGICAL UNIT     , ACTIV----D
IST089I X040A05   TYPE = LOGICAL UNIT     , ACTIV----D
IST089I X040A06   TYPE = LOGICAL UNIT     , ACTIV----D
IST089I X040A07   TYPE = LOGICAL UNIT     , ACTIV----D
IST089I X040A08   TYPE = LOGICAL UNIT     , ACTIV----D
IST089I X040A09   TYPE = LOGICAL UNIT     , ACTIV----D
IST089I X040A00   TYPE = LOGICAL UNIT     , ACTIV----D
IST314I END
```

Figure 3-12. Example of Console Listing with MOVE & Activate DRDS Deck

With VTAM V3R2 the following parameters can be coded on PU statements to be dynamically added:

- ADDR
- PUTYPE
- BNNSUP
- ANS
- MAXOUT

- PASSLIM
- IRETRY
- MAXDATA
- XID
- DATMODE

The following parameters can be coded on LU statements to be dynamically added:

- LOCADDR
- PACING
- BATCH

VTAM only parameters such as ISTATUS can also be coded in a DRDS definition dataset. For any NCP parameters not listed above, NCP default values are used. MAXDATA and LOCADDR must be coded in the deck because there is no default value for these.

Figure 3-13 on page 3-18 shows an example of ADDing a PU and LU using a DRDS deck and the new ability to code DATMODE = FULL in the deck.

```
VTAM Definition Library -- Member DRADD3

         VBUILD TYPE=DR
         ADD TO=L000
 P000C   PU  ADDR=C3,MAXDATA=265,PUTYPE=2,XID=YES,DATMODE=FULL
X000C02  LU  LOCADDR=02,DLOGMOD=M3276,PACING=1
```

```
D NET,ID=L000,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L000, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED             , CONTROL = SDLC
IST134I GROUP = G3276, MAJOR NODE = FRACKX2
IST084I NETWORK NODES:
IST089I P000C    TYPE = PHYSICAL UNIT    , INACT
IST089I X000C02  TYPE = LOGICAL UNIT     , INACT
IST314I END
F CSSVTAM,DR,TYPE=DELETE,ID=P000C,FROM=L000
IST097I MODIFY ACCEPTED
IST241I F DR DEL COMMAND COMPLETE FOR P000C
D NET,ID=L000,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L000, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED             , CONTROL = SDLC
IST134I GROUP = G3276, MAJOR NODE = FRACKX2
IST172I NO NETWORK NODES EXIST
IST314I END
V NET,DRDS,ID=DRADD3
IST097I VARY ACCEPTED
IST487I VARY ACT FOR ID = P000C SCHEDULED BY VARY DRDS
IST670I VARY DRDS PROCESSING FOR ID = DRADD3 COMPLETE
IST093I P000C ACTIVE
D NET,ID=L000,E
IST097I DISPLAY ACCEPTED
IST075I NAME = L000, TYPE = LINE
IST486I CURRENT STATE = ACTIV, DESIRED STATE = ACTIV
IST087I TYPE = LEASED             , CONTROL = SDLC
IST134I GROUP = G3276, MAJOR NODE = FRACKX2
IST084I NETWORK NODES:
IST089I P000C    TYPE = PHYSICAL UNIT    , ACTIV----D
IST089I X000C02  TYPE = LOGICAL UNIT     , ACTIV----D
IST314I END
D NET,ID=P000C,E
IST097I DISPLAY ACCEPTED
IST075I NAME = P000C, TYPE = PU_T2
IST486I CURRENT STATE = ACTIV----D, DESIRED STATE = ACTIV
IST081I LINE NAME = L000, LINE GROUP = G3276, MAJNOD = FRACKX2
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST355I LOGICAL UNITS:
IST080I X000C02  ACTIV----D
IST314I END
```

Figure 3-13. Example of Console Listing with ADD DRDS Deck

The NCP must be generated correctly to allow for Dynamic Reconfiguation to take place. Following is a list of the NCP V4R3 or NCP V5R2 definition statements needed to support Dynamic Reconfiguration.

Table 3-2. NCP V4R3 and NCP V5R2 Dynamic Reconfiguration Statements

DEFINITION STATEMENT	OPERAND	EXPLANATION
BUILD	DR3270	Specify YES if SDLC 3270 PU T1s have not been generated but may be DR added
PUDRPOOL	NUMBER	Number of PU control blocks needed for dynamic addition of PUs
LUDRPOOL	NUMTYP1	Number of PU T1 LU control blocks for DR addition and SDLC switched
LUDRPOOL	NUMTYP2	Number of PU T2 LU control blocks for DR addition and SDLC switched
LUDRPOOL	NUMILU	Number of Independent LU control blocks for DR addition and SDLC switched
LINE	MAXPU	Number of generated PUs on this line plus number to be dynamically added
SERVICE	MAXLIST	Number of entries in Service Order Table--needs to include number for DR added PUs
PU	PUDR	Code YES if this PU may be deleted dynamically
LU	LUDR	Code YES if this LU may be deleted dynamically

With VTAM V3R2 and NCP V4R3 or V5, a PU retains the same network address when it is moved from one line to another. The same control blocks are used, and the pointers updated to reflect the correct line. Additionally, if one resource is DR DELETED, its address can be reused for another DR ADDED resource. The need for RESOEXT, therefore, has disappeared and RESOEXT is no longer a valid parameter in NCP V4R3/V5. Also, the MAXLU operand is invalid on PU or PUDRPOOL statements. The maximum number of LUs allowed per PU is dependent on the addressing capabilities of the node.

3.2.3 Implicit Dynamic Reconfiguration

Implicit Dynamic Reconfiguration is the third enhancement to Dynamic Reconfiguration in VTAM V3R2. This is the ability for the user to code additional T1, T2, or T2.1 PU and LU NCP definition statements in the NCP source deck contained in the VTAM Definition Library. Upon activation of the NCP, any PU and/or LU statements encountered in the NCP source deck will be dynamically added by

VTAM V3R2, providing the NCP has been generated with sufficient resources as documented in Table 1-1 or Table 1-2.

Figure 3-14 on page 3-21 shows our NCP source deck for the Line L000 and two 3276s, PU P000C/LU X000C03 and PU P000D/LU X000C02. When we generated this NCP, FRACKX3, the statements for PU P000C and LU X000C03 were not included. The PU and LU statements were added to the source deck after the generation was completed.

Following the NCP source deck, the console listing showing the activation of FRACKX3 is shown. Note that there is no special message indicating that PU P000C is being dynamically added. The PU is shown as ACTIVE. A subsequent display of PU P000C shows its status as "**ACTIV----D**," dynamically reconfigured.

```
*-------------------------------------------------------------------*
*        LINE GROUP  'G3276'                                        *
*-------------------------------------------------------------------*
*
G3276    GROUP CLOCKNG=EXT,DIAL=NO,DLOGMOD=M3276,DUPLEX=FULL,LNCTL=SDLC*
               ,LOGTAB=LOGONTAB,MAXDATA=265,MAXOUT=7,MODETAB=AMODETAB,N*
               PACOLL=YES,PACING=0,PASSLIM=7,PAUSE=0.3,PUTYPE=2,REPLYTO*
               =5.0,RETRIES=(5,2,4),SPEED=9600,SSCPFM=USSSCS,TYPE=NCP,U*
               SSTAB=USS3270A,VPACING=0
*
L000     LINE ADDRESS=(000,HALF),LPDATS=LPDA1,ISTATUS=ACTIVE,MAXPU=3
         SERVICE ORDER=(P000D),MAXLIST=3
*
P000C    PU    ADDR=C3,PUTYPE=2,ISTATUS=ACTIVE,                         +
               PUDR=NO,                                                 +
               XID=NO
X000C03  LU    LOCADDR=02,                                              +
               ISTATUS=ACTIVE
*
P000D    PU ADDR=C4,PUTYPE=2,ISTATUS=INACTIVE,PUDR=NO,XID=YES
X000C02  LU LOCADDR=02,ISTATUS=INACTIVE
```

```
V NET,ACT,ID=FRACKX3
IST097I VARY ACCEPTED
IST461I ACTIVATE FOR U/RNAME ENTRY ID = 011-S STARTED
IST464I LINK STATION 011-S HAS CONTACTED FRACKX3 SA 107
IST093I FRACKX3 ACTIVE
IST093I PNPA ACTIVE
IST093I P000C ACTIVE
IST093I P040A ACTIVE
IST093I TRNPU1 ACTIVE
IST093I TRNPU2 ACTIVE
IST380I ERROR FOR ID = L124 - REQUEST: ACTLINK, SENSE: 08220000
D NET,ID=P000C
IST097I DISPLAY ACCEPTED
IST075I NAME = P000C, TYPE = PU_T2
IST486I CURRENT STATE = ACTIV----D, DESIRED STATE = ACTIV
IST081I LINE NAME = L000, LINE GROUP = G3276, MAJNOD = FRACKX3
IST654I I/O TRACE = OFF, BUFFER TRACE = OFF
IST314I END
```

Figure 3-14. Example of Implicit Dynamic Reconfiguration

3.3 Migration Considerations for Dynamic Reconfiguration

In order to use the new "MOVE" command or "MOVE" in a DRDS deck, VTAM must be at a V3R2 level and NCP must be V4R3 or V5R2. The same VTAM/NCP requirements are true in order to use any new statements in a DRDS deck such as DATMODE= or XID=.

The new VTAM V3R2 MODIFY DR TYPE=DELETE command or Implicit Dynamic Reconfiguration can be used in conjunction with any supported NCP. (NCP V2 or higher).

There is a compatibility PTF available for VTAM V3R1 systems to have a message issued to the operator in case resource mismatches occur between VTAM and NCP. The Software Compatibility Considerations Chapter of this manual contains a description of this PTF.

NOTE: In environments where NCP ownership is shared between two or more VTAMs, extreme care should be exercised when using Dynamic Reconfiguration. Installations should institute procedures to ensure that similar Dynamic Reconfiguration is performed from each owning host.

3.4 Backup/Recovery Considerations

Please see the note in the preceding paragraph. If proper procedures are not in place, it is possible that a backup VTAM will have an NCP source deck and RRT indicating that a certain resource is a particular network address. If a previous owning VTAM had DR DELETED that resource and DR ADDED a new resource, the backup VTAM would have no way of knowing that. For this reason, installations need to develop backup procedures to use in combination with Dynamic Reconfiguration.

Chapter 4. Subarea Dial

The next three chapters examine three new design options available for configuring Subarea Networks. These are Subarea Dial (also known as Switched Subarea Support), Subarea Multipoint (also known as Multipoint Subarea Support), and Token-Ring Subarea Support. Although most of the content is applicable to VTAM V3R2, NCP V4R3 and NCP V5R2, most of the examples are based on VTAM V3R1.1 and the NCP V4R2 and V4 Subset Features. This is primarily because of limited documentation and code for the other products at the time of publication of this document.

4.1 Subarea Dial Overview

NCP V4R3 and V5R2 support switched links between subarea nodes. Refreshed copies of NCP V4R2 and the MVS NCP V4 Subset have also been announced which extend this function to those environments. The refreshed versions of NCP V4R2 and the V4 Subset can be obtained by reordering these products. A refreshed copy of SSP V3R2 containing the required SSP Feature can also be obtained by reordering SSP V3R2. See the **NCP and SSP Installation** Section of this Chapter for more information on the NCP and SSP features.

These enhancements allow switched connections between communication controllers as well as 9370s using a Telecommunication Subsystem Controller and 4361s using an Integrated Communications Adapter.

The SNA Switched Subarea support is modeled after SNA Dial Support for NCP boundary node devices (PU1 and PU2 devices). The VTAM Switched Major Node support has been extended to allow NCPs (PU4s) and host nodes (PU5s) to be specified in these definitions in addition to the PU1s and PU2s allowed with the current products.

The connections are established with a new operator command:

VARY NET, DIAL, ID = puname

Where **puname** is the label on the PU4/PU5 macro in the VTAM Switched Major Node definition representing the linkstation to which the connection is to be made.

This command causes a Connect-Out RU containing all of the dial related parameters to flow from VTAM to NCP.

In SNA boundary node dial environments, the dialing operation is initiated either by a terminal operator placing a call or is triggered by an application requesting an LU session. When VTAM discovers that the LU resides on a switched PU, VTAM then performs the dial operation based on information contained in the Switched Major Node definition. In the case of Subarea Dial, the physical connections have to be established prior to attempting any session setups that are dependent on the connection. An operator must initiate the dial activity using the above command.

In general, an autocall unit is required but a dial line can be shared (serially used) by many stations. The one exception to the autocall requirement is the 4941 modem. This modem uses an in-stream dial approach which eliminates the need for a separate autocall interface. The coding considerations for using the 4941 will be discussed in the **Installation Considerations** section.

Once a dial connection is made, NCP contact procedures begin. A major difference between leased line and subarea dial contact procedures is that two XID exchanges take place. The first set is analogous to that for boundary node dial connections. This XID processing insures that the calling station is a legitimate user. In the case of subarea dial, it is the combination of IDNUM, SUBAREA and NETID that provide this unique identification. IDBLK is not coded (or used) for subarea dial as it is for boundary nodes. The same IDNUM value must be specified in the VTAMs at each end of the switched connection. The details of this will be addressed in the **Coding Considerations** discussion.

After a successful security XID exchange, the subareas involved enter normal configurable mode XID processing. Communications parameters are exchanged and the primary/secondary link station roles are established.

For NCP-to-NCP connections, the normal configurable station rules apply and the highest subarea will become the primary.

For VTAM-to-NCP and VTAM-to-VTAM connections, the calling station will become the primary station.

At the conclusion of this second XID processing, CDRM sessions can be established and normal cross domain/net traffic can flow. From this point on the flows are normal IRN flows until it is time to break off the connection.

The connection can be broken off in one of two ways. The first is a result of inactivity and will be discussed in the **Installation Considerations** section of this document in connection with a new parameter, **BRKCON**.

The second method of terminating the connection is with the operator command:

VARY NET,HANGUP,ID = puname

Where **puname** is the label on the PU4/PU5 macro in the Switched Major Node definition representing the linkstation.

4.2 Supported Environments

Figure 4-1 on page 4-5, Figure 4-2 on page 4-5, and Figure 4-3 on page 4-5 depict the supported Subarea Dial configurations.

In these figures the VTAM subareas at either end can initiate the switched connection but either a 4941 modem or an Autocall unit is required at the calling end.

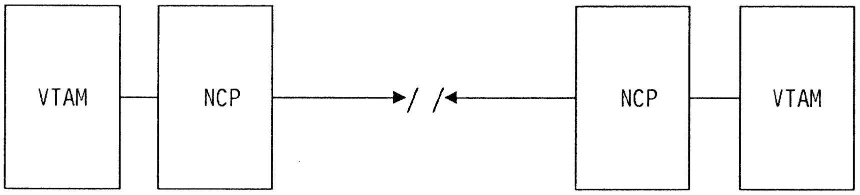


Figure 4-1. Subarea Dial: NCP-to-NCP. Switched connection

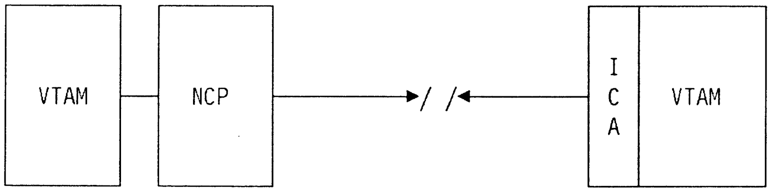


Figure 4-2. Subarea Dial: NCP-to-VTAM/ICA. Switched connection

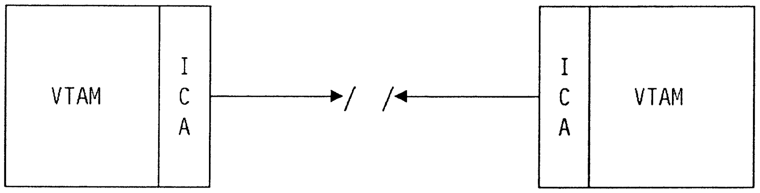


Figure 4-3. Subarea Dial: VTAM/ICA-to-VTAM ICA. Switched connection

The configurations in Figure 4-1, Figure 4-2, and Figure 4-3 are supported in the MVS, MVS/XA, VM and VSE environments; the VTAM and NCPs must be the following releases in order to be participants in the Subarea Dial:

VTAM V3R1.1 with PTFs (MVS, see the Product Installation topic)

VTAM V3R1.2 (VM/SP or VSE)

VTAM V3R2 (MVS, VM/SP, or VSE)

NCP V4R2 with the ACF/NCP V4R2 Feature (MVS)

NCP V4 Subset with the ACF/NCP Subset Feature (MVS)

NCP V4R3 (MVS, VM/SP or VSE)

NCP V5R2 (MVS, VM/SP or VSE)

4.3 Installation Considerations

4.3.1 Coding Considerations

Subarea Dial support is modeled after Boundary Node Dial and hence the VTAM and NCP coding is similar to that environment. The following discussion examines the VTAM and NCP coding required to implement Subarea Dial.

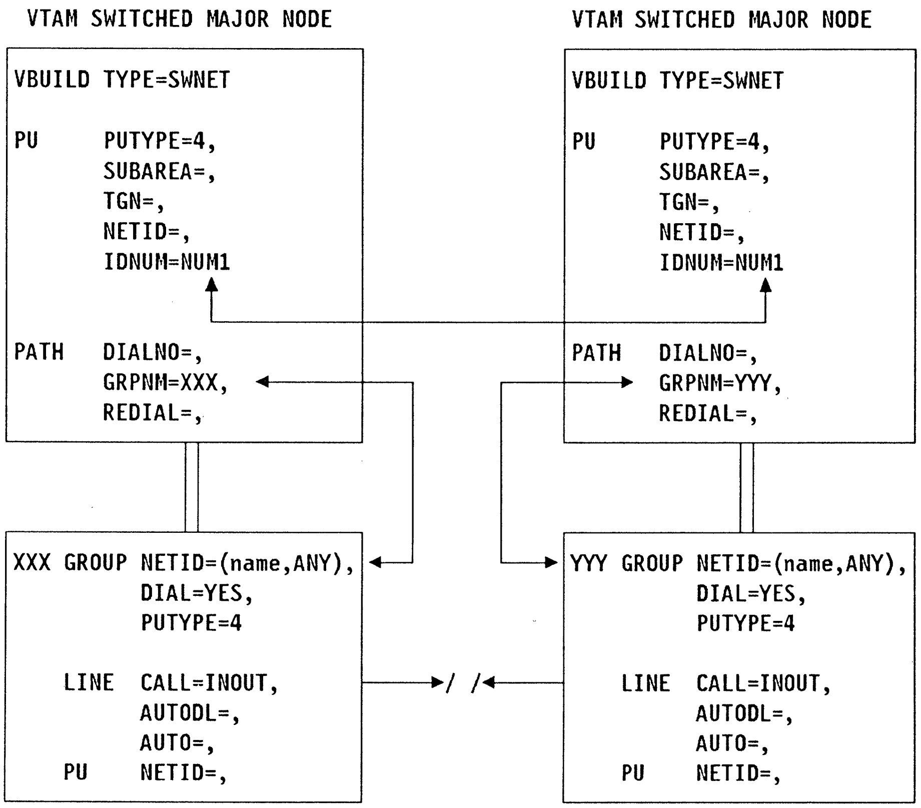
The Switched Major Node definition contains the critical parameters for all SNA dial implementations. These definitions are filed in VTAMLST and contain parameters that would normally be coded on the PU and LU definitions in an NCP or communications adapter major node for leased line environments. The corresponding definitions in the NCP or Communications Adapter major node contains a GROUP, LINE and a dummy PU macro. After completion of a call, VTAM uses the information contained in the received XID to locate the appropriate Switched Major Node by comparing the received XID information with that coded in the list of active Switched Major Nodes.

When a match is found, VTAM dynamically updates the dummy PU control block coded in the NCP or Communications adapter node using the information from the Switched Major Node definition. For boundary nodes (PU1/PU2), the appropriate LU control blocks are also built using the information contained in the Switched Major Node definition.

One major difference between the coding required for Subarea Dial and the coding for Boundary Dial is that a Switched Major Node definition is required at each end of the subarea connection.

In the case of an NCP-to-NCP connection, a Switched Major Node definition would appear in each of the owning VTAMs, and each of the NCPs would contain a GROUP, LINE and dummy PU definition.

See the coding prototype in Figure 4-4 on page 4-7 for an NCP-to-NCP connection.



VTAM SWITCHED MAJOR NODE

```
VBUILD TYPE=SWNET

PU      PUTYPE=4,
        SUBAREA=,
        TGN=,
        NETID=,
        IDNUM=NUM1

PATH    DIALNO=,
        GRPNM=XXX,
        REDIAL=,
```

```
XXX GROUP NETID=(name,ANY),
          DIAL=YES,
          PUTYPE=4

    LINE  CALL=INOUT,
          AUTODL=,
          AUTO=,
    PU    NETID=,
```

VTAM SWITCHED MAJOR NODE

```
VBUILD TYPE=SWNET

PU      PUTYPE=4,
        SUBAREA=,
        TGN=,
        NETID=,
        IDNUM=NUM1

PATH    DIALNO=,
        GRPNM=YYY,
        REDIAL=,
```

```
YYY GROUP NETID=(name,ANY),
          DIAL=YES,
          PUTYPE=4

    LINE  CALL=INOUT,
          AUTODL=,
          AUTO=,
    PU    NETID=,
```

Figure 4-4. Subarea Dial Prototype coding. NCP-to-NCP

As with boundary node dial, NCP contains a GROUP, LINE and a place holder PU statement to define the dial port.

The following discussion reviews the dial related VTAM and NCP parameters and indicates which have been changed or require a different interpretation in the Subarea Dial environment.

4.3.2 NCP Switched Line Parameters

Most of the coding required to implement Subarea Dial involves the GROUP, LINE and PU macros; however, the **VERSION** parameter on the **BUILD** macro must be specified as one of the following releases in order to allow NDF to accept the Subarea Dial parameters:

VER=4R2F For V4R2 and V4 Subset Feature

VER = V4R3 For NCP V4R3

VER = V5R2 For NCP V5R2.

Note: NCP V5R1 does not support subarea dial.

The prototype coding and dial related parameters coded in NCP are:

```
GROUP   ACTIVTO=    (no change but see note 1)
        DIAL=       (no change)
        PUTYPE=     (PU4 and PU5 can now be specified)
        NETID=(name,ANY) (changed see note 2)
LINE    AUTO=       (changed see note 3)
        CALL=       (no change)
        AUTODL=     (switched X.21 only)
        LPDA2DL=    (new for 4941 see note 4)
        RING=       (no change)
PU      BRKCON=     (new see note 1)
        NETID=      (must match the NETID of the attaching linkstation)
        TGN=        (no change)
        .
        .
```

Notes:

1. ANS = CONT can be coded for PU4s and PU5s for VTAM V3R1.2 and VTAM V3R1.1 with the proper maintenance. ANS = CONT can be coded for any PUTYPE with VTAM V3R2 in order to support switched session continuation. This parameter can be used in conjunction with BRKCON and ACTIVTO to control the breaking of the connection. See the discussion of the BRKCON parameter in the **Design Considerations** section of this document.
2. NETID is only required if this is an SNI connection. This format of the NETID parameter allows switched connections to multiple networks through a single port. See the NETID discussion in the Restrictions and Design Considerations section of this document. The value coded in "name" must be the same as that for the native network (the NETID specified on the BUILD macro) or the NETID coded on a NETWORK statement contained in the NCP. This "name" will be assumed to be the NETID of the calling subarea unless the calling subarea specifies a different NETID in the enhanced XID2. This requires NCP V4R3 or NCP V5R2 or higher.
3. AUTO = YES should be coded if a 4941 is being used instead of an autocall unit. If an autocall unit is being used AUTO = (XXX) should be coded where XXX is the line number of the autocall interface.

4. This parameter must be coded if a 4941 auto dialing modem is to be used instead of an autocall unit. If a 4941 modem is used the following parameters should be coded as specified:

   LPDA2DL = YES

   AUTO = YES

   RING = YES

   LPDA2 = NO (or omitted)

   LPDA2DL should be omitted if a 4941 is not being used.

4.3.3 VTAM Switched Major Node Parameters

The following coding prototype should be filed in Switched Major Node definitions at each end of the subarea link:

```
VBUILD  TYPE=SWNET
        MAXGRP=    (no change)
        MAXNO=     (no change)
PU      ANS=       (ANS= can be specified on PU4s and PU5's)
        IDNUM=     (no change, but must match IDNUM at the other end of
                   subarea link)
        IDBLK=     (not required for subarea dial)
        NETID=
        SUBAREA=
        PUTYPE=    (PUTYPE 4 and 5 can now be specified)
PATH    DIALNO=    (no change)
        GID=       (no change)
        GRPNM=     (no change)
        REDIAL=    (no change)
        PID=       (no change)
```

4.3.4 Unsupported Parameters in a Switched Subarea Environment

The following is a list of macros and the parameters which cannot be specified in a Switched Subarea environment.

GROUP

MAXLU (LUs are not applicable to PU4/PU5 definitions)

SHOLD (X.21 Short Hold Mode is not supported for switched subarea connections.

LINE

ISTATUS (The operator must issue a VARY ACT for the switched link)

MAXLU (samė as on the GROUP macro above)

PU (NCP and CA major node)

ISTATUS (no need to issue an ACTIVATE)

TGN (this information is taken from the Switched Node def.)

MAXLU (same as on the GROUP macro above)

PU (switched Major Node)

IDBLK

ISTATUS (The operator must issue a VARY ACT for this PU)

SECNET

PATH (Switched Major Node)

SHOLD (not supported for Subarea Dial)

4.3.5 Coding Example

The following discussion reviews some sample coding for implementing Subarea Dial and is only intended to be used as a guide. The following is an example of the coding that would reside in the subarea nodes in Figure 4-5 on page 4-11.

The configuration is an NCP-to-NCP Switched Subarea connection where the NCP associated with the VTAM in HOST1 is NCP9. The NCP associated with the VTAM in HOST2 is NCP7. The switched connection between the two NCPs operates at 4800 bps.

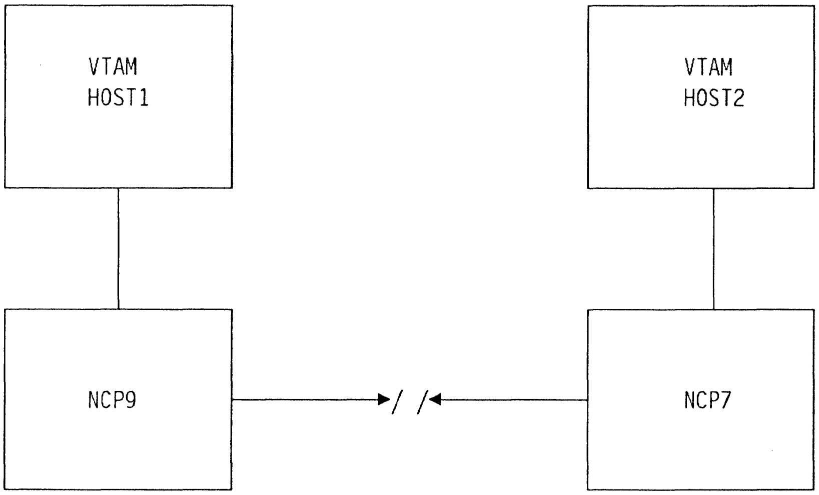


Figure 4-5. Subarea Dial Sample Configuration. NCP to NCP Coding sample

```
VTAM SWITCHED MAJOR NODE HOST1

MVSSMN   VBUILD TYPE=SWNET,MAXGRP=6,MAXNO=6

NCPFE    PU    NETID=NETZ,
               MAXDATA=265,
               SUBAREA=007,
               IDNUM=00001,
               MAXPATH=2,
               PUTYPE=4,
               TGN=1,
               ANS=CONT


         PATH  DIALNO=5428,
               GID=2,
               PID=2,
               REDIAL=4,
               GRPNM=G02SADGR
```

Figure 4-6. Switched Major Node definition for HOST1. Subarea Dial example

```
G02SADGR GROUP ACTIVTO=480.0,
               ISTATUS=ACTIVE,
               DIAL=YES,
               PUTYPE=4,
               LNCTL=SDLC,
               BRKCON=CONNECTO,
               REPLYTO=(3.0,30.0),
               TYPE=NCP
L02008   LINE ADDRESS=008,
               SPEED=4800,
               CALL=INOUT,
               CLOCKNG=EXT,
               AUTO=25,
               DUPLEX=HALF,
               SDLCST=(S02PRI,S02SEC),
               NRZI=YES,
               RETRIES=15,
               TRANSFR=41
P02008A  PU AVGPB=256,
               PUTYPE=4,
               NETID=NETZ
```

Figure 4-7. NCP9 Definitions. Subarea Dial example

```
  VTAM SWITCHED NODE DEFINITIONS HOST2

MVSSMN   VBUILD TYPE=SWNET,MAXGRP=6,MAXNO=6

NCP1D    PU    NETID=NETZ,
               ANS=STOP,
               MAXDATA=265,
               SUBAREA=009,
               IDNUM=00001,
               MAXPATH,
               PUTYPE=4,
               TGN=1


         PATH  DIALNO=5451,
               GID=2,
               REDIAL=4
               GRPNM=G02SNAD3
```

Figure 4-8. Switched Major Node Definitions for HOST2. Subarea Dial example

```
G02SNAD3 GROUP ACTIVTO=240.0,
               ISTATUS=ACTIVE,
               DIAL=YES,PUTYPE=4,
               LNCTL=SDLC,BRKCON=NONE,
               REPLYTO=(3.0,30.0),
               TYPE=NCP

L02006   LINE  ADDRESS=006,
               SPEED=4800,CALL=INOUT,
               CLOCKNG=EXT,BRKCON=NONE,
               DUPLEX=HALF,AUTO=15,
               SDLCST=(S02PRI,S02SEC),
               NRZI=YES,
               RETRIES=15,
               TRANSFR=41
P02006A  PU    AVGPB=256,
               PUTYPE=4
```

Subarea Dial example

Figure 4-9. NCP7 Definitions.

4.3.6 Operations

Subarea Dial connections are established with a new operator command:

VARY NET, DIAL, ID = puname

Where **puname** is the label on the PU4/PU5 macro in the VTAM Switched Major Node definition representing the linkstation to which the connection is to be made.

This command causes a Connect-Out RU containing all of the dial related parameters to flow from VTAM to NCP.

Before the **VARY DIAL** can be issued the following resources must be activated at both ends of the LINK:

- Switched Major Node.
- Switched PU must be connectable (VARY ACT issued against the PU contained in the Switched Major Node definition)
- NCP switched link

In addition, the Dial-Out Path must be enabled on the calling VTAM by issuing a **VARY NET,PATH=USE** command and the line in the called NCP should be enabled for "answer" by issuing a **VARY NET,ANS=ON,ID=line_name**.

The dial connection can be broken by issuing either of the following commands:

V NET,HANGUP,ID=puname

V NET,INACT,F,ID=linename

where **puname** is the name of the PU in the VTAM Switched Major Node definition and **linename** is the NCP line name. Both of these commands take effect immediately and can be issued from either end of the connection.

4.3.7 Design Considerations

4.3.7.1 General

The following discussion enumerates some of the characteristics of Subarea Dial support and the impact of these on network design. The purpose of this discussion is to provide additional information for the planning and use of the Subarea Dial functions.

NCP ownership over a switched link is not supported unless there is at least one leased connection between the NCPs at the time that the **VARY ACT** for ownership is issued.

There may be multiple switched links and multiple leased links in the same (or different) TG between NCPs. Notice that the IDNUM associated with the link stations at the end points of a given subarea link must be different than those at the end points of any other switched links between the same NCP subareas. The combination of IDNUM, NETID, and SUBAREA number must be unique.

Note: For 9370s or 4361s there can only be one link in the TG, leased or switched, between any two adjacent subareas. See the 9370/TSC and 4361/ICA Considerations below.

A remote NCP cannot be loaded or dumped over a Subarea Dial line. This means that a VTAM and a 37xx controller at one location cannot dial a remote location, load and activate an NCP, and begin INN communications.

The announcement documentation stated that an autocall unit is required for Subarea Dial. Because of the way that the 4941 dial support is implemented, it is also supported with Subarea Dial. The 4941 uses an in-stream dial approach that is transparent to VTAM. It uses a subset of the LPDA2 commands (DIAL and DISCONNECT) together with some new code in NCP in such a manner that VTAM is not aware that in-stream dial is being used instead of an autocall unit. The 4941 dial environment is communicated to NCP by specifying AUTO=YES and LPDA2DL=YES LINE parameters. The obvious advantage of the 4941 is the potential for cost reduction since single line interface is needed and no autocall unit is required.

A switched connection cannot be made to a remote unowned NCP. One major reason is that there is no SNA monitor mode facility (SMMF) for switched INN links.

LINK DISCONNECTION

The ANS= parameter can be used to determine the disposition of a switched link after NCP has gone into ANS to the owning VTAM. If ANS=STOP has been specified for the PU currently associated with the link, the line will be disconnected when NCP goes into ANS to the owning host. If ANS=CONT is specified the link will stay up and traffic will continue to flow, however, no other SSCP can activate the link until the connection has been terminated. Even though the loss of the owning SSCP is non-disruptive, there is no equivalent of SSCP takeover for subarea link stations. After ANS processing, the BRKCON parameter determines the circumstances under which the connection will be broken if ANS=CONT was specified. The BRKCON parameter is described below.

Switched links between NCP subareas will not be automatically disconnected when there are no active VR's.

The handling of the switched connection during idle periods is under the control of the user. The user can provide a timer value through the coding of the ACTIVTO parameter. This parameter is used in conjunction with the **BRKCON** parameter to determine whether the subarea link should be disconnected when the time interval between I frames exceed **ACTIVTO**.

The switched connection may be broken by:
Inactivity Timer
SSCP at the opposite end of the link (DISCNT parm coded in that SSCP)
Force INACT of the line

INACTIVITY TIMER

The user can provide a timer value which NCP (or an ICA) will use to determine whether to disconnect a line based on idle conditions. NCP (ICA) will disconnect the line if no traffic has flowed on the line for the specified period. This inactivity period is specified in the ACTIVTO keyword on the GROUP macro for subarea dial lines.

The activity timer usage is specified in a new parameter, **BRKCON**, which can be specified on the GROUP, LINE, or PU statements.

The three options are:

CONNECTO	Enable inactivity timer function at connection time.
NOWNER	Start the inactivity function timer after ANS.
NONE	Ignore the inactivity timer. This is also the default.

FORCE INACT

This command can be issued by any SSCP that owns the NCP containing the switched line. This command will result in a breaking of the dial connection.

DISCNT

If this parameter is coded "DISCNT=YES," the connection will be broken when the last LU session ends and no new sessions have been started before the time interval specified in the zappable constant "RASSDTO" has expired. This constant has a default of 30 seconds.

Note: These disconnection rules only apply to switched connections involving the 9370 Telecommunications Subsystem Controller or a 4361 Integrated Communication Adapter. The connection will not be broken for NCP-to-NCP connections.

It is recommended that "DISCNT=YES" not be coded for VTAM subareas that are acting as an IRN for traffic from other subareas. Any sessions using the connection but not terminating in the IRN VTAM will be terminated when the connection is broken.

4.3.7.2 9370/TSC and 4361/ICA Considerations

Most of the Subarea Dial considerations for 9370/TSC and 4361/ICA are based on the same restrictions that exist for leased lines.

ICA machines are restricted to single link TGs and a single active route between adjacent subareas in SDLC leased line environments. These same restrictions hold for switched subarea links in the 9370/TSC and 4361/ICA environments. A TG number of 1 must be specified for this single link TG just as for the leased line case.

A 9370/TSC or 4361/ICA originating a call can only contact subareas within its own network. The primary reason is that VTAM ICA and TSC support does not provide NETID in the XID. This means that the default NETID of the called port must match that of the calling VTAM.

Note: ICA machines can place a call to a GWNCP and use a GWNCP to establish cross network communications.

4.3.7.3 SNI Considerations

In an SNI environment all of the subarea nodes sharing a dial port must be in the same network when the NCP feature or VTAM ICA connections are involved. In this case the NETID is coded on the PU4/PU5 macro associated with the dial port.

For NCP releases V4R3 and V5R2 and higher, a dial port can be shared by multiple networks. These releases include a NETID in the XID allowing NCP to resolve NETID at station connect time. The NETID, SUBAREA AND IDNUM uniquely define the the link station.

In an SNI environment the **NETID=(name,ANY)** parameter should be coded on the GROUP macro and the LINE macro should be followed with a PU macro representing the link station in the attaching network(s) that will share the port.

The NCP coding would appear as:

```
GROUP   ACTIVTO=
        DIAL=
        PUTYPE=
        NETID=(name,ANY)
LINE    AUTO=
        AUTODL=
        CALL=
        LPDA2D=
        RING=
PU      BRKCON=
        MAXOUT=
        NETID=
        TGN=
        .
        .
```

Note: The NETID on the PUs following the dial group must appear on a NETWORK statement that appears in the NCP or must be equal to the NETID specified on the BUILD macro.

The **NETID=(name,ANY)** on the GROUP macro is interpreted as follows:

name This is the default NETID to be associated with linkstations defined under this GROUP macro. If the ANY parameter is not coded, all of the NETIDs specified on dummy PUs following this GROUP macro must match this **name**.

ANY This parameter allows the dummy PUs following this GROUP macro to represent link stations in any valid network allowing dial ports to be shared by multiple networks.

For Dial-Out, the NETID in the Switched Major Node definition will override the NETID coded on the dummy PU in NCP and this NETID will be sent in the XID.

For Call-IN, the NETID in the received XID will override the NETID coded on the NCP dummy PU. This received NETID must match the NETID in a currently active Switched Major Node definition.

The above discussion applies to NCP V4R3 and V5R2 and higher. There is nothing equivalent to this in VTAM 9370/4361 ICA support, hence, only subareas in the same network as the called 9370/4361 can share such a dial port.

Chapter 5. Multipoint Subarea

5.1 Enhanced Multipoint Support Overview

NCP V4R3 and V5R2 enhance SNA Architecture by allowing the attachment of multiple 4361 and 9370 systems and NCPs to a single SDLC or X.21 line. These functions are also available for NCP V4R2 and the NCP V4 Subset if the NCP V4R2 and NCP V4 Subset *Features* are respectively installed.

This function together with the appropriate level of ACF/VTAM will provide the following:

- Multiple SNA subareas (NCP and VTAM) can be attached to the same non-switched line.
- SNA PU Types 5, 4, 2.1, 2.0 and 1.0 can be attached to the same multipoint line.

This new function allows the same non-switched SDLC multipoint line to be used to attach: host nodes (9370s via a Telecommunications Subsystem Controller or a 4361 via an Integrated Communications Adapter), communication controllers (37xxs), or peripheral nodes (PU T2s, T2.1s, or T1s).

Dynamic Reconfiguration of PU Type 4s and 5s is not allowed for these lines. PU Types 1 and 2 can be added to the line if a corresponding number of these devices have been deleted from the line. The primary reason is that the MAXLST parameter cannot be coded on the SERVICE macro in this environment. This means that the total number of entries in the Service Order Table cannot exceed the number of drops defined on the line at NCP generation time.

Several multipoint lines can exist within the same transmission group. However, 9370s or 4361s only support a single link in a TG.

These functions are discussed from a 9370 perspective in the Washington System Center Bulletin, *9370 Integration into an SNA Networking Environment* (GG66-0277).

The NCP V4R2 Feature and above reintroduces an NCP function that was available with Version 1 NCPs. This function provides the ability to determine by SYSGEN parameters which NCP on a subarea link will be primary and which will be secondary. In a multipoint environment only one station on the line can be the multipoint master (primary). This is not a problem until there is a need to have more than one subarea node on the same line.

In order to effectively implement Multipoint Subarea support there is a need to be able to "predefine" the primary/secondary relationship.

The Multipoint Subarea support is such that once this master station has been defined, all of the subarea nodes on the line are secondary stations and can communicate with this master station; however, the communication between any two secondary stations on the line must flow to the master station first and then on to the other secondary station involved. Therefore, the PATH statements in the tributaries must reflect a traversal through the primary NCP.

5.2 Supported Multipoint Environments

The releases of VTAM and NCP that support the multipoint enhancements are listed below. Note that T2.1 nodes which can be supported on multipoint lines with subarea nodes require VTAM V3R2 and NCP V4R3 or NCP V5R2 in order to operate in T2.1 mode. Otherwise, the T2.1 nodes will be supported as PU Type 2.0 nodes.

If the NCP to which the multipoint link is attached is a gateway NCP and the associated VTAM is a gateway VTAM, then the 9370 can be defined as residing in a different network.

Note: The VTAM owner of the primary NCP must be either VTAM V3R1.1 (MVS) or VTAM V3R2 (MVS, VSE, VM). A 9370 or 4361 using VTAM V3R2 can be associated with a subarea master station but must be used in conjunction with an NCP. The 9370 Transmission Subsystems Controller and 4361 ICA do not support the multipoint subarea function as a primary so an NCP must always be the master station on a line.

The subareas which can be multidropped on the same line are:

VTAM V3R1 (VSE)

VTAM V3R1.1 (VM/SP)

VTAM V3R1.2 (VM/SP or VSE)

VTAM V3R2 (MVS,VM/SP or VSE)

NCP V4R2 (with the ACF/NCP V4R2 feature) MVS only

NCP V4 Subset (with the ACF/NCP Subset feature) MVS only

NCP V4R3 (MVS,VM/SP or VSE)

NCP V5R2

In cases where NCPs are the secondary station, the owning VTAM can be any compatible supported release.

5.3 Installation Considerations

5.3.1 Coding Considerations

There are no new statements associated with Multipoint Subarea support. The most significant change is the reintroduction of the concept of predefined link station primary/secondary relationships. The major coding changes are in the use of (or omission of) existing NCP macros to predefine the primary/secondary relationships of the subareas involved.

Note: Predefined primary/secondary link station support can be used even when Multipoint Subarea support is not being used. This essentially means that on a point to point subarea line the link station role can be determined at generation time and that the stations do not go through configurable state. This support is implemented by coding a single **SDLCST** and **GROUP**macro pair representing the desired link station status (primary/secondary) instead of coding a set for primary mode and a set for secondary mode as is currently done for configurable mode environments.

The NCP coding required to support multipoint subareas is what one would expect, except for those LINE, GROUP and PU parameters that are unique to subarea stations. As discussed above the NCP that is to be designated the multipoint master must be predefined as the the primary. Since this station by definition is the primary, there is no need for the subareas involved to go through any configurable state, and, as such, there is no need to code the configurable station macro SDLCST. The basic coding prototype for defining a multipoint link as it would appear in the primary station is as follows:

```
GROUP  MODE=PRI
LINE
PU     ADDR=A1
PU     ADDR=A2
.
.
```

From the above prototype it is seen that there is one GROUP and one LINE macro representing the line and a PU statement for each drop on the line including any subarea drops.

The prototype coding that would appear in the subarea drops would appear as follows:

```
GROUP  MODE=SEC
LINE   TADDR=A2
PU
```

where **A2** must be equal to the value coded in the **ADDR**= parm of the corresponding PU definition in the primary NCP.

See the example in Figure 5-1 on page 5-6. The figure depicts an NCP master station, NCP1, which has a multipoint line defined that contains an NCP drop, NCP2, with a station address of A1 and a 9370 drop which has a station address of A2.

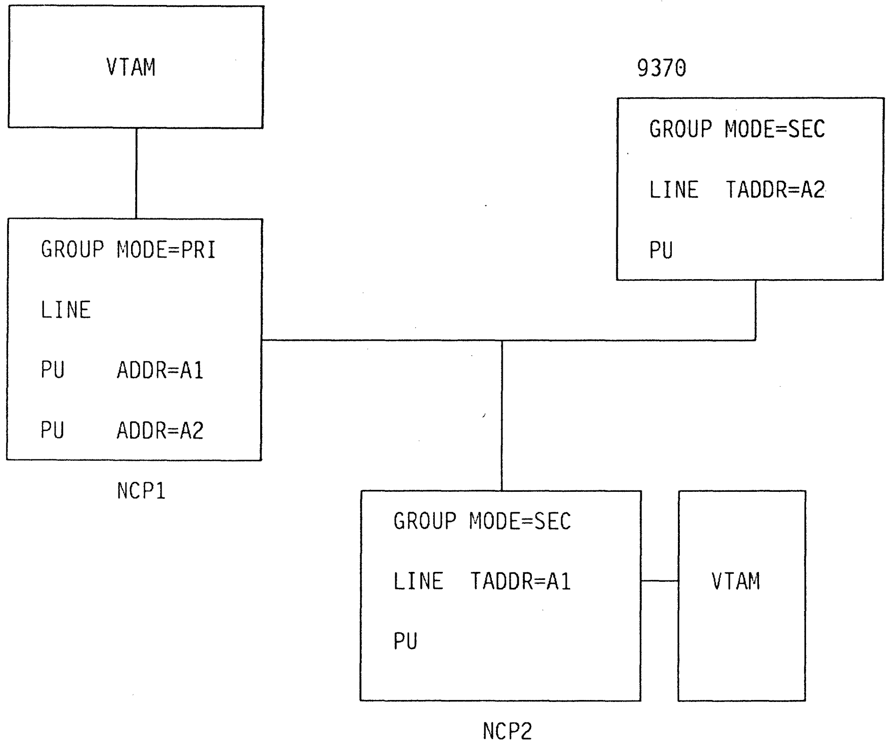


Figure 5-1. VTAM/NCP Multipoint Subarea Support. Prototype Coding

5.3.2 Coding Example

The following is an example of the kind of coding required to implement the multipoint configuration shown in Figure 5-2 on page 5-7. This sample is only provided as a guide as to what parameters are required.

In Figure 5-2 on page 5-7, NCP1 is the multipoint master on a line having two 3174s with polling addresses C1 and C2. NCP2 is the third drop and is a secondary station.

In this example the GROUP, LINE and PU coding in NCP1 is shown in Figure 5-3 on page 5-8.

PU2064A is the PU in NCP1 representing the NCP2 link station. PT64C1 and PT64C2 represent the two 3270 drops.

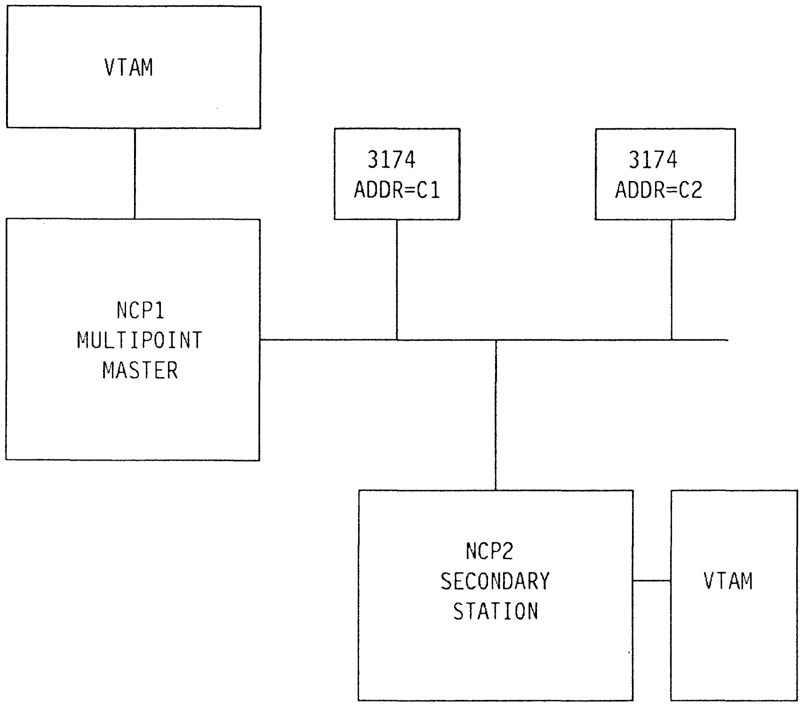


Figure 5-2. Multipoint Subarea Sample Configuration. Coding example

```
G02FRST  GROUP ACTIVTO=60.0,
               DIAL=NO,
               LNCTL=SDLC,
               ANS=CONT,
               MODE=PRI,
               REPLYTO=(3.0,30.0),
               TYPE=NCP
L02064   LINE ADDRESS=(064,FULL),
               SPEED=56000,
               CLOCKNG=EXT,
               DUPLEX=FULL,
               NEWSYNC=NO,
               IPL=NO,
               MONLINK=YES,
               MODULO=8,
               NRZI=YES,
               RETRIES=(5,5,5),
               TRANSFR=41
P02064A  PU AVGPB=256,
               ADDR=01,
               IRETRY=NO,
               PUTYPE=4,
               MAXOUT=7,
               TGN=1,
               NETID=NETZ
PT64C1   PU ADDR=C1,
               IRETRY=NO,
               MAXDATA=265,
               MAXLU=10,
               MAXOUT=7,
               PASSLIM=7,
               PUTYPE=2,
               USSTAB=USSALL,
               ISTATUS=INACTIVE
TT64C1A  LU LOCADDR=1,
               ISTATUS=INACTIVE
TT64C1B  LU LOCADDR=2,
               ISTATUS=INACTIVE
PT64C2   PU ADDR=C2,
               IRETRY=NO,
               MAXDATA=265,
               MAXLU=10,
               MAXOUT=7,
               PASSLIM=7,
               PUTYPE=2,
               USSTAB=USSALL,
               ISTATUS=INACTIVE
TT64C2A  LU LOCADDR=1,
               ISTATUS=INACTIVE
TT64C2B  LU LOCADDR=2,
               ISTATUS=INACTIVE
```

Figure 5-3. NCP1 (Primary end) coding. Multipoint example

Figure 5-4 on page 5-9 is an example of the coding that would appear in NCP2, the secondary subarea station.

The 3174 drops are configured in the same manner as they would be for a multipoint line that did not contain any subarea drops.

```
G02SNAS  GROUP ACTIVTO=420.0,
               DIAL=NO,
               LNCTL=SDLC,
               MODE=SEC,
               REPLYTO=3,
               TYPE=NCP

L02048   LINE  ADDRESS=(048,FULL),TADDR=01,
               SPEED=56000,
               CLOCKNG=EXT,
               DUPLEX=HALF,
               NEWSYNC=NO,
               IPL=YES,
               MONLINK=YES,
               MODULO=8,
               NRZI=YES,
               RETRIES=15,
               TRANSFR=41
P02048A  PU    AVGPB=256,
               IRETRY=YES,
               MAXOUT=7,
               PUTYPE=4,
               TGN=1
```

Figure 5-4. VTAM/NCP Multipoint example. NCP2 (secondary end) coding

5.3.3 Design Considerations

The primary station on a multipoint link must be an NCP. In the previous discussion it was mentioned that in the multipoint environment all of the primary/secondary relationships are predetermined and fixed at system generation time. The central issue is that the primary station must be an NCP although it could be an NCP attached to a 9370 or 4361 running an appropriate level of VTAM.

The secondary stations must code DUPLEX = HALF. The secondary stations on a multipoint link must use Request-for-Send/Clear-to-Send logic. Coding DUPLEX = HALF will enforce this mode of operation for the subarea drops. Other drops on the link such as 3174/3274's should be configured in the same manner as that for operation on any other multipoint line.

For the NCP V4R2 Feature and the V4 Subset Feature the modulo is restricted to 8. NCP V4R3 and NCP V5R2 removes this restriction and the Primary NCP will operate at the the modulo capability of the particular station that it is communicating with.

A remote NCP cannot be a primary on a multipoint line. A primary NCP must be in control of the line at all times, including during loading and dumping. This essentially means that a remote NCP must always be defined as a secondary and must be loaded and dumped by the primary NCP.

The station address specified in the IPL ports table for a 37xx line that will be associated with a secondary station must be equal to the station address specified in the TADDR of the PU representing the link station for the line. This will allow the primary NCP to contact and load the secondary NCP. A lot of thought and planning must be given to environments where secondary NCPs will be link loaded because of the potential performance impact on other drops on the line. The NCP load process is batch oriented and is generally time consuming because of the NCP load module size. The enhancements introduced in VTAM V3R1 and NCP V4R2 which increase the frame size for the load process will reduce the load time, but the most effective means of dealing with this problem is the disk load functions of the 3720 and 3745 environments. When a VTAM host is associated with an NCP subarea drop, a channel load is probably the next best choice for performing the load.

The load may take longer because NCP has to schedule work to other stations on the line.

TGs are supported in the Subarea Multipoint environment. Each PU4/PU5 is on a separate TG. If there is more than one subarea drop on the line there will be more than one TG associated with the line. See Figure 5-5 on page 5-11 and Figure 5-6 on page 5-12 for an example of a multipoint subarea line and some of the resulting PATH definitions. Figure 5-5 on page 5-11 shows a physical view of the line and Figure 5-6 on page 5-12 depicts the logical view of the line and some of the relevant PATH definitions.

These TGs associated with the multidrop line can coexist with other TGs between the multipoint master and the subarea drops such as leased or switched point to point INN links. This support is essentially only available between NCP subarea nodes since the VTAM/ICA environment only supports one single link TG between subareas.

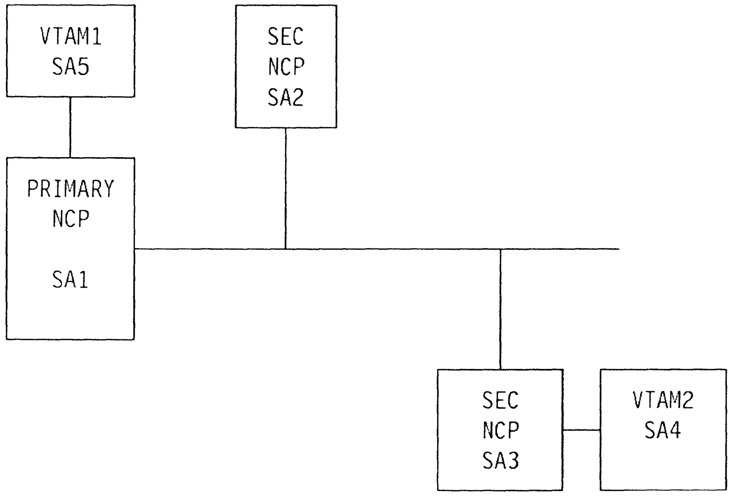


Figure 5-5. Multipoint Subarea Physical View. Pathing Example

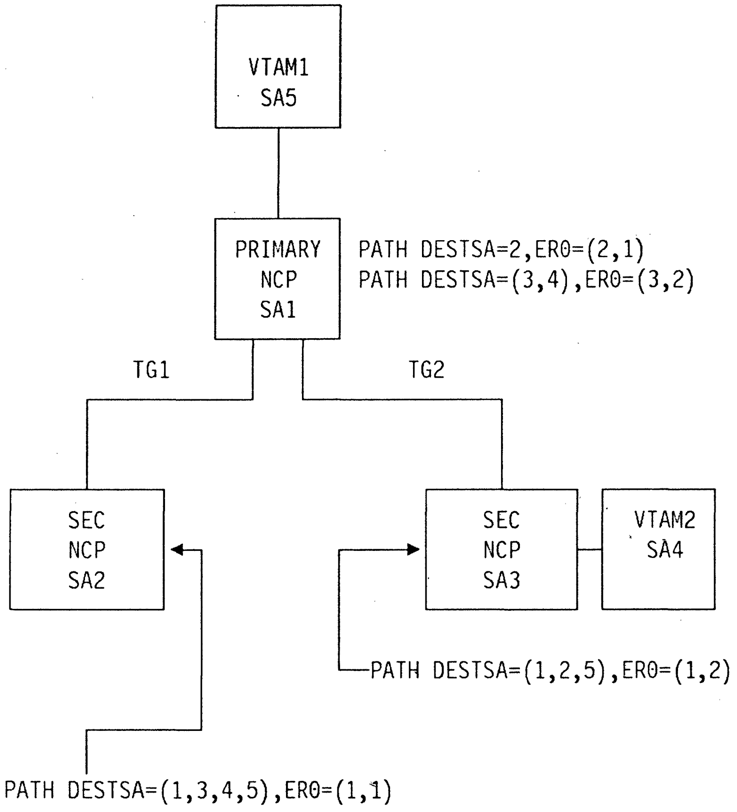

VTAM1
SA5
PRIMARY
NCP
SA1
PATH DESTSA=2,ER0=(2,1)
PATH DESTSA=(3,4),ER0=(3,2)
TG1
TG2
SEC
NCP
SA2
SEC
NCP
SA3
VTAM2
SA4
PATH DESTSA=(1,2,5),ER0=(1,2)
PATH DESTSA=(1,3,4,5),ER0=(1,1)

Figure 5-6. Multipoint Subarea Logical View. Pathing Example

5.4 Product Installation

5.4.1 NCP and SSP Installation.

The installation of the Feature is a reinstallation of NCP V4R2 and SSP V3R2. New users of NCP V4R2 or the Subset will receive refreshed copies of of NCP and SSP code which contains the required code. See the ordering discussion below.

The FMIDs associated with the related products are:

HNC4205 NCP V4R2 Base

JNC4204 NCP V4R2 Feature.

HSS4105 NCP V4 Subset Base

JSS4106	**NCP V4 Subset Feature**
HSP3202	**SSP V3R2 Base**
JSP3205	**SSP V3R2 Feature**

These functions are integrated into V4R3, V5R2, and SSP V3R4 so no additional installation is required to use the functions in these environments.

5.4.2 NCP V4 and V4 Subset Feature Ordering.

There is no new feature number associated with the enhancement feature for either NCP or SSP. The enhancements have been integrated into refreshed releases of these products.

Current users of NCP V4R2 and SSP V3R2 should have received a reorder form by GA time for the refreshed versions of these products. If the reorder form is not available, current users of NCP V4R2 or the V4 Subset must reorder these products in order to obtain the refreshed code. A reinstallation of NCP is required in order to pick up the new functions.

New users of NCP V4R2 or the V4 Subset will automatically receive these enhancements.

As noted above, the SSP V3R2 Feature is also required for installation of the V4 Subset and the V4R2 Features. The above procedures, outlined for NCP, should be followed to get the refreshed code. A reinstall of SSP is required in order to pick up the new functions.

Note: SSP V3R3 does not support the Subarea Dial and Multipoint functions and either SSP V3R2 (with the feature) or SSP V3R4 is required.

5.4.3 VTAM V3R1.1 Requir?d Maintenance

In order to be a multipoint master or to support the Subarea Dial functions the following PTFs need to be applied to MVS VTAM V3R1.1.

MVS/XA	MVS
UY90202	UY90208
UY90203	UY90209
UY90204	UY90210
UY90205	UY90211
UY90206	UY90212
UY90207	UY90213

Note that these PTFs are only available in the MVS and MVS/XA environments and as such the switched and multipoint VTAM support is only available for VTAM V3R1.1 in the MVS and MVS/XA environments. All other environments will require VTAM V3R1.2 or V3R2 as is appropriate.

5.4.4 Required PTFs For VM/VTAM V3R1.1 Environments

For a 9370 or 4361 ICA machine using VM/VTAM V3R1.1 and installed on an NCP multipoint line that has been coded ADDRESS=(xxx,FULL) see APAR VM32126. When NCP is coded as above it sends an XID indicating that it is capable of two way simultaneous transmission. VTAM V3R1.1 and earlier were designed to reject such an XID, primarily because the ICAs were not designed to support duplex data mode. This fix allows the 9370/4361 to accept the XID and indicate in its response that it is capable of two way alternate transmission which the originating NCP will accept and honor. This fix allows these machines to coexist on a line with devices that are capable of duplexing data. This support has been integrated into VTAM V3R1.2 and V3R2.

In conjunction with the VTAM maintenance the 9370 TSC microcode patches should be reviewed for recommended fixes.

5.5 Documentation

The normal VTAM and NCP documentation has been supplemented with the following documents to describe the installation and use of the Subarea Dial and the Multipoint functions.

These are:

SD21-0020	NCP/SSP Supplement for Enhanced Subarea Connectivity
LD21-0021	NCP Reference Supplement for Enhanced Subarea Connectivity
LD21-0019	VTAM V3R1.2 Expanded Networking Capabilities Support

SD21-0020 and LD21-0021 apply to the Feature and NCP V4R2 or the V4 Subset.

The Resource Installation Guide and Reference for NCP V4R3 and NCP V5R2 have been updated to reflect the multipoint enhancements. These publications are:

SC30-3254	ACF NCP Installation and Resource Definition Ref.
SC30-3349	ACF NCP Installation and Resource Definition Guide

Chapter 6. Token-Ring Subarea Support

6.1 Token-Ring Subarea Support Overview

A new function with VTAM Version 3 Release 2 (V3R2) and NCP V4R3.1/V5R2.1 is the ability to use token-ring connections between subareas (between VTAMs and NCPs). The new function is available with the following levels (or higher) of VTAM and NCP:

- VM/VTAM V3R1.2
- VM/VTAM V3R2
- NCP V4R3.1
- NCP V5R2.1

Prior to the new functions Switched Subarea Support, Multipoint Subarea Support and Token-Ring Subarea Support, subareas could **ONLY** be connected using leased point-to-point lines. With the above levels of VTAM and NCP, 9370 processors with token-ring hardware, and 3725, 3720, or 3745 communication controllers with token-ring hardware, can communicate between subareas using token-rings.

Earlier levels of VTAM and NCP supported token-ring hardware on 9370s or communication controllers--however, the type of communication supported was Boundary Network Node (BNN) communication. This means that VTAM or NCP could communicate across a token-ring with peripheral nodes only (physical unit type 2s), using the earlier levels of software.

Figure 6-1 on page 6-4 shows a token-ring with 9370s and 3745, 3725, or 3720 communication controllers attached and communicating with one another using the token-ring. The subareas that can communicate are:

- NCP to NCP
- NCP to 9370
- TSC to TSC (9370)

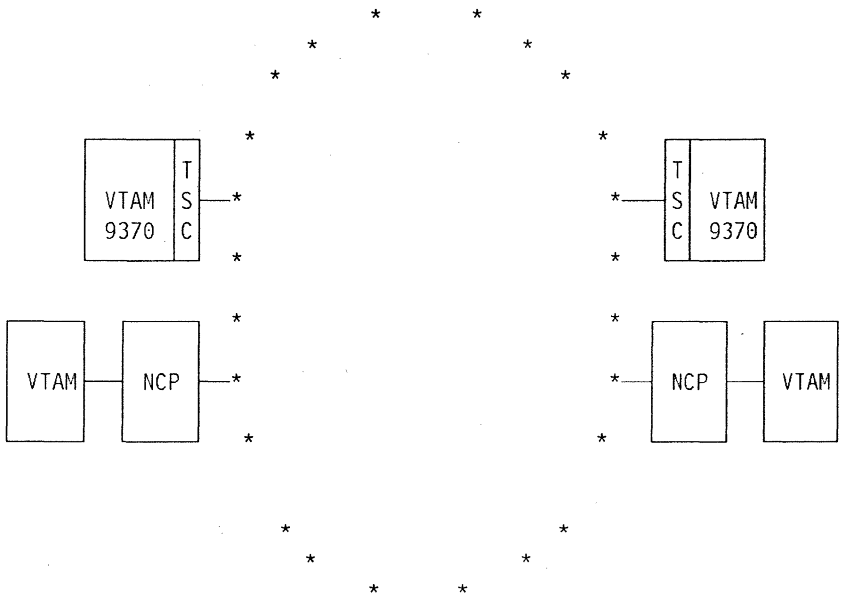


Figure 6-1. Token-Ring INN Support

6.2 Token-Ring INN Restrictions

The following two restrictions must be kept in mind when planning to use token-ring subarea support. These are:

- An NCP cannot be loaded or dumped over the token-ring
- INN and Boundary traffic cannot use the same Token-Ring Adapter Type 1

As shown in Figure 6-1, NCPs using token-ring subarea support need a channel attached VTAM to do the loading and dumping since this is not supported using the token-ring. Another option would be to have the token-ring attached NCP connected via a link to a remote NCP/VTAM which would perform the loading and dumping.

NOTE: With token-ring adapter type 2 (TRA2) and newer software, VTAM V3R3 and NCP V5R3, these two restrictions no longer apply. An NCP can be loaded or dumped over the token-ring using the newer VTAM and NCP, and INN and BNN traffic can use the same token-ring adapter if it is a TRA2 and the newer VTAM and NCP are used.

6.3 Token-Ring INN Planning Considerations

The Token-Ring Subarea Support in VTAM V3R1.2/V3R2 and NCP V4R3.1/V5R2.1 supports subarea to subarea connectivity (PU4-->PU4 or PU5). Connections are represented as a single link TG using multilink protocols. This means that a token-ring INN link cannot be part of the same Transmission Group

(TG) as another INN link connecting to the same destination subarea. Make sure that your token-ring INN link uses a unique Transmission Group Number (TGN). Figure 6-2 on page 6-6 shows a coding example of this.

Large PIUs may need to be segmented and reassembled for transmission over the Token Ring media. This is caused by buffer size limitations on the Token Ring Interface Card (TIC). This limitation is eased with the Token Ring Adapter-2 (TRA2). The TRA2, with a larger buffer available, allows larger PIUs to be transmitted without segmentation.

The following example has not been submitted to any test, is not all inclusive, and is provided for example only.

```
         OPTIONS NEWDEFN=(YES,ECHO)
NCPXYZ   BUILD LOADLIB=xxxxxx,
               MXRLINE=2,      * Exact Number of Physical Lines (TICs)
               MXVLINE=9       * Exact Number of Logical Lines
STPRIM   SDLCST GROUP=PRIMGRP,MODE=PRI
STSECD   SDLCST GROUP=SECDGRP,MODE=SEC
PRIMGRP  GROUP  LNCTL=SDLC,DIAL=NO,MODE=PRI
SECDGRP  GROUP  LNCTL=SDLC,DIAL=NO,MODE=SEC
------------------------------------------------------------------------
GINNP    GROUP  ECLTYPE=(PHYSICAL,SUBAREA),  * Physical INN link
                ANS=CONTINUE                 * Maintain Lines thru ANS
LN1088   LINE   ADDRESS=(1088,FULL),         * Line Addr in 3745 Chassis
                LOCADD=400000000040,   * This TIC's ring station address
                MAXTSL=1108,                 * Recommended Value for TRA1
                RCVBUFC=4095,                * Recommended Value for TRA1
                PORTADD=1                    * User assigned ID (0-99)
                                             * matches PHYPORT below
PU1088   PU
LU1088   LU     ISTATUS=INACTIVE
------------------------------------------------------------------------
GBNNP    GROUP  ECLTYPE=(PHYSICAL,PERIPHERAL)  * Physical BNN link
LN1089   LINE   ADDRESS=(1089,FULL),         * Line Addr in 3745 Chassis
                LOCADD=400000000041,   * This TIC's ring station address
                PORTADD=2                    * User assigned ID (0-99)
                                             * matches PHYPORT below
PU1089   PU
LU1089   LU     ISTATUS=INACTIVE
------------------------------------------------------------------------
GINNL    GROUP  ECLTYPE=(LOGICAL,SUBAREA),   * Logical INN link
                ANS=CONTINUE,                * Maintain Lines thru ANS
                ISTATUS=INACTIVE,            * Don't activate when loaded
                SDLCST=(STPRIM,STSECD),
                PHYPORT=1                    * User assigned ID (0-99)
                                             * matches PORTADD above
INNLN1   LINE
INNPU1   PU     ADDR=04400000000030,   * This is the ring station address
                TGN=2                  * of the adjacent 37XX on the ring
                                       * that this logical link will
                                       * communicate with.  TGN must be
                                       * unique since single link TG only
------------------------------------------------------------------------
GBNNL    GROUP  ECLTYPE=(LOGICAL,PERIPHERAL), * Logical BNN link
                PHYPORT=2,                    * User assigned ID (0-99)
                                              * matches PORTADD above
                CALL=INOUT,                  * NCP or Devices can initiate
                AUTOGEN=8                    * NDF adds LINE and PU pairs
```

Figure 6-2. NCP Generation Example with BNN and INN token-ring definitions

NOTE: For additional logical links to NCP's over this same physical TIC, you must code additional LINE and PU statements, each with unique ADDR values. You can connect to the same adjacent 37XX on a token ring using multiple logical links as long as the TG number for each logical link (TGN = x) is unique.

NOTE these two requirements:

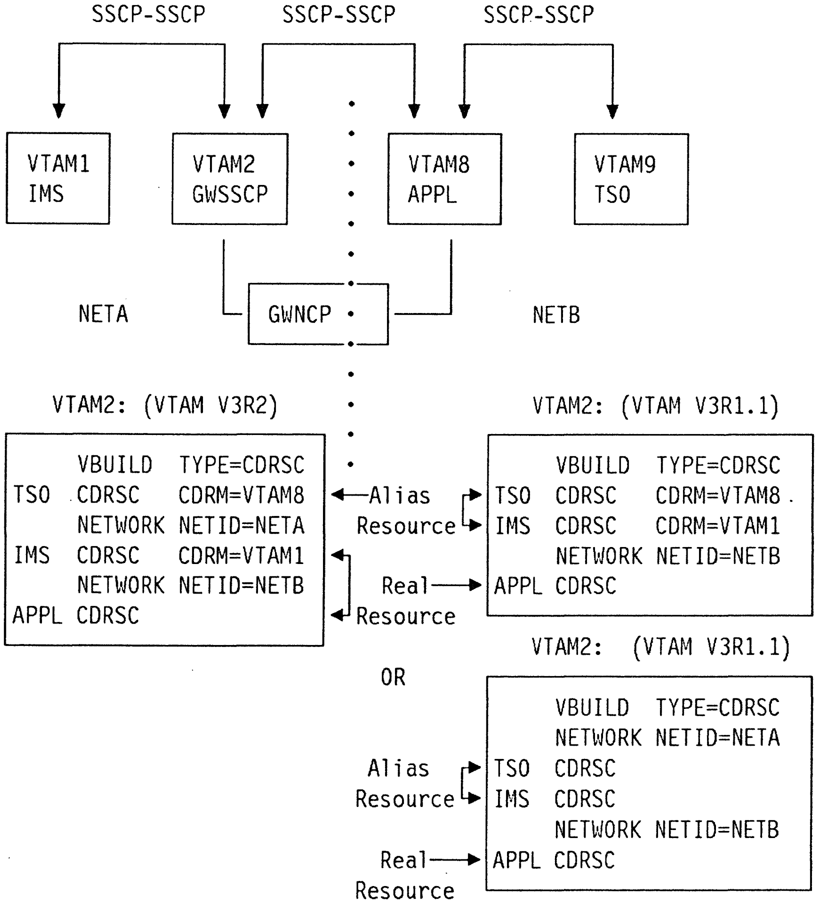
- Each node communicating with NCP over a specific physical link must have a unique ring station address.
- Only Single Link TGs are supported by Token-Ring Subarea Support (multiple token-rings in the same TG are not supported).

Chapter 7. SNI: VTAM V3R2 Considerations

7.1 Defining Real Resources in the Host Network

A new function with VTAM Version 3 Release 2 (V3R2) is the ability to define cross-domain resources (CDRSCs) as **real** resouces in the **same** network. This is a significant improvement over earlier VTAMs for two reasons. First, the overhead incurred by VTAM assuming that same-network resources are alias resources can be avoided. Second, and more importantly, a specific same-network ADJSSCP list can be used for same-network resources, avoiding routing into other networks and allowing more control over ADJSSCP routing. Refer to Section 1.3 for a description of same-network ADJSSCP lists.

In VTAM V3R2 real resources in the same network are defined by preceding the CDRSC statements with a NETWORK statement specifying the same NETID as VTAM. In Figure 7-1, on the left is an example of VTAM2's list of CDRSCs (at a VTAM V3R2 level). In this list, IMS is defined as a real resource in VTAM2's network because it is preceded by a NETWORK statement specifying the same NETID as VTAM2.



VTAM2: (VTAM V3R2)

```
      VBUILD  TYPE=CDRSC
TSO   CDRSC   CDRM=VTAM8
      NETWORK NETID=NETA
IMS   CDRSC   CDRM=VTAM1
      NETWORK NETID=NETB
APPL  CDRSC
```

VTAM2: (VTAM V3R1.1)

```
      VBUILD  TYPE=CDRSC
TSO   CDRSC   CDRM=VTAM8
IMS   CDRSC   CDRM=VTAM1
      NETWORK NETID=NETB
APPL  CDRSC
```

VTAM2: (VTAM V3R1.1)

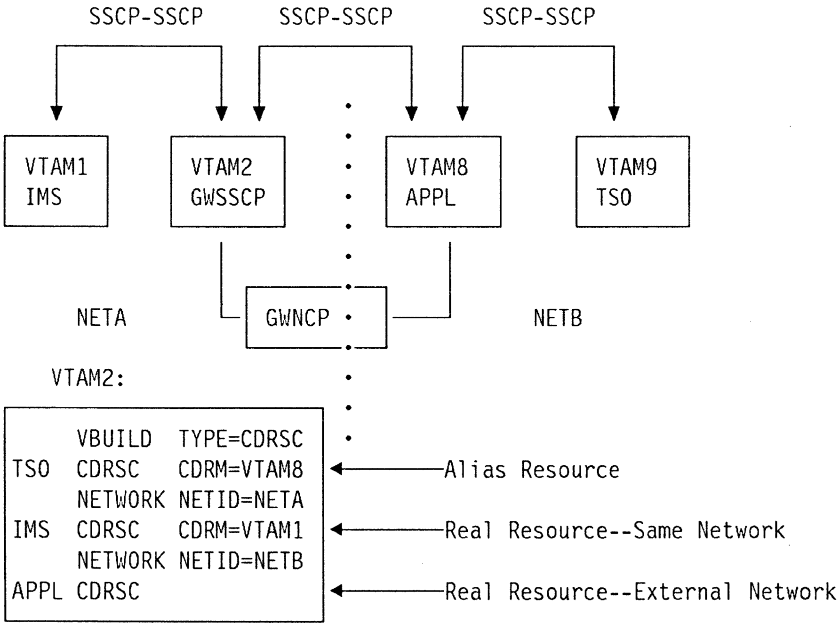
```
      VBUILD  TYPE=CDRSC
      NETWORK NETID=NETA
TSO   CDRSC
IMS   CDRSC
      NETWORK NETID=NETB
APPL  CDRSC
```

Figure 7-1. Example--Real Resources vs. Alias Resources

Although with previous levels of VTAM CDRSCs can be preceded by a NETWORK statement which uses the same NETID as VTAM, these CDRSCs are assumed by VTAM to be alias resources; that is, the resource can reside in any network. (Additionally, if a Netview Alias Name Translation Facility application is active, VTAM calls the Alias application to determine the real name of the resource for CDRSCs coded as alias resources.) With pre-V3R2 VTAM systems, resources that actually reside in the same network must be coded the same way as **alias** resources. In Figure 7-1 on page 7-3, on the right, are examples of how the same list of CDRSCs must be coded at a VTAM V3R1.1 or earlier level. Both TSO and IMS are assumed by VTAM to be alias resources--they are coded with no preceding NETWORK statement in the first example, or following a NETWORK statement with the same NETID as VTAM2 in the second example. The inability to define **real** resources in the same network as VTAM, therefore, can cause overhead in pre-V3R2 VTAM systems.

With pre-V3R2 VTAM systems, two kinds of predefined CDRSCs can exist-- **alias** resources or **real** resources in an external network. This is demonstated in the examples on the right in Figure 7-1 on page 7-3. With VTAM V3R2, three kinds of resources can be defined, **alias** resources, **real** resources in the **same** network as VTAM, and **real** resources in an **external** network. These three kinds of CDRSCs are demonstrated in Figure 7-2.

VTAM uses the way the CDRSC is defined to determine whether to call the Netview Alias Name Translation Facility and to choose an adjacent SSCP (ADJSSCP) list. If the name of another VTAM is coded via the CDRM= operand of the CDRSC definition, the kind of CDRSC definition governs whether the CDRM must be the actual owner of the resource or is to be used as an indication of an SSCP to select for routing purposes.



```
     VBUILD  TYPE=CDRSC
TSO  CDRSC   CDRM=VTAM8
     NETWORK NETID=NETA
IMS  CDRSC   CDRM=VTAM1
     NETWORK NETID=NETB
APPL CDRSC
```

Figure 7-2. Example--three kinds of predefined CDRSCs with VTAM V3R2

Alias Resource:

In Figure 7-2 on page 7-4, the first kind of CDRSC defined, TSO, is known as an **alias** resource because the CDRSC definition statement is not preceeded by a NETWORK statement. TSO could be located in either the same network as VTAM2 or in another network. (If a Netview Alias Name Translation Facility application is active, VTAM2 calls the alias application to determine the real name of the resource before routing the session initiation request to another SSCP.) If the CDRM= operand is coded on the CDRSC statement, it does not have to be the actual owner of TSO. VTAM2 uses the CDRM coded plus the default ADJSSCP list for all networks for routing session initiation requests. In the Figure, VTAM8 is coded on the CDRM= operand of the TSO CDRSC and specifies an SSCP to which the session initiation request can be sent. VTAM8 is gateway-capable and reroutes the session initiation request to VTAM9. Even though VTAM9 is the actual owner of TSO, VTAM2 will allow the session to set up because TSO is defined as an alias resource.

Real Resource, Host Network:

The second kind of CDRSC defined, IMS, is a **real** resource in the host network. This is the capability that is new with VTAM V3R2. Because IMS is defined as **real**, it must be located in the same network, NETA, as VTAM2, and if the CDRM= operand is coded, the named VTAM must be the actual owner of the resource. If IMS were moved to a backup VTAM in NETA, VTAM3 for example, session initiation requests would fail unless the MODIFY CDRM command were first issued at VTAM2 to change to the new owner of IMS, VTAM3.

For routing purposes, VTAM2 first sends session initiation requests to the CDRM coded on the CDRSC statement. If, for some reason, IMS were not found there, VTAM2 would next use the NETA ADJSSCP list for routing purposes. The ability to define a specific ADJSSCP list for the same network, NETA, is also new in VTAM V3R2 and is discussed in Section 7.3.

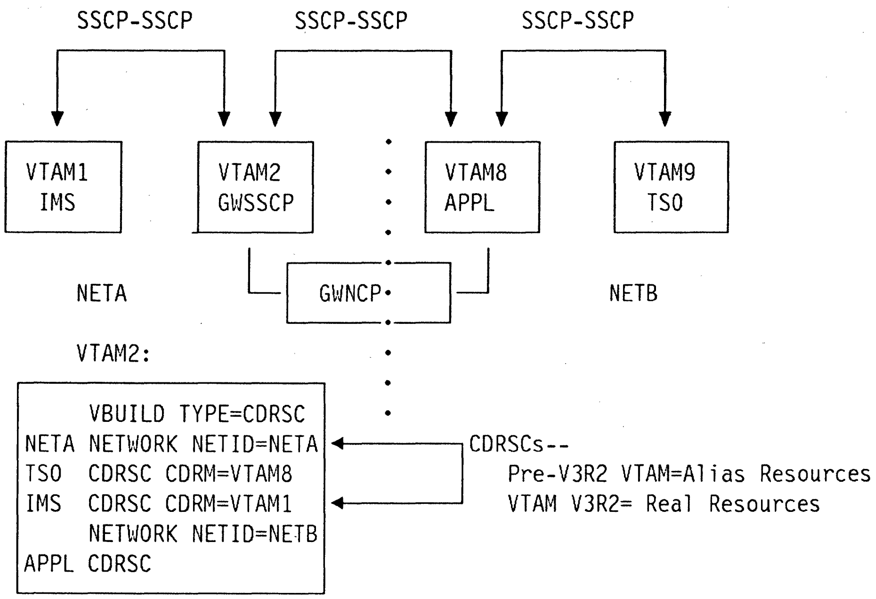
Real Resource, External Network:

The third kind of CDRSC defined, APPL, is a **real** resource in an external network. APPL must be located in the network coded on the NETWORK statement preceeding the APPL CDRSC, NETB. In Figure 7-2 on page 7-4, no CDRM owner is coded. However, if a CDRM= were coded, the CDRM must be the actual owner of APPL. In order to locate CDRSCs following a NETWORK statement for NETB, VTAM2 uses the CDRM, if coded, and the ADJSSCP list for NETB.

7.2 VTAM V3R2 Migration Considerations for CDRSCs

Installations migrating to VTAM V3R2 from previous releases of VTAM should investigate their CDRSC definitions to insure that resources that were previously assumed by VTAM to be **alias** are not assumed by VTAM V3R2 as **real** resources in the host network. Additionally, CDRSCs defined as alias CDRSCs should be examined, and, if they are same network resources, should be coded using the new ability to define real resources in the host network.

Figure 7-3 on page 7-6 illustrates an example of a problem that could arise during VTAM V3R2 migration.



```
     VBUILD TYPE=CDRSC
NETA NETWORK NETID=NETA
TSO  CDRSC CDRM=VTAM8
IMS  CDRSC CDRM=VTAM1
     NETWORK NETID=NETB
APPL CDRSC
```

Figure 7-3. VTAM V3R1 Sample CDRSC Definition

In Figure 7-3, two resources, TSO and IMS, have been defined to VTAM2 as alias resources. With versions of VTAM prior to V3R2, CDRSCs preceded by no network statement, a null network statement, or a NETWORK statement with a NETID of the host network, are assumed to be alias. Therefore, with pre-VTAM V3R2, TSO and IMS can exist in either NETA or NETB, and the CDRM owner does not need to be the actual owning VTAM of the resource.

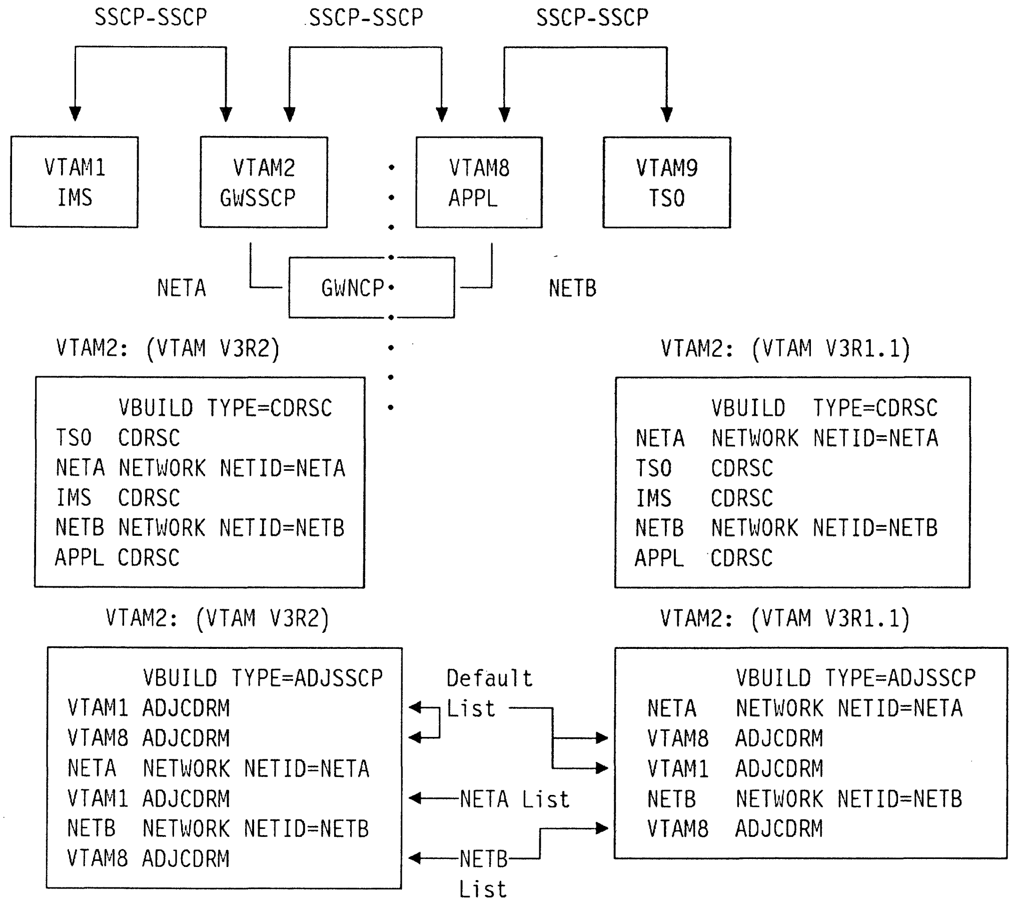
As discussed previously, with VTAM V3R2, CDRSCs following NETWORK statements indicating the same NETID as the host, are now assumed to be real resources in the host network. If the CDRSC statements as shown in Figure 7-3 were used on a VTAM V3R2 system, both TSO and IMS would have to exist in NETA, and VTAM8 would need to be found as the actual owner of TSO, and VTAM1 as the actual owner of IMS. With VTAM V3R2,. the ADJSSCP list used for routing session initiation requests would be the ADJSSCP list for NETA. Adjacent SSCP lists are discussed in more detail in Sections 7.3, 7.4, and 7.5.

To make the definitions shown in Figure 7-3 usable with VTAM V3R2, the TSO CDRSC statement should be moved above the NETWORK NETID=NETA statement. TSO would then be an alias resource and IMS a real resource. Another option is that the NETWORK NETID=NETA statement could be removed, resulting in TSO and IMS both being alias resources.

Whether or not a CDRSC is preceded by a NETWORK statement has implications for adjacent SSCP routing. See Sections 7.3, 7.4, and 7.5 of this bulletin which deal with adjacent SSCP routing.

7.3 Defining an ADJSSCP List for the Host Network

A new function with VTAM V3R2 is the ability to define an ADJSSCP list specifically to use for locating resources in the same network as VTAM. With pre-VTAM V3R2 releases, VTAM can define an ADJSSCP list for use with external networks and can define a default ADJSSCP list to use for all networks; however, there is no way to define a list to use only for resources known to be located in the same network as VTAM.



VTAM2: (VTAM V3R2)

```
     VBUILD TYPE=CDRSC
TSO  CDRSC
NETA NETWORK NETID=NETA
IMS  CDRSC
NETB NETWORK NETID=NETB
APPL CDRSC
```

VTAM2: (VTAM V3R1.1)

```
     VBUILD  TYPE=CDRSC
NETA NETWORK NETID=NETA
TSO  CDRSC
IMS  CDRSC
NETB NETWORK NETID=NETB
APPL CDRSC
```

VTAM2: (VTAM V3R2)

```
      VBUILD TYPE=ADJSSCP
VTAM1 ADJCDRM
VTAM8 ADJCDRM
NETA  NETWORK NETID=NETA
VTAM1 ADJCDRM
NETB  NETWORK NETID=NETB
VTAM8 ADJCDRM
```

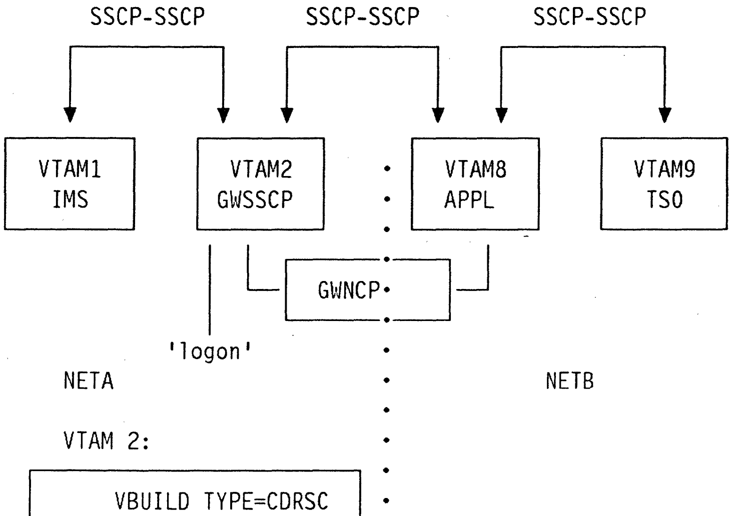
VTAM2: (VTAM V3R1.1)

```
      VBUILD TYPE=ADJSSCP
NETA  NETWORK NETID=NETA
VTAM8 ADJCDRM
VTAM1 ADJCDRM
NETB  NETWORK NETID=NETB
VTAM8 ADJCDRM
```

Figure 7-4. VTAM V3R2 Same Network ADJSSCP List

Figure 7-4 illustrates the differences in the way an ADJSSCP table can be coded between VTAM V3R2 and earlier levels of VTAM. With the VTAM V3R2 ADJSSCP table, shown on the left, there are three types of lists that can be coded--first, a default list for all networks, second, an ADJSSCP list for the same network, and third, an ADJSSCP list for an external network. These three types of lists are explained in more detail in conjunction with Figure 7-5 on page 7-8.

In contrast, with earlier versions of VTAM (illustrated on the right in Figure 7-4), only two types of lists can be defined--first, a default list for all networks and second, an ADJSSCP list for an external network. Figure 7-5 on page 7-8 illustrates a definition of an ADJSSCP list for the host network with VTAM V3R2.



VTAM 2:

```
     VBUILD TYPE=CDRSC
TSO  CDRSC
NETA NETWORK NETID=NETA
IMS  CDRSC
NETB NETWORK NETID=NETB
APPL CDRSC
```

VTAM2:

```
       VBUILD TYPE=ADJSSCP
VTAM1 ADJCDRM                  <--------Default ADJSSCP
VTAM8 ADJCDRM                  <--------List for all networks
NETA  NETWORK NETID=NETA
VTAM1 ADJCDRM                  <--------ADJSSCP List for NETA
NETB  NETWORK NETID=NETB
VTAM8 ADJCDRM                  <--------ADJSSCP List for NETB
```

Figure 7-5. VTAM V3R2 Sample ADJSSCP Table Definition

In Figure 7-5, VTAM2 contains three different types of ADJSSCP lists defined in the VBUILD TYPE=ADJSSCP Major Node. The first list, made up of VTAM1 and VTAM8, immediately follows the VBUILD statement and is **NOT** preceded by a NETWORK statement. This is the default ADJSSCP list for all networks. VTAM2 uses the default list to locate resources for which no network information is available. In the above example, if a VTAM2 user logs on to TSO, VTAM2 uses the default ADJSSCP list for all networks to try to locate TSO, since it has no information about TSO's network at logon. (TSO is defined as an alias resource).

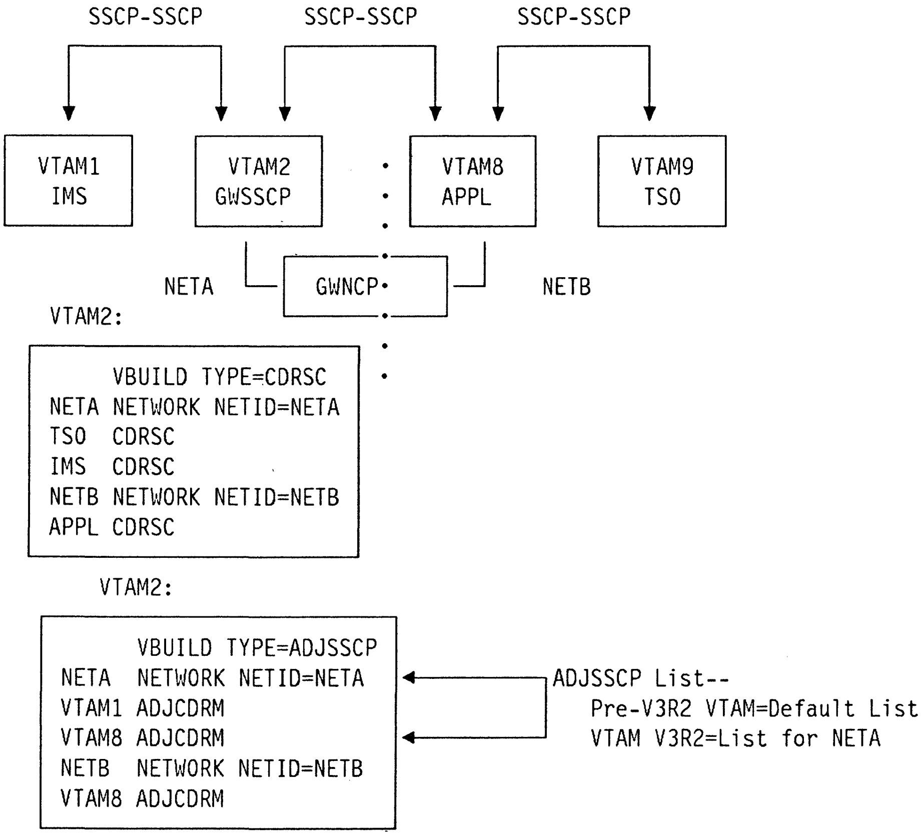
The second list shown in Figure 7-5 is an example of the new ability in VTAM V3R2 to define an ADJSSCP list for the host network. This list of SSCPs follows a NETWORK statement with the same NETID as that of VTAM2. In the example, NETWORK NETID=NETA is followed by a list of one SSCP, VTAM1, which VTAM2 uses when the resource is known to exist in VTAM's own network. VTAM2 uses the ADJSSCP list for NETA when a user logs on to IMS since VTAM2 has a predefined CDRSC for IMS indicating that IMS is a NETA resource. The session initiation request is sent to VTAM1.

The third type of list illustrated in Figure 7-5 is an ADJSSCP list for use with an external network. The list is made up of one SSCP, VTAM8, which follows the NETWORK NETID=NETB statement. If a user at VTAM2 logs on to APPL, VTAM2 uses the NETB ADJSSCP list to route the session initiation request because VTAM2 has a predefined CDRSC for APPL indicating it to be a NETB resource.

7.4 VTAM V3R2 Migration Considerations for ADJSSCP Tables

Installations migrating to VTAM V3R2 from previous releases of VTAM should investigate their ADJSSCP table definitions to insure that the list that was previously used as a default list for all networks is not used by VTAM V3R2 as an ADJSSCP list for the host network.

Figure 7-6 gives an example of how the above problem could arise.



VTAM2:

```
     VBUILD TYPE=CDRSC
NETA NETWORK NETID=NETA
TSO  CDRSC
IMS  CDRSC
NETB NETWORK NETID=NETB
APPL CDRSC
```

VTAM2:

```
      VBUILD TYPE=ADJSSCP
NETA  NETWORK NETID=NETA
VTAM1 ADJCDRM
VTAM8 ADJCDRM
NETB  NETWORK NETID=NETB
VTAM8 ADJCDRM
```

Figure 7-6. Pre-VTAM V3R2 ADJSSCP Table Migration Example

With VTAMs previous to V3R2, a default ADJSSCP list is defined by a list of SSCPs either directly following the VBUILD TYPE=ADJSSCP statement, a null NETWORK statement, or a NETWORK statement indicating the same NETID as VTAM. VTAM uses the default ADJSSCP list to locate resources for which it has no network information. VTAM has no network information if the CDRSC is dynamically created or is predefined as an alias resource. If the default ADJSSCP list shown in Figure 7-6 were used when VTAM is migrated to a V3R2 level, problems could arise because now VTAM only uses this list for resources located in NETA and VTAM has no default ADJSSCP list for all networks.

In Figure 7-6 on page 7-9, two problems exist which must be fixed in order for the CDRSC and ADJSSCP definitions to work with VTAM V3R2. The first problem is that TSO will be assumed by VTAM2 to be a **real** resource in NETA with VTAM V3R2. (See Section 7.1). The second problem is that there is no default ADJSSCP list for all networks. The list of CDRSCs and the ADJSSCP table would work the same way with VTAM V3R2 as it did with previous levels if the NETWORK NETID=NETA statement were removed from both the CDRSC Major Node Definition and the ADJSSCP Major Node Definition. A better resolution would be to define IMS as a real resource in NETA and TSO as an alias resource, and to define a default ADJSSCP list consisting of VTAM1 and VTAM8 and an ADJSSCP list for NETA consisting of VTAM1.

7.5 VTAM V3R2 ADJSSCP Routing Logic

A significant enhancement in VTAM V3R2 ADJSSCP Routing Logic is that VTAM now sends session initiation requests directly to the resource's owner, if possible, if it has an SSCP-to-SSCP session with the owner, and even if there are existing sessions with the resource using a different route. VTAM V3R2 uses the logic detailed in Figure 7-7 on page 7-11 and Figure 7-8 on page 7-11 for **each and every** session initiation request. With previous levels of VTAM, a list of adjacent SSCPs is pointed to from the CDRSC and is used for each session initiation as long as the CDRSC exists or as long as there are existing sessions with a particular resource. If started with SSCPORD=PRIORITY, pre-V3R2 VTAMs use logic to reorder ADJSSCPs based on knowledge of successful routes; however, as long as there are existing sessions with a resource, the last successful route is used for session initiation requests, even if a direct route to the resource's owner becomes available. See Technical Bulletin GG66-0258 "SNA Network Interconnection (SNI) Session Initiation Request Routing" for more information concerning routing with VTAMs previous to V3R2.

With VTAM V3R2 the routing order for adjacent SSCPs that VTAM uses for each session initiation request is shown in Figure 7-7 on page 7-11 and Figure 7-8 on page 7-11. Figure 7-7 on page 7-11 shows the routing order for adjacent SSCPs if VTAM has been started with the SSCPORD=DEFINED start option. DEFINED tells VTAM to always use a list of ADJSSCPs to be tried based on the way the user has defined the ADJSSCP table in the VBUILD TYPE=ADJSSCP definition. Figure 7-8 on page 7-11 shows the order in which VTAM constructs a list of adjacent SSCPs if VTAM has been started with the SSCPORD=PRIORITY start option. PRIORITY tells VTAM to always use a list of ADJSSCPs to be tried based on knowledge of which SSCPs have been successful with past session initiation requests.

NOTE: Both of these figures assume that VTAM has also been started with SSCPDYN=YES. SSCPDYN=YES tells VTAM to add ADJSSCPs to the list, even if they have not been defined, if VTAM becomes aware of them. To use these two lists if VTAM has been started with SSCPDYN=NO (dynamic additions of ADJSSCPs will not be performed), ignore the sections labeled "learned."

1. The VTAM that owns the resource (the destination logical unit).**See Note 1**
2. The CDRM on the CDRSC definition, if it is coded and no NETID specified.
3. **One** of these ADJSSCP lists, defined in VBUILD TYPE=ADJSSCP: **See Note 2**
   a. The resource owner's NETWORK specific list, if coded, if owner and NETID are known.
   b. The NETWORK specific list, if coded, if NETID is known.
   c. The resource owner's default ADJSSCP list, if coded, if owner is known.
   d. The default ADJSSCP list.
4. Learned ADJSSCPs (not coded in the ADJSSCP table) in this order:
   ADJSSCPs that have been successful in the past.
   ADJSSCPs that have been unsuccessful in the past.

Figure 7-7. List of SSCPs selected if SSCPORD = DEFINED

1. The VTAM that owns the resource (the destination logical unit). **See Note 1**
2. The CDRM on the CDRSC definition, if it is coded and no NETID specified.
3. ADJSSCPs either learned or defined that have been successful in the past.
4. **One** of these ADJSSCP lists, defined in VBUILD TYPE=ADJSSCP: **See Note 2**
   a. The resource's owner NETWORK specific list, if coded, if owner and NETID are known.
   b. The NETWORK specific list, if coded, if NETID is known.
   c. The resource's owner default ADJSSCP list, if coded, if owner is known.
   d. The default ADJSSCP list.
5. Learned ADJSSCPs (not coded in the ADJSSCP table) in this order:
   ADJSSCPs that have been successful in the past (learned since #3).
   ADJSSCPs that have been unsuccessful in the past.

Figure 7-8. List of SSCPs selected if SSCPORD = PRIORITY

Note 1: The VTAM that owns the resource is the actual VTAM where the resource resides. If there are existing sessions with the resource, then VTAM knows the actual owner. In the case that there are not existing sessions with the resource, if the resource is defined with NETID and CDRM (a real CDRSC with CDRM coded), VTAM uses the CDRM as the actual owner; the CDRM coded will be the first ADJSSCP that VTAM attempts to use.

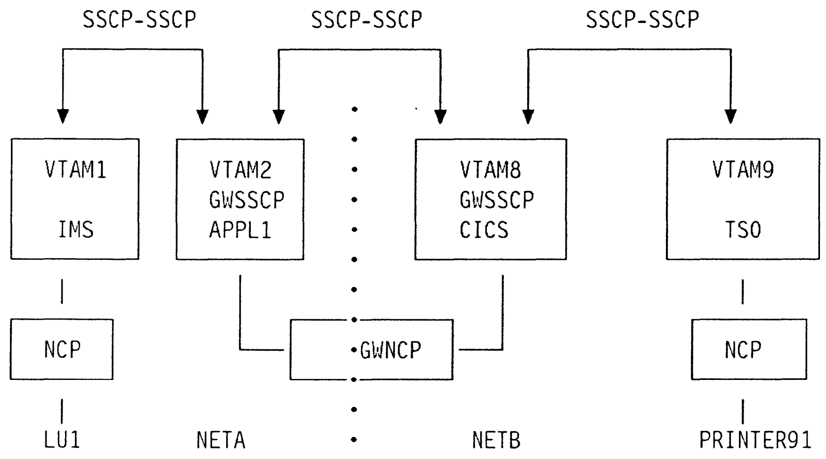
Note 2: When VTAM is selecting an ADJSSCP list, it will select one of the lists (a. through d.) as shown in the above Figures. If one or more of the lists does not exist, the next list shown would be used instead. For example, if a resource's owner were known but network was not, VTAM would try to select resource owner's default ADJSSCP list (shown as item c.). If there were no resource owner's default ADJSSCP list, then item d., the default ADJSSCP list, would be used.

Following are examples of the circumstances under which the various ADJSSCPs are used. The examples are given in the order detailed in Figure 7-7.

1. Using the VTAM that owns the resource:

Using Figure 7-9 as an example, assume that LU1, owned by VTAM1 is already in session with APPL1. In this case VTAM1 knows the actual owner of APPL1. If a second LU owned by VTAM1 logs on requesting a session with APPL1, VTAM1 sends the session initiation request to VTAM2, the owner of APPL1, since VTAM1 has an SSCP-to-SSCP session with VTAM2.

However, assuming a session exists between LU1 and CICS, even though VTAM1 knows the owner of CICS is VTAM8, a second session initiation request for CICS could not be sent because no SSCP-to-SSCP session exists between VTAM1 and VTAM8. VTAM1 would proceed in logic to try to route to other SSCPs as described in Figure 7-7 on page 7-11 and Figure 7-8 on page 7-11 to locate CICS.



```
ADJSSCP Table for VTAM1:

         VBUILD TYPE=ADJSSCP
VTAM2    ADJCDRM
VTAM8    ADJCDRM
VTAM9    CDRM
VTAM2    ADJCDRM
VTAM8    ADJCDRM
NETA     NETWORK NETID=NETA
VTAM2    ADJCDRM
NETB     NETWORK NETID=NETB
VTAM2    ADJCDRM
VTAM8    ADJCDRM
```

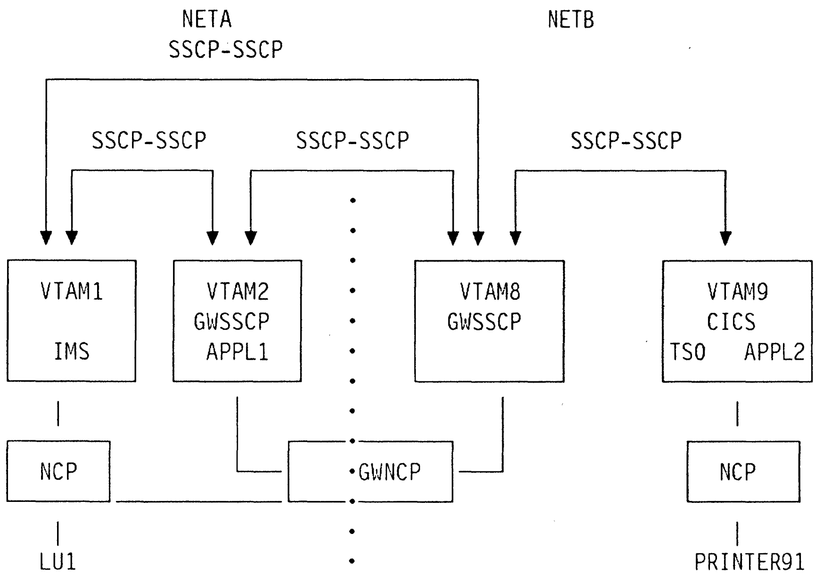
```
CDRSCs for VTAM1:

      VBUILD TYPE=CDRSC
TSO   CDRSC CDRM=VTAM9
NETA  NETWORK NETID=NETA
APPL1 CDRSC CDRM=VTAM2
NETB  NETWORK NETID=NETB
CICS  CDRSC CDRM=VTAM8
```

Figure 7-9. VTAM V3R2 ADJSSCP Routing Logic Example 1

2. Using the CDRM coded on the CDRSC Definition if no NETID specified:

Again using Figure 7-9 as an example, assuming no sessions with TSO currently exist, if LU1 logs onto TSO, VTAM1 does not know the owner of the resource. Even though there is a CDRM, VTAM9, coded on TSO's CDRSC definition, TSO is defined as an alias resource since the definition is not preceded by a NETID, and VTAM9, therefore, is not necessarily the actual owner of TSO. If VTAM1 had an SSCP-to-SSCP session with VTAM9 (it does not in Figure 7-9), VTAM1 would route the session initiation request to VTAM9. Since, in this example, VTAM1 has no SSCP-to-SSCP session with VTAM9, VTAM1 would proceed in logic to selecting an ADJSSCP list.



ADJSSCP Table for VTAM1:

```
        VBUILD TYPE=ADJSSCP
VTAM2   ADJCDRM
VTAM8   ADJCDRM
VTAM9   CDRM
VTAM8   ADJCDRM
NETA    NETWORK NETID=NETA
VTAM2   ADJCDRM
NETB    NETWORK NETID=NETB
VTAM2   ADJCDRM
VTAM9   CDRM
VTAM8   ADJCDRM
```

Resource Owner's Network Specific ADJSSCP List

CDRSCs for VTAM1:

```
      VBUILD TYPE=CDRSC
TSO   CDRSC CDRM=VTAM9
NETA  NETWORK NETID=NETA
APPL1 CDRSC
NETB  NETWORK NETID=NETB
CICS  CDRSC CDRM=VTAM9
```

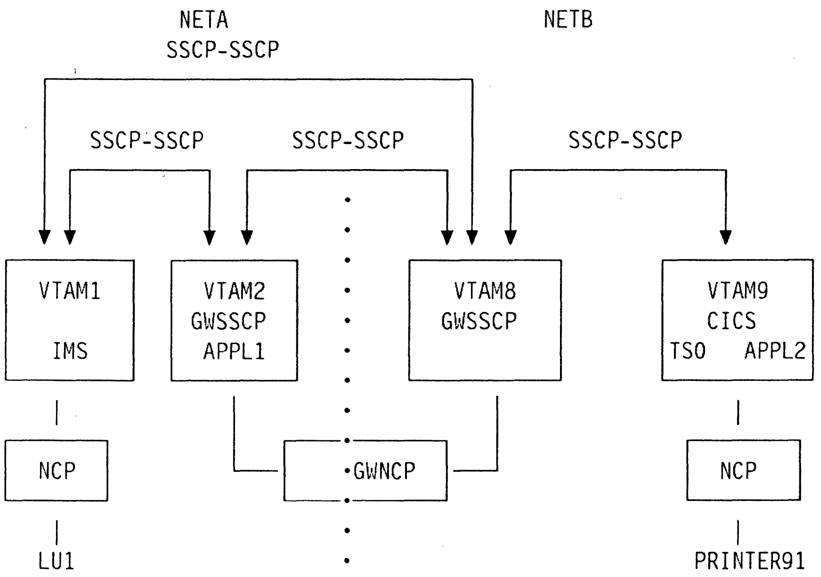
Figure 7-10. VTAM V3R2 ADJSSCP Routing Example 2

3a. Using a Resource Owner's Network Specific ADJSSCP list:

Figure 7-10 shows a configuration where there are two different gateway VTAMs controlling one gateway NCP. The routing has been designed so that VTAM1 routes almost all of its session initiation requests, even for NETB resources to VTAM2. However, faster session setup time is desired for CICS so a specific entry for VTAM9 has been coded in the NETB ADJSSCP list.

Assume that LU1 logs on to CICS. At this point, the actual owner of CICS is known, VTAM9, the CDRM coded on the CICS CDRSC definition (CICS is defined as a **real** resource). However, VTAM1 has no SSCP-to-SSCP session with VTAM9, so it must procede in logic to selecting an ADJSSCP list. Since CICS's

NETID as well as its owner are known to VTAM1, the resource owner's network specific list is used. This list follows the CDRM VTAM9 statement coded following the NETWORK NETID=NETB statement and consists of one ADJSSCP, VTAM8. The session initiation request for CICS is forwarded to VTAM8 which reroutes it to VTAM9.



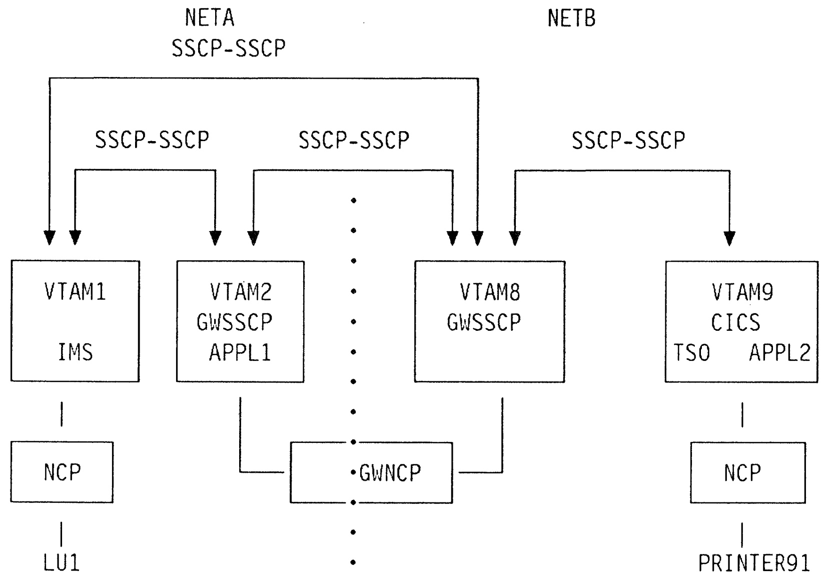
```
          ADJSSCP Table for VTAM1:          CDRSCs for VTAM1:

                   VBUILD TYPE=ADJSSCP              VBUILD TYPE=CDRSC
          VTAM2    ADJCDRM                   TSO    CDRSC CDRM=VTAM9
          VTAM8    ADJCDRM                   NETA   NETWORK NETID=NETA
          VTAM9    CDRM                      APPL1  CDRSC
          VTAM8    ADJCDRM                   NETB   NETWORK NETID=NETB
ADJSSCP   NETA     NETWORK NETID=NETA        CICS   CDRSC CDRM=VTAM9
List ───► VTAM2    ADJCDRM
for       NETB     NETWORK NETID=NETB
NETA      VTAM2    ADJCDRM
          VTAM9    CDRM
          VTAM8    ADJCDRM
```

Figure 7-11. VTAM V3R2 ADJSSCP Routing Example 3

3b. Using a Network Specific ADJSSCP list:

Using Figure 7-11 assume LU1 has logged off CICS and now logs on to APPL1. At this point, APPL1's owner is not known, there is no CDRM coded on the CDRSC definition for APPL1, and since the CDRSC for APPL1 is defined following the NETWORK NETID=NETA statement, the ADJSSCP list for NETA is used. This is the list of ADJSSCPs that follows the NETWORK NETID=NETA statement, and only consists of VTAM2. The session initiation request for APPL1 is sent to VTAM2.



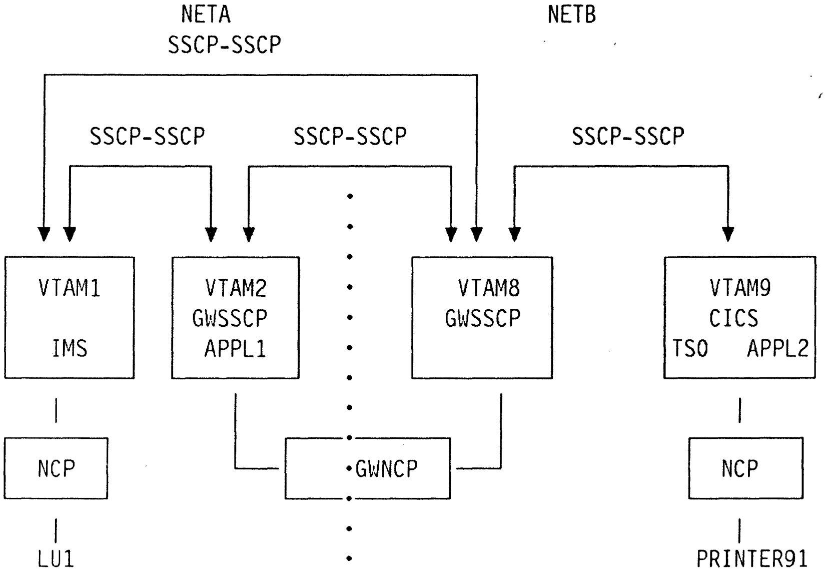
```
                ADJSSCP Table for VTAM1:
                        VBUILD TYPE=ADJSSCP
Resource        VTAM2   ADJCDRM
Owner's         VTAM8   ADJCDRM
Default         VTAM9   CDRM
ADJSSCP───►     VTAM8   ADJSSCP
List            NETA    NETWORK NETID=NETA
                VTAM2   ADJCDRM
                NETB    NETWORK NETID=NETB
                VTAM2   ADJCDRM
                VTAM9   CDRM
                VTAM8   ADJCDRM
```

```
CDRSCs for VTAM1:
      VBUILD TYPE=CDRSC
TSO   CDRSC CDRM=VTAM9
NETA  NETWORK NETID=NETA
APPL1 CDRSC
NETB  NETWORK NETID=NETB
CICS  CDRSC CDRM=VTAM9
```

Figure 7-12. VTAM V3R2 ADJSSCP Routing Example 4

3c. Using the resource owner's default ADJSSCP list:

Using Figure 7-12 assume that LU1 logs on to TSO. In this case, the resource owner's default ADJSSCP list is used. Following the logic in Figure 7-7 on page 7-11, TSO's actual owner is not known (TSO is defined as an **alias** resource). A CDRM, VTAM9, is coded on TSO's CDRSC definition; however, since VTAM1 has no session with VTAM9, the session initiation request cannot be sent to VTAM9. TSO's NETID is not known because TSO is defined as an alias resource; therefore, a network specific ADJSSCP list cannot be used. In the default ADJSSCP list there is a CDRM entry for VTAM9, the same CDRM that is coded on the CDRSC definition for TSO. The list of ADJSSCPs following the VTAM9 CDRM entry in the default ADJSSCP list is the resource owner's default ADJSSCP list. This list consists of one ADJSSCP, VTAM8, to which VTAM1 sends the session initiation request for TSO. VTAM8, then, uses similar logic to reroute the session initiation request to VTAM9.



ADJSSCP Table for VTAM1:

```
Default          VBUILD TYPE=ADJSSCP
ADJSSCP ──► VTAM2  ADJCDRM
List     ──► VTAM8  ADJCDRM
            VTAM9  CDRM
            VTAM8  ADJCDRM
            NETA   NETWORK NETID=NETA
            VTAM2  ADJCDRM
            NETB   NETWORK NETID=NETB
            VTAM2  ADJCDRM
            VTAM9  CDRM
            VTAM8  ADJCDRM
```

CDRSCs for VTAM1:

```
      VBUILD TYPE=CDRSC
TSO   CDRSC CDRM=VTAM9
NETA  NETWORK NETID=NETA
APPL1 CDRSC
NETB  NETWORK NETID=NETB
CICS  CDRSC CDRM=VTAM9
```

Figure 7-13. VTAM V3R2 ADJSSCP Routing Example 5

3d. Using the default ADJSSCP list:

Using the example shown in Figure 7-13, .if the IMS application running at VTAM1 acquires PRINTER91, the default ADJSSCP list is used. It is used because VTAM1 dynamically creates a CDRSC for PRINTER91 and has no owner, CDRM, or network information about PRINTER91 at the time it builds the session initiation request. The default list is the list of ADJSSCPs immediately following the VBUILD TYPE=ADJSSCP statement and consists of VTAM2 and VTAM8. The session initiation request is sent to VTAM2. VTAM2, being a gateway SSCP, is capable of rerouting session initiation requests to other SSCPs. Assuming the session initiation request is sent to VTAM8, VTAM8, being a gateway SSCP, can again reroute the request to VTAM9.

Even if this CDRSC is coded for PRINTER91 at VTAM1:

```
           VBUILD TYPE=CDRSC
PRINTER91  CDRSC
```

the default ADJSSCP list is used because nothing is known about PRINTER91's owner, CDRM is not coded, network is not known, and there is no CDRM coded to allow possible use of a resource owner's default ADJSSCP list.

"Learned" ADJSSCPs:

Using Figure 7-13 on page 7-16 as an example, assume that APPL2 initiates a session with IMS. The session initiation request is sent from VTAM8 to VTAM1. Note that VTAM1 has not predefined a CDRSC for APPL2. VTAM8 is a "learned" ADJSSCP and VTAM1 will use the ADJSSCP of VTAM8 for subsequent session requests for APPL2 if SSCPDYN=YES has been coded as a VTAM1 start option.

If LU1 logs on to APPL2, assuming VTAM1 is started with SSCPDYN=YES and SSCPORD=PRIORITY, the logic VTAM1 uses for routing is:

1. The owner (VTAM9) of APPL2 is known, but VTAM1 has no SSCP-SSCP session with VTAM9.

2. There is no CDRSC for APPL2 and, therefore, no CDRM coded.

3. VTAM8 is a learned, successful ADJSSCP to VTAM1 for use with APPL2 since VTAM8 is the SSCP that the session initiation request was received from. The request will be sent to VTAM8 which should forward it to VTAM9.

4. If, for some reason VTAM8 should respond negatively (its ADJSSCP table did not route to VTAM9, for example), VTAM1 would next use the resource owner's network specific ADJSSCP list since NETID is known. (In this case, VTAM8 would be the resulting SSCP and would not be tried again). ADJSSCP routing would stop at this point unless some other ADJSSCP had been learned during the routing.

7.6 Start Options & PCCU Statement

The VTAM start options, **NETID** and **SSCPNAME**, must be coded with VTAM Version 3 Release 2 (V3R2). Prior to VTAM V3R2 these two start options are not required; rather, they are are used to indicate that VTAM is capable of performing SNA Network Interconnection (SNI) functions. With VTAM V3R2, a new start option, **GWSSCP**, indicates whether VTAM is capable of performing SNI functions. NETID and SSCPNAME do **not** indicate gateway-capability with VTAM V3R2.

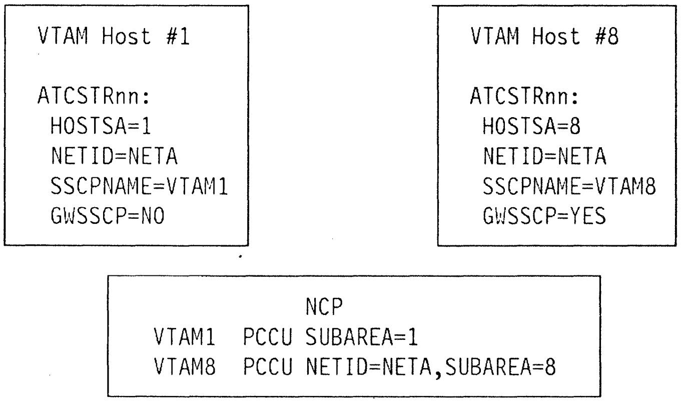
GWSSCP indicates whether this VTAM is gateway-capable. VTAMs which are gateway-capable can reroute session initiation requests, can provide SNI session setup if they are in session with a gateway-NCP, and can use the Netview Alias Name Translation Facility. The default for GWSSCP is GWSSCP=YES. Any VTAM V3R2 which is a gateway-SSCP or which provides functions that only gateway-capable VTAMs do, must be started with GWSSCP=YES. GWSSCP is not applicable to VSE VTAM systems. Although VSE VTAM systems can participate in networks connected via SNI, they cannot perform SNI functions.

SSCPNAME is the name of the SSCP. It should match the name of the CDRM statement pertaining to the VTAM being started, and match the NAME operand of the corresponding GWNAU definition statement, if coded, in the gateway NCP.

NETID is the name of the network in which the VTAM being started resides. A VTAM V3R2 started with NETID and GWSSCP=NO is capable of utilizing NETWORK statements in VTAM definitions, such as CDRSCs and ADJSSCP lists, but it is not gateway-capable.

Changes in the use of NETID and GWSSCP to indicate gateway-capability cause the need for compatiblity PTFs for VTAMs prior to V3R2. The compatibility PTFs are discussed in Section 7.7. Because all VTAM V3R2s are started with NETID, there are some considerations for migrating ADJSSCP lists and CDRSC definitions, discussed in Sections 7.2 and 7.4.

V3R2 VTAMs started with GWSSCP=NO do not recognize an NCP PCCU statement coded with NETID and will fail NCP activation. NCPs must contain a PCCU statement with matching SUBAREA number and **NO** NETID when defining V3R2 VTAMs started with GWSSCP=NO. See Figure 7-14.



```
VTAM Host #1

ATCSTRnn:
 HOSTSA=1
 NETID=NETA
 SSCPNAME=VTAM1
 GWSSCP=NO
```

```
VTAM Host #8

ATCSTRnn:
 HOSTSA=8
 NETID=NETA
 SSCPNAME=VTAM8
 GWSSCP=YES
```

```
              NCP
VTAM1  PCCU SUBAREA=1
VTAM8  PCCU NETID=NETA,SUBAREA=8
```

Figure 7-14. VTAM V3R2 PCCU Definition Statements

In Figure 7-14, VTAM Host #1 can activate the NCP because the NCP contains an applicable PCCU statement:

VTAM1 PCCU SUBAREA=1

If the PCCU statement were changed to:

VTAM1 PCCU NETID=NETA,SUBAREA=1

NCP activation would fail.

For VTAM Host #8, the PCCU statement can be coded with NETID as shown in Figure 7-14, or without NETID.

7.7 GW-Capability Compatibility PTF

Because VTAM V3R2 is using a new option, GWSSCP=YES or NO, to determine gateway-capability and because V3R2 VTAM is always started with NETID, earlier versions of VTAMs that have SSCP-to-SSCP sessions with VTAM V3R2s may require a compatibility PTF. Without the PTF, pre-V3R2 VTAMs use NETID to indicate and detect gateway-capability. Without the maintenance applied, VTAM V3R2s may not properly detect gateway-capability of same-network pre-V3R2 VTAMs; additionally, without the maintenance all pre-V3R2 VTAMs detect **all** V3R2 VTAMs as gateway-capable.

Following are examples of configurations in which the SNI gateway compatibility PTF is required:

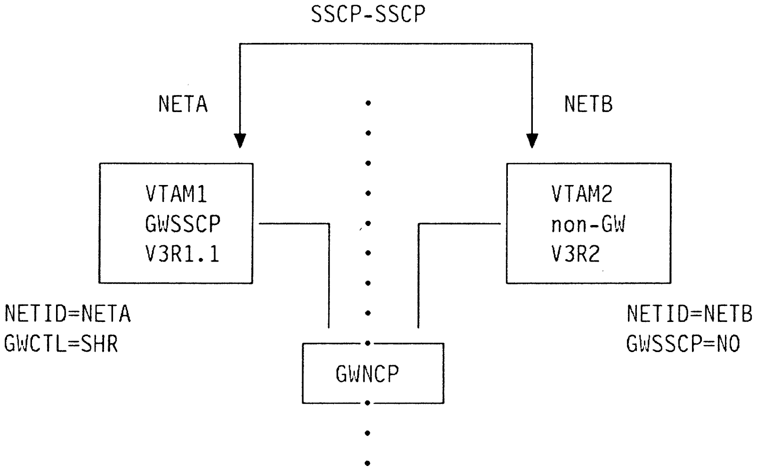


Figure 7-15. GW-VTAM V3R1.1 in Cross-Net Session with Non-GW VTAM V3R2

Figure 7-15 shows an example of a VTAM V3R1.1 gateway SSCP with a cross-network session with a VTAM V3R2 non-gateway system. Without the maintenance applied, the VTAM V3R1.1 system assumes the V3R2 system to be a gateway because NETID is coded in the start options of VTAM2. With GWCTL=SHR coded on VTAM1's PCCU statement, VTAM1 will only perform part of the SNI session setup for session initiation requests originating in NETA, expecting VTAM2 to share responsibility for session setup. The session setup will fail because VTAM2 is not a gateway SSCP. In this configuration, for sessions originating in NETB, VTAM1 will perform the gateway functions with or without the maintenance applied.

The PTF is also required to prevent VTAM1 from sending session initiation requests for resources not located at VTAM2 to VTAM2, expecting VTAM2 to reroute them. These session initiation requests will be rejected by VTAM2 because VTAM2 is not gateway-capable and cannot reroute the requests.

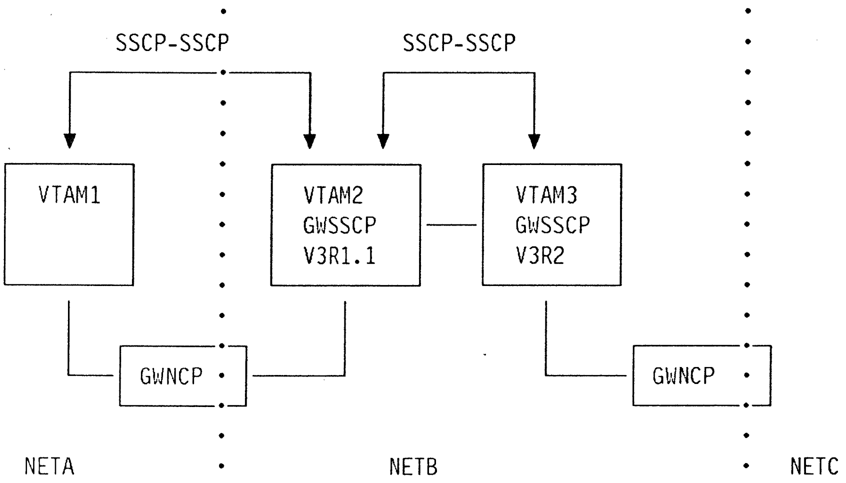


Figure 7-16. Same network GWSSCP VTAM V3R1.1 and VTAM V3R2

Figure 7-16 illustrates another problem that could occur without the PTF installed. Figure 7-16 shows an example of a VTAM V3R1.1 system and V3R2 system in the same network where both VTAMs are gateway-SSCPs. Without the PTF on VTAM2, VTAM3 will think that VTAM2 is **NOT** a gateway-SSCP. Session initiation requests originating in NETC or at VTAM3, known to involve resources in NETA will fail. This is because VTAM3 will not send session initiation requests involving resources in NETA to VTAM2 because VTAM3 believes VTAM2 to be incapable of rerouting them.

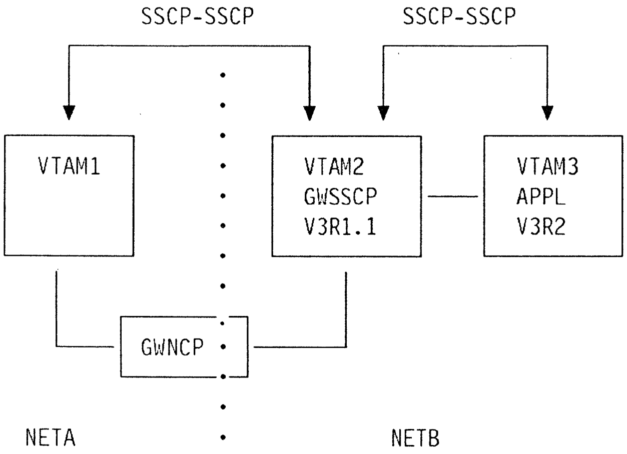


Figure 7-17. Same network GWSSCP VTAM V3R1.1 and VTAM V3R2 Application Host

Figure 7-17 illustrates another problem that could occur without the needed maintenance installed on VTAM2. If the application program, APPL, running at VTAM3, tries to initiate sessions with resources known to exist in NETA, the session requests will fail. Because VTAM3 does **NOT** believe VTAM2 to be gateway-capable, VTAM3 will not send VTAM2 session requests involving resources known to be in another network.

Applications might initiate session requests for a variety of reasons. First, the request might be for an application-to-application session. Second, the application might have some devices it automatically acquires. Third, with applications such as TSO, the logon from terminal results in a CLSDST PASS and TSO then issuing a session initiation request with the destination logical unit being the terminal. With CLSDST PASS, the origin logical unit (OLU) - destination logical unit (DLU) roles become reversed. The OLU, terminal logging on, becomes the DLU after the CLSDST PASS is issued. Therefore, if a user in NETA logged on to APPL (TSO) in Figure 7-17 on page 7-20, the session would fail. After the CLSDST PASS occurred, the TSO application at VTAM3 would issue a request for a session with the NETA user. VTAM3 would not send the session initiation request to VTAM2, believing VTAM2 incapable of rerouting, unless the maintenance were installed at VTAM2.

7.8 VTAM V3R2 GWCTL=SHR Enhancement

One of the configuration options with SNI is to have two gateway VTAMs which share responsibility of cross-network session setup in conjunction with one gateway NCP. This type of configuration is illustrated in Figure 7-18. When the PCCU macros within the gateway NCP for two gateway VTAMs in two different networks contain the GWCTL=SHR statement, the gateway VTAMs share responsibility for cross-network session setup. These responsibilities include establishing alias and real addresses, alias and real names, and virtual routes for cross-network sessions.

VTAM Version 3 Release 2 is enhanced in the shared control environment in order to avoid a problem that occurred in some cases with previous releases of VTAM. The problem that could occur is that if the gateway VTAM on the destination logical unit (DLU) side had not yet activated the gateway NCP, session setup would fail. Even though the gateway VTAM on the origin logical unit (OLU) side was completely capable of performing the session setup, since GWCTL=SHR was coded, session setup would fail if the DLU side gateway VTAM had not activated the gateway NCP.

Using Figure 7-18 as an example, with VTAM V3R1.1 and earlier releases, if LU1 logged on to APPL, and VTAM8 had not yet activated the GWNCP, the session setup would fail. With VTAM V3R2 and the same example of LU1 logging on to APPL, upon receiving an error because VTAM8 had not yet activated the GWNCP, VTAM2 performs all functions needed for the cross-network session setup.

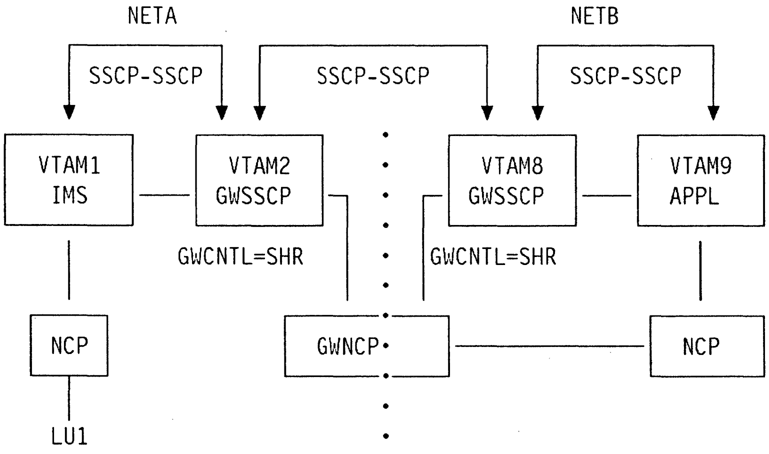
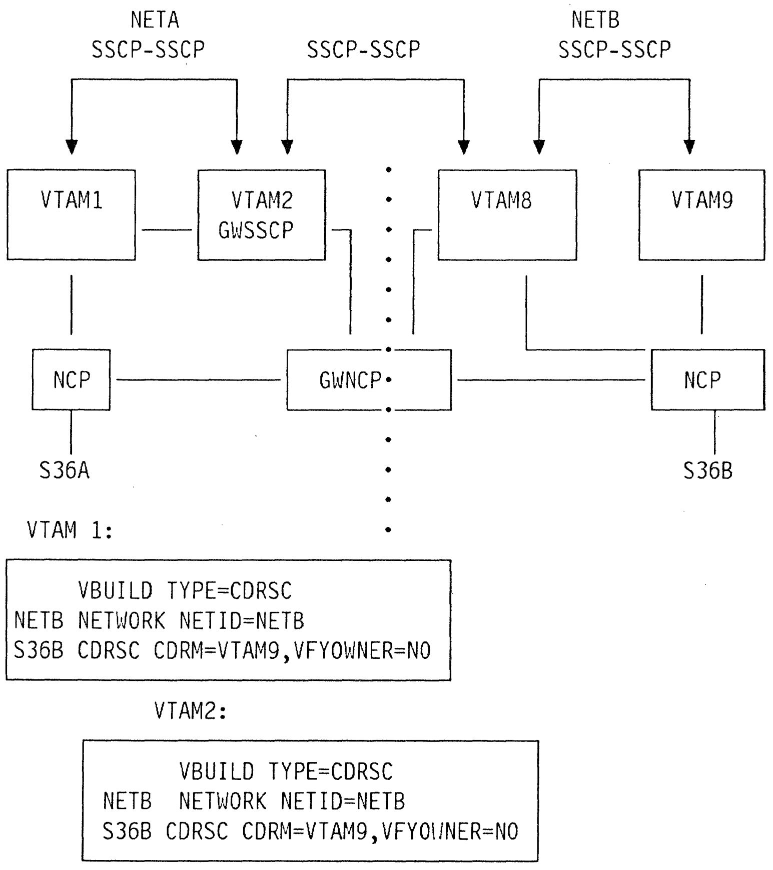


Figure 7-18. VTAM V3R2 GWCTL=SHR Example

7.9 MODIFY CDRM,IMMEDIATE and VERIFY OWNER

Because of some of the characteristics of T2.1 nodes and independent LUs, VTAM V3R2 contains enhanced function for the MODIFY CDRM command, and a new function called "owner verification." In the past, the only type of logical unit that was capable of more than one session simultaneously was a VTAM application program. With VTAM application programs, when the owning VTAM goes down, all sessions end, and the application program is restarted with a new owning VTAM.

With VTAM V3R2 and support for independent logical units in Type 2.1 nodes, logical units in peripheral nodes can have multiple sessions and when the owning VTAM goes down, a backup VTAM can gain ownership of the peripheral node and logical unit **without** disrupting existing sessions. In order to facilitate changes in VTAM ownership of independent logical units, two new functions were added to VTAM V3R2, the **IMMEDIATE** operand on the MODIFY CDRM command and the **VFYOWNER = YES|NO** operand on the CDRSC definition statement.



VTAM 1:

```
    VBUILD TYPE=CDRSC
NETB NETWORK NETID=NETB
S36B CDRSC CDRM=VTAM9,VFYOWNER=NO
```

VTAM2:

```
      VBUILD TYPE=CDRSC
NETB  NETWORK NETID=NETB
S36B CDRSC CDRM=VTAM9,VFYOWNER=NO
```

Figure 7-19. VTAM V3R2 MODIFY CDRM,I and VFYOWNER

Using Figure 7-19 as an example, assume that a session has been established between S/36A and S/36B. S/36B's owning VTAM, VTAM9, is lost and ownership of S/36B is taken over by VTAM8. Through the loss of the owning VTAM and takeover by the backup VTAM, the session between S/36A and S/36B continues. If S/36A wants to initiate a subsequent session with S/36B, the operator at VTAM1 must issue a MODIFY CDRM for S/36B in order to change the owner to VTAM8. If the MODIFY CDRM command were not performed, the session setup would fail at VTAM8. The failure would occur because the CDINIT to initiate the session carries both the name of the resource (S/36B) and the CDRM owner (VTAM9 as coded on the CDRM= operand of the S/36B CDRSC definition statement in VTAM1). The CDRM is carried in the CDINIT because S/36B is defined as a real resource. Although S/36B is now owned by VTAM8, since the CDRM name carried in the CDINIT is VTAM9, the session setup request would fail at VTAM8.

Since there is an existing session between the S/36s, the new IMMEDIATE operand is necessary on the MODIFY CDRM command. With VTAM V3R1.1 and earlier, a MODIFY CDRM does not take effect until all sessions are terminated. The IMMEDIATE operand allows a VTAM operator to change the CDRM owner of a CDRSC immediately without disrupting existing sessions. In the example, if S/36A needed to start a second session with S/36B after S/36B had been taken over by VTAM8, the operator at VTAM1 must issue the command:

"MODIFY NET,CDRM=VTAM8,ID=S/36B,IMMEDIATE."

NOTE: There are PTFs available for pre-V3R2 VTAMs to allow them to have the MODIFY CDRM,IMMEDIATE function. See the Software Compatibility section of this manual for more information on the maintenance.

The purpose of the new function, **owner verification**,is so the operator at the intermediate VTAM (the gateway, VTAM2) does **NOT** have to issue the MODIFY command. Since the CDRSC for S/36B is coded at VTAM2 with VFYOWNER=NO, if a second session with S/36B was initiated by S/36A, the operator at VTAM2 would not have to issue the MODIFY command. Assume that S/36B has been taken over by VTAM8, and that the VTAM1 operator has issued the MODIFY CDRM command. The CDINIT for S/36B would carry VTAM8 as the owner, but the S/36B CDRSC definition at VTAM2 is coded with VTAM9 as the CDRM. Since VFYOWNER=NO is specified on the S/36B CDRSC, VTAM2 will replace the owner as VTAM8 during the session setup.

Additionally, assuming there was an existing session between S/36A and S/36B, that S/36B had been taken over by VTAM8, and that the MODIFY CDRM **had not** been performed at VTAM1, if S/36B initiated a session with S/36A, it would succeed since VFYOWNER=NO is coded on the CDRSC definition at VTAM1.

VFYOWNER=YES|NO can **only** be coded on a CDRSC definition of a **REAL** resource (one following a network statement) that also has the CDRM= operand coded. All other types of CDRSCs--

- Dynamic CDRSCs,
- CDRSCs for alias resources,
- CDRSCs with no CDRM= operand coded

default to VFYOWNER=NO.

CDRSCs with VFYOWNER=YES coded will cause session setup to fail if the CDRM owner is changed unless the operator issues a MODIFY CDRM command. Using Figure 7-19 on page 7-23 as an example, if VFYOWNER=YES was coded at VTAM1 and VTAM2, and if S/36B was taken over by VTAM8, subsequent session setups would fail. If S/36B tried to initiate a session with VTAM1, VTAM1 would fail the setup because the owner is different than that coded on the CDRSC. If VTAM1 issued the MODIFY CDRM command and changed the CDRM to VTAM8, a session setup originating with S/36A would fail at VTAM2 because VFYOWNER=YES is coded for S/36B.

7.10 Parameter Changes in NCP V4R3 and NCP V5R2

NETID is a required parameter in the BUILD definition statement in NCP V4R3, NCP V5R1, and NCP V5R2. The coding of NETID no longer causes SNI code to be included during the NCP generation-- it is the coding of HSBPOOL that causes SNI code to be included.

With these NCPs, channel adapters can be coded using a new method. The new method is to code a GROUP, LINE, and PU statement with LNCTL=CA to represent the channel adapter, rather than CA= on the BUILD statement. For SNI, it is necessary to indicate the NETID of the node connected to the channel adapter. With previous NCPs, this was done with a CANETID= statement. With the new type of channel definitions, the NETID of the node connected to the channel adapter can be coded on the GROUP, LINE or PU statement representing the channel adapter.

See Section 7.6 for considerations for coding NETID in the PCCU definition for use with VTAM V3R2.

7.11 Alias Name Translation Facility Enhancement

With levels of VTAM prior to V3R2, the Netview Alias Name Translation Facility, if active, is always called to see if translation is needed for the OLU name, the DLU name, the COS entry name, and the Logmode Entry name. An enhancement in VTAM V3R2 is that the VTAM Constants Module (ISTRACON) contains bits that may be set to tell VTAM to call the Netview Alias Name Translation Facility for a specific purpose. Bits may be to set to specify only calling the Alias Facility for the following purposes:

- LU alias names
- LU real names
- CDRM names
- COS entry names
- Logmode entry names

7.12 Dynamic Adjacent SSCPs

A new function called "Dynamic Adjacent SSCPs" is being provided for VTAM V3R2 users via PTF. This new function is packaged along with the USERVAR management PTF described in Ivory Letter 288-505. The APARs associated with this maintenance are:

• OY16021	VTAM V3R2 for MVS/XA	available March 1989
• OY16022	VTAM V3R2 for MVS/370	available March 1989
• VM33000	VTAM V3R2 for VM & VM/9370	available June 1989
• DY37697	VTAM V3R2 for VSE	available September 1989

After installation of the maintenance, users of VTAM are not required to define an Adjacent SSCP table. A new start option, DYNASSCP, directs VTAM whether or not to build lists of adjacent SSCPs dynamically. If DYNASSCP=YES is coded, and an appropriate adjacent SSCP list is not found for a session initiation request, VTAM will route the session initiation request to other CDRMs with which VTAM has an SSCP-to-SSCP session. VTAM constructs this "dynamic" list of SSCPs in the order in which the SSCP-to-SSCP sessions become active.

DYNASSCP = YES is the default.

If an appropriate adjacent SSCP list is found, VTAM uses the defined list rather than building a "dynamic" one. An appropriate adjacent SSCP list is the default list for all networks, or a specific list of adjacent SSCPs for an external network if the targeted resource is known to exist in that network. For example, a session initiation request involving a resource for which no network information is known is routed using a dynamically built adjacent SSCP list if no default adjacent SSCP list has been defined.

Additionally, the Session Management Exit Adjacent SSCP Selection Function can be used to reorder or shorten a dynamically built adjacent SSCP list.

Users with simple networks may wish to take advantage of the Dynamic Adjacent SSCP function because it simplifies the definitions required for VTAM. Users with more complex networks, where it is important to define lists for performance reasons, should continue to define Adjacent SSCP Tables. Users considering implementing "Dynamic Adjacent SSCPs" should keep in mind performance implications of routing to all CDRMs with which there is an SSCP-to-SSCP session. For example, if a cross-net SSCP-to-SSCP session is active before a same-net SSCP-to-SSCP session, VTAM sends session initiation requests to the cross-net SSCP before it sends to the same-net SSCP.

****

Following is a copy of the PTF cover letter for this new function:

****

7.12.1 Dynamic Adjacent SSCP Table Function

VTAM Version 3 Release 2

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such products, programming, or services in your country.

Comments, suggestions, or questions about this document should be addressed to: IBM Corporation, Department E15, P.O. Box 12195, Research Triangle Park, NC 27709, USA. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

This cover letter is intended to help you plan for, install and use the new Dynamic Adjacent SSCP Table function. It does not contain any programming interfaces.

With the dynamic adjacent SSCP table function, you are no longer required to code adjacent SSCP tables to establish cross-domain and cross-network sessions. With this function, VTAM will dynamically route session establishment requests (both CDINIT and DSRLST requests) to all active adjacent SSCPs until the correct SSCP is found. If you do code adjacent SSCP, tables, VTAM will still use them.

7.12.2 VTAM Publications Affected by Dynamic Adjacent SSCPs

7.12.2.1 VTAM Installation and Resource Definition

Chapter 7 of *VTAM V3R2 Installation and Resource Definition* contains a section with the heading "Defining Adjacent SSCP Tables." You may continue to define adjacent SSCP tables. If VTAM locates an appropriate adjacent SSCP table, it will continue to use this table with no effect from this function.

New Start Option: Chapter 6 of *VTAM Installation and Resource Definition* contains listings for the start options for VTAM V3R2. With dynamic adjacent SSCP tables, a new start option (DYNASSCP) is available.

DYNASSCP = YES|NO determines whether VTAM will dynamically route session establishment requests to all active adjacent SSCPs if no appropriate adjacent SSCP table is defined.

DYNASSCP = YES -- VTAM will determine the adjacent SSCPs when routing a session establishment request across a domain or across a network. If VTAM does not locate an appropriate adjacent SSCP table, it will dynamically route the session establishment request to all active adjacent SSCPs until the correct SSCP is found.

DYNASSCP = NO-- VTAM will not perform dynamic routing. If you have not coded adjacent SSCP tables, session establishments may fail.

7.12.2.2 VTAM Customization

By using the dynamic adjacent SSCP table function in conjunction with the adjacent SSCP selection function of the session management exit routine, you can avoid defining adjacent SSCP tables while maintaining control of session routing. If these functions are both used, the adjacent SSCP selection function is passed information from either the user-defined adjacent SSCP table or, if there is no user-defined table applicable to the request, the dynamic table of all active adjacent SSCPs. You can alter this table to control session routing.

Chapter 4 of *VTAM V3R2 Customization* describes the adjacent SSCP selection function of the session management exit routine.

7.12.2.3 VTAM Operation

DISPLAY ADJSSCPS Command: Chapter 4 of *VTAM V3R2 Operation* describes the DISPLAY ADJSSCPS command. This command displays only adjacent SSCP tables that you have defined. Dynamically defined tables are not displayed.

New Start Option: Chapter 4 of *VTAM Operation* describes the start options available for V3R2.. The new start option DYNASSCP is available with this function. The syntax is the same as described above in "VTAM Installation and Resource Definition."

Chapter 8. Session Continuation

VTAM V3R2, NCP V4R3, and NCP V5R2 provide several enhancements that improve network backup and recovery. Most of these new functions provide enhancements to SSCP ownership changes.

These enhancements are in the areas of :

- Switched PU Session continuation
- Token Ring Session continuation
- BSC Session Continuation (added to NCP V4R2 by PTFs)
- Resource Giveback
- Enhanced Takeover information

The following discussion reviews these enhancements.

8.1 Switched Session Enhancements

VTAM V3R2 with NCP V4R3 or NCP V5R2 supports two new functions for switched links that were previously available only for non-switched SDLC links: non-disruptive loss of SSCP ownership and non-disruptive takeover of SSCP ownership.

In the first case, SSCP ownership is lost when the controlling VTAM terminates; that is, the SSCP-LU session between an LU attached to an NCP and an owning VTAM no longer exists. If, however, the LU currently has an LU-LU session established, then the loss of the controlling SSCP will not disrupt that LU-LU session. This support was previously available only for non-switched SDLC devices. VTAM and NCP now provide this support for switched SDLC devices. This support allows **ANS = CONTINUE** to be specified in the Switched Major Node definition for switched SDLC devices. VTAM passes this automatic network shutdown (ANS) information to NCP after a dial connection has been made. If the NCP enters Automatic Network Shutdown processing for a given host, any LU-LU sessions associated with switched resources owned by that host are not disrupted if **ANS = CONT** has been coded.

The current levels of VTAM and NCP allow another SSCP to "takeover" (activate) PU/LU resources previously owned by an SSCP for which NCP has completed ANS processing. This takeover process is non-disruptive to existing LU-LU sessions on a non-switched SDLC link. This function has been extended to switched SDLC lines in VTAM V3R2, NCP V4R3 and NCP V5R2 environments.

8.2 Token Ring

8.2.1 Token Ring Support in NCP V4R2

Earlier releases of NCP only supported session continuation for SNA devices in a leased line environment. Session continuation enhancements began to emerge for other environments with NCP V4R2. The following discussion addresses the NTRI session continuation capability available with NCP V4R2 and outlines some design enhancements in NCP V4R3 and NCP V5R2 which address some of those limitations.

NCP's implementation of Token Ring support (NTRI) is based on SNA dial. VTAM is not aware of the Token Ring environment. From a design point of view it was desirable to have LU sessions in the token ring environment continue after the owning SSCP was lost.

NCP's handling of LU sessions associated with a particular PU during NCP Automatic Network Shutdown (ANS) processing is implied in the ANS= parameter coded on the PU macro.

Prior to NCP V4R2, ANS=STOP was the only valid option for PUs defined in a switched environment.

NCP V4R2 support was extended to allow ANS=CONT to be specified for the switched Programmed Resources such as those associated with NTRI resources. These extensions allow NTRI LU sessions to survive the loss of an owning host.

One important function that V4R2 Token Ring continuation did not incorporate was resource takeover. This essentially meant that once the original SSCP owner was lost, no other SSCP could activate the PU without forcing the termination of all of the remaining LU-LU sessions.

If ANS=CONT is coded, cross domain NTRI sessions continue after ANS to the owning VTAM; however, any attempt to gain ownership of the NTRI PU will be unsuccessful unless the user is willing to disrupt existing sessions. In order to get ownership of the PU, a user first has to get ownership of the link by issuing an ACTLINK. An ACTLINK to the NTRI virtual line will fail with an 08010004 sense code to any host including the recovered original owning host.

A VARY INACT, FORCE will clean up the LINE and free the PUs and LUs but all of the cross domain sessions will be broken.

This VARY INACT,F could be issued by any VTAM owner of the NCP.

Part of the processing for real switched links (which have to be coded ANS=STOP prior to NCP V4R2) was to break the session and clean up the ownership of the link. It is the omission of the switched link clean up that allows the NTRI sessions to continue. The link is left active but unowned. This is the meaning of the 08010004 sense code returned if a host attempts to activate the link before it has been forced deactivated. Any control blocks such as LUDRPOOL control blocks associated with the switched resources are not available for reuse until the link has been forced deactivated.

The other option for NCP V4R2 NTRI resources is to code ANS=STOP which will cause all of the cross domain sessions to end when the owning host is lost. In this case NCP cleanup is triggered and the NTRI resources are available for use without issuing a forced deactivation.

8.2.2 Token Ring Support Enhancements in NCP V4R3/V5R2

NCP V4R3 and NCP V5R2 enhancements provide better session continuation and recovery characteristics for the NTRI environment. The new releases allow cross domain sessions to continue as they currently do; however, a new SSCP can get ownership of the NTRI resources without disrupting the existing sessions. The new SSCP (or recovered original SSCP) can get control of the resources by first activating the LINK and then the PUs and LUs. In the new environment NCP will not return the 08010004 sense code when the unowned NTRI link is activated. Once a new owner has been established for the NTRI resources this VTAM owner (if it is V3R2) can now issue the new VTAM **GIVEBACK** command and return the link to an unowned state, making it possible for other eligible VTAMs to become the owner. This **GIVEBACK** command is discussed in Section 8.5.

8.3 NPSI Session Continuation

The NPSI V1 products provided session continuation for X.25 resources defined on Permanent Virtual Circuits (PVCs). These NPSI releases (V1R4, V1R4.2 and V1R4.3) are used with their compatible NCP release of NCP V4R2 and below. NCP V4R3 and NCP V5R3 and their compatible NPSI release exploit the new session continuation enhancements by extending session continuation to NPSI resources connected through Switched Virtual Circuits (SVCs). Both NPSI PVCs and SVCs can take advantage of the **GIVEBACK** function to be discussed later if VTAM V3R2 is used. The NPSI session continuation support is essentially the same as that for real SNA resources when the new releases of VTAM, NCP and compatible NPSI products are used.

8.4 BSC Session Continuation

One additional enhancement to session continuation for the NCP V4R2 environment is the Session Continuation capability for BSC devices. This support is implemented by PTFs in the NCP V4R2 environment and is incorporated into the subsequent NCP releases.

NCP action during ANS processing is dictated by the ANS= specification on the LINE macro. A third option has been added to this parameter in order to support BSC session continuation.

The valid options for **ANS** are now:

ANS= **STOP**

CONTINUE

CONT

DELAY

ANS = DELAY should be specified in order to invoke BSC Session Continuation. This will defer the ANS processing of the BSC devices until all of the sessions are ended or until an SSCP attempts to activate the link. After one of the above conditions is met, normal ANS processing for the BSC line is invoked. Even though existing sessions are allowed ·to continue, no SSCP can activate the CLUSTER without disrupting the remaining sessions. This means that no new session can be started for the devices attached to the CLUSTER since it is unowned. Once the link and CLUSTER are reactivated, the devices are again able to participate in sessions.

The BSC Session Continuation function applies only to BSC 3270s.

8.5 Resource GIVEBACK Enhancements

Prior to VTAM V3R2, NCP V4R3, and V5R2, an SSCP could take over ownership of resources on non-switched lines but that ownership could never be returned to the original SSCP or any other SSCP in a non-disruptive manner. Several techniques, using releases of VTAM prior to V3R2, allow existing ownership to be relinquished so as to allow the original or another SSCP to regain ownership. One such technique is simulating an SSCP failure by deactivating a channel-link station for a channel-attached NCP. This causes the NCP to enter Automatic Network Shutdown (ANS) and thus to free the resource ownership without disrupting existing sessions. In other words, NCP's ANS process removes the name of the owning SSCP for a resource when communication with that SSCP is lost. If LU-LU sessions exist which are not affected by the lost SSCP, NCP allows the LU-LU sessions to continue as long as ANS = CONT or DELAY has been specified. If ANS = STOP has been specified, the LU-LU sessions are terminated by NCP. The removal of the owning SSCP from a resource allows another VTAM to activate that resource. Another technique for changing the ownership of SNA resources was through the use of the RELEASE and ACQUIRE options of the VTAM VARY command. The VARY with the ACQ option is non-disruptive for existing sessions. VTAM V3R2, NCP V4R3, and V5R2 extend this support to switched environments. The problem common to all of the earlier VTAMs and NCPs was that there was no way of returning resources to the original owner nondisruptively.

A new VTAM/NCP facility allows **GIVEBACK** of SSCP ownership to an alternate or to the original SSCP without disrupting existing sessions. A new VTAM command together with facilities in NCP V4R3 and V5R2 implement this GIVEBACK function. This enhancement applies to both switched and non-switched SNA real resources and for Programmed resources that want to take advantage of it.

GIVEBACK is initiated by the operator command:

VARY INACT,TYPE = GIVEBACK,ID = linkname

GIVEBACK generates a new form of DACTLINK and must be issued for each link in order to relinquish resource ownership. The command does not support the specification of an NCP, PU, or LU operand in the ID operand. Resource ownership is relinquished so that a different SSCP (including the original owner) can gain ownership of the resources by using the normal "VARY ACTIVATE." In contrast to this **GIVEBACK** facility, the VTAM **VARY RELEASE** process is disruptive to existing sessions with levels of software prior to those explained in the next paragraph.

With NCP V4R3.1, NCP V5R2.1, SSP V3R4.1 and the following levels of VTAM V3R2:

- VTAM V3R2 for MVS/XA with FMID JVT3215
- VTAM V3R2 for MVS/370 with FMID JVT3214
- VTAM V3R2 for VM at level 5664280D
- VTAM V3R2 for VSE at level G70

enhancements have been made to the VARY RELEASE command to allow non-disruptive GIVEBACK of resources. GIVEBACK is initiated by the operator command:

VARY NET,REL,ID = ncpname or puname,TYPE = GIVEBACK

The advantage of using the RELEASE rather than the INACTIVATE command is that an NCP can be specified, rather than every link within an NCP as required on the INACTIVATE command.

8.6 Takeover/Giveback Example with ACTIVATE/INACTIVATE

Using Figure 8-1 on page 8-8 as an example, assume that VTAM1 is a CMC, has activated NCP11 and owns all its resources, the PUs and LUs on Leased and Switched SDLC links, on BSC lines, attached to the Token-Ring, and attached using an X.25 packet-switched data network. Assume also that all of the resources except the T2.1 nodes have LU-LU sessions with the VTAM application program, APPL, running on the backup host, VTAM2. The T2.1 nodes have LU-LU sessions with each other.

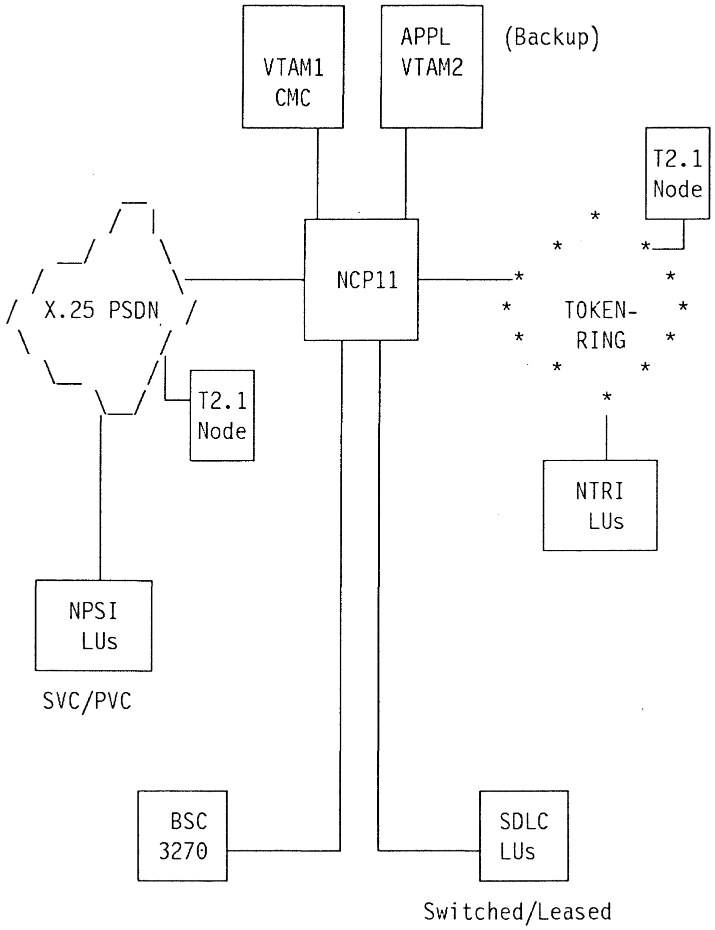


Figure 8-1. Session Continuation Example Using Activation/DeActivation Processing

If VTAM1 goes down, NCP goes into ANS and the LU-LU sessions continue, assuming that ANS=CONT (ANS=DELAY for BSC) has been coded for all the resources. VTAM2 can then activate NCP11, SCOPE=ALL and gain ownership of all the resources. The LU-LU sessions can continue through this process with the exception of the BSC line, where the LU-LU sessions would be terminated when VTAM2 activated the line.

After VTAM1 is brought back up, VTAM2 can issue:

```
VARY NET,INACT,TYPE=GIVEBACK,ID=each token-ring virtual line name
VARY NET,INACT,TYPE=GIVEBACK,ID=each T2.1 node line name
VARY NET,INACT,TYPE=GIVEBACK,ID=each SDLC link name
VARY NET,INACT,TYPE=GIVEBACK,ID=each X.25 VC name
```

This will result in the devices, except the BSC one, becoming unowned so that another VTAM can gain ownership of them. The LU-LU sessions can continue through this process, although no new session initiation requests can occur until a device has an owning VTAM.

After the V NET,INACT,TYPE=GIVEBACK commands have completed, VTAM1 can then regain ownership of the NCP and all resources by issuing the command:

```
V NET,ACT,SCOPE=ALL,ID=NCP11
```

8.7 Takeover/Giveback Example with ACQUIRE/RELEASE

Using Figure 8-2 as an example, assume that VTAM1 is a CMC, has activated NCP11 and owns all its resources, the PUs and LUs on Leased and Switched SDLC links, on BSC lines, attached to the Token-Ring, and attached using an X.25 packet-switched data network. In the NCP coding, all of the devices have been coded using the statement OWNER=VTAM1. In the PCCU statement pertaining to VTAM1, OWNER=VTAM1 has been coded. In the PCCU statement pertaining to VTAM2, OWNER=VTAM2 has been coded, as has BACKUP=YES.

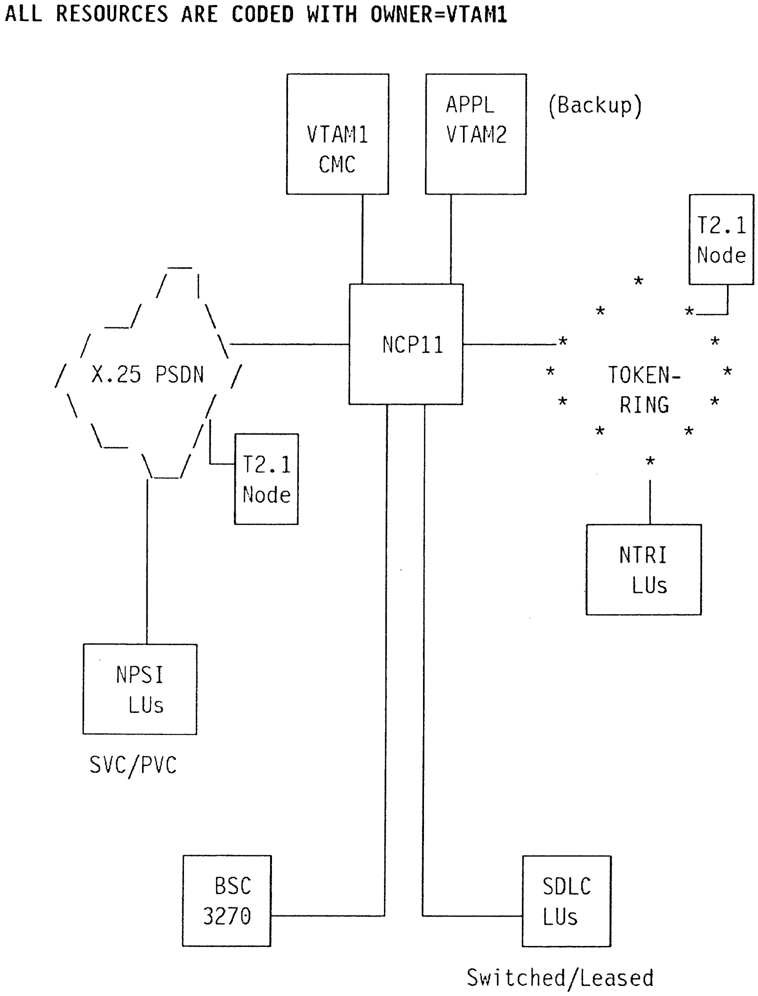


Figure 8-2. Session Continuation Example Using Acquire/Release Processing

Using this type of coding, when both VTAMs activate NCP11, VTAM1 will gain ownership of the NCP and all the resources. VTAM2 will gain ownership of the NCP, but not the PUs and LUs. VTAM2 will build a partial control block structure for the PUs and LUs, and when the links are displayed, they will show up with a status of "RESET".

Assume also that all of the resources except the T2.1 nodes have LU-LU sessions with the VTAM application program, APPL, running on the backup host, VTAM2. The T2.1 nodes have LU-LU sessions with each other.

If VTAM1 goes down, NCP goes into ANS and the LU-LU sessions continue, assuming that ANS=CONT (ANS=DELAY for BSC) has been coded for all the resources. The operator at VTAM2 then needs to issue the VARY ACQUIRE command for the NCP and all of the PUs before issuing a VARY ACTIVATE command for the NCP SCOPE=ALL.

A further enhancement with VTAM V3R2 is that the operator can now specify PUSUB on the VARY ACQUIRE command and acquire the NCP and all PUs with one command, rather than having to issue multiple commands. The new form of the command is

```
VARY NET,ACQ,ID=ncpname,PUSUB
```

The LU-LU sessions can continue through this process with the exception of the BSC line, where the LU-LU sessions would be terminated when VTAM2 activated the line.

After VTAM1 is brought back up, VTAM2 can issue:

```
VARY NET,RELEASE,TYPE=GIVEBACK,ID=NCP11
```

This will result in the devices, except the BSC one, becoming unowned so that another VTAM can gain ownership of them. The operator will need to issue a VARY NET INACT command for the BSC device. The LU-LU sessions can continue through this process, although no new session initiation requests can occur until a device has an owning VTAM.

8.8 Enhanced Session Takeover Information

NCP V4R3 and NCP V5R2 are capable of supplying additional information to a VTAM V3R2 which takes over resources previously owned by another host. This information includes such things as the LU session partner names as well as the owning SSCP names. This extended takeover information is available for any boundary LU that supports extended BINDs. An LU's ability to support extended BINDs is indicated in the response to an ACTLU. NCP passes this information to VTAM in several new control vectors. For independent LUs this information flows in a new boundary function RU. For dependent LUs these control vectors flow in the ACTLU response.

An example of how useful this additional information can be is discussed below.

The VTAM RELREQ facility in the current product allows a VTAM application to voluntarily give up its session with an LU in order that some other PLU can start a session with that LU. In an environment where the LU had been involved in a cross domain session and the NCP had gone into ANS to the original owner, the session partner information is lost to any subsequent owner of the LU (including the original owner) after takeover occurs. If an application now requests a session with the LU, the owner would not know who the session partner was and hence would not know where to route the RELREQ.

This would result in a VTAM MSG "IST784I SESSION(s) EXIST WITH UNKNOWN PARTNER". The new products (VTAM V3R2, NCP V4R3, and NCP V5R2) provide some immunity against this exposure. If the boundary LU involved supports extended BINDs, this partner LU information would be available to the new owner and the RELREQ could be routed on to the session partner. CICS implements RELREQ support and hence is a good candidate for taking advantage of this enhancement.

8.9 Session Continuation Matrix

The following chart summarizes the session continuation support for various NCP resources as a function of NCP level. The chart also assumes VTAM V3R2 when the function itself requires VTAM V3R2 such as in the case of **GIVEBACK**.

	BSC 3270	**SNA LUs**	**NTRI LUs**	**NPSI LUs**
SESSION CONTINUATION LEASED	• V4R2 • V4R3 • V4R3.1 • V5R1 • V5R2 • V5R2.1	ALL RELEASES	N/A	• V4R2 • V4R3 • V4R3.1 • V5R1 • V5R2 • V5R2.1
SESSION CONTINUATION SWITCHED	N/A	• V4R3 • V4R3.1 • V5R2 • V5R2.1	• V4R2 • V4R3 • V4R3.1 • V5R1 • V5R2 • V5R2.1	• V4R3 • V4R3.1 • V5R2 • V5R2.1 • **NOTE 1**
GIVEBACK	NO	• V4R3 • V4R3.1 • V5R2 • V5R2.1	• V3R3 • V4R3.1 • V5R2 • V5R2.1	• V4R3 • V4R3.1 • V5R2 • V5R2.1 • **NOTE 1**

SESSION CONTINUATION MATRIX

Notes:

1. The NPSI releases that correspond to the various NCP releases are:

NCP V4R3	NPSI V2R1
NCP V4R3.1	NPSI V2R1
NCP V5R1	NPSI V3R1
NCP V5R2	NPSI V3R2
NCP V5R2.1	NPSI V3R2

Chapter 9. NCP V4R3 Storage Usage

9.1 Design Overview

NCP V4R3 and NCP V5 introduce many significant functional enhancements. The following discussion examines the impact that these enhancements have on NCP storage requirements and how storage is used in the new releases.

NCP V4R3 and NCP V5 have undergone control block restructures in order to support the new NCP functions. In some cases the storage increases have been significant as in the case of the amount of storage that it takes to represent an LU. This increase is about 170 bytes per LU. In addition to increases in the amount of storage required, the new releases use storage differently than previous NCP releases.

Earlier NCP load modules consisted of the NCP control code and control blocks, including the ones representing the NCP SNA resources. Included in the control block portions of the NCP were the control block pools used to represent resources for which control blocks were acquired dynamically. The pooled resources in these early releases were in support of SNA dial, Dynamic Reconfiguration and SNI cross network sessions.

The control block restructure in NCP V4R3 and NCP V5 is primarily in support of the enhancements in the NCP boundary functions required for Type 2.1 Node Support.

In previous NCP releases there was essentially one address associated with a boundary LU and all of the LU related information was contained in a single control block called the LUB. In order to support the Type 2.1 Node environment, where boundary LUs could potentially have a total of 128K sessions, a fixed single control block approach would have resulted in wasted storage when an LU was involved in less than the architected maximum number of sessions. The new control block implementation separates the LU functions from the LU address(s) and session(s).

The new design is based on a pooled control block concept. Control blocks are taken from the pools on demand. Some of these control block pools reside at the high storage addresses above the NCP buffer pool. This new approach uses several control blocks to represent LUs . There are three categories of control blocks: those representing the LU itself; those representing the LU's address(s); and those representing the LU's session(s). All of these control blocks are pooled resources and are dynamically allocated. These allocations can occur at NCP generation time, NCP initialization time or when the LU acquires sessions. This results in a more efficient overall use of storage since the pools only have to be capable of satisfying peak demand.

This control block restructure is applicable to both dependent and independent LUs.

9.2 Effects of NCP Parameters on Control Block Pools

The control blocks used to hold LU information are built in pools at either NCP generation time or NCP initialization time with NCP V4R3 and NCP V5. Those built at initialization time reside in high storage above the NCP buffer pool. The amount of storage used by these pools is not included in the NDF LNKEDT load module size.

The LU control block pool sizes are determined by NCP generation parameters. For each independent LU generated (LOCADDR = 0), there are control blocks allocated at NCP initialization time from the pools in high storage. The amount of storage allocated totals 208 bytes. For each dependent LU generated, there are also control blocks allocated at NCP initialization time from these pools. The amount of high storage allocated totals 264 bytes.

On the LU statement for an independent LU there is an RESSCB keyword which specifies the number of LU-LU session control blocks which should be reserved from the pool for this particular independent LU. RESSCB is not valid for dependent LUs. One session control block is needed for every session with which this LU is involved. A generated LU is automatically given a session control block even if RESSCB = 0 and more can be obtained from the pool via the ADDSESS keyword on the BUILD statement. For each value specified, 208 bytes of high storage will be allocated.

On the BUILD statement there are four parameters which affect the amount of high storage formatted into LU-related control blocks at NCP initialization. They are ADDSESS, AUXADDR, NAMTAB, and SESSACC.

- The ADDSESS keyword specifies the number of LU-LU session control blocks available for use by any independent LU in addition to those reserved by the RESSCB keyword on the LU statement and generated by the AUXADDR keyword. This keyword should be defaulted to zero if there are only dependent LUs in this NCP. For each value specified, 184 bytes of high storage will be allocated.

- The AUXADDR keyword specifies the number of additional PLU addresses needed by all independent LUs for use with parallel sessions. The address each LU is given at NCP initialization is used for all sessions in which the LU is a SLU. For each independent LU which can be a PLU, one should be added to this keyword. If any of these independent LUs are involved in parallel sessions with the same SLU, one should be added to this keyword for every parallel session. This keyword generates an additional LU-LU session control block to be used with the generated address. This keyword should be defaulted to zero if there are only dependent LUs in this NCP. For each value specified, 208 bytes of high storage will be allocated.

- NAMTAB specifies the number of entries in the network names table. The names stored in this table are the unique names received in various session activation PIUs - SSCP, Network, and CP names. For each value specified, 10 bytes of high storage will be allocated.

- The third operand of the SESSACC keyword specifies control blocks used for the NPA Session Accounting function. One block is needed for every session for which accounting information is needed. For each value specified, 40 bytes of high storage will be allocated.

Each keyword on the LUDRPOOL statement affects the amount of high storage formatted into LU-related control blocks at NCP initialization.

- The NUMTYP1 keyword specifies the number of dependent LUs for Type 1 PUs that can be DR-added. For each value specified, 264 bytes of high storage will be allocated.
- The NUMTYP2 keyword specifies the number of dependent LUs for Type 2.0 PUs that can be DR-added. For each value specified, 264 bytes of high storage will be allocated.
- The NUMILU keyword specifies the number of independent LUs for Type 2.1 PUs that can be DR-added. For each value specified, 208 bytes of high storage will be allocated.

9.3 NCP Storage Organization

Figure 9-1 contains a generalized map of NCP storage for NCP V4R3 and the V5 NCPs. The NCP load module and control blocks which are built at generation time occupy low storage. The NCP buffer pools reside above this region. The dynamic control block pools containing most of the LU and LU session related control blocks discussed in this section reside above the NCP buffer pool. These pools, as well as the NCP buffer pools are built at NCP initialization time. The MOSS Mail Box, which is used to communicate between NCP and MOSS is located in high storage above the dynamic control pools.

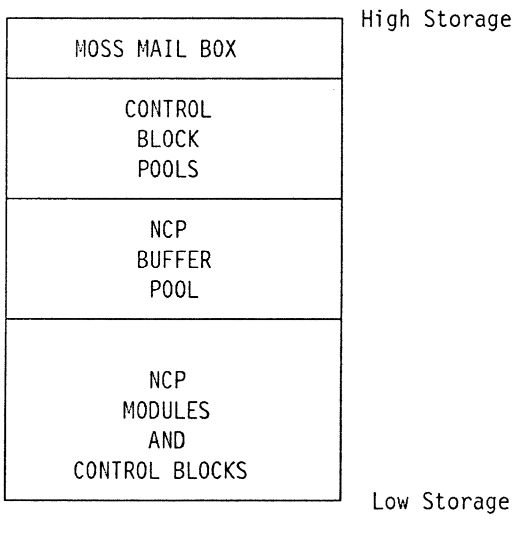


Figure 9-1. NCP V4R3 and V5 Storage Map.

9.4 NCP V4R3 and NCP V5 Storage Considerations

From the previous discussion it is seen that the NDF load module LNKEDT size does not account for significant amounts of NCP storage. The NDF LNKEDT load module size provided a good estimate of the NCP storage (exclusive of buffers) required for previous releases of NCP, but it is not a reliable source for NCP V4R3 and NCP V5. In fact, it is possible that the NDF load module size could indicate the storage required for NCP V4R3 or V5 is significantly less than that required for NCP V4R2. The primary reason for this discrepancy is that the load module size for NCP V4R2 contains the LU control blocks, while that for NCP V4R3 and NCP V5 does not. The pools from which these control blocks are taken are not allocated until NCP initialization time for NCP V4R3 and NCP V5.

A great deal of detail planning is required before selecting the NCP parameters that influence the sizes of the LU control block pools.

For example, consideration must not only be given to the number of sessions that an independent LU has, but consideration must also be given to the session polarity (PLU/SLU) and the number of parallel sessions an LU will have with a target LU. Underestimation will limit the number of sessions that independent LUs can have; however, overestimation will result in unnecessary storage being reserved. This overestimation reduces the amount of storage available for NCP buffers.

The storage increases for NCP V4R3 and NCP V5 are significant and the storage calculations are different from all previous releases of NCP. **It is imperative that the appropriate configurator (CF3725/3745) be run for all network designs involving the new NCP releases.**

9.5 NCP Loadmodule Size Restriction

All current NCP (up to and including V5R3) load modules are limited to a maximum size of 4 Megabytes (4,194,304 bytes).

This restriction is a result of a 22 bit addressing limitation of the current 37XX LA and BAL Assembler instructions. This limitation does not appear to be documented in any of the current NCP manuals, although the 37xx Principles of Operations manuals, such as SA33-0102, document the fact that these instructions only have 22 bit addressing capability.

All NCP executable modules including user code and selected control blocks are included in the 4 Megabyte load module limit. Some NCP control blocks, such as some of the LU and session control blocks, are allocated dynamically at initialization time for NCP V4R3 and NCP V5R2 and are not a consideration in this 4Meg calculation. See the storage discussion in section 9.1.

Those control blocks dynamically generated at initialization time are not included in the loadmodule size.

HSCBs (half session control blocks) are not included in the load module when the following maintenance is applied:

Table 9-1. PTFs AND RELATED APARs FOR SSP. SSP PTFs

PTF	APAR	RELEASE
UR25284	IR82703	MVS V3R4
UR25283	IR82703	MVS V3R4.1
UR25479	IR82827	VM V3R4
UR25480	IR82827	VM V3R4.1
UR25790	IR82828	VSE V3R4
UR25791	IR82828	VSE V3R4.1

Table 9-2. NCP V5R2 V5R2.1 PTFs AND RELATED APARS

PTF	APAR	RELEASE
UR25317	IR82737	MVS V5R2
UR25316	IR82737	MVS V5R2.1
UR25319	IR82737	VM V5R2
UR25320	IR82837	VM V5R2.1
UR25318	IR82837	VSE V5R2
UR26678	IR82830	VSE V5R2.1

Table 9-3. NCP V4R3 and V4R3.1 PTFs AND RELATED APARS

PTF	APAR	RELEASE
UR25322	IR82746	MVS V4R3
UR25321	IR82746	MVS V4R3.1
UR25324	IR82746	VM V4R3
UR25325	IR82745	VM V4R3.1
UR25323	IR82746	VSE V4R3
UR26676	IR82829	VSE V4R3.1

The 37XX configurators have been updated to be more useful in evaluating NCP storage requirements. See Figure 9-2 on page 9-8 for an example of the new storage and performance report generated by CF3745.

```
        **SAMPLE STORAGE OUTPUT REPORT**

     OCT 19,1989      CF3745 REPORT 3         PAGE: 2


           STORAGE AND PERFORMANCE SUMMARY

      STORAGE ESTIMATES                      BYTES

      EP/3745 PROGRAM MODULE                 55860
      NCP V5R2.1 PROGRAM MODULE             479592
      PRPQ OR USER WRITTEN PROGRAMS         600000
      EP CONTROL BLOCK                       16288
      CONTROL BLOCK                         255980
      TOTAL BUFFERING REQUIREMENT            91440*
                                        ----------
      TOTAL STORAGE                        1499160

    **NEW LOAD MODULE ESTIMATE**

      LOAD MODULE SIZE                     1264360


  If your NCP load module size is approaching 4 MB, please
  consult either the NCP qbucket or the 3745 qbucket on
  INFO/SYSTEM for additional information.

  **END OF SAMPLE STORAGE OUTPUT REPORT**
```

Figure 9-2. CF3745 Output Storage Report

The "TOTAL STORAGE" value shown includes all NCP executable code, all user executable code, all control blocks built at generation time, all control blocks built at NCP initialization time and all buffers required.

The "LOAD MODULE SIZE" value shown includes all NCP executable code, all user executable code, and all control blocks built at generation time. The control blocks blocks built at NCP initialization time and buffer requirements need not be included in this 4Meg calculation. They can be placed within the first 4MB of storage or in the next increment of storage. The load module size determines where these control blocks and buffers are placed.

The configurator has been updated to issue the message shown in Figure 9-3 on page 9-9 when the critical 4Meg limit has been reached:

```
THE AIDS STOP EXECUTING DUE TO THE FIRST REASON IT ENCOUNTERS,
OTHER REASONS MAY ALSO EXIST:
ERROR - NCP HAS A CRITICAL STORAGE REQUIREMENT - FOR CCU A

On the 3745, it is a hardware requirement that the load module size
not exceed 4,194,304 bytes.  Your load module currently is 6609951.
In general, link control blocks for SDLC, NPSI Virtual Circuits,
NTRI Logical Lines, any additional storage for IBM RPQs or User Programs
(Q:655 or Q:660), etc. contribute to the load module size.
For NCP (V5R2.1 only), the combination of ERLIMIT (q.465), NUMNETWORKS
q.465),
and SALIMIT (q.465) will have a large effect on the load module size.

TO REVIEW AND CHANGE THE NECESSARY RESPONSES, PRESS THE EDIT PF KEY,
OR PRESS THE MANAGER PF KEY AND ENTER 3 IN THE OPTION
FIELD TO RE-EXECUTE IN STEP MODE
```

Figure 9-3. CF3745 Output Message when 4MEG Restriction Exceeded

Please see Chapter 12, Section 12.7 of this manual for more information concerning the impact of SALIMIT and ERLIMIT values on NCP storage.

9.5.1 Symptoms of Exceeding 4MEG Loadmodule Limitation

NDF provides a Gen time message if the limit is reached. When loading a load module larger than 4M, NCP abends with 92B ABEND code upon initialization.

When transferring a load module to disk that is larger than 4M, the following messages will be presented:

IST961I NON-DISRUPTIVE LOAD OF ncpname (WITH load mod name) FAILED

IST523I REASON = LOAD MODULE TOO LARGE

The above information will help the user to better understand the current NCP loadmodule size restrictions and to recognize the associated symptoms at both NCP generation and load time.

9.6 Configuring the NCPs with CF3725/CF3745

Each run of CF3725 assumed that there is one INN link (56KB, full-duplex) and 50 BNN links (9600bps half-duplex) attached to the 372X. Each BNN link had a PU T2.0 node attached to it; for these runs, a 3174 was used. That configuration remained constant throughout all permutations of CF3725 executions. It is a small configuration for a 3725, but it facilitates uniformity when discussing both the 3720 and the 3725. The NCP V4 Subset was not used because this configuration is larger than the Subset supports, i.e., it consisted of more than 28 lines.

The one variable in the CF3725 runs is the number of sessions supported. *NPP Storage Estimates*, SC30-3403, uses 1600 LUs in its example. The tables that follow show the storage requirement for 1000, 1600, 2000, and 3000 LUs. NPM data was collected for all LUs, which led to a higher storage requirement than might otherwise have been needed. NPA Session Accounting was not used.

CAUTION:
The following tables do not include the buffering requirements for each particular NCP. Furthermore, these figures are estimates for a specific configuration and are not to be construed as authoritative for all configurations or environments. The tables are not to be used as a substitute for running CF3725.

9.7 Migrating to NCP V4.3

9.7.1 NCP V4.2

Table 9-4 sets the stage for the ensuing storage comparisons. All scenarios involve migrating from NCP V4.2, which is the last NCP that will run on 3720s as well as 3725s. Thus, it is a platform to use when planning migrations for either communications controller.

Table 9-4. NCP V4.2 Storage Matrix. 3720/3725 storage required for code and control blocks

NCP V4.2 Storage	**372X storage required for LUs (in bytes)**			
Type of NCP storage	1000 LUs	1600 LUs	2000 LUs	3000 LUs
NCP Program Module	254830	254830	254830	254830
NCP Control Blocks	276459	392379	469659	662859
Total	**531289**	**647209**	**724489**	**917689**

9.7.2 NCP V4.3

The results of CF3725 runs for a 3725 and NCP V4.3 are shown in Table 9-5.

Table 9-5. NCP V4.3 Storage Matrix. 3725 storage required for code and control blocks

NCP V4.3 Storage	**3725 storage required for LUs (in bytes)**			
Type of NCP storage	1000 LUs	1600 LUs	2000 LUs	3000 LUs
NCP Program Module	445047	445047	445047	445047
NCP Control Blocks	484548	701268	845748	1206948
Total	**929595**	**1146315**	**1290795**	**1651995**

9.7.3 Comparison of NCP V4.2 and NCP V4.3

Table 9-6 conveys the relative storage increase for the 3725 when migrating to NCP V4.3 from NCP V4.2.

Table 9-6. NCP V4.2/V4.3 Storage Comparison. Increase in 3725 storage required by NCP V4.3

Total NCP storage	**3725 storage required for LUs (in bytes)**			
NCP Version	1000 LUs	1600 LUs	2000 LUs	3000 LUs
NCP Version 4.3	929595	1146315	1290795	1651995
NCP Version 4.2	531289	647209	724489	917689
Total storage increase	**398306**	**499106**	**566306**	**734306**

9.8 Migrating to NCP V5.1

9.8.1 NCP V5.1

The 3720 can migrate either to NCP V5.1 or NCP V5.2 from NCP V4.2. The tables that follow show the storage impact for each migration path. The path that will need the most careful attention will be the migration from NCP V4.2 to NCP V5.1 or NCP V5.2; migrating from NCP V5.1 to NCP V5.2 has relatively minor impact on 3720 storage (see Table 9-11 on page 9-13).

Table 9-7 shows the results of a CF3725 analysis for NCP V5.1 on a 3720.

Table 9-7. NCP V5.1 Storage Matrix. 3720 storage required for code and control blocks

NCP V5.1 Storage	**3720 storage required for LUs (in bytes)**			
Type of NCP storage	1000 LUs	1600 LUs	2000 LUs	3000 LUs
NCP Program Module	413406	413406	413406	413406
NCP Control Blocks	480088	696808	841288	1202488
Total	**893494**	**1110214**	**1254694**	**1615894**

9.8.2 Comparison of NCP V4.2 and NCP V5.1

Table 9-8 on page 9-12 outlines the storage increase for a 3720 when migrating to NCP V5.1 from NCP V4.2.

Table 9-8. NCP V4.2/V5.1 Storage Comparison. Increase in 3720 storage required by NCP V5.1

Total NCP storage	3720 storage required for LUs (in bytes)			
NCP Version	1000 LUs	1600 LUs	2000 LUs	3000 LUs
NCP Version 5.1	893494	1110214	1254694	1615894
NCP Version 4.2	531289	647209	724489	917689
Total storage increase	**362205**	**463005**	**530205**	**698205**

9.9 Migrating to NCP V5.2

9.9.1 NCP V5.2

Results of CF3725 for a 3720 and NCP V5.2. Note that the storage requirement for the 3720 and NCP V5.2 is equal to that of the 3725 and NCP V4.3.

Table 9-9. NCP V5.2 Storage Matrix. 3720 storage required for code and control blocks

NCP V5.2 Storage	3720 storage required for LUs (in bytes)			
Type of NCP storage	1000 LUs	1600 LUs	2000 LUs	3000 LUs
NCP Program Module	445047	445047	445047	445047
NCP Control Blocks	484548	701268	845748	1206948
Total	**929595**	**1146315**	**1290795**	**1651995**

9.9.2 Comparison of NCP V4.2 and NCP V5.2

The relative storage increase for the 3720 when migrating to NCP V5.2 from NCP V4.2 is shown in Table 9-10.

Table 9-10. NCP V4.2/V5.2 Storage Comparison. Increase in 3720 storage required by NCP V5.2

Total NCP storage	3720 storage required for LUs (in bytes)			
NCP Version	1000 LUs	1600 LUs	2000 LUs	3000 LUs
NCP Version 5.2	929595	1146315	1290795	1651995
NCP Version 4.2	531289	647209	724489	917689
Total storage increase	**398306**	**499106**	**566306**	**734306**

9.9.3 Comparison of NCP V5.1 and NCP V5.2

The relative storage increase for the 3720 when migrating to NCP V5.2 from NCP V5.1 is in Table 9-11.

Table 9-11. NCP V5.1/V5.2 Storage Comparison. Increase in 3720 storage required by NCP V5.2

Total NCP storage	**3720 storage required for LUs (in bytes)**			
NCP Version	1000 LUs	1600 LUs	2000 LUs	3000 LUs
NCP Version 5.2	929595	1146315	1290795	1651995
NCP Version 5.1	893494	1110214	1254694	1615894
Total storage increase	**36101**	**36101**	**36101**	**36101**

9.10 NCP V4.3 and V5.2 with PU T2.1 Support

The following storage comparisons are based on the 372Xs studied above and a 3745. The configuration differs in that the communications controllers now have a T2.1 device with 5 independent LUs (ILUs) attached via an SDLC 9600 bps half-duplex line. Each ILU is capable of three sessions. Three storage analyses are shown:

1. A 37XX with the same configuration as examined previously, but with a PU T2.0 and 5 dependent LUs (DLUs) in place of a PU T2.1 and 5 ILUs. This will show the incremental storage increase when going to a T2.1 environment where the number of LUs are the same, but the control blocks required for those LU's sessions differs because of the T2.1 implementation of ILUs. This is the **COMP** case.
2. A 37XX with a PU T2.1 and 5 ILUs with no extra session control blocks (ADDSESS and AUXADDR in the BUILD macro, NUMILU in the LUDRPOOL macro, and RESSCB in the LU macro) defined. This is the **BASE LEN** case.
3. A 37XX with a PU T2.1 and 5 ILUs that are defined in the NCP to have three sessions (RESSCB=3), with the capability to add one additional LU-LU session (ADDSESS=1) and one extra PLU session with the same SLU (AUXADDR=1). This is the **LEN** case.

Each case assumes 1600 DLUs in addition to those being configured here.

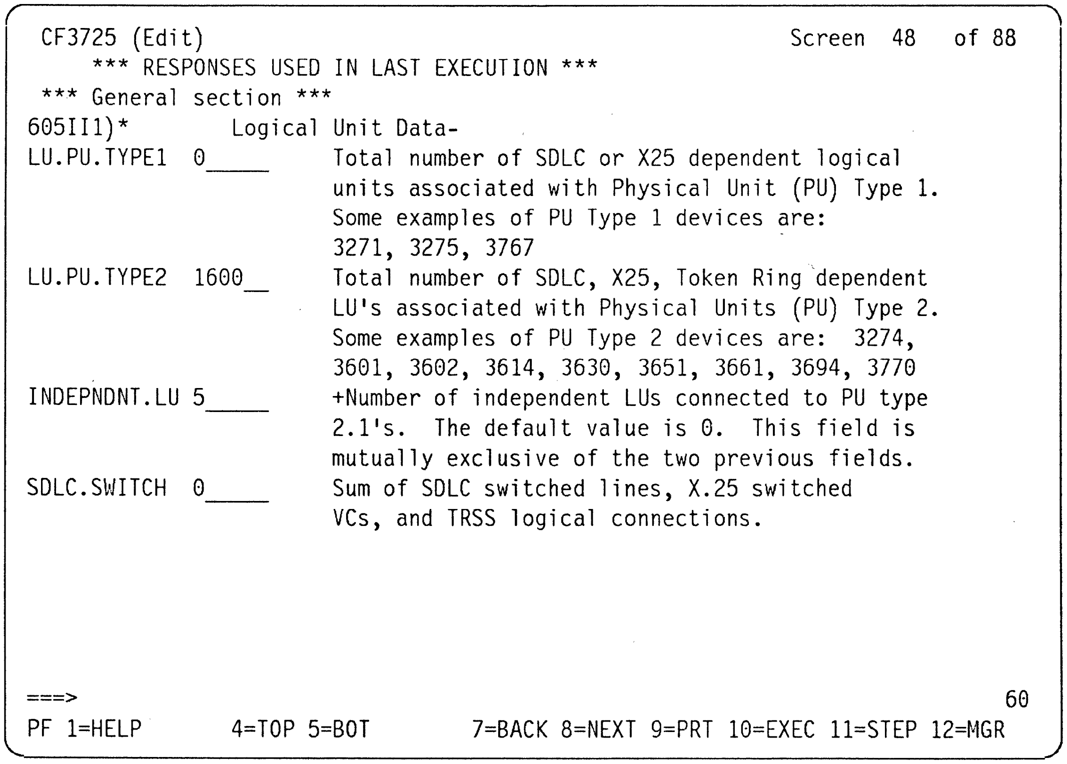
The intent is to demonstrate the storage considerations of this new NCP function in light of the 'old' function and also to demonstrate the effect of the new parameters in the NCP BUILD and LU macros. These configurations are very limited in their scope, due to the number of resources and sessions defined, but they should give an indication of the NCP storage required in order to use them.

CAUTION:
The following tables do not include the buffering requirements for each particular NCP. Furthermore, these figures are estimates for a specific configuration and are not to be construed as authoritative for all configurations or environments. The tables are not to be used as a substitute for running CF3725/CF3745.

9.10.1 CF3725 Answers

The following three screens from CF3725 show the key questions to answer when configuring PU T2.1 resources. The same questions are in CF3745, too, but the questions are numbered differently. The answers to questions 607 and 610 have a direct impact on controller storage because they are used to determine the size of the NCP pools used to provide control blocks for these sessions. Inaccurate answers to these questions in CF3725/CF3745 can result in either insufficient or excessive storage allocation.

Ensure that changes to an existing CF3725 or CF3745 file are accurately reflected in the configurator output. It is possible that the configurator will not update its reports if, for example, a new line using PU T2.1 is configured without verifying Q605 has the correct number of ILUs, resulting in a report that will not show the correct number of LUs in the configuration. This is particularly true if changes are made via EDIT rather than STEP EXECUTION mode. The resulting incorrect reports could lead to false assumptions about the storage required to support the desired configuration.



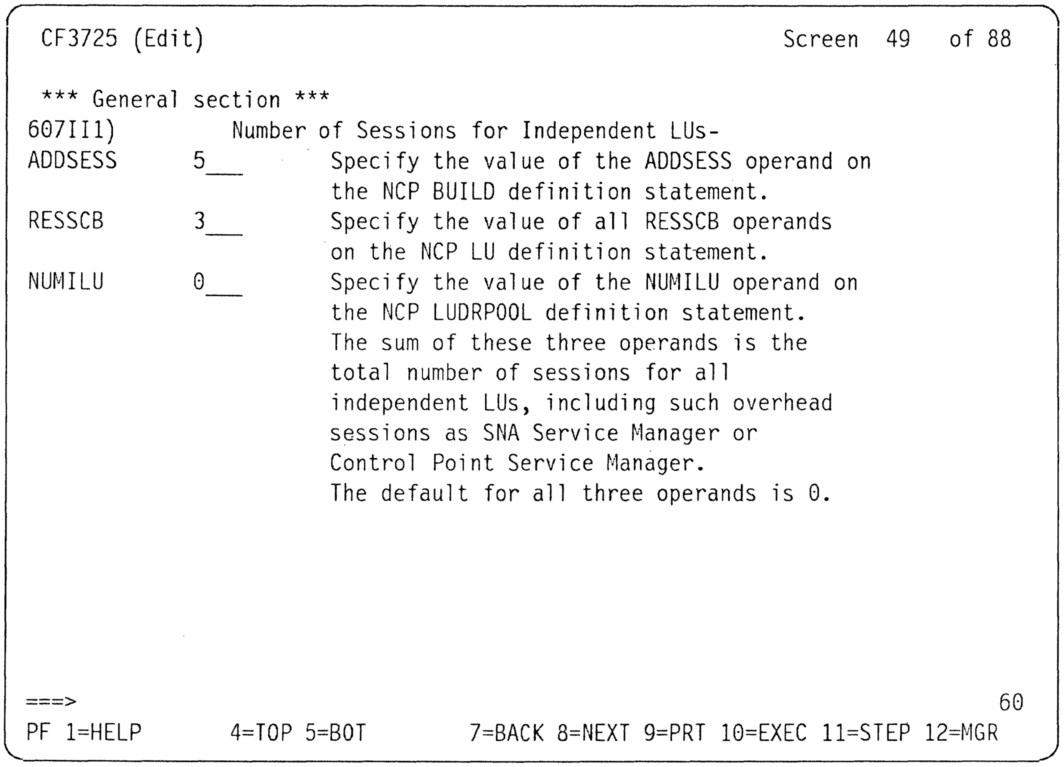
```
 CF3725 (Edit)                                          Screen  48   of 88
     *** RESPONSES USED IN LAST EXECUTION ***
 *** General section ***
605II1)*       Logical Unit Data-
LU.PU.TYPE1  0_____     Total number of SDLC or X25 dependent logical
                        units associated with Physical Unit (PU) Type 1.
                        Some examples of PU Type 1 devices are:
                        3271, 3275, 3767
LU.PU.TYPE2  1600__     Total number of SDLC, X25, Token Ring dependent
                        LU's associated with Physical Units (PU) Type 2.
                        Some examples of PU Type 2 devices are:  3274,
                        3601, 3602, 3614, 3630, 3651, 3661, 3694, 3770
INDEPNDNT.LU 5_____     +Number of independent LUs connected to PU type
                        2.1's.  The default value is 0.  This field is
                        mutually exclusive of the two previous fields.
SDLC.SWITCH  0_____     Sum of SDLC switched lines, X.25 switched
                        VCs, and TRSS logical connections.





===>                                                                    60
PF 1=HELP       4=TOP 5=BOT       7=BACK 8=NEXT 9=PRT 10=EXEC 11=STEP 12=MGR
```

Figure 9-4. CF3725 question 605



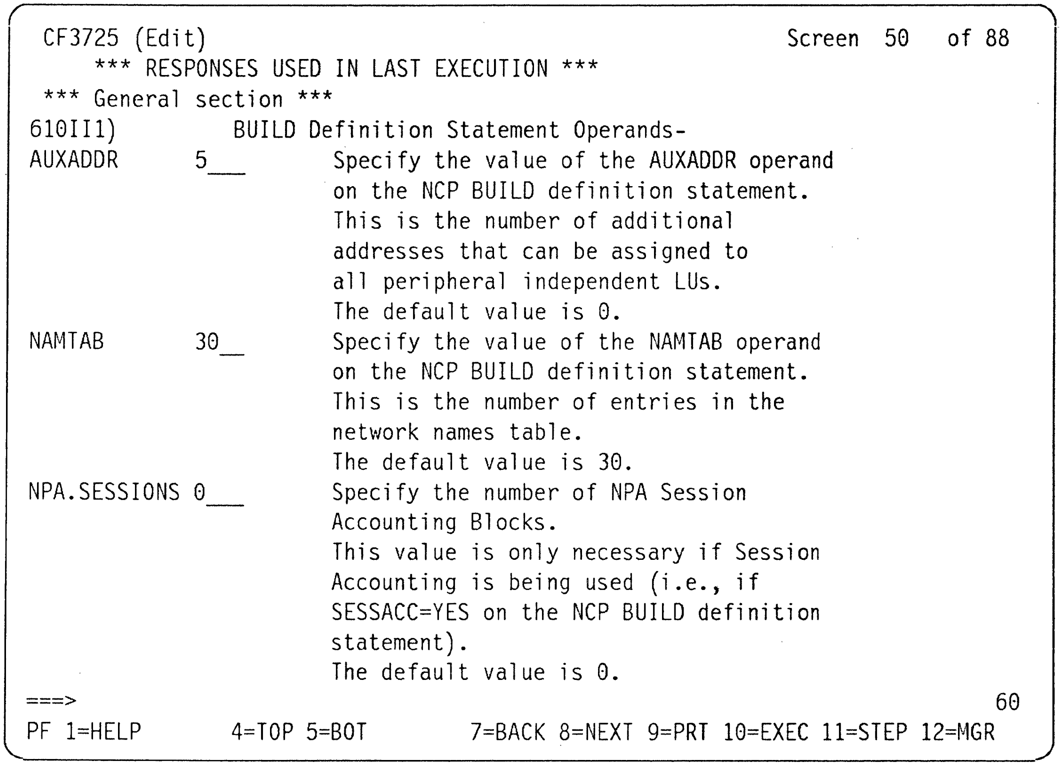
```
 CF3725 (Edit)                                          Screen  49   of 88

 *** General section ***
607II1)          Number of Sessions for Independent LUs-
ADDSESS      5___        Specify the value of the ADDSESS operand on
                         the NCP BUILD definition statement.
RESSCB       3___        Specify the value of all RESSCB operands
                         on the NCP LU definition statement.
NUMILU       0___        Specify the value of the NUMILU operand on
                         the NCP LUDRPOOL definition statement.
                         The sum of these three operands is the
                         total number of sessions for all
                         independent LUs, including such overhead
                         sessions as SNA Service Manager or
                         Control Point Service Manager.
                         The default for all three operands is 0.




===>                                                                    60
PF 1=HELP       4=TOP 5=BOT       7=BACK 8=NEXT 9=PRT 10=EXEC 11=STEP 12=MGR
```

Figure 9-5. CF3725 question 607



```
 CF3725 (Edit)                                            Screen  50   of 88
     *** RESPONSES USED IN LAST EXECUTION ***
 *** General section ***
610II1)         BUILD Definition Statement Operands-
AUXADDR       5___      Specify the value of the AUXADDR operand
                        on the NCP BUILD definition statement.
                        This is the number of additional
                        addresses that can be assigned to
                        all peripheral independent LUs.
                        The default value is 0.
NAMTAB        30__      Specify the value of the NAMTAB operand
                        on the NCP BUILD definition statement.
                        This is the number of entries in the
                        network names table.
                        The default value is 30.
NPA.SESSIONS 0___       Specify the number of NPA Session
                        Accounting Blocks.
                        This value is only necessary if Session
                        Accounting is being used (i.e., if
                        SESSACC=YES on the NCP BUILD definition
                        statement).
                        The default value is 0.
===>                                                                   60
PF 1=HELP       4=TOP 5=BOT       7=BACK 8=NEXT 9=PRT 10=EXEC 11=STEP 12=MGR
```

Figure 9-6. CF3725 question 610

9.10.2 NCP V4.3 LEN Storage

Table 9-12 demonstrates the relative storage increases when implementing the PU T2.1 function in NCP V4.3. Case BASE LEN shows that when RESSCB, ADDSESS, NUMILU, and AUXADDR are set to 0 (zero), with 5 independent LUs defined, more memory is still required for PU T2.1 versus a comparable configuration that implements only dependent LUs. Case LEN shows that when values are assigned to those NCP parameters, yet more storage is required.

Table 9-12. NCP V4.3 Storage Matrix. 3725 storage required for code and control blocks

NCP V4.3 Storage	**3725 storage required for LUs by CASE (in bytes)**		
Type of NCP storage	COMP	BASE LEN	LEN
NCP Program Module	445047	445047	445047
NCP Control Blocks	702037	702562	704300
Total	**1147084**	**1147609**	**1149347**
Increase from previous case	**0**	**525**	**1738**

9.10.3 NCP V5.2 LEN Storage

The next two tables (Table 9-13,Table 9-14) show the relative storage increases when implementing the PU T2.1 function in NCP V5.2 on the 3720 and the 3745. Case BASE LEN shows that when RESSCB, ADDSESS, NUMILU, and AUXADDR are set to 0 (zero), with 5 independent LUs defined, more memory is still required for PU T2.1 versus a comparable configuration that implements only dependent LUs. Case LEN shows that when values are assigned to those NCP parameters, yet more storage is required.

9.10.3.1 NCP V5.2 and the 3720

Storage increase for PU T2.1 in NCP V5.2 on the 3720.

Table 9-13. NCP V5.2 Storage Matrix for the 3720. 3720 storage required for code and control blocks

NCP V5.2 Storage	**3720 storage required for LUs by CASE (in bytes)**		
Type of NCP storage	COMP	BASE LEN	LEN
NCP Program Module	445047	445047	445047
NCP Control Blocks	702037	702562	704590
Total	**1147084**	**1147609**	**1149637**
Increase from previous case	**0**	**525**	**2028**

9.10.3.2 NCP V5.2 and the 3745

Storage increase for PU T2.1 in NCP V5.2 on the 3745.

Table 9-14. NCP V5.2 Storage Matrix for the 3745. 3745 storage required for code and control blocks

NCP V5.2 Storage	**3745 storage required for LUs by CASE (in bytes)**		
Type of NCP storage	COMP	BASE LEN	LEN
NCP Program Module	471969	471969	471969
NCP Control Blocks	853455	853980	855719
Total	**1325424**	**1325949**	**1327688**
Increase from previous case	**0**	**525**	**1739**

Again, keep in mind that this configuration is for a very limited use of NCP's PU T2.1 support. The storage requirement grows as the number of supported ILUs increases. It is imperative that the account SE configure a customer's 37XXs using CF3725/CF3745.

9.11 Sizing the Load Module

Table 9-15 provides a summary of the load module size for each version of NCP, from V4.2 to V5.2, required to support the same NCP configuration. NCP V5.1 and V5.2 were configured for the 3720. LEN is not represented in these load module sizes.

Table 9-15. Storage Comparison by NCP Version. Increase in 372X storage required by NCP V4.3, V5.1, and V5.2.

Total NCP storage	**Storage required for LUs (in bytes)**			
NCP Version	1000 LUs	1600 LUs	2000 LUs	3000 LUs
NCP Version 4.2	531289	647209	724489	917689
NCP Version 4.3	929595	1146315	1290795	1651995
NCP Version 5.1	893494	1110214	1254694	1615894
NCP Version 5.2	929595	1146315	1290795	1651995

NCP V4.3 and NCP V5.2 use more storage than earlier versions of NCP. Indeed, the increase in storage usage when migrating to NCP V4.3/V5.2 is probably greater than any other prior version or release of NCP. The customer can no longer rely on SSP NDF output to gauge the NCP load module size. It is the SE's responsibility to use CF3725 and/or CF3745 to determine the total storage required to execute NCP V4.3/V5.2 on the customer's 3720, 3725, or 3745.

Chapter 10. Type 2.1 Node Support

10.1 Overview -- Type 2.1 Node Support

Type 2.1 Node Support, a function available when both VTAM V3R2 and the appropriate NCP (V4R3 for the 3725 or V5R2 for the 3720/3745) are installed, allows peripheral nodes attached to the NCP to establish sessions with each other or with VTAM application programs. Type 2.1 Nodes, which are systems such as a Series/1 or AS/400, can establish multiple sessions with many different session partners or with the same session partner. Type 2.1 Node Support allows nodes to use the connectivity of a VTAM/NCP Subarea Network and fully exploit the facilities available with LU6.2. VTAM and NCP are participants in the IBM System Application Architecture. VTAM V3R2 in conjunction with the appropriate NCP adds support for Type 2.1 nodes as part of the Common Communications Support element of the IBM System Application Architecture.

The bulletin "A Technical Overview: VTAM Version 3 Release 2, NCP Version 4 Release 3, NCP Version 5 Release 2" (GG66-0283) contains an overview of Type 2.1 Node Support.

10.1.1 What is a Type 2.1 Node?

Starting in the late 1970s, IBM developed a number of systems that were capable of communicating peer-to-peer, without attaching to a host network running VTAM and NCP. Examples of early systems are the 8100, the 5520, and the Displaywriter. Each different product tended to use an implementation that was specific only to that product for peer-to-peer communication, and only a few of the products developed protocols that allowed unlike systems to communicate peer-to-peer.

In June of 1986, IBM announced an architecture called "SNA Low Entry Networking." The purpose of the architecture is to standardize upon a common set of protocols so that unlike systems (for example, a Personal Computer and a System/36) can communicate with each other. SNA Low Entry Networking defines path control network connectivity between **ADJACENT** Type 2.1 nodes. (SNA Low Entry Networking defines an architecture for communication between nodes adjacent to each other. The architecture does **not** define communication between non-adjacent nodes. In other words, SNA Low Entry Networking does not define communication through intermediate routing nodes.) Customers can use a variety of connection media such as the IBM Token-Ring Network, SDLC switched or leased lines, or X.25 networks to connect peer systems that are Type 2.1 nodes. Figure 10-1 on page 10-4 shows an example of some IBM systems that are Type 2.1 nodes.

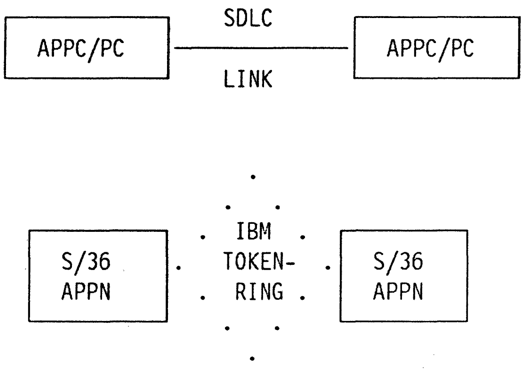


Figure 10-1. IBM Peer Systems--T2.1 Nodes

The PCs and S/36s shown in Figure 10-1 use the SNA Low Entry Networking architecture for communication between themselves. This architecture is explained in "SNA Formats and Protocols Reference Manual: Architecture Logic for Type 2.1 Nodes" (SC30-3422).

Low Entry Networking architecture is intended to support communication between Logical Units (LUs) Type 6.2 within the Type 2.1 Nodes. LU6.2, also known as Advanced Program-to-Program Communication (APPC), is an IBM standard architecture designed to allow communication between programs, or LUs, in different kinds of systems. The intent of LU6.2 is to have a common protocol between differing systems so that the programs running in these systems can communicate with each other without having to implement special support to accomodate product differences. In other words, a PC implementing LU6.2, an intermediate system implementing LU6.2 (such as an AS/400), a microcoded system (such as a 3820 printer) implementing LU6.2, and a VTAM application program in an MVS/370 system implementing LU6.2 should all be able to communicate.

Please refer to the following documents for more information on LU6.2:

- GG24-1584 An Introduction to APPC
- GC30-3084 SNA Transaction Programmer's Reference Manual for LU6.2
- SC30-3269 Format & Protocol Reference Manual: Architecture Logic for LU6.2

At the time of the writing of this bulletin, the following IBM systems implement Low Entry Networking Architecture and exploit the Type 2.1 Node Support provided by VTAM V3R2 and NCP V4R3/V5R2:

- AS/400
- S/36 using R5 or higher
- S/38 with MTR MT06007
- PC Support 400
- PC running APPC/PC V1.11 or OS/2 Extended Edition
- Series/1 running EDX V6 or RPS V7.1

- S/88
- RT/PC R2
- TPF V2R4

10.1.2 Type 2.1 Node Communication

A Format 3 XID is used by Type 2.1 nodes during link activation. The nodes exchange XID3s to negotiate protocols such as which node is the primary link station, whether or not an SSCP-to-PU session is requested, whether the nodes operate in full-duplex or half-duplex mode, how many frames can be received before acknowledgement, maximum frame size, etc..

LU6.2, also called APPC, is used for the communication between the logical units (the programs) in the Type 2.1 nodes. The logical units communicate using sessions. An LU6.2 is capable of having more than one simultaneous session with another LU6.2, called **parallel** sessions, and is also capable of having simultaneous sessions with many different LUs, called **multiple** sessions. Each session is started by the Primary Logical Unit (PLU) sending the Secondary Logical Unit (SLU) a BIND.

10.2 APPN--Advanced Peer-to-Peer Networking

Advanced Peer-to-Peer Networking is supported by two products, the S/36 and the AS/400. Each S/36 or AS/400 uses the Type 2.1 and LU6.2 architectures in conjunction with some additional functions. The additional functions allow S/36s and AS/400s that are Network Nodes to provide many of the functions that VTAM and NCP do--functions such as locating resources and acting as Intermediate Routing Nodes.

The example in Figure 10-2 uses the S/36. However, wherever a S/36 is shown, an AS/400 could also be used.

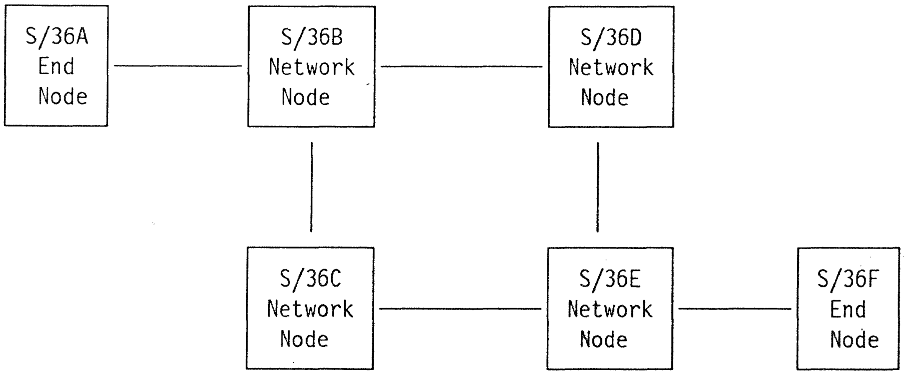


Figure 10-2. Advanced Peer-to-Peer Networking

Using Figure 10-2 as an example, each S/36 shown connects to the other S/36s using Low Entry Networking Architecture. The programs that are used to communicate Peer-to-Peer, such as Display Station Pass Through, Distributed Data Management, Personal Services/36, and user-written APPC programs use LU6.2 architecture. The Network Nodes, S/36s B, C, D, and E, use functions provided by

APPN to locate resources, select routes, and act as Intermediate Routing Nodes. The Network Node functions are not defined as part of Low Entry Networking Architecture; they are functions implemented by the S/36 as extensions to the architecture. A brief description of some APPN functions follows. For more information on APPN, please refer to:

- GG66-0216 SNA Networks of Small Systems
- SC21-9471 S/36 APPN Guide

If LUA, an LU in S/36A, an End Node, wants to communicate with LUF, an LU in S/36F, LUA sends a BIND naming the session partner, LUF, to the adjacent network node, S/36B. S/36B, using Network Node functions, doesn't necessarily have LUF defined, and sends a search request to its adjacent Network Nodes, S/36C and S/36D. S/36C and S/36D, likewise not necessarily having knowledge of LUF, can send search requests to their adjacent nodes. S/36E has LUF defined because S/36F is an End Node; S/36E, therefore, responds to the search request. S/36B, using APPN facilities, calculates the best route for the session between LUA and LUF, and the session is subsequently started over that route.

With the advent of the AS/400, APPN terminology changed somewhat so that non-Network Nodes are now known as either **LEN** Nodes or **END** Nodes.

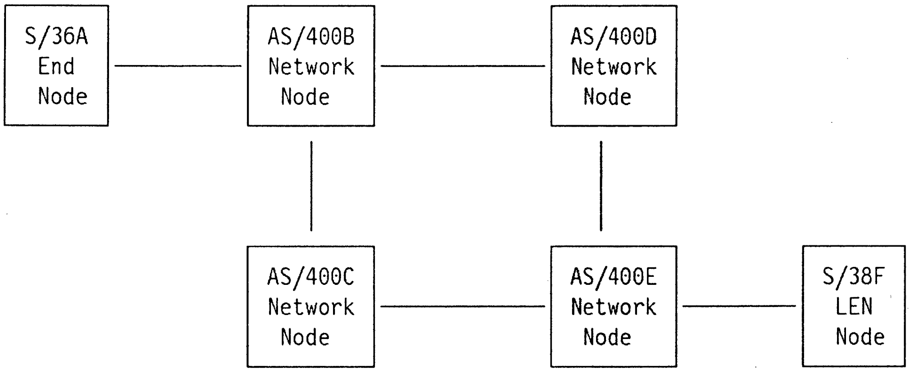


Figure 10-3. APPN -- LEN nodes and END nodes

Using Figure 10-3 as an example, there are two non-Network Nodes in the APPN network, S/36A, an **END** Node, and S/38F, a **LEN** Node. The difference between them is that the S/36 can return a search request and the S/38 cannot. This means that resources (LUs) located in S/36A **DO NOT** have to be defined in AS/400B. AS/400B can simply send a search request for an unknown resource to S/36A; S/36A would then positively respond if it contained that resource. However, since the S/38 cannot be sent a search request, all resources located in S/38F have to be defined in AS/400E as being located at the end of the link to S/38F.

An APPN network can be connected to a subarea network (an NCP). In this case, the subarea network has the appearance of a **LEN** Node to the APPN network.

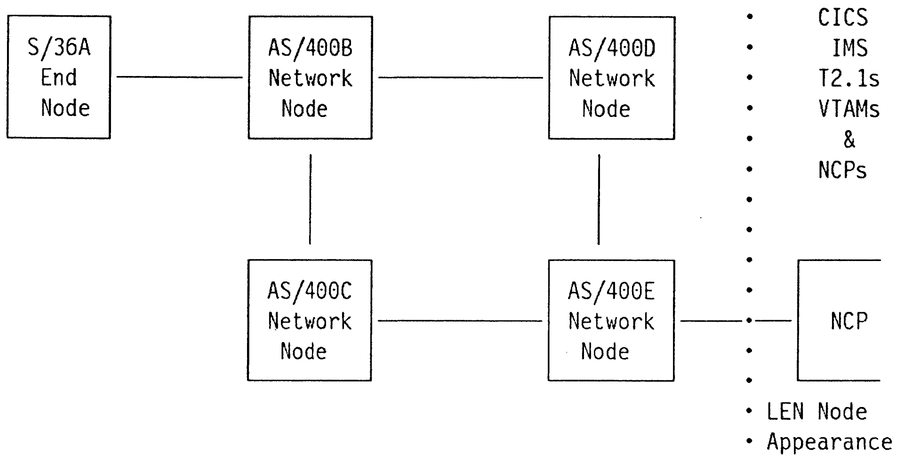


Figure 10-4. APPN network connected to subarea network

Using Figure 10-4 as an example, the APPN network is link-connected to an NCP. The NCP is part of a subarea network containing many VTAMs, NCPs, and to which other T2.1 nodes or APPN networks are attached. Since the subarea network has the appearance of a LEN node, all the resources to which the APPN network wishes to communicate (CICS, IMS, LUs in other T2.1 nodes or APPN networks, etc.), must be defined as if they were located right at the end of the link to NCP. The protocols used on the link between the APPN network and NCP are adjacent node protocols as defined by SNA Low Entry Networking architecture. The protocols used between the network nodes are network node protocols as defined by APPN. Some of the network node protocols (the search requests, for example) are also used between AS/400B and S/36A.

10.3 VTAM/NCP Support of Type 2.1 Nodes

With the enhancements contained in VTAM V3R2 and NCP V4R3/V5R2, systems that implement Low Entry Networking Architecture can attach to NCP; Logical Units within these Type 2.1 Nodes can have multiple sessions, can have sessions with other peripheral nodes in addition to VTAM application programs, and can be Primary Logical Units (initiate sessions by sending BIND).

Figure 10-5 on page 10-8 illustrates the additional function available with Type 2.1 Node Support. In the Figure, S/36A and S/36B are Type 2.1 nodes and 3174A is a PU Type 2. LUs on 3174A, because it is supported by VTAM and NCP as a PU Type 2, are capable of having only a single session, and the session partner **must be** a VTAM application program (in this example APPL1). An LU on a PU Type 2 is always a secondary logical unit and the VTAM application program is the primary logical unit (the application sends the BIND that initiates the session).

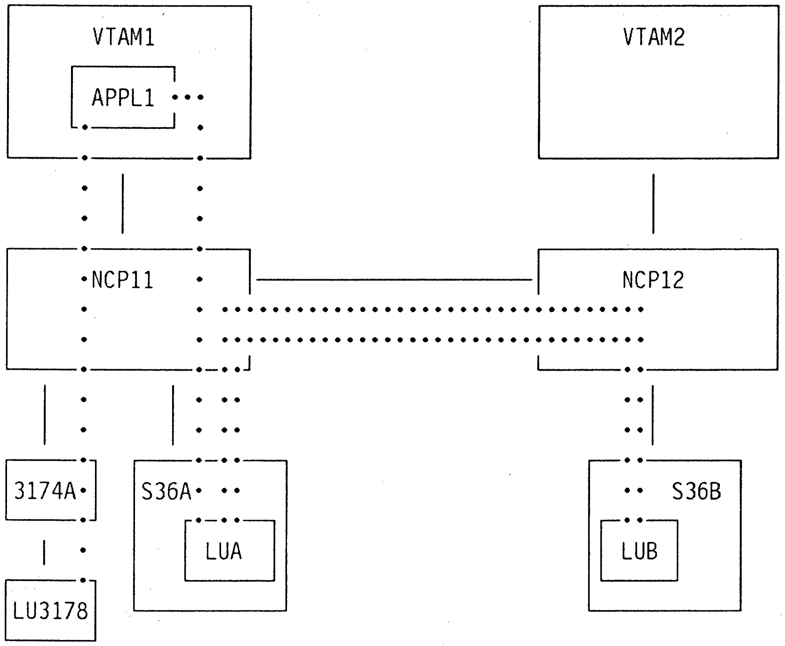


Figure 10-5. Example of Type 2.1 Node Support

In Figure 10-5, the two S/36s are Type 2.1 Nodes. LUA in S/36A has multiple sessions, one with APPL1 and two (parallel sessions) with LUB in S/36B. Either LUA or LUB can be the PLU (i.e. can send BIND, to initiate sessions). Type 2.1 nodes, in addition to having both PLU and SLU capabilities, can have session partners that are VTAM application programs or that are other peripheral LUs and can have more than one session simultaneously. The Logical Units within the Type 2.1 nodes that are capable of initiating sessions and having multiple sessions are known as **independent** LUs. The logical units contained in a device like the pictured 3174 are known as **dependent** LUs.

INDEPENDENT LUS:

- Are capable of initiating sessions as a PLU (sending BIND)
- Can have many different session partners simultaneously (multiple sessions)
- Can have many sessions with the same partner simultaneously (parallel sessions)
- Must be within a Type 2.1 Node (SSCP-to-PU session is optional)
- Have no SSCP-to-LU session

DEPENDENT LUS:

- Have an SSCP-to-LU session
- Are limited to a single session only
- Can only be a SLU
- Can be within a Type 2.1 Node, a PU Type 2, or a PU Type 1

Independent LUs are defined to VTAM and NCP as an LU with **LOCADDR = 0** under a PU Type 2 that has indicated **XID = YES.** A Type 2.1 Node can have multiple independent LUs. The NCP definition statements necessary to define the 3174 and S/36s in Figure 10-5 are shown in Figure 10-6 on page 10-9.

NCP11:

```
3174A  PU PUTYPE=2 (XID=YES optional)
LU3178 LU LOCADDR=2
          .
          .
S36A   PU PUTYPE=2,XID=YES
LUA    LU LOCADDR=0
```

NCP12:

```
S36B  PU PUTYPE=2,XID=YES
LUB   LU LOCADDR=0,RESSCB=2
```

Figure 10-6. NCP Definitions for T2.1 Nodes

Each session that an independent LU has requires a boundary session control block. The new parameter **RESSCB** can be coded to reserve a certain number of boundary session control blocks for exclusive use by this particular LU within the NCP. If boundary session control blocks are not reserved by RESSCB, they come from the pool defined by another new parameter on the NCP BUILD definition statement, ADDSESS (see the Section of this Chapter entitled "NCP Changes for Type 2.1 Node Support" for more information on ADDSESS). Please refer to the Chapter in this bulletin entitled "NCP Storage Usage" for more information about changes in NCP storage and control blocks.

A Type 2.1 Node can support **both** Independent and Dependent LUs. Recall that the Independent LU can initiate sessions, is capable of having multiple sessions and has no SSCP-to-LU session. A Type 2.1 Node that supports both Independent and Dependent LUs would be defined:

```
NODEA    PU PUTYPE=2,XID=YES
INDLUA   LU LOCADDR=0,(optional RESSCB=) ◄─────┐Independent LUs
INDLUB   LU LOCADDR=0,(optional RESSCB=) ◄─────┘
DEPLU    LU LOCADDR=1                    ◄─────┐Dependent LUs
DEPLUB   LU LOCADDR=2                    ◄─────┘
```

Figure 10-7. Type 2.1 Node Definition for Dependent and Independent LUs

10.3.1 T2.1 Node Attachment Restrictions

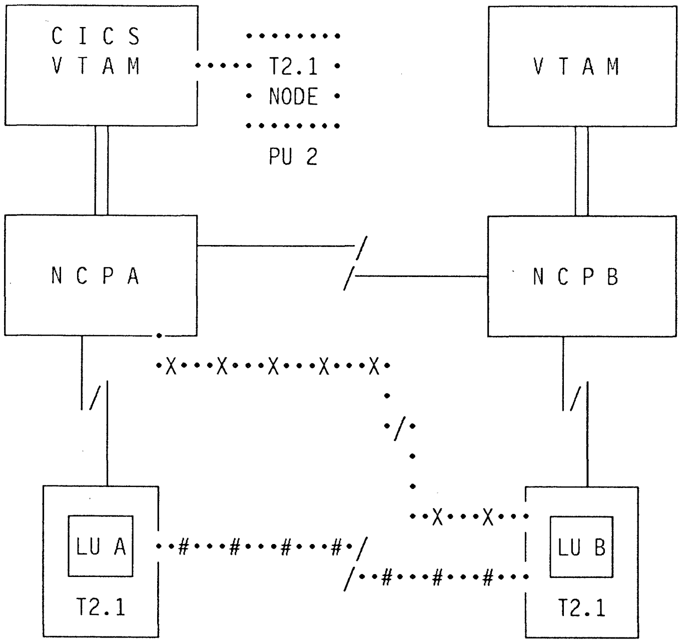


Figure 10-8. T2.1 Node Attachment Restrictions

Figure 10-8 points out T2.1 node attachment restrictions using VTAM V3R2 and NCP V4R3 or NCP V5.

A T2.1 node is limited to a single attachment to the subarea network. Since the T2.1 node containing LUB has a connection to NCPB in Figure 10-8, the connection (shown with Xs) to NCPA is invalid and should not be attempted. More than one connection from a T2.1 node to a subarea network could result in duplicate resource problems. The first problem encountered would be that LUB would have to be defined in both NCPA and NCPB, causing a duplicate LU name in the subarea network. The second problem would be that the subarea network presents the appearance of an **ADJACENT** T2.1 node, a **LEN node**--therefore, all resources to which LUB wishes to communicate need to be defined in LUB's node as at the end of the link to the subarea network. If LUB wanted to communicate with CICS and were attached to both NCPA and NCPB, it would be impossible to define CICS in LUB's node as a resource at the end of both links.

T2.1 nodes should not be connected to each other when these nodes are also attached to the subarea network. The connection between the T2.1 nodes (shown with #s) would be invalid. Having both the subarea connections along with direct connections between the T2.1 nodes causes unpredictable results and is not recommended. Again, duplicate resource definitions would present a problem. If LUB wished to communicate with LUA, LUA could not be defined in LUB's node as a resource at the end of both the link to NCPB and the link to LUA's node.

Only the NCP (V4R3 & V5) has boundary support for T2.1 nodes; any T2.1 nodes connected directly to the VTAM V3R2 boundary are supported as T2.0 nodes only, as shown in Figure 10-8 on page 10-10.

NOTE: With VTAM V3R3, T2.1 nodes can be supported when directly attached to VTAM. For example, an AS/400 which is link or token-ring connected to a 9370 running VTAM V3R3, can be supported as a T2.1 node with independent logical units, multiple sessions, etc..

10.3.2 VTAM/NCP Contact Flow for Type 2.1 Nodes

Figure 10-6 on page 10-9 illustrates that the Type 2.1 nodes, S/36A and S/36B, are defined as PUTYPE=2, the same as the 3174. The difference is that the operand XID=YES **must** be coded for Type 2.1 Nodes. Coding XID=YES results in NCP sending the peripheral node an XID prior to sending SNRM (Start Normal Response Mode). If the peripheral node responds with an XID3 and the XID negotiation is successful, then VTAM and NCP support the peripheral node as a Type 2.1. If the node responds with anything other than an XID3, the node will be treated as a PU Type 2.0 (dependent LUs only, SSCP-to-PU session, SSCP-to-LU sessions, etc.).

XID=YES can be coded for non-Type 2.1 Nodes, although there is no reason to. Most IBM Type 2.0 devices accept an XID but do not respond with an XID3. IBM devices which **CANNOT** accept an XID (XID=NO must be coded) are the 3271-11, 3271-12, 3275-11, 3275-12, 3614,3624,3710, and the 3791.

The default for the XID keyword is YES if using SSP V3R3; the default is NO if using SSP V3R4 or higher.

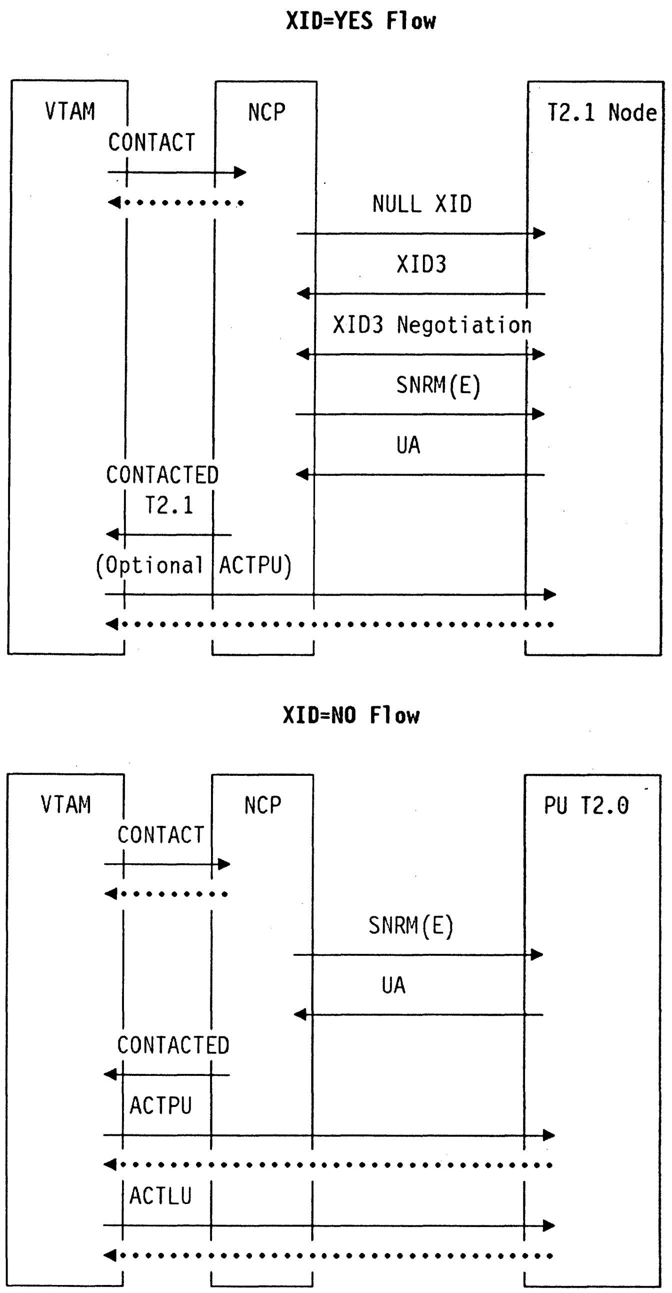


Figure 10-9. NCP V4R3/V5R2 Peripheral Node Contact Flow

Figure 10-9 shows the effect of coding XID=YES. If XID=YES is coded, the request/response units shown on the top flow between VTAM, NCP and the peripheral node. This flow **must** occur and the peripheral node **must** respond with an XID3 in order for VTAM and NCP to treat it as a Type 2.1 node. If XID=NO is coded or defaulted, then the flow on the the bottom will occur. This flow is the same as has existed with levels of VTAM prior to V3R2 and levels of NCP prior to V4R3 or V5R2.

It is not necessary to code the following parameters, because values are exchanged between NCP and the peripheral node during the XID3 negotiation:

- MAXDATA
- MAXOUT
- MODULO
- DATMODE

MAXDATA--The MAXDATA (largest PIU that NCP will send to the node) that NCP will use is based on what the peripheral node sends in on the XID. If a MAXDATA value is coded on the PU definition for the T2.1 node, it is ignored.

MAXOUT--The MAXOUT (number of PIUs that NCP will send to the node before requiring a response) that NCP will use is based on what the peripheral node sends in on the XID.

MODULO--The MODULO that NCP will use will be based on what is sent in by the peripheral node for MAXOUT on the XID. If the NCP is defined as MODULO 128 and the peripheral node specifies MAXOUT < =7 in the XID, the lower of the two values will be used.

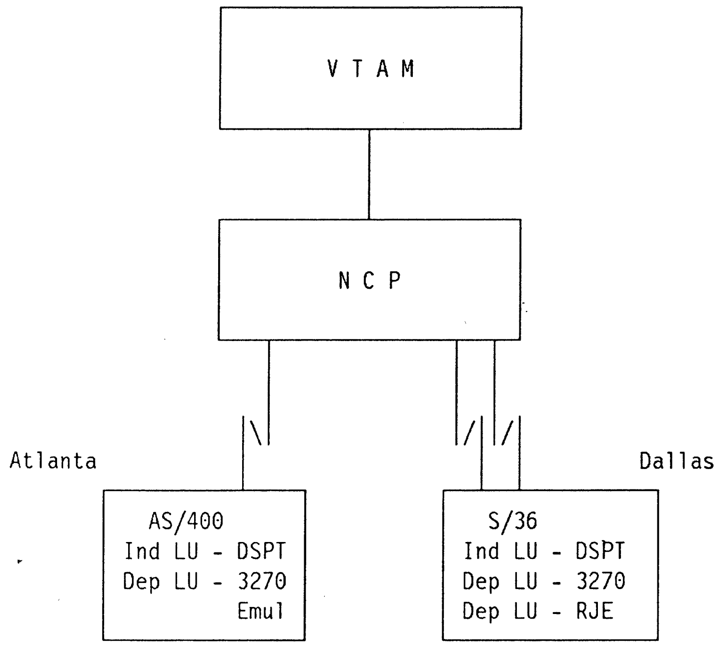
DATMODE--In order to use DATMODE=FULL, the NCP must define DATMODE=FULL, and the XID must indicate DATMODE=FULL. Otherwise, DATMODE=HALF is used.

Other information exchanged between NCP and the peripheral node using the XID3 is whether or not the node should be sent an ACTPU and whether or not the nodes support BIND segmentation and BIND pacing.

The importance of whether an ACTPU should be sent to establish an SSCP-PU session can be appreciated with respect to ALERT processing. Since ALERTS flow on the SSCP-PU session, if a T2.1 node does not request an ACTPU be sent, then no ALERTS will be forwarded from this node to the NetView focal point.

Currently, the SYS/36 does not allow both dependent and independent traffic to flow to one PU, and does not request the ACTPU in its XID3 exchange with the host. For the SYS/36 to forward ALERTs to the host, another PU definition would be required, defined for support of dependent LU traffic. The AS/400 does allow both independent and dependent traffic to flow to the same PU, and it does request an ACTPU in the XID3 exchange. The PC and PS2, defined as T2.1 nodes also request an ACTPU.

10.3.3 T2.1 (AS/400 & SYS/36) Node NCP Definition



**Need 2 Line-Adapters

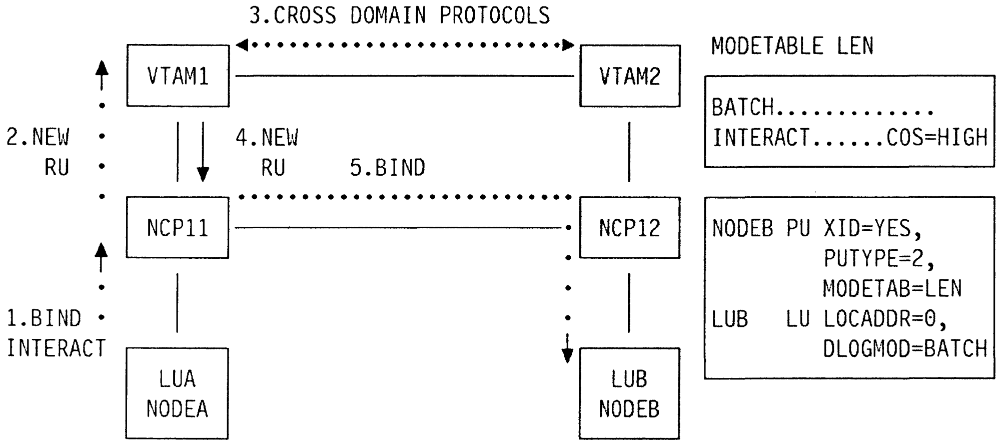
NCP DEFINITIONS

```
ATL400  PU PUTYPE=2,XID=YES          DALS36  PU PUTYPE=2,XID=YES
ATLLU62 LU LOCADDR=0                 DALLU62 LU LOCADDR=0
ATL3270 LU LOCADDR=2                 DALS36A PU PUTYPE=2,XID=NO
                                     DAL3270 LU LOCADDR=2
                                     DALRJE  LU LOCADDR=3
```

Figure 10-10. AS/400 and SYS/36 Coding Differences

Figure 10-10 depicts the differences between the AS/400 and the SYS/36 with respect to the definitions required to allow both of these T2.1 nodes to forward ALERTs to the host. Notice that the AS/400 defines one PU which includes both Independent LU and Dependent LU statements. The SYS/36 is required to separate the LUs between the PU definitions, one PU for Independent LU statements and another for the Dependent LU statements. The SYS/36 requires two different ports for the two PUs, if both are used simultaneously. Although two links are shown in Figure 10-10 to the SYS/36, this could also be a multipoint configuration.

10.3.4 Type 2.1 Node LU-to-LU Session Initiation Flow



```
MODETABLE LEN

BATCH............
INTERACT......COS=HIGH

NODEB PU XID=YES,
         PUTYPE=2,
         MODETAB=LEN
LUB   LU LOCADDR=0,
         DLOGMOD=BATCH
```

Figure 10-11. Type 2.1 Node Session Initiation

Figure 10-11 illustrates how a session is initiated by an Independent LU in a Type 2.1 node. The Independent LU, in the example LUA, sends a BIND naming its intended session partner, LUB. The Type 2.1 Node, NODEA, is using the same protocols between itself and NCP as it would if it were connected to NODEB. NODEA, therefore, views the NCP it is attached to as another Type 2.1 node, and views any intended partner LUs as if they were contained in that Type 2.1 Node.

The BIND includes a Mode name and, among its control vectors, a control vector for Class of Service name. A control vector containing a Fully Qualified Procedure Correlation ID (PCID) will also be included if the session is being initiated with an Extended BIND. To date, only the AS/400 and SYS/36 have the capability as peripheral devices to start sessions using an Extended BIND.

The fully qualified PCID is the network name followed by the control point name, followed by the PCID, a unique identifier generated by the peripheral node to uniquely identify this particular session.

In order for the session setup to work, both LUA and LUB must be **ACTIVE** to their owning VTAMs. Although no SSCP-to-LU sessions exist, and SSCP-to-PU sessions may or may not exist, the PUs and LUs representing the Type 2.1 Nodes and the Independent LUs must be active to VTAM. In Figure 10-11, assume that NODEA and LUA are owned by VTAM1 and that NODEB and LUB are owned by VTAM2.

Once NCP receives a BIND from a T2.1 peripheral node, it discards the COS control vector and forwards the BIND with the remaining control vectors to the owning VTAM. The Mode name in the BIND is used to locate a corresponding LOGMODE entryname in the LOGMODE table of the SLU. If a corresponding entryname is not found, the session is rejected.

In Figure 10-11, at label "1.BIND," the BIND from LUA is received by NCP11 and packaged into a new RU and forwarded to VTAM1 at label "2.NEW RU." VTAM1 extracts the destination LU name and the MODE name from the BIND and sends them, at label "3," using normal cross-domain protocols (CDINIT, CDCINIT) to locate the SSCP owner of the destination session partner. Once found, the MODETABLE in the SLU definition is searched for a LOGMODE entry that corresponds to the name carried in the BIND. In Figure 10-11, the MODE name carried in the BIND is 'INTERACT'. The default LOGMODE entry (DLOGMOD) specified in the NCP SLU definition is 'BATCH' which would be used if no MODE entry had been specified. In our case, the LOGMODE TABLE defined with a name of 'LEN' must have an entry named 'INTERACT' or the session setup will fail. Our example shows that an entry of 'INTERACT' does exist in the LOGMODE TABLE named 'LEN'. The RUSIZEs, PACING values and COS name from the 'INTERACT' entry are returned at label "3," to VTAM1 from VTAM2. VTAM1 then returns the BIND (including the additional information received from VTAM2) back to NCP11, at label "4.NEW RU" in a new RU. The new RU includes a ROUTING control vector built from the COS entry named 'HIGH'. The entry 'HIGH' **must** exist in the COS table in the PLU's domain. NCP11 rebuilds the BIND using information from the 'INTERACT' LOGMODE entry and forwards it to NCP12, shown at label "5.BIND." The COS entry 'HIGH' provides the routing information necessary to activate a virtual route for the session.

NOTE: In this example, NCP is activating a route between Subarea 11 and Subarea 12. Prior to VTAM V3R2/NCP V4R3-V5R2, sessions between peripheral nodes do not exist and NCPs do not necessarily contain PATH statements indicating other NCPs as the Destination Subarea. NCP PATH statements may need to be modified in order to support a Type 2.1 Node environment. Additionally, unless NCPs are used with SNI, NRF, or XI, they usually are generated with only Explicit Route definitions. When using T2.1 node support, NCPs activate virtual routes for sessions, if not already active, and need to be generated with Virtual Route, as well as Explicit Route, definitions.

As the positive response to the BIND is returned by LUB in NODEB through the subarea network to LUA in NODEA, new RUs are sent from the NCPs to the owning VTAMs informing them that a session has been established between the T2.1 nodes.

Although our example is of a cross domain environment, the two T2.1 nodes could also reside in the same domain or could be located in separate networks using the facilities of SNI for session setup.

Please refer to the NCP Reference manual for NCP V4R3 or NCP V5R2 for more specific information on session initiation and termination in Type 2.1 Node environments.

10.4 NCP Changes for Type 2.1 Node Support

There are new parameters on the NCP **BUILD** definition statement that are used to indicate certain values for Type 2.1 Node support. They are:

- ADDSESS
- MAXSESS
- AUXADDR
- NAMTAB
- SESSACC

ADDSESS indicates the number of LU-to-LU boundary session control blocks to be defined in the NCP that are available for use by any independent LU. If RESSCB values have been supplied on LU statements, the RESSCB values are added to the value of ADDSESS to define the total number of boundary session control blocks in the NCP.

AUXADDR indicates the number of additional addresses that can be assigned to all peripheral LUs. An LU needs an address for each session that it has with the **same session partner** (parallel session) for which it is the PLU.

MAXSESS indicates the maximum number of LU-to-LU sessions that any LU can have.

NAMTAB indicates the number of entries in the network names table. Each network requires an entry, as does each SSCP (VTAM that owns this NCP), as does each Type 2.1 node which is either attached to this NCP or in session with resources attached to this NCP.

SESSACC indicates whether or not session accounting information should be collected for sessions involving Type 2.1 nodes. SESSACC can also be specified for nodes other than Type 2.1. Threshold values can be specified. Session accounting information, such as PIU counts and byte counts, can be collected by NCP V4R3/V5R2 and forwarded to NPM R3. **NPA=YES** must also be specified in order to have the session accounting function.

Other new or changed parameters are:

1. PU
   XID=YES
2. LU
   LOCADDR=0
   RESSCB=
   PACING=
3. LUDRPOOL
   NUMILU=

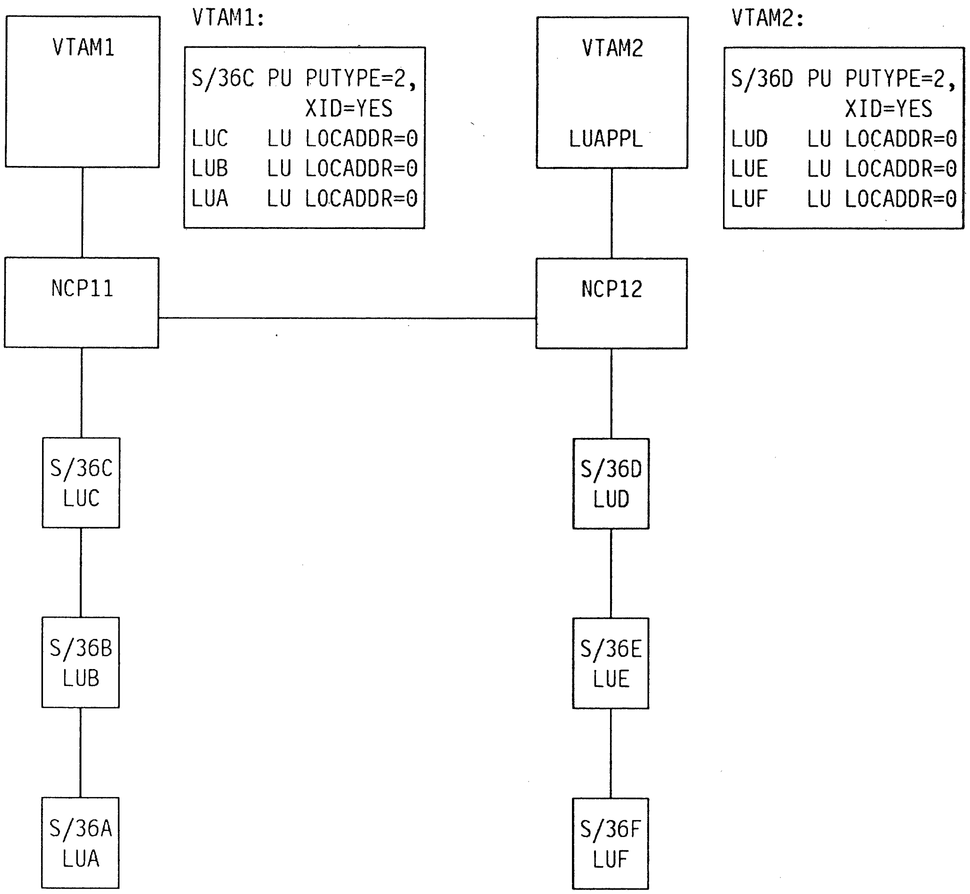
The new parameter, **XID=YES**, must be coded on the PU statement for Type 2.1 Nodes. A value of 0 must be coded for **LOCADDR** on the LU statement for Independent LUs. **RESSCB**, on the LU statement, specifies the number of boundary session control blocks reserved for this LU. RESSCB is an optional parameter.

The **PACING** operand has changed. Only the first value, the number of PIUs sent before receiving a Pace response, can be specified. The second value, specifying in which PIU the Pace request should be turned on, is not valid with NCP V4R3 and NCP V5. The Pace request is turned on in the first PIU.

The LUDRPOOL statement contains a new operand, **NUMILU**, which must be coded if Independent LUs are to be dynamically added to the NCP, or if Independent LUs on switched lines are to be used.

10.5 Using APPN with VTAM/NCP Type 2.1 Node Support

Although VTAM V3R2 and NCP V4R3/V5R2 are only aware of peripheral nodes directly attached to NCP, it is possible connect a number of Type 2.1 nodes to a S/36 (or AS/400) attached to NCP and use the functions of APPN in addition to Type 2.1 Node Support for peer-to-peer communication.



VTAM1:

```
S/36C PU PUTYPE=2,
         XID=YES
LUC   LU LOCADDR=0
LUB   LU LOCADDR=0
LUA   LU LOCADDR=0
```

VTAM2:

```
S/36D PU PUTYPE=2,
         XID=YES
LUD   LU LOCADDR=0
LUE   LU LOCADDR=0
LUF   LU LOCADDR=0
```

Figure 10-12. Using APPN and Type 2.1 Node Support

In Figure 10-12, there are two APPN networks attached to a VTAM/NCP subarea network. The first APPN network consists of S/36A, S/36B, and S/36C. S/36B and S/36C are Network Nodes and S/36A is an End Node. S/36C views the subarea network as an End Node that contains LUD, LUE, LUF, and LUAPPL. S/36C has defined these LUs as though they reside in the adjacent End Node (the attachment to NCP11).

The second APPN network consists of S/36D, S/36E, and S/36F. S/36D and S/36E are Network Nodes and S/36F is an End Node. S/36D views the subarea network as an End Node that contains LUA, LUB, LUC, and LUAPPL. **NOTE: The S/36s attached to the subarea network do not need to define every LU in the subarea network. They only need to define the LUs that they, or another S/36 within that APPN network, need to communicate with.**

VTAM and NCP view an APPN network as if it is an adjacent Type 2.1 Node containing several LUs. Note that three independent LUs have been defined to VTAM1 under a single PU representing the Type 2.1 Nodes; the same is true for VTAM2.

In Figure 10-12 on page 10-18, if LUA wanted to communicate with LUF, the following would occur:

1. LUA would send a BIND naming LUF to its adjacent node, S/36B.
2. S/36B, a Network Node, using APPN functions would send a search request for LUF to its adjacent Network Node, S/36C.
3. S/36C would have defined LUF as residing in its adjacent End Node, the Subarea Network, and respond positively to S/36B for the request for LUF.
4. S/36B would calculate the route and forward the BIND to S/36C.
5. S/36C would forward the BIND to NCP11.
6. NCP11 would notify the owning VTAM of LUA that LUA had sent a session initiation request for LUF.
7. VTAM1 would use normal cross-domain protocols to locate LUF at VTAM2.
8. NCP11 would activate the route for the session and send the BIND on that route to NCP12.
9. NCP12 would send the BIND to its adjacent Type 2.1 Node, S/36D.
10. S/36D would send a search request for LUF to its adjacent Network Node, S/36E.
11. S/36E would respond positively and S/36D would calculate the route.
12. The BIND would be forwarded to S/36F.

NOTE: if a certain Class of Service, "FAST" for example, were to be used for the session between S/36A and S/36F in the above example, definitions would need to be coordinated between the subarea network and the APPN networks. As explained in Section 10.3.4, the COS used in the subarea network, and also forwarded to the destination APPN network, is the COS name defined on the LOGMODE entry of the SLU (or the default). If the origin APPN network sent in "FAST" as the COS name on the BIND, FAST would not be used unless also defined on the LOGMODE entry for LUF at VTAM2.

10.6 Session Accounting & NPM Enhancements

NCP V4R3 and V5R2 have been enhanced in conjunction with NPM R3. One enhancement is the ability to have multiple NCP to NPM sessions so that in the case the primary session fails, there is a hot backup session already active. Other enhancements are the ability to collect accounting data such as PIU counts and byte counts for Independent LU sessions, and the ability to use NPM with Dynamically Reconfigured resources.

10.7 Migration Considerations

Type 2.1 Nodes and Independent LUs can only be supported when both the VTAM that owns them and the NCP to which they are attached are VTAM V3R2 and NCP V4R3 or NCP V5R2. If either VTAM or NCP is at a previous level, the Type 2.1 Nodes are supported as PU Type 2s.

If Independent Logical Units will be establishing sessions with Logical Units in other peripheral nodes, NCP PATH statements should be checked to ensure that there are routes between the NCPs. Since NCP may be activating routes for Independent LU sessions, VRs as well as ERs need to be specified on the PATH statements.

See the Software Compatibility Considerations Chapter of this manual for information on compatibility PTFs.

10.8 Problem Determination

A new VTAM operator command, DISPLAY SESSIONS, is provided with VTAM V3R2 that identifies sessions and session partners. See the Chapter in this manual entitled "Miscellaneous" for a description of the DISPLAY SESSIONS command. Each session has a unique Session Identifier (SID) which can be used with the VTAM operator command, VARY TERM, to terminate the session.

The Generalized PIU trace (GPT) and Trace Analysis Program (TAP) are enhanced to trace all PIU flow for LUs so that PIU flow for sessions and conversations is available.

Netview's Session Awareness (SAW) data for LU-to-LU sessions when both partners are in Type 2.1 nodes is available. The Extended Bind used by Independent LUs to establish sessions is also available in Netview.

Chapter 11. VTAM/NCP Performance and Other Enhancements

There are significant performance improvements with VTAM V3R2 and some performance enhancements with NCP V4R3/V5R2. This chapter discusses performance improvements as well as other new functions that have not been discussed in earlier chapters of this manual.

11.1 VTAM V3R2 Performance Enhancements

The session services component of VTAM V3R2 was rewritten providing shorter path lengths and more efficient buffering techniques. This has resulted in performance benefits over prior releases in the functions of session initiation, termination and failure recovery.

Performance test runs by CPD indicate VTAM V3R2 may provide significantly improved performance compared to VTAM V3R1.1. Reductions in CPU time for CMC host activation were in the range of 40% to 50% for networks involving between 10,000 and 30,000 LUs. Elapsed time (wall clock) reductions ranged between 20% and 30% for the same number of LUs. Data host performance improvements were similar to the CMC results.

Another positive accomplishment was the reduction of the 'large network effect'. This effect occurred in the activation of larger networks due to storage constraints and resulted in a non-linear activation curve. VTAM V3R2 has almost eliminated the 'large network effect', consequently there is no longer a need to throttle session services requests using the ITLIM parameter defined in the VTAM Start Options. The ITLIM parameter is not valid with VTAM V3R2 and will be ignored if coded.

The following figures indicate the VTAM V3R2 performance gains achieved. These tests were run in an unconstrained, no bottleneck lab environment on a 3090-200E processor. Using 3725 controllers, 30,000 LUs were activated with VTAM V3R1.1 and 60,000 LUs were activated with VTAM V3R2. Although these were separate test runs, the results were combined on a single graph for easier interpretation. Customers may not experience the same results due to differences in environments.

VTAM V3R2 uses more virtual storage than VTAM V3R1.1; in the range of up to 20% after the network has been brought up. The differences between the two releases in virtual storage depends upon the ITLIM value used in VTAM V3R1.1 and how the network was brought up.

Customers should refer to the Storage Estimates manual (SC30-3403) FOR ESTIMATING VTAM V3R2's use of storage in their particular environment.

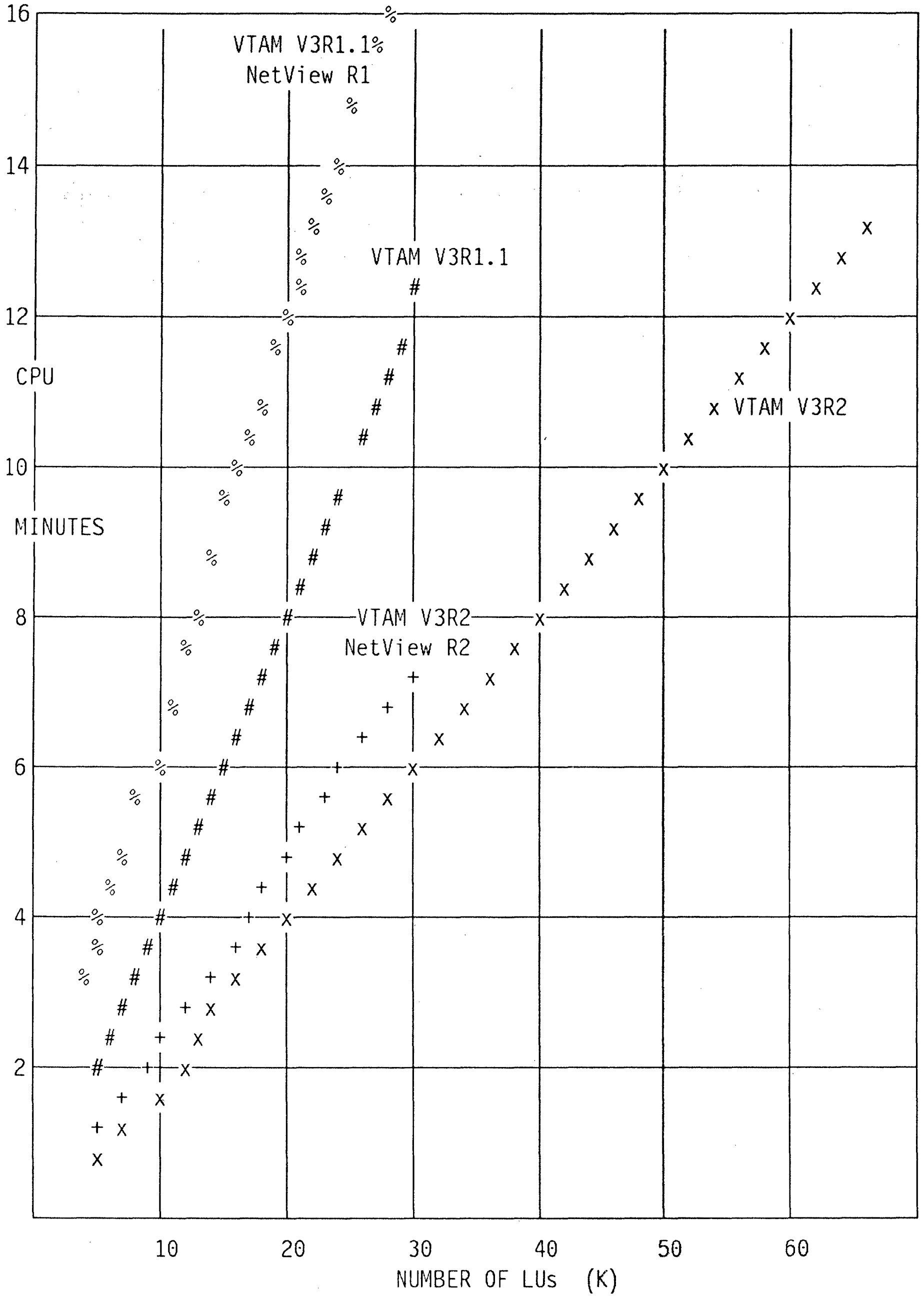
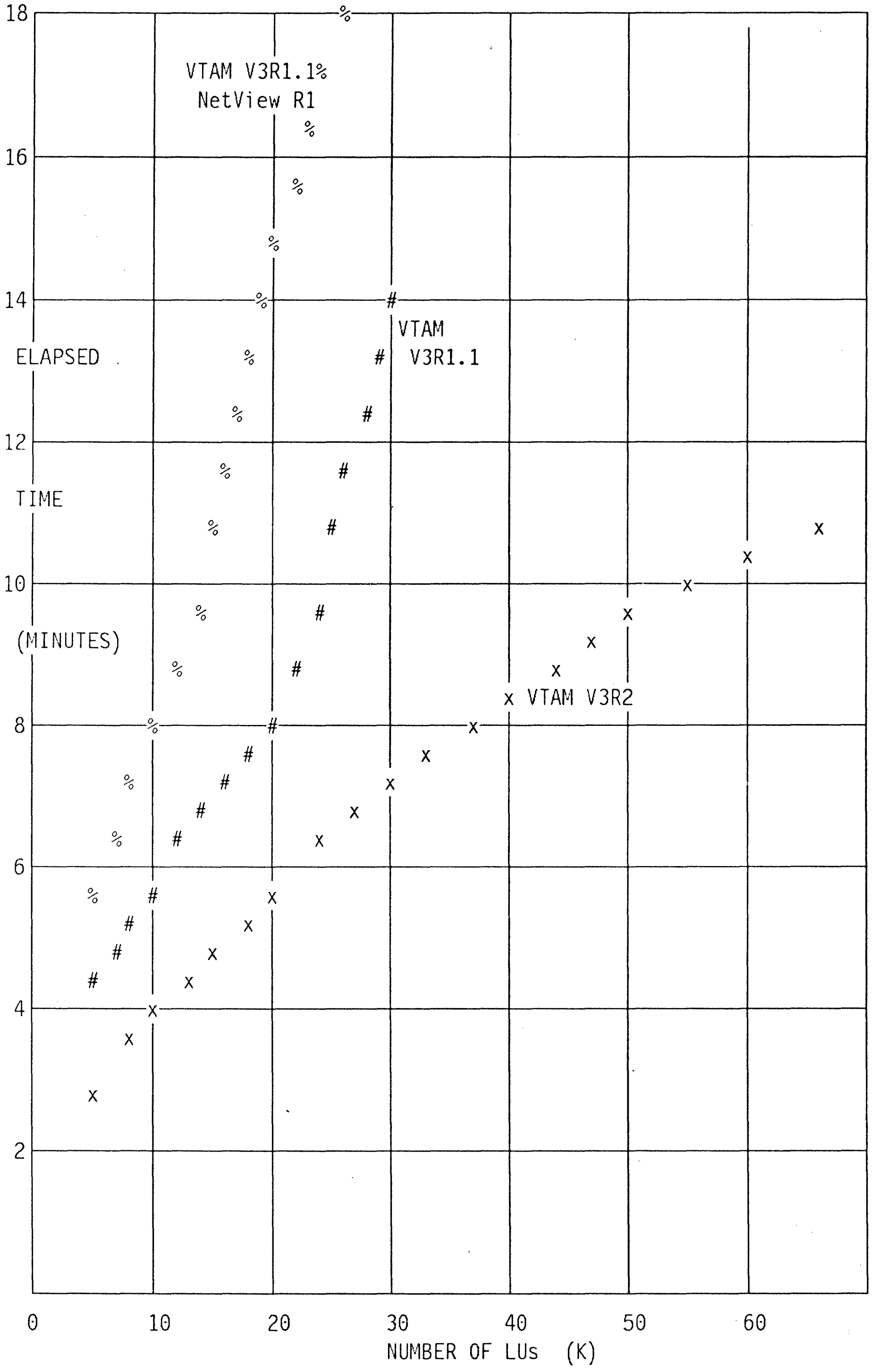


Figure 11-1. Host Processor CPU minutes for Activation

In Figure 11-1, the horizontal axis depicts the number of LUs being activated, while the vertical axis is the amount of CPU time (in minutes) consumed. Symbols are used to differentiate between the tests. A 'X' symbol is used to identify VTAM V3R2; a '#' symbol identifies VTAM V3R1.1. Notice that to complete activation of 20,000 LUs with VTAM V3R1.1 required 8 CPU minutes, while the same number of LUs are activated in 4 minutes with VTAM V3R2. This is a substantial performance improvement. Depending on your outlook, it can be stated that with VTAM V3R2 the time required to activate this network will be cut in half; or, the size of the network could double to 40,000 LUs while CPU time necessary for activation would only equal the time VTAM V3R1.1 required for activation of 20,000 LUs.

The other two lines on the graph, using symbols '+' and '%' display the same activation as discussed previously but with the addition of NetView collecting SAW and PIU data for all resources (no filtering). The test with VTAM V3R1.1 (%) included NetView R1 while the VTAM V3R2 (+) test included NetView R2. Performance enhancements in NetView R2 are evident in this comparison. Using the same 20,000 LUs, NetView R1 had completed collection on all resources at the 12 CPU minute point, 4 minutes after VTAM V3R1.1 had completed activation of the resources. With NetView R2, the collection was completed in approximately 5 CPU minutes, only 1 minute after VTAM V3R2 completed the activation.



VTAM V3R2 with Netview R2 almost the same as VTAM V3R2

Figure 11-2. Host Processor Elapsed time for Activation

Figure 11-2 is a graph of the same activation as Figure 11-1 on page 11-4 but the vertical axis represents elapsed (wall clock) time. Elapsed time to activate 20,000 LUs in VTAM V3R1.1 (#) was 8 minutes while in VTAM V3R2 (X) it was approximately 5 1/2 minutes, a performance gain of 30%. This figure graphically shows the 'large network effect' in the VTAM V3R1.1 curve beginning at the 20,000 LU point while the curve for VTAM V3R2 is more linear.

Notice that a line representing NetView R2 is not indicated; this is due to the fact that it would align exactly over the VTAM V3R2 (X) line, mainly a result of executing on a multi-engine processor.

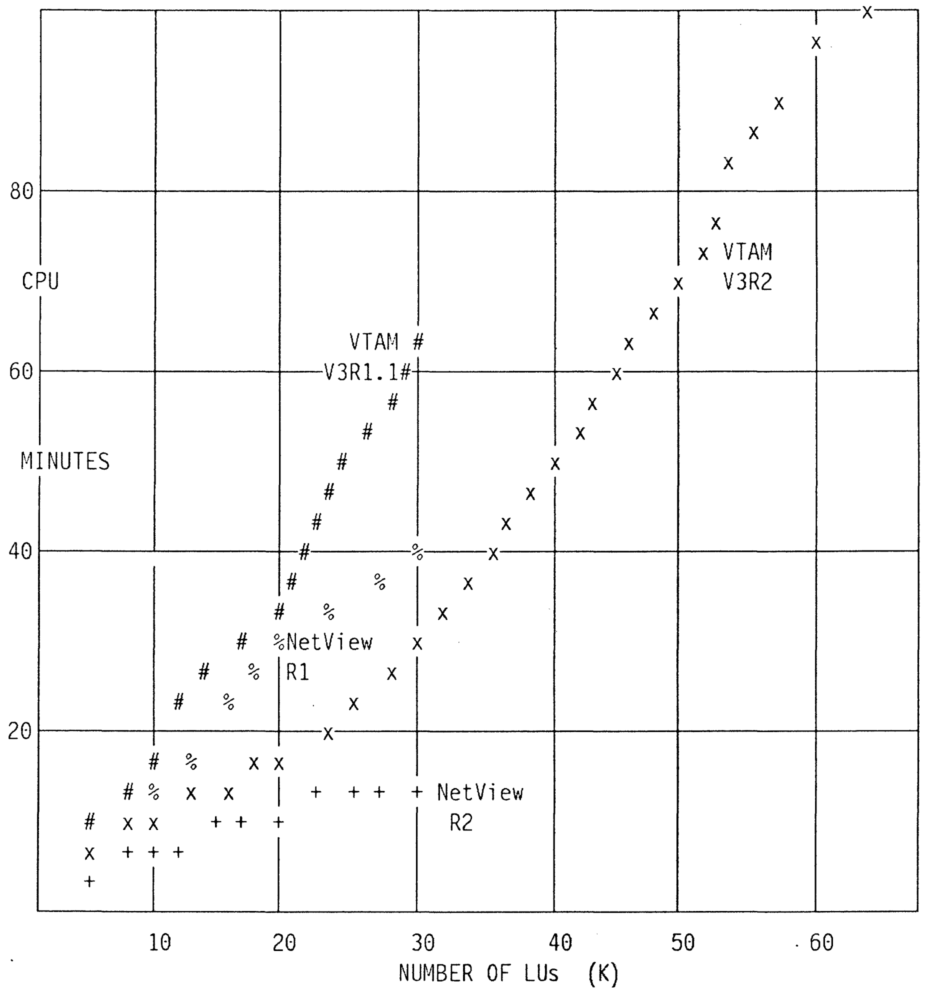


Figure 11-3. Host Processor CPU Minutes for Logon

Figure 11-3 displays CPU minutes required to establish LU-LU (LOGON) sessions with a data host. The times for the VTAM and NetView tests were made from separate runs, in contrast to the previous figures where the runs were combined in one benchmark.

Using the 20,000 LU point, it required approximately 30 CPU minutes to complete the LU-LU session activation for the LUs in VTAM V3R1.1 (#) while with VTAM V3R2 (X), LOGON was completed in approximately 15 minutes; a 50% reduction in setup time with VTAM V3R2.

Pointing out the performance benefit of NetView R2 over R1, notice that the SAW and PIU collection of the 20,000 LUs required 35 CPU minutes with NetView R1 (%), only 10 CPU minutes with NetView R2 (+).

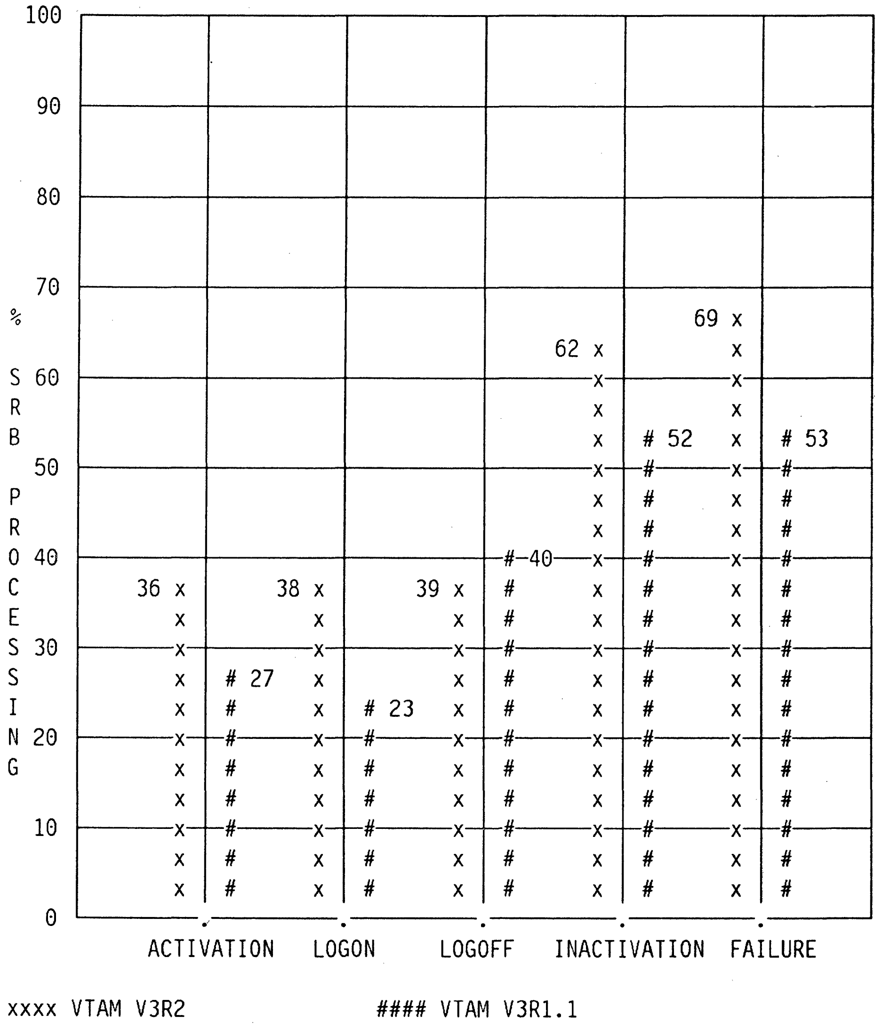


Figure 11-4. Host Processor Percentage of SRB Processing

When executing in a multi-engine processor environment, the greater the amount of TCB/SRB overlap, the better the performance. VTAM V3R2 has improved its ability to perform in a multi-engine environment by adding more function into SRB type processing.

Figure 11-4 depicts the percentage of SRB processing (vertical axis) for each function listed along the horizontal axis. In all but one function, the amount of SRB processing has been increased in VTAM V3R2 (X) over VTAM V3R1.1 (#). This allows more function to be scheduled on an idle engine concurrent with VTAM TCB processing on a separate engine, providing the benefit of reduced elapsed time to perform the functions.

References:

- GG66-0285 Network Management Performance in a CMC Environment

- SC30-3489 Netview/VTAM Benchmark Study Results

The information plotted on these figures was taken from two bulletins. 'Network Management Performance in a CMC Environment' (GG66-0285) has results from the VTAM V3R1.1 and NetView R1 tests. 'NetView/VTAM Benchmark Study Results' (SC30-3489) has results from the VTAM V3R2 and NetView R2 tests.

11.2 31-bit I/O Buffers

VTAM V3R2 for MVS/XA allocates I/O buffers and related control blocks from the Extended Common Storage Area, addressable by a 31-bit address. Whenever possible, VTAM uses the 31-bit I/O system interface. The few exceptions are when VTAM uses the system EXCP function which still only supports 24-bit addressing.

A new internal storage pool (APBUF) is used for control blocks that are used when interfacing with the system EXCP function. APBUF is allocated in CSA. VTAM buffer pool start options include the new APBUF pool as does the resulting messages of the VTAM DISPLAY BFRUSE command.

11.3 Change in VTAM's use of Private Subpools

11.3.1 Loading User modules and tables

In VTAM V3R2, user written modules and tables, which include USS tables and Logon Mode tables are loaded into LSQA in VTAM Private. Previously they were loaded into "low Private." The modules may be loaded above or below the 16M line depending upon how they are link-edited. If AMODE/RMODE = 31, they will be loaded above the line.

11.3.2 Acquiring working storage

11.3.2.1 VTAM V3R1.1

Prior to V3R2 the design of the storage management component of VTAM was to use three subpools for private area GETMAIN calls: SP229, SP230, and SP250. The bulk of the calls are for SP250. The current design of virtual storage management (VSM) in MVS/XA uses a single DQE chain to track allocated virtual storage for a given private area subpool. As the network grows in size, more storage is be allocated in the private area, which causes the length of the DQE chain to increase. A serial search algorithm is currently used to process this chain, which means that the time spent in GETMAIN/FREEMAIN increases with the size of the network.

Under normal, steady-state network conditions, this does not result in noticeable problems. However, if a network failure occurs, VTAM enters a recovery sequence, which requires many GETMAIN requests for private area storage, in a short amount of time. A large network system has DQE chains that are already long. The additional requests compound this part of the problem, and the number of requests and the short time-frame in which they are issued leads to very significant amounts of CPU time being used by GETMAIN/FREEMAIN processing. The end result is performance degradation, which can become quite severe in extreme cases.

11.3.2.2 VTAM V3R2

To solve this problem, VTAM V3R2 Storage Management has been changed so as to take better advantage of the structure of storage management in the operating system.

VTAM V3R2 no longer GETMAINs private storage solely from subpools 229, 230, and 250. Instead, the subpool for each request is determined from the size of the request, with SP001 being used for the smallest sizes, through SP127 for the largest. This drastically shortens the length of the DQE chain for a given subpool, which reduces GETMAIN/FREEMAIN processing time in like proportion.

There are APARs for MVS/XA VTAM V3.1.1 and V3.1.0 that provide the same function. These are:

OY10575/UY17446 for V3.1.0 (8803)

OY08077/UY15616 for V3.1.1(8801)

11.4 VTAM COATTAILING:

Coattailing is the function by which VTAM attempts to send multiple PIUs with a single Start I/O, as opposed to performing a Start I/O for each PIU. VTAM does this by delaying the Start I/O when it has a PIU to write, in order to wait for other PIUs which VTAM may be able to write with a single Start I/O.

Prior to VTAM V3R2, the user can specify an interval that VTAM is to delay before writing for Channel-to-Channel attachments only. The interval can be specified between 0 and 9.999 seconds. All other channel attachments are statically set for an interval of .2 seconds.

A performance enhancement in VTAM V3R2 provides the ability to specify a delay interval for channel attached NCPs and local SNA attached devices as well. The coattailing enhancement is available with the following levels of VTAM:

- VTAM V3R2 for MVS/XA with FMID JVT3215
- VTAM V3R2 for MVS/370 with FMID JVT3214
- VTAM V3R2 for VM at level 5664280D
- VTAM V3R2 for VSE at level G70

VTAM now performs PIU collection for the user specified interval or until 75% of the number of PIUs sent in the previous interval has been reached. Virtual Route pacing responses and TP2 traffic still result in an immediate write channel program being issued as was the case in prior releases of VTAM. The DELAY= parameter is the means of specifying the interval with the default set at .2 seconds. It may be specified in the PCCU statement, in the GROUP statement of the Channel Attached Major Node VBUILD definition for LNCTL=NCP, and in the Local SNA Major Node VBUILD definition for TYPE=LOCAL.

Please refer to LD35-0270, "VTAM Enhancements" for definition details.

11.5 NCP Performance Improvements

Several enhancements have been made in the NCP boundary support functions starting with NCP V4R3 and NCP V5R2 aimed at improving performance.

11.5.1 3174 Group Poll Support

Group Polling is an NCP facility designed to enhance the performance of a 3174 which is acting as a Token Ring gateway. The 3174 microcode contains the complimentary group polling support.

In a token ring gateway environment, the 3174 and the stations on the ring present the image of a multipoint SDLC line to NCP. Each station on the ring is associated with a specific station address (polling address).

Prior to these 3174 and NCP polling enhancements, data flowing from the stations on the ring arrived asynchronously at the 3174 but was queued at the control unit until a poll containing the station's specific device address arrived. The data would then be forwarded to NCP. This wait time at the 3174 had the potential for increasing the response time.

NCP's Group Poll support together with the corresponding 3174 support will decrease this wait time by allowing data from any station, whose address matches the Group Poll address, to flow inbound in response to a Group Poll. This reduction in queueing time at the controller has the potential for enhancing the performance of stations operating in a 3174 token ring gateway environment.

The Group Poll function is invoked by coding a new parameter on the PU statement:

GP3174=(group poll address|NO)

Where the "group poll address" above corresponds to a group poll value set in the 3174. The default, NO, implies that the group poll function is not to be used.

Each of the ring stations still has a unique address associated it as before. This address is defined to NCP with the **ADDR=** parameter for the corresponding PU. NCP still uses this address for some of its scheduling operations to the device.

The 3174 must be upgraded to Configuration Support B Release 1 and customized for the Group Poll function before this support can be generated in NCP.

This NCP enhancement is described in the cover letter for PTF UR90157 and the following related APARs.

IR83735	SSP V3R4.1 MVS
IR83776	SSP V3R4.1 VSE
IR83778	SSP V3R4.1 VM
IR83751	NCP V5R2.1 MVS/VSE/VM
IR83826	NCP V4R3.1 MVS/VSE/VM

At the publication date of this document the NCP SRLs have not been updated to reflect this function.

11.5.2 Reduced Path Length for Segmented PIUs

There have been significant path length reductions in the coding associated with the PIU segmentation portion of NCP. This will result in more efficient processing for V4R3 and V5R2 NCPs than for previous NCP releases with comparable traffic passing through the NCP boundary function.

11.5.3 Enhanced NTRI Performance

NTRI (NCP Token-Ring Interface) performance has been enhanced by redesigning some of the functions that ran at NCP level 5, the lowest dispatching priority, such that they now run at interrupt level 3. This is the same level at which much of the NCP scanner and timer code currently operates. This should enhance the NTRI performance over that for previous NCP levels. This performance enhancement is only applicable with a minimum NCP level of NCP V4R3.1 and NCP V5R2.1.

11.5.4 Class of Service for Peripheral Nodes (Boundary COS)

Previous NCP releases provide only limited user facilities for controlling the scheduling priorities of boundary (peripheral) node sessions.

BATCH=YES/NO

Prior to NCP V4R3 and NCP V5R2 the only user parameter that exists for influencing the relative priority of LU session traffic on boundary links is the **BATCH=** keyword.

This parameter groups all LUs into two categories, high priority and low priority. All NCP outbound boundary node traffic is subject to this prioritization scheme. If BATCH=YES is specified, the LU is placed in the low priority category. BATCH=NO, the default, places the LU in the high priority category. The LUs in each category compete with each other for outbound traffic scheduling. Service is provided to the high priority LUs first.

If a high priority LU has data queued to it at the same instant that a low priority LU has queued data, then the high priority LU's data are processed first.

The BATCH=YES/NO function is completely removed from NCP V4R3 and V5R2 and a new function **COS Extension** is added for NCP V5R2/V4R3. This new facility is discussed below.

COS EXTENSION

This enhancement extends the transmission priority scheme associated with virtual routes to boundary nodes in much the manner as earlier NCP releases exploited priorities on NCP subarea links. NCP schedules the outbound LU traffic based on which of the three transmission priorities (tp0, tp1, tp2) is associated with the session's virtual route.

Starting with NCP V4R3 and NCP V5R2 this function is invoked by specifying a new keyword, **LSPRI** , on the NCP GROUP statement.

The format of this new parameter is:

LSPRI = NO|YES|PU|LINK where:

LSPRI = NO This option allows NCP to schedule the link traffic based on the order in which PIUs arrive in the NCP and no consideration is given to the transmission priority.

LSPRI = YES This option implies that user line control is being used and that it supports segmented PIUs. Note that this parameter is only valid if user line control or NTRI is being used.

LSPRI = PU. This option forces NCP to schedule the outbound traffic for each PU in accordance with the priority of the virtual route used by the session. The traffic for all LUs on a given PU is scheduled based on transmission priority. Traffic for all LUs with TP0 is scheduled first. That for TP1 is scheduled next and that for TP2 is scheduled last. Outbound traffic is prioritized on a PU-wide basis. For example, if a PU has a high priority LU and a medium priority LU, and NCP has output data to be send to both LUs, the high priority LU's data will be sent out on the line first.

LSPRI = LINK This option is only available in NCP V4R3.1 and V5R2.1. It allows all of the LU sessions using a given **line** to compete for the line based on their session priority. Notice that with LSPRI = PU LUs only competed for service with other LUs associated with the same PU. The LINK option allows the user to insure that all LUs using a given line can contend for the line based on their relative priorities.

This function is implemented using output service queues representing the three transmission priorities instead of the service order table for output processing.

In order to use this LINK option, HDXSP = YES must also be specified for the link.

This COS extension function, while not a direct replacement for BATCH = YES, provides an effective means of prioritizing outbound LU traffic.

11.6 Session Awareness Data Filter Overview

VTAM V3R2 with the Enhancements provides a new function, the Session Awareness Data Filter in VTAM. The Session Awareness Data Filter in VTAM is available in the following levels of VTAM:

- VTAM V3R2 for MVS/XA with FMID JVT3215
- VTAM V3R2 for MVS/370 with FMID JVT3214
- VTAM V3R2 for VM at level 5664280D
- VTAM V3R2 for VSE at level G70

Prior to this function in VTAM, VTAM passed all Session Awareness Data over to Netview, as shown in Figure 11-5. The figure illustrates that when sessions are established or terminated, VTAM passes information about the sessions to Netview over an LU-LU session between Netview and VTAM. With Netview, a user-coded table, called a filter table, can be used to determine which session awareness data should be saved or not saved by Netview, which data should be recorded to DASD, and other parameters. Figure 11-5 shows that the Filter Table in Netview specifies SAW=YES to indicate that session awareness data should be kept and specifies SAW=NO to indicate that session awareness data should be discarded.

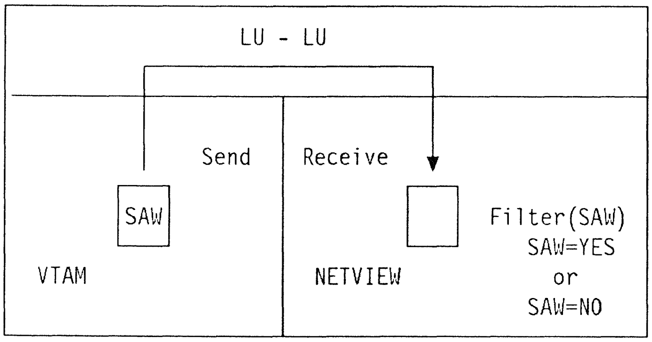


Figure 11-5. SAW Data Filter prior to VTAM V3R2

SAW=YES **must** be specified in order for Netview to have the data necessary to display things like session status and configuration, to do session tracing, to collect RTM data, and to record session data to SMF.

Figure 11-6 on page 11-15 shows that with the VTAM Enhancements, a filter table can be used to allow VTAM to selectively pass Session Awareness Data to Netview. System performance can be improved by eliminating the overhead of passing the session awareness data to Netview which had been thrown away in the past. Netview can perform additional filtering and can also obtain information from VTAM concerning the number of sessions for which session awareness data was not passed. Netview Release 3's command SESSIMON and SMF records contain information concerning the number of sessions filtered.

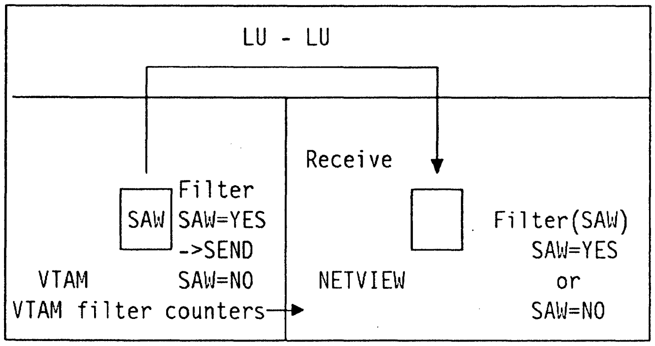


Figure 11-6. SAW Data Filter in VTAM V3R2

A Session Awareness Data Filter table is shipped with the VTAM V3R2 Enhancements called ISTMGC10, shown in Figure 11-7. The table as shipped specifies that no data should be filtered, that is, all session awareness data should be passed over to Netview.

```
ISTMCG10  KEEPMEM  START
DOSAW     KCLASS   SAW=YES
          MAPSESS  KCLASS=DOSAW,PRI=*,SEC=*
          KEEPMEM  STOP
```

Figure 11-7. SAW Data Filter Table, ISTMGC10

If the user wishes to filter session awareness data before passing it to Netview, they can use one of three methods:

1. Code, assemble and replace ISTMCG10 in SYS1.VTAMLIB. At the next VTAM startup, ISTMCG10 will be loaded into storage and used for filtering.

2. Code, assemble and replace ISTMCG10 in SYS1.VTAMLIB. A "refreshed" copy of the filter table can be loaded into storage through use of VTAM's Dynamic Table Replacement command in the form:

   ```
   F procname,TABLE,TYPE=FILTER,OPTION=LOAD,NEWTAB=ISTMCG10
   ```

3. Code, assemble and place in SYS1.VTAMLIB a filter table using a different name. The new table is loaded into storage and used by VTAM when the Dynamic Table Replacement command is issued in the form:

   ```
   F procname,TABLE,TYPE=FILTER,OPTION=LOAD,NEWTAB=newname
   ```

NOTE: TYPE = FILTER is mandatory on the modify table command with OPTION = LOAD for the filter table.

Whether or not data should be filtered is determined at session setup time and lasts while the session exists. Therefore, if awareness data were not filtered at session setup time and the VTAM filter table were changed to specify filtering during the life of the session, awareness data for that session should continue to be passed to Netview.

If the user wishes to take an existing Netview table and use it in VTAM, they can take the table as coded and add the KEEPMEM START and KEEPMEM STOP statements at the beginning and end of the table. VTAM ignores extranneous operands such as KEEPPIU, DASD, ER, VR, and TP which are meaningful to Netview only.

Figure 11-8 shows a simple example of a SAW Filter Table from Netview, modified for use with VTAM by preceding the table with the statement

```
label  KEEPMEM START
```

and following the table with the statement:

```
  KEEPMEM STOP
```

```
FILTER1  KEEPMEM  START
SAWYES   KCLASS   SAW=YES
SAWNO    KCLASS   SAW=NO
         MAPSESS  PRI=HOST*,SEC=HOST*,KCLASS=SAWYES
         MAPSESS  PRI=*,SEC=*,KCLASS=SAWNO
         KEEPMEM  STOP
```

Figure 11-8. Example of use of existing NetView SAW Filter with VTAM

The table must then be assembled, linkedited, and placed into VTAMLIB (using the name FILTER1 in our example). The table can then be subsequently loaded into storage for use by VTAM using the MODIFY TABLE command in the form:

```
MODIFY CSS1VTAM,TABLE,TYPE=FILTER,OPTION=LOAD,NEWTAB=FILTER1
```

Please refer to LD35-0270, "VTAM Enhancements" for definition and operation details.

NOTE: Plus signs (+) are valid continuation characters in NetView. They are NOT valid continuation characters to the Assembler. When we tried to assemble the sample SAW Filter table that comes with NetView R3, we received assembler errors because the sample table uses plus signs for continuation characters. We remedied the situation by placing all parameters for one statement on one line or by following proper assembler rules of indicating continuation by the placement of a character in column 72.

11.7 USERVAR

VTAM V3.1 introduced USERVAR (User Variable) which is used to map a generic name specified in a terminal logon to a specific application, based on the value of the variable. USERVARs are used by the Extended Recovery Facility (XRF) to direct user logons to the currently active IMS or CICS subsystem. They can also be used for application switching, for instance to facilitate either migration from one application release to another or for load balancing.

Prior to the VTAM V3R2 enhancements, USERVAR implementation involved definitions in the Interpret Table, and managing procedures, either manual (operator), automated (Netview CLISTs) or VTAM application programs (IMS or CICS for XRF) to propagate USERVAR values across the various domains in the network. This could involve a rather complex process.

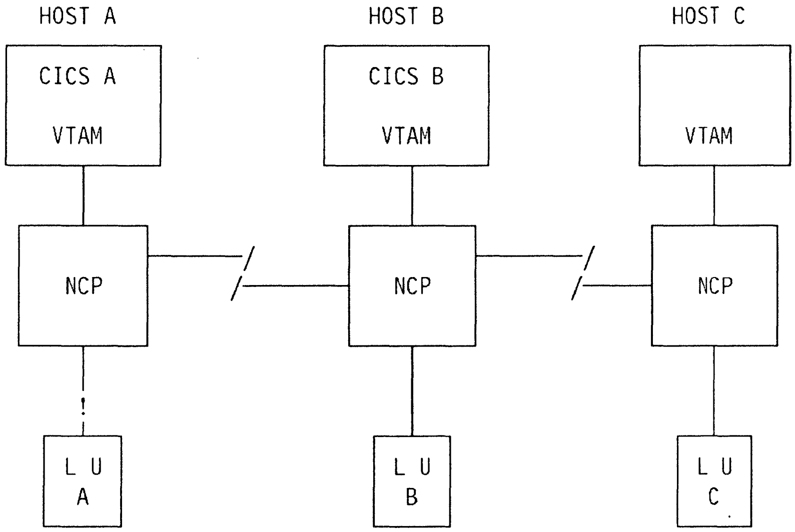
The VTAM V3R2 USERVAR enhancements, available as PTFs, eliminate the need to code Interpret Table definitions or to maintain CLISTs. VTAM itself can now communicate USERVAR values across domains in the network. The enhanced USERVAR support is available via the PTFs associated with the following APAR numbers:

• VTAM for MVS/XA	OY16021
• VTAM for MVS/370	OY16022
• VTAM for VM including 9370	VM33000
• VTAM for VSE	DY37697

The USERVAR enhancements are **not** documented in standard VTAM publications until the V3R3 level is available. The PTF cover letter or WSC Flash 8903.9 should be used for documentation prior to the receipt of the V3R3 VTAM publications.

11.7.1 Prior to Enhancement

The following example shows basic USERVAR implementation before the VTAM V3R2 enhancements:



```
User Definitions:
 HOST A - Interpret Table LOGCHAR APPLID(USERVAR,UVCICS),SEQNCE='CICS'
 HOST B - Interpret Table LOGCHAR APPLID(USERVAR,UVCICS),SEQNCE='CICS'
 HOST C - Interpret Table LOGCHAR APPLID(USERVAR,UVCICS),SEQNCE='CICS'
User Commands:
 HOST A - F NET,USERVAR,ID=UVCICS,VALUE=CICSA
 HOST B - F NET,USERVAR,ID=UVCICS,VALUE=CICSA
 HOST C - F NET,USERVAR,ID=UVCICS,VALUE=CICSA
```

Figure 11-9. Uservar Definition prior to VTAM V3R2

In Figure 11-9 we are assuming that all terminals owned by the three hosts will use USERVAR processing to logon to the active CICS application. An Interpret table macroinstruction must be defined in each host's VTAM. The macroinstruction definition for this example would be:

```
LOGCHAR APPLID(USERVAR,UVCICS),SEQNCE='CICS'
```

If a session is initiated via an unformatted logon or a formatted INIT-SELF from a terminal specifying 'CICS' as the application, the Interpret table entry will direct session setup processing to the USERVAR named 'UVCICS' to obtain the real application name. The USERVAR command would be issued at each host either through operator command, CLIST or by an application. The command for our example would be:

```
F NET,USERVAR,ID=UVCICS,VALUE=CICSA
```

Any terminal logging on to 'CICS' or LOGON APPLID(CICS) will have its session setup completed to CICSA.

If, due to a problem associated with CICSA, or for migration to a different CICS, or for load balancing, it becomes necessary to have 'CICSB' become the active application; a modify USERVAR command would have to be issued at each host, modifying the value of the current USERVAR from CICSA to CICSB.

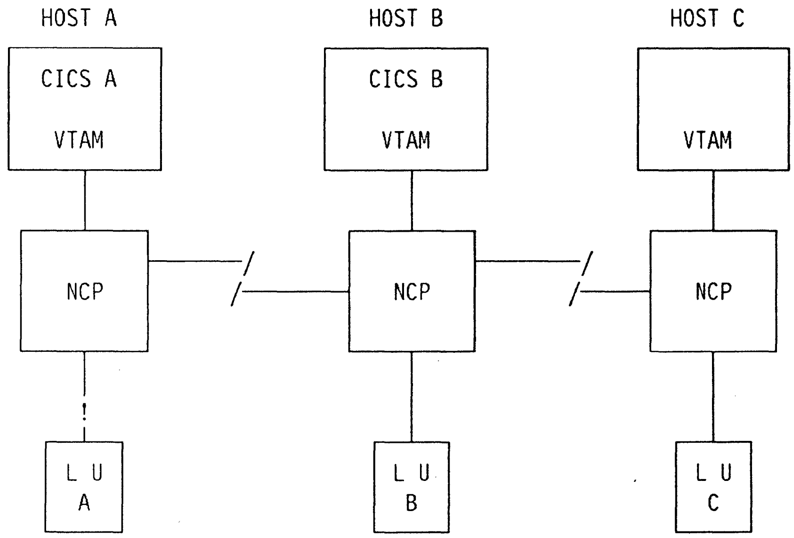
```
F NET,USERVAR,ID=UVCICS,VALUE=CICSB
```

All subsequent logons to 'CICS' would now be directed to CICSB on HOST B for session setup.

Typically, customers provide their own procedures, manual or automated (Netview CLISTs), to propagate USERVAR changes across the various domains in the network. This often involves a complex process in order to accommodate conditions such as systems being unavailable at the time a USERVAR changes or subsequent system restarts.

11.7.2 VTAM V3R2 Enhancement

The following example shows basic USERVAR implementation after the VTAM V3R2 enhancements:



```
User Definitions:
 None
User Commands:
 HOST A - F NET,USERVAR,ID=CICS,VALUE=CICSA
```

Figure 11-10. Uservar definition with VTAM V3R2

With the enhancement, VTAM V3R2 supports two classes of USERVAR. The first class, a user managed USERVAR is set by the MODIFY USERVAR command as explained above. VTAM will not attempt to alter the value of a USERVAR set in this manner, allowing existing CLISTs and procedures for propagating USERVARs across domains to operate as before the enhancements.

The second class, an automatic VTAM managed USERVAR, is created dynamically by VTAM as a copy of a user managed USERVAR in another domain. In Figure 11-10, the active CICS application is CICSA on HOST A. A user managed USERVAR would be set at HOST A either manually (operator command) or by the application:

F NET,USERVAR,ID = CICS,VALUE = CICSA

If LU C issued a logon request for CICS, HOST C would issue a CDINIT request (for APPL CICS) to the other HOSTS. HOST A would respond to the CDINIT with a CDTERM carrying a sense of 0888000F, indicating that the name, CICS, is really a USERVAR, and would provide the real application name, CICSA. HOST C now automatically creates the VTAM managed USERVAR locally, and re-issues the CDINIT with the real application name. For subsequent logons to CICS from HOST C, VTAM uses information in the automatically created USERVAR to translate requests immediately.

If the application (CICSA) fails and a backup takes over in an XRF environment, the new backup application (CICSB) issues the user managed USERVAR command at HOST B indicating it is now the currently active application. VTAMs in other domains, notified that XRF sessions have been switched, delete their automatic USERVARs, requiring the next logon to CICS to go through cross domain search as explained earlier, creating a new automatic USERVAR indicating CICSB as the application to be searched for in response to a logon to CICS. In a non-XRF environment, the automatic USERVARs will also be automatically deleted as part of cleanup process when the application fails. Session setup will then proceed as explained previously-- with the backup application host returning a CDTERM response to the CDINIT, a new automatic USERVAR created and re-issue of the CDINIT using the backup application name (CICSB) from the USERVAR value.

A new operand on the modify USERVAR command may be used to determine if subsequent attempts to logon using the same USERVAR result in a repeat of the cross domain search. The TYPE= parameter is specified in the user managed USERVAR and copied automatically into the VTAM managed USERVARs. The three types are:

STATIC USERVAR- the USERVAR value is not expected to change, and once the automatic USERVAR's are set, they will not be changed unless the automatic USERVAR is deleted first.

DYNAMIC USERVAR- the USERVAR value only changes occasionally, such as XRF takeover following an application failure. With this type, VTAM will recheck the USERVAR value after every abnormal session termination to the application referenced by the USERVAR value. It does this by performing a normal cross domain search to the application, as though the automatic USERVAR did not exist.

VOLATILE USERVAR- the USERVAR value is expected to change often, resulting in a cross domain search to re-establish the USERVAR value each time it is referenced in a session logon instead of creating an automatic copy of the user managed USERVAR. As a result of the additional searches required, this could have a substantial impact on system performance during session establishment.

11.7.3 Planning Considerations

The ability for VTAM to manage USERVAR values can only occur if all hosts on the session establishment path have the enhancement maintenance installed.

At least one host must have a user managed USERVAR from which other hosts obtain information for their automatic USERVARs.

With the exception of XRF, VTAM will not change user managed USERVARs, leaving that responsibility to the user or subsystem that created them.

Existing procedures and NETVIEW CLISTs will continue to work as before with the current release of VTAM, allowing current implementation to continue until all hosts are fully migrated to the enhanced software levels.

The USERVAR name and its value must be unique; for instance, the following command is invalid:

```
F NET,USERVAR,ID=CICSA,VALUE=CICSA
```

11.8 Session Management Exit Enhancements

The Session Management Exit in VTAM V3R2 has been enhanced in the ADJSSCP and GWPATH selection functions. The Session Management Exit routine allows combining of session related functions (Authorization, Accounting, GWPATH Selection, etc.) into one exit. The exit is called in the SSCP of the originating logical unit (OLU), destination logical unit (DLU), and in each SSCP along the session setup path.

In prior VTAM releases, the ADJSSCP selection function of the SME provided the capability to reorder or reduce the list of adjacent SSCPs from which the next SSCP would be chosen for CDINIT routing for LU-LU session setup.

With the VTAM V3R2 enhancement, the ADJSSCP Selection function will be invoked during DSRLST (Direct Search List) routing as well as CDINIT routing. The DSRLST RU is sent during INQUIRE APPSTAT macro processing (checks an application programs ability to establish sessions), and during automatic logon (VARY LOGON, LOGAPPL) processing for a switched SLU. The DSRLST processing uses routing information from CDRSC definitions and ADJSSCP tables. As with CDINIT, when the exit is called during DSRLST processing, the list of adjacent SSCP's may be reordered or reduced in preparation for choosing the next SSCP for LU-LU session setup. Additional information available to the exit is whether the PLU or the SLU is initiating the session, whether the session is being initiated by Autologon, and whether the session is being initiated by CLSDST PASS.

11.9 Direct Search List (DSRLST) Changes

Direct Search List is a method that VTAM uses to determine the status of a resource before sending the session initiation request for that resource. Direct Search List is used when a VTAM application program issues Inquire APPLSTAT, when a VARY LOGON of a Switched SLU occurs, or when LOGAPPL is used in conjunction with a Switched SLU.

In VTAM V3R2, DSRLST routing has been improved. The same routing logic that is used with SNI or MSNF environments is used for DSRLST routing with VTAM V3R2 in order to improve DSRLST routing performance. See the SNI Considerations Chapter of this bulletin for a description of VTAM V3R2 routing logic.

11.10 Pacing

11.10.1 Overview

This will provide an overview of session level pacing for NCP boundary attached LUs. We will not discuss session level pacing to locally attached (channel) LUs on the VTAM boundary.

There has been a slight change in the traditional LU-LU session pacing using VTAM V3R2 with NCP V4R3 or V5R2, along with a new method of session pacing called 'Adaptive Session Pacing'.

11.10.2 Fixed Session Pacing

Using Figure 11-11 on page 11-23, we will review fixed session pacing for LUs attached to an NCP. Notice that the NCP is up-level software (V4R3 - V5R2) and the VTAM is down level (V3R1.1). Also notice the new naming conventions; the connection between the boundary LU and the NCP is named the Route Extension (REX) stage, while the connections between subareas are named Virtual Route (VR) stages.

The method of establishing pacing values has not changed but is included here for review. For LU-LU sessions, pacing values are determined from definitions in the LOGMODE entry or from NCP definitions. For example, the pacing value for traffic flowing from the APPL LU on the left side of Figure 11-11 on page 11-23 to the boundary NCP is determined by the PSND (Primary Send) value from the SLU's logmode entry, or, if the PSND is zero, from the VPACING value on the NCP definition statement for the SLU (LU-A). The determination of a pacing value from the boundary NCP to the LU (LU-A), is from the SRCV (Secondary Receive) value in the SLU's logmode entry, or, if the SRCV value is zero, from the PACING value on the NCP definition statement for the SLU (LU-A). These two pacing stages constitute outbound session pacing.

Inbound session pacing traditionally was a single stage operation, between the two session endpoints. If however, the NCP level is V4R3 or V5R2 or higher, the NCP becomes a staging point for inbound pacing, making it two stage. This change is transparent to users and does not require any modifications to existing pacing definitions. Inbound pacing is determined by the SSND (Secondary Send) value in the SLU's logmode entry which operates as a switch. If the SSND has a non-zero value, then the value coded for VPACING on the APPL VBUILD statement in our example would be used as the inbound pacing value. If the SSND value is zero, no inbound pacing will occur.

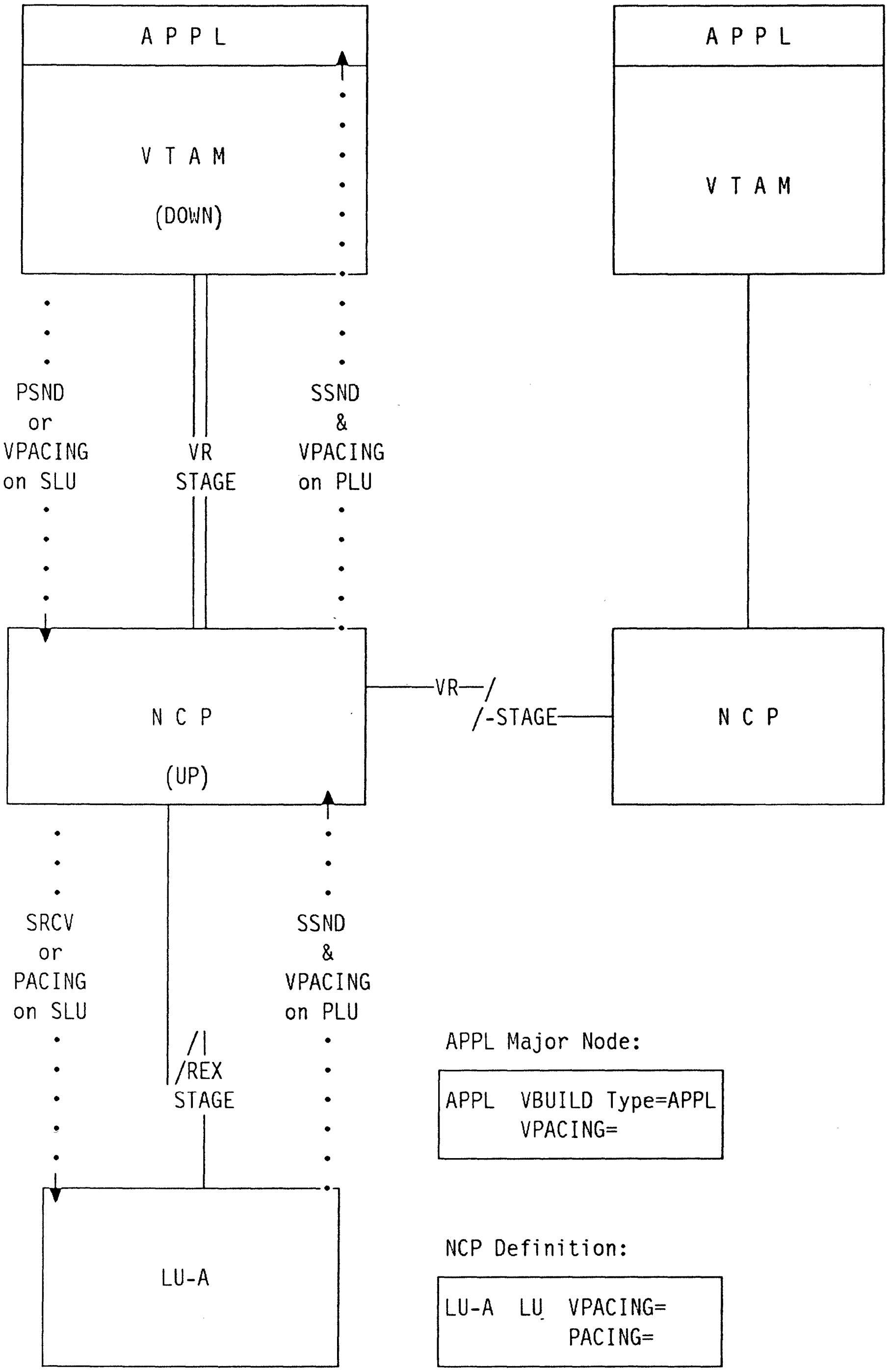


Figure 11-11. Fixed Session Pacing - Dependent LU

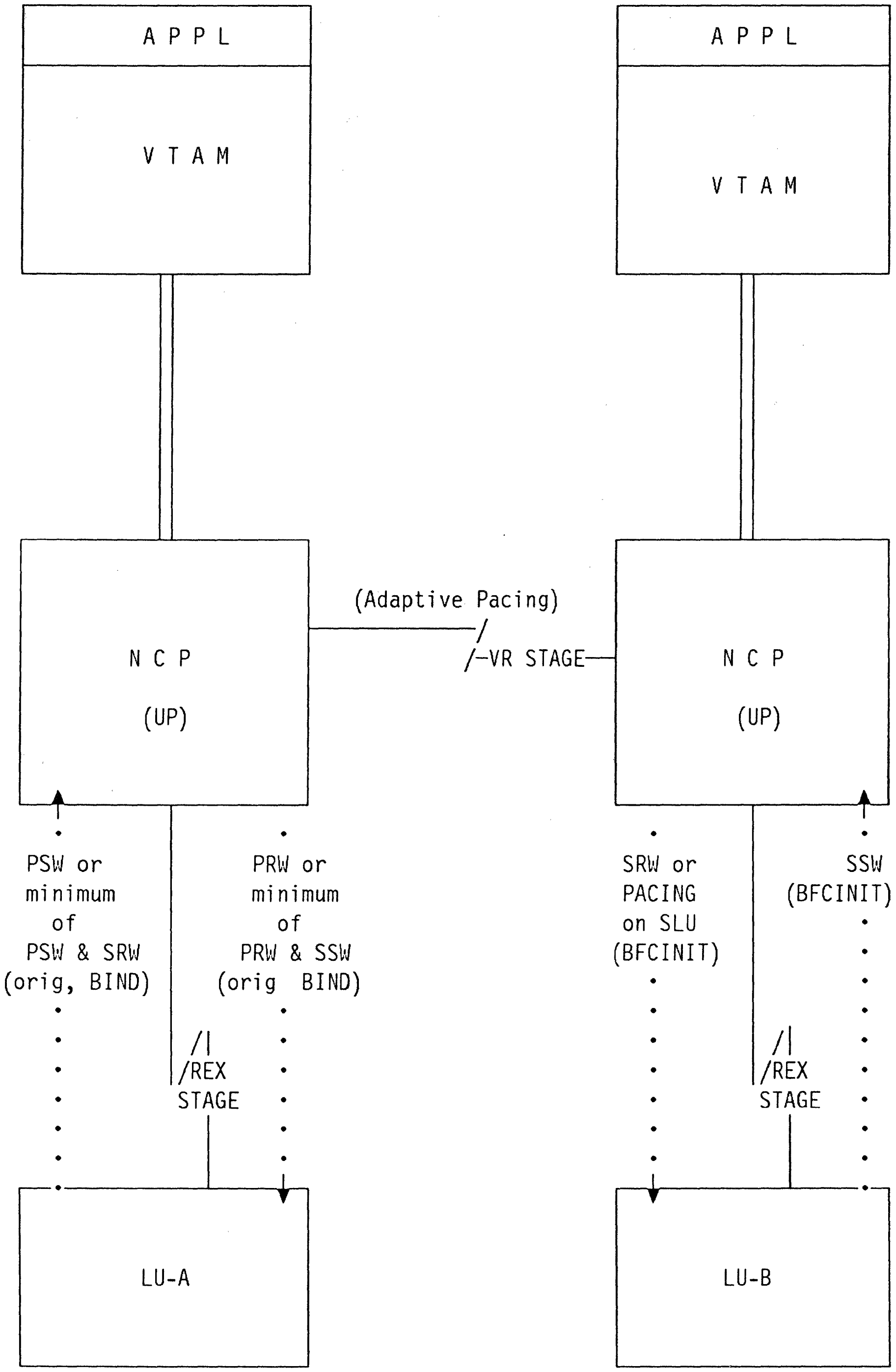

APPL
VTAM
APPL
VTAM
NCP
(UP)
(Adaptive Pacing)
/-VR STAGE
NCP
(UP)
PSW or minimum of PSW & SRW (orig, BIND)
PRW or minimum of PRW & SSW (orig BIND)
/REX STAGE
SRW or PACING on SLU (BFCINIT)
SSW (BFCINIT)
/REX STAGE
LU-A
LU-B

Figure 11-12. Fixed Session Pacing - Independent LU

Figure 11-12 displays an Independent LU (LU-A) establishing a session with its intended partner (LU-B) which is in a different domain. The asumption is that the LU (LU-A) initiating the session is not an AS/400 or SYS/36, and as such does not support Adaptive Session Pacing. For our purposes, we will assume LU-A to be a PC or PS/2 defined as an Independent LU, supporting fixed pacing only.

Notice in Figure 11-12, that the PSW (Primary Send Window) will be established from the PSND value in the ORIGINAL BIND sent to the NCP from LU-A. If the BIND is negotiable, the value for the PSW stage will be the minimum of the PSND or SRCV from the ORIGINAL BIND.

The REX stage between the NCP and LU-B, however, will have the fixed session pacing value determined by the SRCV value in the logmode entry associated with LU-B or, if zero, the value coded for the PACING keyword on the SLU (LU-B) definition in the NCP.

Notice the difference in the method of setting pacing values on the REX stage for the PLU (LU-A) and the REX stage for the SLU (LU-B). Since the PLU REX stage is determined by values in the ORIGINAL BIND only, there is no way of effecting that pacing value with host based definitions. As an example, if the BIND received at the NCP from LU-A had zero values for pacing, the PSW would be zero since we have no method of overriding it, while the SRCV value of zero in the BIND could be overridden by coding a value for SRCV in the SLU's logmode entry or, if zero, by coding a value for PACING on the NCP definition for LU-B.

A potential problem could occur with this scenario if LU-A were sending large blocks of data to the NCP (file transfer), with no pacing mechanism available to throttle it (there is no way to modify the pacing values in the BIND that is sent from a PC or PS/2). The end result could affect performance or, in severe instances, drive the NCP into an overrun condition or even slowdown.

The above problem has been resolved through PTF maintenance for VTAM and NCP. With the maintenance applied, if the BIND from LU-A is received with a PSND value of zero, then a new control vector passes the value of PSND from the logmode table of the PLU (LU-A); or, if the logmode PSND is zero, the VPACING value coded in the PLU (LU-A) definition is used.

The APARs describing the problem and the resolution are:

- VTAM V3R2 - OY24254
- NCP V4R3 or V4R3.1 - IR86037
- NCP V5R2 or V5R2.1 - IR86244

11.10.3 Adaptive Session Pacing

Adaptive session pacing is an alternative to fixed session pacing and is introduced at the subarea level with NCP V4R3/V5R2 and VTAM V3R2 or later VTAM and NCP levels.

Simply stated, adaptive session pacing allows the receiver of PIUs to determine the window size (pacing value) based on resource constraints within the receiver node.

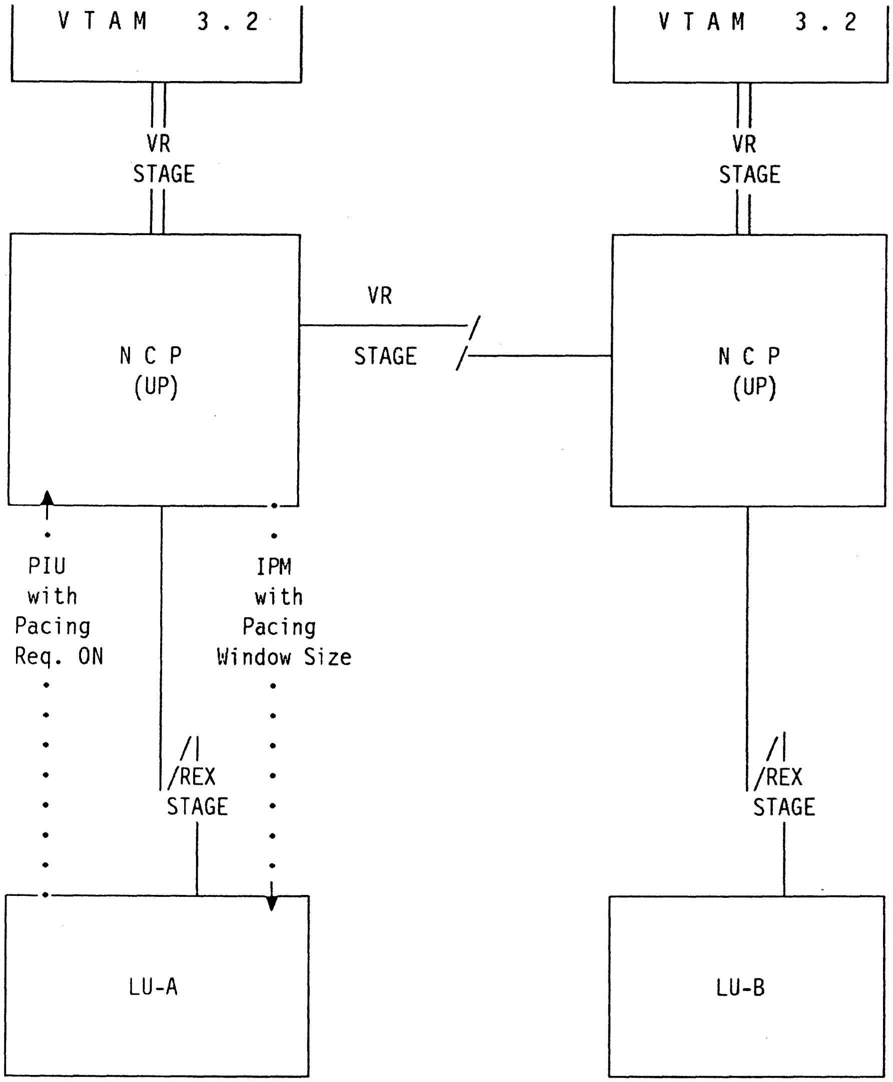


Figure 11-13. Adaptive Session Pacing

As an example of adaptive session level pacing, assume in Figure 11-13, that LU-A is an AS/400 and is the PLU side of the session. The first PIU of the session sent to the boundary NCP will have the pacing request bit ON. The NCP will return an IPM (Isolated Pacing Message) with a window size of one (1). For each subsequent PIU sent from the AS/400 to the NCP with the pacing request bit ON, the NCP will increase the window size in the IPM by a factor of eight (8) if there are no constraints (buffer resources, etc.) within the NCP node. For example, with a non-constrained NCP, the IPM window size values returned from the NCP would be 1, 8, 64, 512, 5096, and then 32,767 which is the maximum (essentially a no pacing value). Naturally, if there are constraints in the receiving node, then the window size would be adjusted accordingly.

If LU-A transmits large numbers of PIUs to the NCP, such that the queue in the NCP reaches a predetermined threshold, the adaptive session pacing algorithm would consider this session to be BATCH oriented. NCP would return an unsolicited IPM to LU-A with a new window size of one (1). The window size of one (1) would remain constant, unless the transmitter of the PIUs (LU-A) turned on the RLWI (Request Larger Window Size Indicator) in the PIU. The receiving node (NCP in our example) has the option of honoring the request and raising the window size or ignoring it.

There has been a problem with sessions being prematurely set to BATCH status creating performance problems with interactive sessions. An NCP APAR, IR86790, describes the problem and the solution.

Only sessions initiated by an Extended BIND support adaptive session pacing. An extended BIND is one in which a new control vector, a Fully Qualified PCID (Procedure Correllation ID), is appended to the BIND. Fully qualified meaning NETID.SSCPNAME.PCID, where the PCID is an eight byte random string of characters generated by the PLU (issuer of the BIND). The PCID is used to uniquely identify the session throughout the path between session endpoints. Currently, the AS/400 and SYS/36 (in APPN mode) are the only peripheral devices capable of supporting Extended BINDs. With up-level VTAMs and NCPs, as shown in Figure 11-13 on page 11-26, all BINDs will be extended by the NCP, whether they originate from host based applications or peripheral nodes capable of issuing BINDs. As a result, on VR stages between up level VTAMs and NCPs, adaptive session pacing will be used, even if the session endpoints are LUs that do not support extended BINDs. Although adaptive pacing will occur between uplevel VTAMs and NCPs, the pacing mechanisms differ based on the direction of session traffic. If the PIU with the pacing request bit ON is received by VTAM from the NCP, VTAM will return an IPM with the maximum value (32,767), which would be considered a no pacing value. If SSND (Secondary Send) has a non-zero value defined in the logmode entry for the SLU, and a value is coded for VPACING on the APPL VBUILD statement, then VTAM will return that value (VPACING) to the NCP in the IPM response to the PIU with the pacing request bit ON sent to VTAM from the NCP. We are assuming in this example that one of the session endpoints is a host application.

In the reverse direction, if the PIU with the pacing request bit ON is received from VTAM by the NCP, the NCP adaptively paces as described earlier, returning an IPM with a window size increasing by a factor of eight for interactive traffic.

A good description of adaptive sesion pacing can be found in the NCP Reference manual (LY30-5569).

11.11 DISPLAY SESSIONS

11.11.1 Using DISPLAY SESSIONS

VTAM V3.2 introduces a new command, DISPLAY SESSIONS. It can be used by the VTAM operator to display session status for network resources. Session status can be displayed for:

1. A single session identified by its session identifier (SID).
2. All sessions in which a specified LU is the PLU.
3. All sessions in which a specified LU is the SLU.
4. All sessions in which a pair of LUs have a specified PLU/SLU relationship as session partners.

5. All sessions in which a specified LU is a session partner, regardless of its being PLU or SLU.
6. All sessions between a pair of LUs regardless of their PLU/SLU relationship as session partners.
7. A count of all sessions involving a specified LU.
8. A count of all sessions in VTAM's domain.
9. All sessions in VTAM's domain.

To display sessions between specific LUs, one of the session partners must reside in the host VTAM network.

Issuing the SCOPE operand of the command allows the operator to determine the session status type that is to be displayed: ALL sessions, all PENDing sessions, all Queued sessions, or all ACTive sessions. SCOPE = PEND is the default. The LIST operand allows the operator to specify whether a COUNT of sessions by session status or a list of ALL sessions is desired. LIST = COUNT is the default.

A full description of the DISPLAY SESSIONS command can be found in *VTAM Messages and Codes, SC23-0114.*

11.11.2 Examples

The following figures, on pages 11-28 through 11-31, show the various uses of the DISPLAY SESSIONS command. Care must be taken in a large network where there can be many sessions to avoid issuing the command in a manner that will result in the VTAM operator's terminal screen being flooded with unwanted session data. An example would be issuing:

```
D NET,SESSIONS,SCOPE=ALL,LIST=ALL
```

A display of every session that particular VTAM is aware of would result.

```
          D NET,SESSIONS,SID=EB53BBA0081E3C4A

IST097I  DISPLAY  ACCEPTED
IST879I  PLU/DLU REAL = CSSNET.N2P1004    ALIAS = CSSNET.N2P1004
IST879I  SLU/OLU REAL = CSSNET.X1A4       ALIAS = CSSNET.X1A4
IST880I  SETUP STATUS = ACTIV
IST314I  END
```

Figure 11-14. Displaying a session by session identifier. The session pair identified by SID = EB53BBA0081E3C4A.

VTAM identifies each session that involves one or more resources within its domain with a session identifier (SID). It is a unique 8 byte field. It can be useful in environments where a PLU may have many sessions with one or more SLUs and information about one specific session is required.

```
          D NET,SESSIONS,PLU=N2P1004,SCOPE=ALL

IST097I  DISPLAY  ACCEPTED
IST873I         PLU               SLU               SID        STATUS
IST874I  CSSNET.N2P1004   CSSNET.X1A4       EB53BBA0081E3C4A ACTIV
IST878I  NUMBER OF PENDING SESSIONS =         0
IST878I  NUMBER OF ACTIVE  SESSIONS =         1
IST878I  NUMBER OF QUEUED  SESSIONS =         0
IST878I  NUMBER OF TOTAL   SESSIONS =         1
IST314I  END
```

Figure 11-15. Displaying all sessions for a primary logical unit (PLU). Sessions where N2P1004 is the PLU.

```
          D NET,SESSIONS,SLU=X1A4,E

IST097I  DISPLAY  ACCEPTED
IST873I         PLU               SLU               SID        STATUS
IST874I  CSSNET.N2P1004   CSSNET.X1A4       EB53BBA0081E3C4A ACTIV
IST878I  NUMBER OF PENDING SESSIONS =         0
IST878I  NUMBER OF ACTIVE  SESSIONS =         1
IST878I  NUMBER OF QUEUED  SESSIONS =         0
IST878I  NUMBER OF TOTAL   SESSIONS =         1
IST314I  END
```

Figure 11-16. Displaying all sessions for a secondary logical unit (SLU). Sessions where X1A4 is the SLU.

```
          D NET,SESSIONS,PLU=N2P1004,SLU=X1A4,SCOPE=ALL

IST097I  DISPLAY  ACCEPTED
IST873I         PLU               SLU               SID        STATUS
IST874I  CSSNET.N2P1004   CSSNET.X1A4       EB53BBA0081E3C4A ACTIV
IST878I  NUMBER OF PENDING SESSIONS =         0
IST878I  NUMBER OF ACTIVE  SESSIONS =         1
IST878I  NUMBER OF QUEUED  SESSIONS =         0
IST878I  NUMBER OF TOTAL   SESSIONS =         1
IST314I  END
```

Figure 11-17. Displaying all sessions for a PLU/SLU pair. Sessions for the PLU/SLU pair N2P1004/X1A4.

```
          D NET,SESSIONS,LU1=N2P1004,SCOPE=ALL

IST097I  DISPLAY  ACCEPTED
IST873I        PLU              SLU                SID        STATUS
IST874I  CSSNET.N2P1004    CSSNET.X1A4       EB53BBA0081E3C4A ACTIV
IST878I  NUMBER OF PENDING SESSIONS =          0
IST878I  NUMBER OF ACTIVE  SESSIONS =          1
IST878I  NUMBER OF QUEUED  SESSIONS =          0
IST878I  NUMBER OF TOTAL   SESSIONS =          1
IST314I  END
```

Figure 11-18. Displaying all sessions for a specific LU. All sessions with the LU N2P1004.

```
          D NET,SESSIONS,LU1=N2P1004,LU2=X1A4,E

IST097I  DISPLAY  ACCEPTED
IST873I        PLU              SLU                SID        STATUS
IST874I  CSSNET.N2P1004    CSSNET.X1A4       EB53BBA0081E3C91 ACTIV
IST878I  NUMBER OF PENDING SESSIONS =          0
IST878I  NUMBER OF ACTIVE  SESSIONS =          1
IST878I  NUMBER OF QUEUED  SESSIONS =          0
IST878I  NUMBER OF TOTAL   SESSIONS =          1
IST314I  END
```

Figure 11-19. Displaying all sessions for a specific LU pair. Sessions between the LU pair N2P1004 and X1A4.

```
          D NET,SESSIONS,LU1=X1A4,LIST=COUNT,E

IST097I  DISPLAY  ACCEPTED
IST878I  NUMBER OF PENDING SESSIONS =          0
IST878I  NUMBER OF ACTIVE  SESSIONS =          1
IST878I  NUMBER OF QUEUED  SESSIONS =          0
IST878I  NUMBER OF TOTAL   SESSIONS =          1
IST314I  END
```

Figure 11-20. Displaying the session count for an LU. The number of sessions for LU X1A4.

```
 * N2P1     D NET,SESSIONS,LIST=COUNT,SCOPE=ALL
 ' N2P1
IST097I  DISPLAY  ACCEPTED
IST878I  NUMBER OF PENDING SESSIONS =          0
IST878I  NUMBER OF ACTIVE  SESSIONS =          6
IST878I  NUMBER OF QUEUED  SESSIONS =          0
IST878I  NUMBER OF TOTAL   SESSIONS =          6
IST314I  END
```

Figure 11-21. Displaying the session count in VTAM's domain. Session counts are by session category.

```
        D NET,SESSIONS,LIST=ALL,E

IST097I  DISPLAY  ACCEPTED
IST873I        PLU              SLU              SID         STATUS
IST874I  CSSNET.N2P1004   CSSNET.X1A4      EB53BBA0081E3C4A ACTIV
IST874I  CSSNET.TS10001   CSSNET.X1A2      EB53BBA0081E3C47 ACTIV
IST874I  CSSNET.N2P1003   CSSNET.X1A1      EB53BBA0081E3C3E ACTIV
IST874I  CSSNET.N2P1LUC   CSSNET.N2P2LUC   EB53BBA0081E3BE9 ACTIV
IST874I  CSSNET.DSIAMLUT  CSSNET.ISTPDCLU  EB53BBA0081E36B6 ACTIV
IST874I  CSSNET.DSIAMLUT  CSSNET.ISTPDCLU  EB53BBA0081E36B5 ACTIV
IST878I  NUMBER OF PENDING SESSIONS =       0
IST878I  NUMBER OF ACTIVE  SESSIONS =       6
IST878I  NUMBER OF QUEUED  SESSIONS =       0
IST878I  NUMBER OF TOTAL   SESSIONS =       6
IST314I  END
```

Figure 11-22. Displaying the sessions in VTAM's domain. All session pairs where one partner is within the domain.

11.12 VTAM Sense Code Enhancements

VTAM sense codes have changed in VTAM V3.2 in comparison to VTAM V3.1.1. New sense codes have been added and some have been deleted or modified in comparison to VTAM V3.1.1. The intent is to have the sense code convey more and better information to the person responsible for network operation or network problem determination.

The following tables show the sense codes that have been added since VTAM V3.1.1 or have been changed through deletion or modification when included in VTAM V3.2.

Note: This list is not authoritative - more sense codes could have been added, deleted, or modified in the time since this list was compiled. It is incumbent upon the reader to read *VTAM Messages and Codes*, SC23-0114, or *SNA Network Product Formats*, LY43-0081, to determine valid sense codes. *The NCP Reference*, LY30-5605, decribes NCP-generated sense codes and their causes.

11.12.1 The Sense Code

Figure 11-23 on page 11-32 below outlines the basic structure of the VTAM sense code. Byte 0 is the Category of the sense code, i.e., the category of error detected by VTAM. Byte 1 modifies the category and bytes 2-3 provide data that make the error description more specific and thus, more useful to the person performing problem determination and problem source identification (PD/PSI). Bytes 0-1 together form the **sense code** while bytes 0-3 comprise the **sense data**.

Byte 0	Byte 1	Byte 2	Byte 3
Category	Modifier	Sense code specific information	
Sense Code →			
Sense Data →			

Figure 11-23. Detail of VTAM Sense Code.

11.12.2 Sense code changes in VTAM V3.2

VTAM V3.2 uses bytes 2-3 more often than prior VTAM releases in an attempt to aid in PD/PSI. In many cases, existing sense codes have been enhanced through the use of these last two bytes.

Table 11-1. VTAM sense codes changed in VTAM V3.2. Added, deleted, and modified sense codes by category.

VTAM Sense Code Category	Added	Deleted	Modified
X'08' Request Reject	081F,0830,0838, 083E,0849,084F, 0854,086A,086C, 086D,086E,086F, 0870,088C,0890, 0891,0892,0893, 0894,0895,0896, 0897,0898,0899, 089A,089B,089C, 089D,08A0,08A2	0885	0801,0805,0806, 0809,080A,080C, 080E,0812,0813, 0815,0817,0818, 081A,081C,081E, 0821,0824,082C, 0832,0833,0834, 0835,0839,083A, 0840,0841,0842, 084B,084C,084D, 0852,0857,0861, 0864,086B,0877, 0878,0879,087D, 0888,088E,088F
X'10' Request Error	1004,1006,1011	NONE	1001,1003,1005, 1007,1008
X'20' State Error	2012	NONE	2002,2003,2011
X'40' Request Header (RH) Usage Error	4002,400E	NONE	NONE
X'80' Path Error	8014,8015,8016, 8018,8020	NONE	8003,8004,8005, 8006,8007,800F, 8013

11.12.3 Example Sense Code Enhancement

The examples below (Figure 11-24, Figure 11-25 on page 11-34) compare the meaning of the VTAM sense code **0817** in VTAM V3.1.1 to its meaning in VTAM V3.2. It shows that the VTAM V3.2 implementation of the sense code conveys more information about the reported error and significantly improves PD/PSI by pinpointing more exactly the error's source.

11.12.3.1 VTAM V3.1.1

Figure 11-24 is the description for sense code 0817 as it exists in VTAM V3.1.1.

```
0817   Link inactive. A request requires use of the link, but
       the link is not active.
```

Figure 11-24. Sense code 0817 as defined in VTAM V3.1.1.

11.12.3.2 VTAM V3.2

Figure 11-25 is an excellent example of the sense code enhancements in VTAM V3.2. It uses bytes 2-3 to specify more granularly the causes of the link activation's failure.

0817 Link or link resource not active. A request requires use of a link or link resource that is not active.

Bytes 2 and 3 following the sense code contain sense-code specific information. Settings allowed are:

0000 No specific code applies
0001 Link inactive
0002 Link station inactive
0003 Switched link connection inactive

Figure 11-25. Sense code 0817 as defined in VTAM V3.2.

11.13 ACF/NCP Version 5 Usage Tiers

The announcement of NCP V5 for the 3720 and the 3745 introduced a new method of pricing the product licenses: **usage tiers**. The usage tier gauges the product's price by the size of the communications controller it supports. The usage tier applies only to NCP V5 executing on the 3720 or the 3745; it is not used for any release of ACF/NCP V4.

11.13.1 Determining the Usage Tier

Usage tiers are determined by the physical components of the 3720 and 3745. The number of scanners, channel adapters, and token-ring adapters is used as a guide to calculate the required usage tier for a given 3720/3745 configuration.

The usage tiers are:

Tiers	Maximum physical components		
1	1 LSS/TSS + 1 TRA + 2 CA		
2	2 LSS/TSS + 1 TRA + 2 CA		
3	8 LSS/HSS + 8 CA	or	4 LSS/HSS + 4 TRA + 8 CA
4	24 LSS/HSS + 16 CA	or	20 LSS/HSS + 4 TRA + 16 CA
5	32 LSS/HSS + 16 CA	or	28 LSS/HSS + 4 TRA + 16 CA

A simpler method to use when determining the appropriate usage tier for a given configuration is as follows:

Tiers	Representative machine(s)
1	3720 Base frame
2	3720 and 3721 Model 1 or 2 expansion unit
3	3745 Base frame
4	3745 and 3746-A11 expansion unit
5	3745 and 3746-A11 and 3746-A12 expansion units

CF3725/CF3745 generates the appropriate usage tier for the customer's configuration on the hardware reports. CFPROGS may be used in configuring the software order.

11.13.2 Installation Considerations

The introduction of usage tiers affects the installation of NCP V5 as well as its licensing. It affects the tapes the customer will receive, the SMP installation process, and the NCP generation. The following sections highlight the changes in the installation and implementation of NCP V5 brought about by usage tiers.

11.13.2.1 Product Tapes

When the customer receives NCP V5, the package will consist of two tapes: the tape for Tier 1 and the tape for the highest Tier ordered by the customer. Thus, if the customer ordered a Tier 3 NCP, a Tier 1 and a Tier 3 tape will arrive. The Tier 1 tape contains all the NCP modules plus the 'keys' for usage tier 1. The Tier 3 tape provides the 'keys' for tiers 3 and 2. The highest tier tape includes support for itself and all the tiers between itself and tier 1. However, a tier tape will not provide the 'keys' for a higher tier: a Tier 3 tape will not have the 'keys' needed for Tier 4.

If a customer wishes to migrate to a tier higher than the existing tier license, the license can be upgraded to that tier by adding the necessary tier feature code to the NCP V5 license. The feature codes on the license will be for each tier up to and including the highest tier.

11.13.2.2 Installation

Each tier of the NCP has its own FMID. When installing NCP V5, the only FMIDs installed will be those tiers the customer ordered. Upgrading to a new tier will not require a re-installation of NCP V5, but rather the installation of the new FMID that tier represents.

11.13.2.3 Coding Usage Tier in the NCP

A new NCP keyword must be used when creating an NCP V5 load module to indicate the usage tier for the NCP. That keyword is **USGTIER**, which is coded in the NCP's BUILD macro. USGTIER = 1 is the default.

USGTIER can be coded for a tier equal to or less than the usage tier for which the NCP is licensed. Coding a usage tier value that exceeds the licensed usage tier will result in NCP generation failure.

Consider a customer who has an NCP license with the Usage Tier 3 feature codes on the license:

- The customer generates an NCP load module for a 3720 specifying USGTIER = 1 in the NCP source. The NCP generation will succeed.
- The customer generates an NCP load module for a 3745 specifying USGTIER = 3. The 3745 has 4 LSS, 1 TRA, and 2 CA. The NCP generation will succeed.
- The customer generates an NCP load module for a 3745 specifying USGTIER = 3. However, 10 LSS are coded in the NCP source. The NCP generation will fail because the resources defined in the source deck exceed those supported by Usage Tier 3. The NCP generation will fail.
- The customer generates an NCP load module for a 3745 specifying USGTIER = 4. The NCP generation will fail because the usage tier specified exceeds that supported by the NCP libraries.

When coding USGTIER in the NCP source, keep in mind the limitations on line addresses imposed by the usage tiers. The value in USGTIER will affect the

ADDRESS that can be coded in the NCP LINE macro. Coding USGTIER = 1|2 for a 3745 means that line addresses 1024-1039 cannot be used because usage tiers 1 and 2 do not support HSS lines. Furthermore, only addresses 1088 and 1089 are valid for TRAs. These restrictions extend to the addresses used for 3745 channel adapters: the USGTIER value restricts the ADDRESS that can be coded in the LINE macro for the channel adapter logical address. These restrictions can be examined in more detail in the *NCP Resource Definition Reference*, SC30-3448.

Chapter 12. Extended Subarea Addressing and 16 Explicit Routes

12.1 Extended Subarea Addressing

A new function with VTAM Version 3 Release 2 (V3R2) and NCP V4R3.1/V5R2.1 is the ability to address 65,535 subareas (VTAMs and NCPs) within the same network. The new function, called Extended Subarea Addressing (EXSA), is available in the following levels (or higher) of VTAM and NCP:

- VSE/VTAM V3R2
- VM/VTAM V3R2 with APAR VM32259
- MVS/370 VTAM V3R2 with APAR OY14997
- MVS/XA VTAM V3R2 with APAR OY14996
- NCP V4R3.1
- NCP V5R2.1

Prior to Extended Subarea Addressing, VTAMs and NCPs with the correct level of software, called Extended Network Addressing (ENA), can use subarea addresses from 1 to 255. The levels of VTAM and NCP necessary to support ENA are:

- VTAM V3R1 and higher for MVS/370, MVS/XA, and VSE
- VTAM V3R1.1 and higher for VM
- NCP V4R1 and higher for MVS and VSE
- NCP V4R2 and higher for VM

With levels of VTAM and NCP prior to Extended Network Addressing, the number of subareas allowed within a network is governed by the coding of a parameter, MAXSUBA. MAXSUBA indicates to VTAM and NCP how many bits of a sixteen bit field should be used to address the subarea and how many bits should be used to address the element.

Figure 12-1 on page 12-4 shows the addressing structure used in pre-ENA levels of VTAM and NCP, ENA levels, and EXSA levels. With pre-ENA levels of VTAM and NCP, two bytes (16 bits) are carried in the transmission header (TH) of PIUs. The figure shows that when MAXSUBA=63 has been specified, that 6 bits of the two bytes are used to designate the subarea number, and the remaining 10 bits are used to designate the element number (element numbers through 1024 can be used). When the coding of MAXSUBA is changed, it affects how many bits are used for subarea and how many are used for element. For example, if MAXSUBA=31 is coded, then five bits (up to 31 subareas) are used to address the subarea, and 11 bits (up to 2048 elements per subarea) are used to address the element. MAXSUBA=15 results in 4 bits (though 15) for subarea, and 12 bits (through 4096) for element.

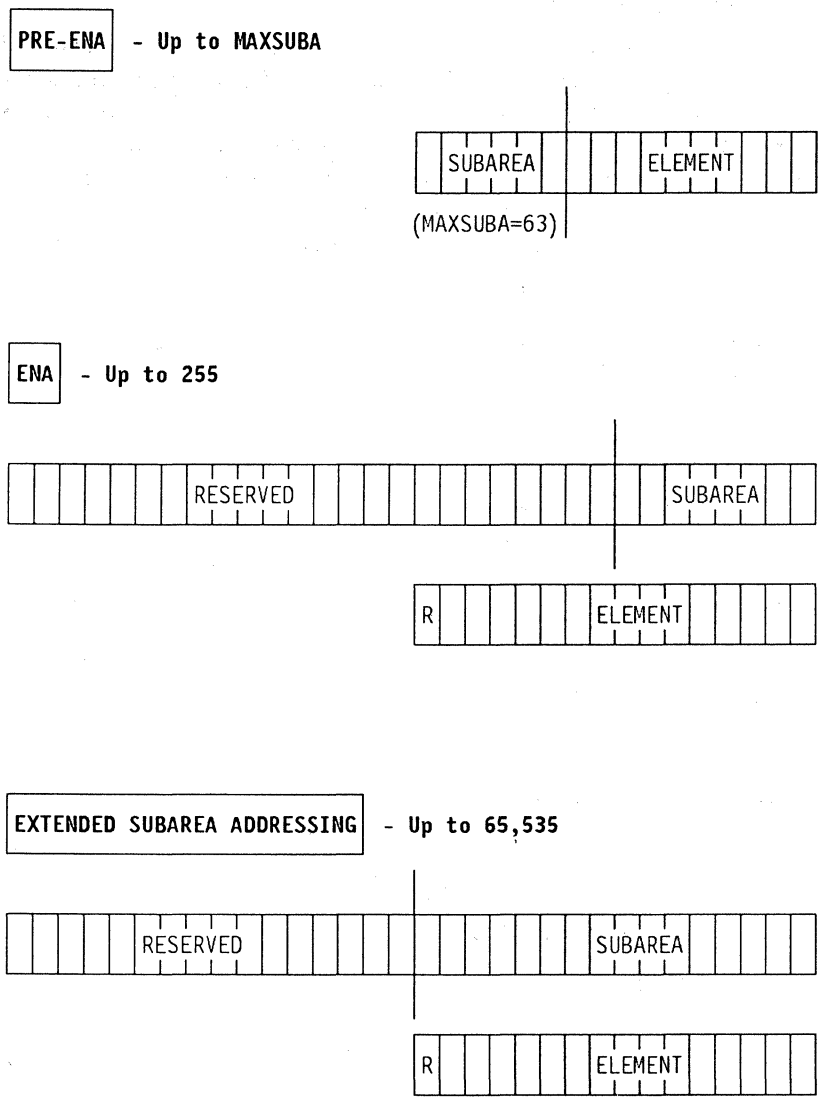


Figure 12-1. EXSA, ENA, and pre-ENA Addressing Structure

What changes with VTAM V3 and NCP V4 (ENA), is that the 16 bits, formerly split between element and subarea, are used to address the element with the high order bit reserved, resulting in the ability to address up to 32,768 elements per subarea. Four bytes are allocated for subarea address but only the low order byte is used with ENA level software, resulting in the ability to address up to 255 subareas.

With Extended Subarea Addressing (EXSA), two bytes (16 bits) can be used to address subareas, resulting in the ability to address up to 64K subareas. Four bytes are actually carried in the transmission header, but EXSA levels of software only use two bytes, and ENA levels only use one byte.

12.2 Coding Subarea Addressing Limits

When Extended Network Addressing levels of VTAM and NCP are installed, subarea numbers 1 through 255 can be used. Extended Subarea Addressing differs from Extended Network Addressing in that the user can specify limits on the number of subareas that can be addressed through installation or generation parameters. Parameters are being provided to allow the user to control the variables so that NCP storage can be conserved if it is not needed. The increase in number of subareas and number of explicit routes increases the amount of NCP storage needed to build routing tables.

The last section in this Chapter contains information regarding the effect on NCP storage usage of varying subarea limits.

When the Extended Network Addressing level of VTAM is installed, the VTAM constants module, ISTRACON, is used to indicate the highest subarea number that is addressable. By default, the value in ISTRACON is set to 511. If the user wishes to address more subareas, they can "ZAP" more bits on. By zapping the next higher order bit on, 1023 subareas can be addressed; by zapping the next higher order bit on, 2047 subareas can be addressed, and upward in powers of two (4095, 8191, 16383, 32767, and 65535).

For NCP, a parameter called SALIMIT can be coded on the BUILD definition statement to specify the highest subarea number that is addressable by this NCP. For gateway-NCPs, the SALIMIT parameter can also be coded on NETWORK definition statements. SALIMIT is coded with numbers in powers of two--255, 511, 4095, 8191, 16383, 32767, and 65535. The default value for SALIMIT is 255.

12.3 Migration Considerations for Extended Subarea Addressing

The suggested migration path for users who wish to use Extended Subarea Addressing is:

- Continue to use subarea addresses of 255 or lower while upgrading VTAM and NCP to EXSA levels.
- When installing EXSA levels, set VTAM's ISTRACON and NCP's SALIMIT to the maximum value of subareas planned for the future.
- After all nodes are at EXSA level, start using subarea numbers greater than 255.

If users need to use subarea numbers higher than 255 prior to upgrading all VTAMs and NCPs to EXSA level software, they must be sure to adhere to the following rules:

- If a pre-EXSA node is ADJACENT to an EXSA node, the EXSA node's subarea number must be 255 or lower.
- VTAM must be EXSA level if it owns any NCPs that are link attached to another NCP with a subarea number greater than 255.

When pre-EXSA nodes and EXSA nodes are present in a network, pre-EXSA nodes can route through EXSA nodes with high subarea numbers, as long as the pre-EXSA nodes are NOT adjacent to the EXSA nodes with high subarea numbers. An easy way to figure out what is allowed is to determine if a valid PATH statement can be coded. A PATH statement is coded as:

```
PATH  DESTSA=nnn,
      ERn=(ADJSA,TGn)
```

A pre-EXSA level VTAM or NCP cannot use a subarea number higher than 255 in the destination subarea (DESTSA) portion of the PATH statement, or in the adjacent subarea (ADJSA) portion of the PATH statement. If there is a high subarea EXSA node between the adjacent subarea and destination subarea, routing can be accomplished.

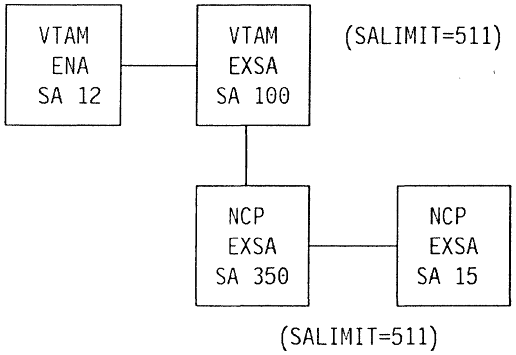


Figure 12-2. EXSA Migration--Supported Configuration

Using Figure 12-2 as an example, terminals attached to NCP subarea 15 could logon to applications running on VTAM subarea 12, even though NCP subarea 350 is in the route. Note that both nodes adjacent to NCP subarea 350 have EXSA level software with SALIMIT set to 511 (subarea 350 is addressable by both). What makes this configuration viable, is that VTAM subarea 12 can have a PATH statement with destination subarea of 15 and adjacent subarea of 100. Since VTAM subarea 12 is only ENA level, both destination subarea and adjacent subarea must be within its subarea addressing range of 255.

VIOLATION

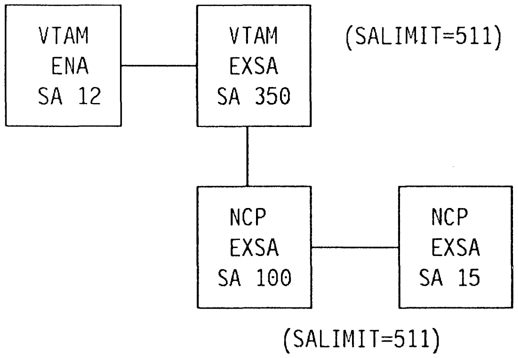


Figure 12-3. EXSA Migration--Configuration in Violation

Figure 12-3 shows a violation of one of the rules that must be followed when mixed EXSA and pre-EXSA level software exists. VTAM SA 12, which has ENA level software, CANNOT be adjacent to a node with a higher subarea address than 255 (in this example VTAM SA 350). VTAM SA 12 would not be able to have a path statement that coded SA 350 as either an adjacent subarea or destination subarea.

VIOLATION

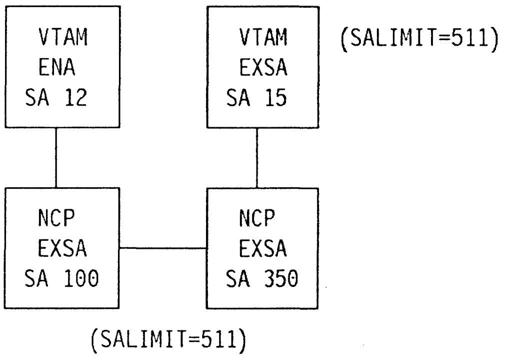


Figure 12-4. EXSA Migration--Configuration in Violation

Figure 12-4 shows another violation of one of the migration rules. In the figure, VTAM SA 12 owns NCP SA 100 and VTAM SA 15 owns NCP SA 350. However, in order for NCP SA 100 to contact NCP SA 350, the link station in NCP SA 100 used to communicate with NCP SA 350 must be owned by an EXSA VTAM. In Figure 12-4, since VTAM SA 12 is only ENA level, NCP to NCP communication cannot be accomplished since there is an NCP subarea number (350) outside of the addressing range of the ENA level VTAM.

If mixed SALIMITs are used within networks connected via SNI, the rules are similar to those for mixing ENA and EXSA nodes. The Network Program Products Planning Manual for VTAM V3R2 has additional information regarding migration considerations for mixed environments.

12.4 16 Explicit Routes

A new function with VTAM Version 3 Release 2 (V3R2) and NCP V4R3.1/V5R2.1 is the ability to use explicit route numbers between 0 and 15. Prior to these levels of software, only eight explicit routes were available, explicit routes 0 through 7. Virtual route numbers must remain in the 0 to 7 range with the new software; however, a virtual route can be mapped to more than one explicit route. The new function is available in the following levels (or higher) of VTAM and NCP:

- VSE/VTAM V3R2
- VM/VTAM V3R2 with APAR VM32259
- MVS/370 VTAM V3R2 with APAR OY14997
- MVS/XA VTAM V3R2 with APAR OY14996
- NCP V4R3.1
- NCP V5R2.1

Prior to these levels of VTAM and NCP, explicit route numbers and virtual route numbers between 0 and 7 can be used. There is an NCP parameter, ERLIMIT, that specifies whether only ER numbers less than 7 may be used (ERLIMIT= 8), or if ER numbers greater than 7 may be used (ERLIMIT= 16). Like the parameter used to specify subarea limits, ERLIMIT affects how much storage is used to build routing tables. Figure 12-6 on page 12-10 and Figure 12-7 on page 12-10 show several examples of the effect on NCP storage of varying ERLIMIT.

12.5 Migration Considerations for 16 Explicit Routes

The suggested migration path for users who wish to use Explicit Routes with numbers greater than 7 is:

- Continue to use explicit route numbers of 7 or lower while upgrading VTAM and NCP to EXSA levels.
- When installing the EXSA level of NCP, set NCP's ERLIMIT equal to 16.
- After all nodes are at EXSA level, start using explicit route numbers greater than 7.

If users need to use explicit numbers higher than 7 while their network consists of mixed EXSA and pre-EXSA levels of VTAM and NCP, they must adhere to the following rules:

- In order to use an explicit number higher than 7, all nodes on the route must have EXSA level software.
- In order to use an explicit number higher than 7, all NCPs on the route must have ERLIMIT= 16 coded.
- If there is a pre-EXSA level of software on the route, only explicit route numbers 0 through 7 can be used.

Remembering that a PATH statement is coded as:

```
PATH  DESTSA=nnn,
      ERn=(ADJSA,TGn)
```

helps clarify the above rules. An explicit route is distinguished by a number, and every node on the route must have the proper level of code (EXSA) in order to be able code a PATH statement with an explicit route number higher than 7.

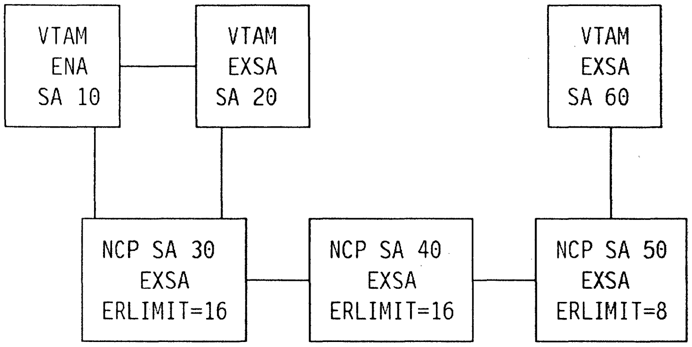


Figure 12-5. Explicit Route Numbers 0-16 -- Migration Example

Using Figure 12-5 as an example, routes between VTAM Subarea 10 and any other VTAM or NCP in the diagram can only use explicit route numbers 0 through 7 because VTAM Subarea 10 does not have EXSA level software. Routes between VTAM Subarea 20, NCP Subarea 30 and NCP Subarea 40 can use explicit route numbers 0 through 16 because they all have EXSA level software and the NCPs have ERLIMIT=16 coded. Routes between VTAM Subarea 60 and all other VTAMs and NCPs in the diagram can only use explicit route numbers 0 through 7 because a node in the route, NCP Subarea 50, is only capable of using explicit route numbers 0 through 7 (ERLIMIT=8 is coded). If NCP Subarea 50 were pre-EXSA level code, the same restriction would exist-- routing between VTAM Subarea 60 and any other node could only use explicit route numbers 0 through 7.

12.6 SNI Considerations

Both SALIMIT and ERLIMIT are coded in the NCP BUILD definition statement. SALIMIT can also be coded in conjunction with different NETWORKs in a gateway-NCP and can have varying values to accomodate different subarea number requirements in various interconnected networks. ERLIMIT, however, must be set for a gateway-NCP as a whole. Therefore, if ER numbers above 7 are required for routing between the gateway-NCP and **ANY** interconnected node, ERLIMIT=16 must be coded for the gateway-NCP.

12.7 NCP Storage Considerations for SALIMIT and ERLIMIT

Following are two formulas which can be used to estimate NCP storage required for most of the routing control blocks. To use the formulas, the following values need to be determined:

- **NUMSA** - Number of subareas to which this NCP will route in this network. This is the sum of all DESTSAs on PATH statements plus the value on PATHEXT for the native network. For other networks in a GWNCP, this value can be determined by adding all DESTSAs on PATH statements and PATHEXT following the NETWORK statement.
- **SALIMIT** - The SALIMIT value from the BUILD or NETWORK statement.
- **ERLIMIT** - The ERLIMIT from the BUILD statement.
- **NUMTG** - Number of transmission groups in this network. This is the value of all the TGs (unique ADJSA,TGN pairs) on ERx keywords on PATH statements in the network (in the native net, or following NETWORK statements), plus the value of TGBXTRA for the network.

For gateway-NCPs, the calculations should be made for each network separately and then added together.

The formula for NCP V4R3 or NCP V5R2 is:

```
NUMSA*16 + 256*(7 + 2*NUMTG)
```

The formula for NCP V4R3.1 or NCP V5R2.1 is:

```
NUMSA*ERLIMIT*3 + (SALIMIT + 1) * (4 + (ERLIMIT/8)*(3+2*NUMTG))
```

The following figure shows how the storage requirements within an NCP are affected by the variables for ERLIMIT and SALIMIT.

For NCP V4R3/V5R2 Storage for routing tables is 3568 bytes.

For NCP V4R3.1/V5R2.1:

SALIMIT	ERLIMIT=8	ERLIMIT=16
255	3688	6352
511	7016	11984
1023	13672	23248
2047	26984	45776
4095	53608	90832
8191	106856	180944
16383	213352	361168
32767	426344	721616
65535	852328	1442512

Figure 12-6. NCP Storage for Routing Tables Example - 15 Subareas and 3 TGs

For NCP V4R3/V5R2 Storage for routing tables is 12832 bytes.

For NCP V4R3.1/V5R2.1:

SALIMIT	ERLIMIT=8	ERLIMIT=16
255	13232	25440
511	25264	48480
1023	49328	94560
2047	97456	186720
4095	193712	371040
8191	386224	739680
16383	771248	1476960
32767	1541296	2951520
65535	3081392	5900640

Figure 12-7. NCP Storage for Routing Tables Example - 50 Subareas and 20 TGs

Chapter 13. New VTAM/NCP Parameters

13.1 VTAM Start Options

GWSSCP= default=**YES**

A new option with VTAM V3R2 that specifies if the SSCP is gateway-capable or not, and allows use of the alias name translation facility.

NETID=

Is now a required option in VTAM V3R2. It contains the name of the network this host resides in, and should be unique within a set of interconnected networks.

SSCPNAME=

is now a required option in VTAM V3R2. It contains the name of the VTAM SSCP. In SNI configurations, the SSCPNAME must match the NAME on the corresponding GWNAU statement. It is also strongly recommended that SSCPNAME match the host CDRM name for improved usability and network management.

ITLIM=

This start option has been removed. VTAM V3R2 has restructured session initiation to more efficiently utilize private storage making ITLIM unnecessary.

VTAM INTERNAL TRACE (VIT)

There are three new options on the VTAM Internal Trace:

APPC (LU 6.2 API)
ESC (Execution Sequence Control)
NRM (Network Resource Management)

Information on these trace options can be found in the VTAM Diagnosis manual (LY30-5601).

13.2 VTAM VBUILD Statements

13.2.1 VBUILD TYPE=APPL operand changes

APPC= default=**NO**

Tells VTAM whether this application program can issue APPCCMD (LU 6.2) macro-instructions.

DMINWNL= default=**1**

Only valid if APPC=YES; defines the minimum number of contention-winner parallel sessions for this application program. The actual number is negotiated by VTAM. See Chapter 6 of "VTAM PROGRAMMING FOR LU 6.2" (SC30-3400).

DMINWNR = default = **1**

Only valid if APPC = YES; defines the minimum number of contention-winner parallel sessions for the remote LU. The actual number is negotiated by VTAM. See Chapter 6 of "VTAM PROGRAMMING FOR LU 6.2" (SC30-3400).

DSESLIM = default = **2**

Only valid if APPC = YES; defines the maximum number of sessions allowed between the application LU and remote LU using a particular mode name. When a CNOS (Change Number of Sessions) request for non-zero session limits is received, VTAM generates a negotiated CNOS reply for session limits which is the lessor of the requested limit (SESSLIM) or the value specified for DSESLIM. Note that the DSESLIM value is always greater than or equal to the combined values of DMINWNR and DMINWNL. See Chapter 6 of "VTAM PROGRAMMING FOR LU 6.2" (SC30-3400).

DRESPL = default = **NALLOW**

Only valid if APPC = YES; defines whether this application program will accept a request that makes it the LU that initiates session DEACTIVATION. Unless DRESPL = ALLOW, VTAM will always negotiate the session deactivation role to be that of the remote LU (Source). See Chapter 6 of "VTAM PROGRAMMING FOR LU 6.2" (SC30-3400).

DDRAINL = default = **NALLOW**

Only valid if APPC = YES; specifies whether this application will accept a request asking that it drain waiting allocation requests on its side. See Chapter 6 of "VTAM PROGRAMMING FOR LU 6.2" (SC30-3400).

AUTOSES = default = **0**

Only valid if APPC = YES; specifies the maximum number of contention-winner sessions to activate automatically. New sessions beyond the AUTOSES value can be activated in response to allocation requests. See Chapter 6 of "VTAM PROGRAMMING FOR LU 6.2" (SC30-3400).

LMDENT = default = **19**

Only valid if APPC = YES. VTAM places the remote LUs defined to this local application program in a set of definitions known to the ACB. VTAM then uses an Open-Hashing technique to locate specific LU definitions based on their LU names. LMDENT allows tuning the size of the HASH TABLE based on the number of LUs. A method of determining this value can be found in Chapter 5 of "VTAM INSTALLATION & RESOURCE DEFINITION"

(SC23-0111).

EAS =

The VTAM V3R2 maximum number of concurrent sessions this application can have with other LUs has increased to 65,535. This value tells VTAM the amount of storage to allocate for an FMCB directory table. Since FMCBs represent sessions, VTAM can locate the FMCBs more efficiently by using this hashing table. If the table is not large enough, then the FMCBs are chained thereby increasing the path length and adversely affecting performance. Too large a table wastes VTAM storage. A recommended value for EAS is slightly higher than the peak period number of concurrent users. Note that for TSO you should specify EAS = 1.

PARSESS =

Coding YES informs VTAM that this application program may participate in multiple concurrent (parallel) sessions with another application program. The default in VTAM V3R2 depends on the setting of the APPC = operand. If APPC = NO, the PARSESS default is NO; if APPC = YES, the default is YES.

13.2.2 VBUILD TYPE = CA definition changes for CTCA support

MAXBFRU = default = 1

Coded on the GROUP or LINE definition statement, this operand is changed in VTAM V3R2 to specify the number of 4K pages allocated for read operations. If communicating with a pre-V3R2 VTAM, the MAXBFRU operand must specify the number of buffers allocated instead of the number of 4K pages. Note: This operand change also applies, with PTFs, to VTAM V3.1.0 and V3.1.1 in MVS environments. Check INFOSYS for a CTC Flash with applicable PTF numbers.

13.2.3 VBUILD TYPE = SWNET definition changes for SWITCHED MAJOR NODE

ANS = CONTINUE default = STOP

Coded on the PU definition statement, this operand specifies that an LU-LU session not be disrupted if the owning VTAM to NCP session is lost, and allows non-disruptive ownership takeover by another VTAM. Prior to VTAM V3R2, these functions only applied to non-switched SDLC devices.

DATMODE = default = HALF

Coded on the PU definition statement, this operand specifies whether the PU communicates in FULL or HALF DUPLEX data mode. This operand supports

PU Type 2.1 nodes if their XID exchange indicates Full-Duplex mode. Note: DATMODE is not valid for Subarea Links.

CPNAME=

Coded on the PU definition statement, this operand specifies a control-point name for dial-supported TYPE 2.1 peripheral nodes. This enhances the XID processing by allowing the CPNAME to be supplied instead of the IDNUM and IDBLK. You must identify the PU by coding CPNAME, or IDBLK and IDNUM, or both.

LOCADDR=

Coded on the LU definition statement, with VTAM V3R2 an address of 0 specifies this definition is for an independent LU in a TYPE 2.1 node.

RESSCB= default=**0**

Coded on the LU definition statement, this operand specifies the number of boundary session control blocks reserved by the NCP for an independent LU. These reserved SCBs are drawn from the number specified in the MAXSESS operand of the BUILD statement.

13.3 VM VSCS DTIGEN Statements

For VM VSCS (VTAM SNA Console Services) the following **DTIGEN** operands are changed or new:

DPXMTL=

The default size has been increased from 1724 bytes to 1948 bytes in VTAM V3R2. This is the buffer size for data sent to displays in line mode.

KPXMTL=

The default size has been increased from 256 bytes to 284 bytes in VTAM V3R2. This is the buffer size for data sent to keyboard terminals and printers.

RCVBFRL=

The default size has been increased from 256 bytes to 284 bytes in VTAM V3R2. This is the buffer size for data received from a VTAM LU.

STCHKTM= default=**0** (No Checking)

Specifies the time interval for checking the VSCS storage pool for available segments (all its blocks are on the free queue).

STRELTM= default=**0**
Specifies the time interval between releasing available segments of storage.

Three new trace operands **(EXTRACE, FRTRACE, GETRACE)** and two changed trace operands **(VTTRACE, TRASIZE)** are available in VTAM V3R2. Information on these DTIGEN trace operands can be found in LY30-5601 'VTAM DIAGNOSIS'.

13.4 Logon Manager

With VTAM V3R2, TPF V2R4 may be defined as a TYPE 2.1 node. Since TPF terminals are generally dependent LUs, the **LOGON MANAGER** in VTAM V3R2 allows them logon access to applications residing in the TPF host. Information on defining the **LOGON MANAGER** and TPF APPLICATIONS to VTAM can be found in Appendix F of the "VTAM V3R2 INSTALLATION & RESOURCE DEFINITION" manual (SC23-0111).

13.5 NCP Statements

13.5.1 On the BUILD definition statement,

the following keywords are new or changed in NCP V5R2:

ADDSESS= default=**0**
Specifies the number of LU-LU BSBs (Boundary Session Control Blocks) available to any independent LU in addition to those reserved for it by the RESSCB operand of the LU statement. If an independent LU exhausts all the BSBs reserved (in the BSBPOOL) by the RESSCB operand, any additional BSBs needed will be allocated from the non-reserved area of the BSBPOOL created by the ADDSESS and AUXADDR keywords of the BUILD statement. If the number of active sessions equals the number specified in the RESSCB operand for that LU and there are no more session control blocks available in the unreserved portion of the BSBPOOL, any new BINDS for this LU will be rejected. Dependent LUs have a 2 session maximum (SSCP-LU & LU-LU), so 2 Session Control Blocks are always reserved in the BSBPOOL to prevent rejection. A peripheral TYPE 2.1 node requires a BSB for each independent LU having more than one session. This figure can initially be determined by including a control block for each independent LU involved in parallel sessions plus several more for PU TYPE 2.1 session overhead. Each ADDSESS specified requires 108 bytes of NCP storage.

AUXADDR= default=**0**
Specifies the number of additional addresses (one is

automatically assigned) that can be assigned to peripheral LUs establishing additional parallel sessions with the same partner. The single address assigned to the LU can be used for SLU sessions or any number of PLU sessions with different session partners. However, any additional PLU sessions with the same session partner require an additional PLU address which is obtained from the pool of addresses specified by the AUXADDR operand. The value of ADDSESS and AUXADDR produce the pool of session control blocks.
Each entry requires 24 bytes of storage.

MAXSESS = default = **5000**

Specifies the maximum number of LU-LU sessions that any single NCP boundary LU can have. Coding MAXSESS prevents any one independent LU from allocating all the control blocks and insures that all independent LUs have the chance to participate in sessions. This value should be initially determined based on the number of independent LUs, the potential number of LU sessions plus several more for PU TYPE 2.1 session overhead.
This operand does not require NCP storage but may impact the storage requirements for ADDSESS and/or RESSCB.

NAMTAB = default = **30**

Specifies the number of entries in the NETWORK NAMES TABLE. The value is determined by the total number of networks, SSCPs and Type 2.1 PUs with which this NCP may concurrently have sessions.
Each entry in the table requires 10 bytes of storage.

PATHEXT = default = **254**

Specifies the number of additional TRT (Transit Routing Table) rows generated for this network after the PATH definition statements are processed. This keyword allows the DYNAMIC PATH TABLE UPDATE to add routing definitions to a new destination subarea (DSA).
Each entry requires 16 bytes.

TGBXTRA =

Specifies the number of extra Transmission Group Control Blocks (TGB's) that can be associated via a DYNAMIC PATH TABLE change with a different adjacent subarea. If TGBXTRA is not coded, it defaults to the total number of subarea links and subarea channels defined in the NCP.
Each entry requires 580 bytes of NCP storage.

13.5.2 On the LUDRPOOL definition statement,

the following keyword is new in NCP V5R2:

NUMILU = default = **0**

Specifies the number of independent LU control blocks reserved for dynamic reconfiguration. See the LUDRPOOL description in the 'NCP Resource Definition Reference' (SC-30-3448) for information on including control-point to control-point session overhead in this value.

13.5.3 On the GROUP definition statement,

the following has been changed in NCP V5R2, V5R1 and V4R3:

Running NCP V5R2 or V5R1 in a 3745, you must define the channels as channel links on the **GROUP** and **LINE** definition statements. They are no longer defined in the BUILD statement. Both channel subarea links (NCP to subarea host) and channel peripheral links (NCP to TYPE 2.1 host) may be defined. The 3720 allows defining the channels as links on the GROUP and LINE statements or as channels on the BUILD statement. See Chapter 6 of the "NCP Resource Definition Guide" (SC30-3447) for more information and sample definitions.

13.5.4 On the LINE definition statement,

the following is new in NCP V5R2, V5R1, V4R3 and with PTF's V4R3:

ANS = **DELAY** default = **STOP**

This keyword allows session continuation for non-switched BSC 3270 devices in the event the owning SSCP is lost. ANS processing is delayed, allowing existing application sessions to continue.

13.5.5 On the PU definition statement,

the following is new in NCP V5R2, V5R1 and V4R3:

If you are defining a TYPE 2.1 PU, you do not have to code the following keywords:

MAXDATA =
MAXOUT =
MODULO =
PASSLIM =

Type 2.1 PUs create values for these keywords in their Format 3 XID exchange. Values coded for these definitions will be ignored with the exception of MODULO. If you choose to code a value for MODULO, the NCP uses the smaller of the generated XID or defined value.

13.5.6 On the LU definition statement,

the following keywords are new or changed in NCP V5R2, V5R1 and V4R3:

LOCADDR = 0
Specifies an independent LU. Only the controller storage size limits the number of LUs coded as LOCADDR = 0.

RESSCB = default = **0**
Specifies the number of boundary session control blocks reserved by the NCP for an independent LU. These reserved SCBs are from the number specified in the MAXSESS operand of the BUILD statement.

Chapter 14. VTAM V3R2 Software Compatibility Considerations

14.1 Software Compatibility Maintenance Requirements

When VTAM Version 3 Release 2 (V3R2) is installed on a host processor in a network, compatibility PTFs may be required on other software products. The VTAM, NCP and SSP Program Directories should be checked for compatibility information as should be the Preventative Service Planning (PSP) bucket. Other information may be found in VTAM and NCP QBUCKETs in INFOSYSTEM.

Following is a description of several different areas where maintenance on software products may be required.

14.1.1 VTAM -- Dynamic Reconfiguration

With VTAM V3R2 and NCP V4R3/NCP V5, dynamic reconfiguration has changed in that addresses originally used for genned resources that have been deleted can now be reused for dynamically added resources. When using dynamic reconfiguration, a DR ADD, DR DELETE, or DR MOVE should be performed from all VTAMs owning an NCP to avoid network address mismatches. Maintenance has been developed for VTAM Version 3 Release 1.1 systems so that the VTAM operator will receive message IST870I, indicating an address mismatch, if there is an address incompatibility between VTAM and NCP.

An address incompatibility could arise in the following situation:

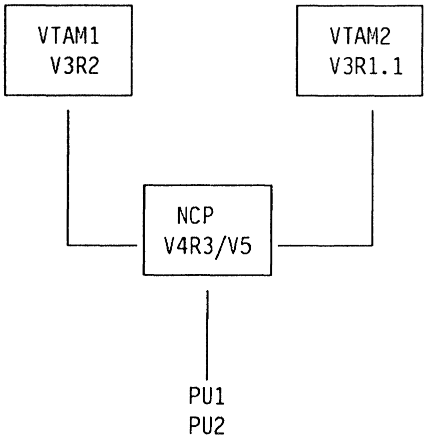


Figure 14-1. Dynamic Reconfiguration Example

In Figure 14-1, VTAM1 is the original owner of the NCP and PU1. PU1 is a resource that has been genned as part of the NCP and has been dynamically deleted. Subsequently, VTAM1 has dynamically added PU2 and the address that was originally used for PU1 has been used for PU2. VTAM1 has been canceled and VTAM2 has taken over the NCP and its resources.

With the compatibility PTF applied on VTAM2, message IST870I is displayed during NCP activation indicating that the address which VTAM2 is attempting to use for PU1 is in use for another resource, PU2. The needed dynamic delete of PU1 and add of PU2 can then be performed at VTAM2.

Maintenance is not required for VTAM V3R1.2 or VTAM V3R2 systems because the logic to issue message IST870I is in the base code. The maintenance to issue IST870I is not available for VTAMs earlier than V3R1.1 so address inconsistences may exist without VTAM message IST870I being issued. The result will be an "unusable" resource.

14.1.2 VTAM -- LOCADDR

Independent Logical Units (LUs) are genned in NCP V4R3 or NCP V5R2 using LOCADDR = 0 following a PU type 2 definition statement. VTAM systems earlier than V3R1.1 will fail NCP activation upon encountering an independent LU unless compatibility maintenance is applied.

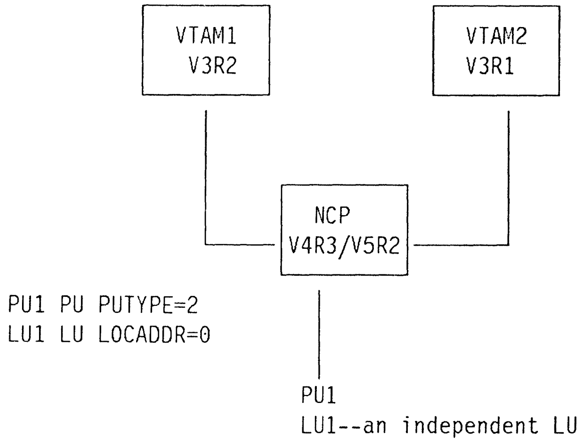


Figure 14-2. Independent LUs--LOCADDR = 0 Example

Using Figure 14-2 as an example, without the maintenance, VTAM2 would not be able to activate the NCP that contains an independent LU. With the maintenance applied, messages IST322I and IST323I should be displayed at VTAM2 and NCP activation will continue. The independent LU will only be usable if PU1 and LU1 are ACTIVE to VTAM1, which is at the proper VTAM level to support independent LUs. Maintenance is not required for VTAMs at level V3R1.1 or V3R1.2 as the logic to continue NCP activation upon encountering an independent LU is in the base code.

14.1.3 VTAM -- MAXLU

With NCP V4R3 and NCP V5, one of the changes in NCP definitions is that the MAXLU keyword on the PU definition statement is no longer valid if VIRTUAL=NO is coded or defaulted on the GROUP or LINE statement. When switched connections are made involving NCP V4R3 or V5 lines, VTAM's default value for MAXLU is used since there is no longer a value supplied by NCP. With VTAMs previous to V3R1.1 the default for MAXLU is 0 and, when used in conjunction with NCP V4R3/V5, causes switched line connection failures unless maintenance is applied to VTAM.

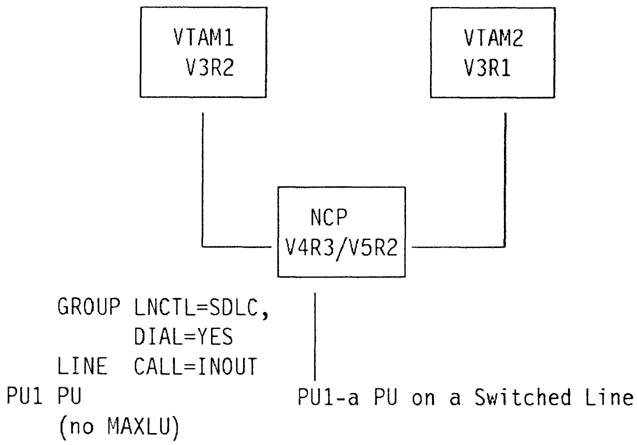


Figure 14-3. Switched Lines--MAXLU Example

Using Figure 14-3 as an example, without the maintenance, if VTAM2 activated the NCP, MAXLU would default to 0 and the switched connection of PU1 would fail. With the maintenance applied to VTAM2, the default for MAXLU is 255. Maintenance is not required for VTAM levels of V3R1.1 or V3R1.2 as the logic has been included in the base code.

14.1.4 VTAM -- SSCP Gateway Capability Detection

When VTAM V3R2 is installed in an environment where SNA Network Interconnection (SNI) is used, compatibility maintenance must be applied to VTAM V2R2, V3R1, or V3R1.1 gateways under certain circumstances. Without the maintenance applied, V3R2 VTAMs do not properly detect the gateway-capability of **same network** pre-V3R2 VTAMs; additionally, without the maintenance, pre-V3R2 VTAMs detect **all** V3R2 VTAMs as gateway-capable. With configurations similar to the following three examples, maintenance is required.

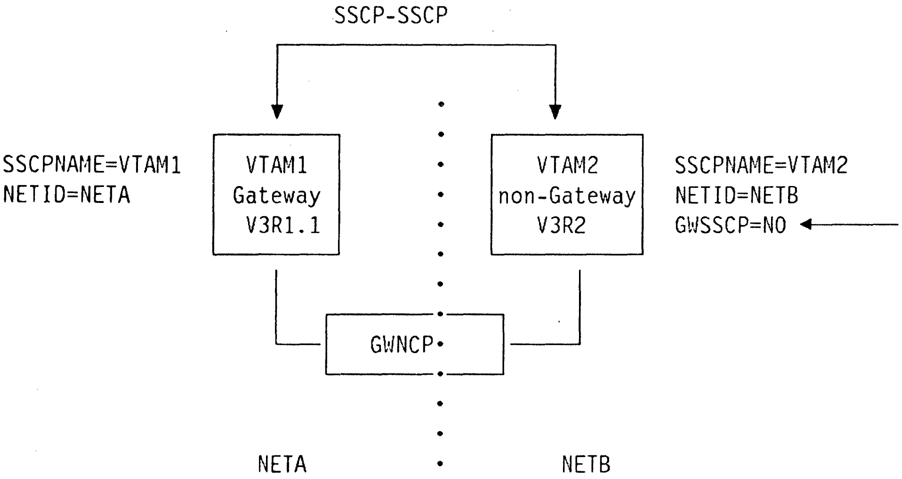


Figure 14-4. Configuration 1--Cross-network VTAM V3R2

In configurations similar to Figure 14-4, where a VTAM V3R2 non-gateway system (indicated by GWSSCP=NO) has a **cross-network** SSCP-to-SSCP session with a gateway VTAM V3R1.1 or earlier system, maintenance is required on the earlier systems. Without the maintenance, VTAM1 will understand VTAM2 to be gateway-capable since VTAM2 is started with NETID (required for V3R2). VTAM1 will therefore expect VTAM2 to provide SNI functions such as rerouting session initiation requests and participating in cross-network session setup. With the maintenance, VTAM1 will understand that VTAM2 is a non-gateway system.

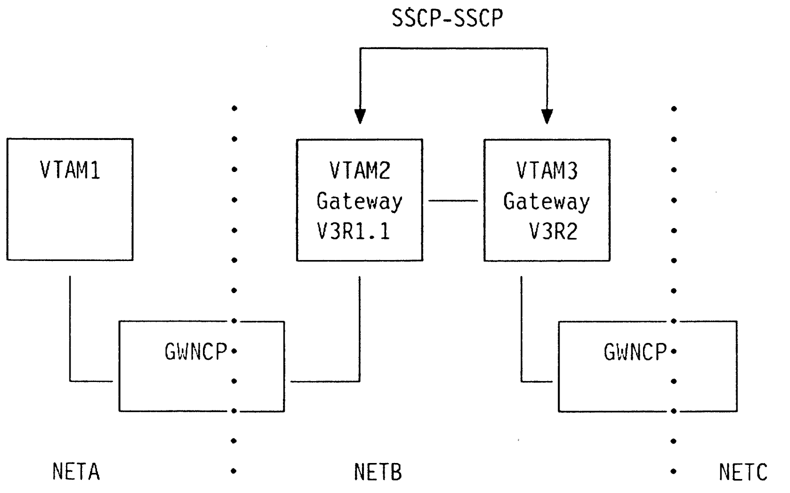


Figure 14-5. Configuration 2--Same-network gateway VTAM V3R2

In configurations similar to Figure 14-5, where a gateway VTAM V3R2 and gateway VTAM V3R1.1 or earlier in the **same network** have an SSCP-to-SSCP session, the VTAM V3R1.1 or earlier systems will require maintenance. In the figure, VTAM2 and VTAM3 have an SSCP to SSCP session. VTAM3 will not understand VTAM2 to be gateway-capable unless the maintenance is applied on VTAM2. The presence of NETID in VTAM2 is not sufficient for VTAM3 to properly understand VTAM2's gateway-capability unless the maintenance is applied. VTAM3 will not route session requests for resources known to exist in NETA to VTAM2 if VTAM3 does not understand VTAM2 to be gateway-capable.

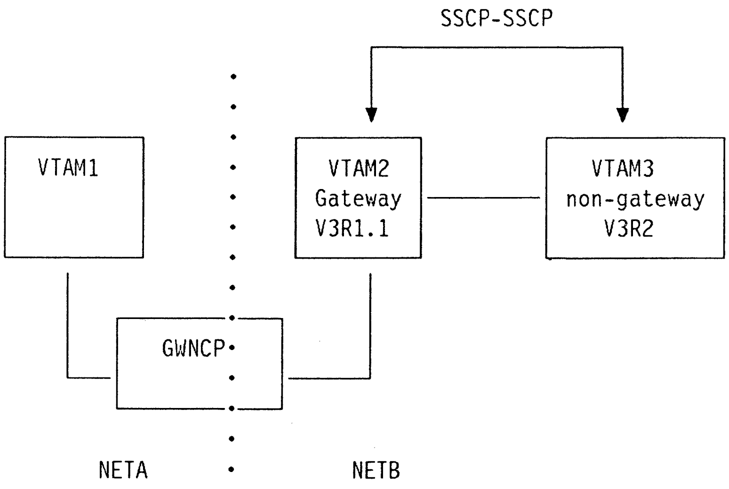


Figure 14-6. Configuration 3--Same-network non-gateway VTAM V3R2

In configurations similar to Figure 14-6, where a non-gateway VTAM V3R2 system has an SSCP-to-SSCP session with a VTAM V3R1.1 or earlier system in the **same network**, the maintenance is required. The reason is that VTAM3 in the above example will not understand VTAM2 to be gateway-capable unless the maintenance is applied. VTAM3 will not send session initiation requests to VTAM2 for resources known to exist in NETA if VTAM3 does not believe VTAM2 to be gateway-capable.

Refer to the SNI chapter in this bulletin for examples of problems that could occur without the needed maintenance.

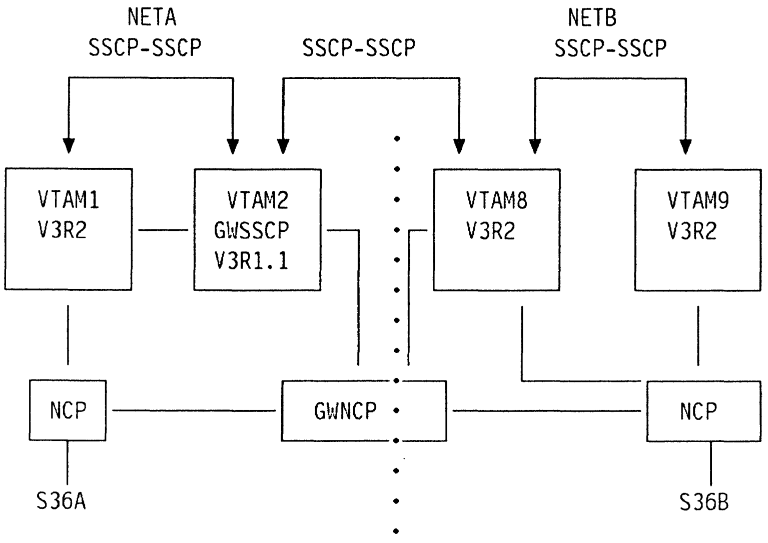
14.1.5 VTAM -- Modify CDRM and Owner Verification

With VTAM V3R2, two new functions are available to facilitate support of independent LUs. The first of these is a new operand, IMMEDIATE, on the VTAM "MODIFY CDRM" command. The IMMEDIATE operand allows the CDRM (owner) of a cross domain resource to be changed while sessions with that resource exist. In earlier levels of VTAM, the owner could not be changed until all existing sessions ended.

The second function, owner verification, allows the coding of VFYOWNER=YES|NO on a CDRSC. With VFYOWNER=NO coded (or defaulted), it is not necessary for an operator at a gateway VTAM to issue a MODIFY CDRM if the VTAM owner of an LU has changed. With levels of VTAM prior to V3R2, an operator at a gateway-VTAM needs to issue a MODIFY CDRM command if the CDRSCs are predefined and the owner has changed.

Maintenance is available for VTAM V2R1, V2R2, V3R1 and V3R1.1 to give them equivalent function. **Note:** VFYOWNER is not applicable to systems that do not support SNI function. With systems not providing SNI function (for example, VTAM V2R1), the maintenance only provides MODIFY CDRM,TYPE=IMMED support.

The SNI Chapter in this bulletin contains more information on owner verification and the MODIFY CDRM,TYPE=IMMED command.



VTAM 1:

```
    VBUILD TYPE=CDRSC
NETB NETWORK NETID=NETB
S36B CDRSC CDRM=VTAM9,VFYOWNER=NO
```

VTAM 2:

```
    VBUILD TYPE=CDRSC
NETB NETWORK NETID=NETB
S36B CDRSC CDRM=VTAM9,VFYOWNER=NO
```

Figure 14-7. VTAM V3R2 MODIFY CDRM,I and VFYOWNER

Using Figure 14-7 as an example, assume that S/36A is owned by VTAM1 and that S/36B is owned by VTAM9 and that a session exists between them. VTAM9 has gone down and S/36B has been taken over by VTAM8. The operator at VTAM1 must issue the modify CDRM command in order to change the owner of S/36B from VTAM9 to VTAM8 in order for S/36A to be able to establish another session with S/36B. The IMMEDIATE (available in VTAM V3R2 or in earlier VTAMs with the maintenance applied) function of the MODIFY CDRM commands allows the operator at VTAM1 to change the owner while there is an existing session.

With the appropriate maintenance applied at VTAM2, the gateway SSCP, the CDRSC for S/36B can be coded as shown specifying VFYOWNER = NO. In the case that S/36B has a different owner (now VTAM8) than is specified on the CDRM = VTAM9 operand, VTAM2 will allow sessions to be established between S/36A at VTAM1 and S/36B at VTAM8 because VFYOWNER = NO has been coded.

Additionally, with the maintenance applied at VTAM2, the operator could specify MODIFY CDRM,TYPE = IMMED to change the owner of S/36B from VTAM9 to VTAM8 while sessions existed between S/36B and S/36A. It would be necessary to change the owner of S/36B via the MODIFY CDRM command if VFYOWNER = YES were coded for S/36B at VTAM2. If the command were not issued, the session setup between S/36A and S/36B would fail.

14.1.6 VTAM -- USS, LOGMODE, INTERPRET and COS Tables

With VTAM V3R2, some changes have been made to VTAM tables in order to use them with the new function, Dynamic Table Replacement. USS and INTERPRET tables need to be reassembled using VTAM V3R2 libraries before they can be dynamically added, replaced or deleted. In some cases, once tables are assembled using VTAM V3R2 libraries, maintenance is required if the tables are then used on pre-V3R2 VTAM systems. Specifically, if LOGMODE or INTERPRET tables are assembled using VTAM V3R2 libraries, maintenance is required on VTAM V2R1, V2R2, V3R1, or V3R1.1 systems if the tables are distributed to these systems for use. With USS tables, VTAM V3R2 provides a FORMAT=V3R2|OLD parameter on the USSTAB Macro instruction. If OLD is specified, the USS table can be used on a pre-VTAM V3R2 system after assembly with V3R2 libraries. If COS tables are assembled using V3R2 libraries and subsequently distributed to pre-V3R2 VTAMs for use, maintenance is **NOT** required.

The chart below summarizes the VTAM table considerations: The acronym "DTR" stands for Dynamic Table Replacement.

Table 14-1. VTAM V3R2 Tables

	USS	**INTER-PRET**	**LOGMODE**	**COS**
Table must be reassembled to use DTR	YES FORMAT = V3R2	YES	NO	NO
If DTR not used, pre-V3R2 assembled table will work with V3R2	YES	YES	YES	YES
V3R2 assembled table will work on pre-V3R2	FORMAT = OLD	PTF	PTF	YES

14.1.7 DFP/XA -- IEWFETCH Load Error

In environments in which VTAM V3R2 runs in conjunction with DFP/XA, maintenance is **REQUIRED** on DFP/XA V1R2.1, V2.1, and V2R2 systems to avoid incorrect loading of tables used by VTAM. Maintenance is not required on DFP/XA V1R2.3 and DFP/XA V3 systems because the maintenance is in the base code.

14.1.8 NCP V3 & NCP V2 -- Activation By VTAM V3R2

Any NCP V3 or V2 which is activated by a VTAM V3R2 host requires maintenance. The maintenance is **REQUIRED** on NCP V3 or V2 or activation of the NCP by a VTAM V3R2 host will fail. The maintenance allows NCP to correctly calculate the length of one of the control vectors exchanged between it and VTAM V3R2.

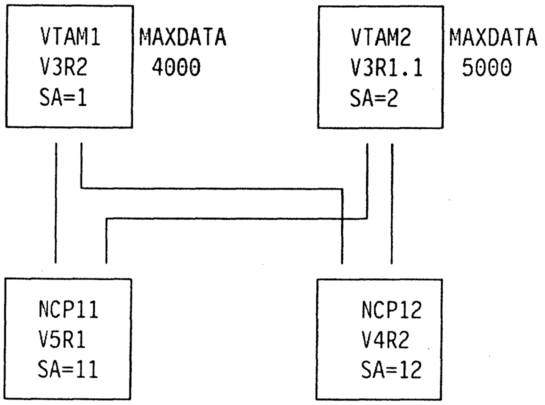
14.1.9 NCP & VTAM -- XID Changes

With VTAM V3R2, there are changes in the way that VTAM determines the largest PIU it can send to NCP. VTAM V3R2 determines the largest PIU that NCP can receive based on the value coded for MAXDATA on the PCCU definition statement, or based on a field, XIDMXPIU, in the XID sent to VTAM by NCP. XIDMXPIU is a value calculated by NCP and based on values coded on certain NCP definition statements. VTAM V3R2, VTAM V3R1.2, and VTAM V3R1.1 with PTF UY90127/UY90128 applied (necessary for 3745 support) use the smaller value of MAXDATA or XIDMXPIU to determine the largest size PIU that can be sent to NCP.

With VTAMs prior to VTAM V3R2, V3R1.2, or V3R1.1 with PTFs, VTAM only uses the value coded on MAXDATA. Because of the way that VTAM V3R2 determines the maximum PIU size that NCP can receive, channel attached NCPs may require maintenance.

Using Figure 14-8 on page 14-12 as an example, NCP12 (NCP V4R2 without maintenance described by APAR IR76227) sets the field, XIDMXPIU, to the lowest value determined by (MAXBFRU X UNITSZ) - BFRPAD on all the HOST statements. In the case shown above, there are two host statements, one for VTAM1 where the calculated value is 4000 and one for VTAM2 where the calculated value is 5000. NCP12 sends the lower value, 4000, in the XID to both VTAM1 and VTAM2. With VTAMs prior to V3R2, V3R1.2, or V3R1.1 without maintenance, VTAM1 would have used the MAXDATA value of 4000 and VTAM2 would have used the MAXDATA value of 5000 to determine maximum PIU size, ignoring the value NCP sent in. With VTAM V3R2, V3R1.2, or VTAM V3R1.1 with the maintenance, VTAM uses the lower value of XIDMXPIU or MAXDATA. With the above configuration, VTAM2 would use a maximum PIU size of 4000 rather than 5000 once VTAM V3R2/V3R1.2 is installed or the maintenance is applied to V3R1.1. VTAM V3R2/V3R1.2 or VTAM V3R1.1 with the maintenance do not pass a PIU larger than the lower of MAXDATA and XIDMXPIU out to the NCP.

Additionally, there is maintenance available for NCP V2, V3, and V4, in order for NCP to send in a large value. With a large value in the XIDMXPIU, the MAXDATA value will probably be smaller and this will be the value used to determine maximum PIU size.



NCP11:

```
PCCU ..
   MAXDATA=4000
   SUBAREA=1

PCCU ..
   MAXDATA=5000
   SUBAREA=2

HOST ..
   MAXBFRU=8
   UNITSZ=500
   BFRPAD=0
   SUBAREA=1

HOST ..
   MAXBFRU=10
   UNITSZ=500
   BFRPAD=0
   SUBAREA=2
```

NCP12:

```
PCCU ..
   MAXDATA=4000
   SUBAREA=1

PCCU ..
   MAXDATA=5000
   SUBAREA=2

HOST ..
   MAXBFRU=8
   UNITSZ=500
   BFRPAD=0
   SUBAREA=1

HOST ..
   MAXBFRU=10
   UNITSZ=500
   BFRPAD=0
   SUBAREA=2
```

Figure 14-8. VTAM/NCP XID Example

NCP V4R3 and V5 use a different calculation to determine what value to supply in the XIDMXPIU, depending on how the channel adapters are coded. A summary of the different ways NCP calculates XIDMXPIU is as follows:

NCP 4.2 and previous WITHOUT the PTF

The value for XIDMXPIU is derived at NDF generation time by taking the lowest of MAXBFRU * UNITSZ - BFRPAD coded on all the HOST definition statements.

NCP 4.2 and previous with the PTF applied

The value for XIDMXPIU is derived from the following formula.
(254 * NCP Buffer Size) - 18

NCP V4R3 and V5

There are two ways to calculate the XIDMXPIU value depending on how the channel adapter is defined.

1. If the channel adapter is defined on the BUILD statement:
   (254 * NCP Buffer Size) - 18

2. If the channel adapter is defined on the GROUP or LINE statement:
   TRANSFR * NCP Buffer Size

The maintenance should be applied to NCP V4R2 and earlier if:

- NCP is channel attached to VTAM V3R1.1 with UY90127 or UY90128
- NCP is channel attached to VTAM V3R2 or VTAM V3R1.2

  AND

- The largest PIU destined for the NCP is larger than the lowest coding of MAXBFRU * UNITSZ on **any** HOST statement.

14.1.10 IMS/DC Assembly Error

Without the needed maintenance on IMS, an assembly of IMS with VTAM V3R2 fails. (Note: with IMS V2R1 and V2R2, the module that fails assembly, DFSCVMR0, is automatically assembled at GEN time). The maintenance is required for IMS V2 as well as for V1R3.

14.1.11 NPM R3

NPM R3 requires maintenance in order to support NCP V4R3 and/or NCP V5. NPM R2 does **NOT** support VTAM V3R2 or NCP V4R3/V5.

14.1.12 Netview R2

There are a number of PTFs available to fix Netview R2 problems that were encountered in a VTAM V3R2 environment.

14.1.13 VTAM V2R1 OS/VS1

VTAM V2R1 for OS/VS1 needs a PTF in order to be able to communicate with resources on a VTAM V3R2 system, or with resources on other VTAM systems when the sessions set up through a gateway VTAM V3R2 system.

14.1.14 Other Products

Maintenance is needed on other products to fix problems encountered when running in a VTAM V3R2 environment. A list of products known to require maintenance when used with VTAM V3R2 follows:

1. S/36 Release 5.1
2. Series/1 RPS Releases 7.1 and 7.2
3. CICS V1R7.1
4. APPC/PC
5. S/38 Release 8.0

Installations should check the PSP buckets, program directories and other available sources for information regarding other VTAM application programs or T2.1 nodes that may require compatibility maintenance.

14.2 APAR List

The following table provides APAR information concerning the software maintenance described above.

Table 14-2. APAR LIST -- Part 1

PRODUCT	DESCRIPTION	VERSION	APAR NUMBER
VTAM	Dynamic Reconfiguration	• MVS V3R1.1 • MVS/XA V3R1.1 • VM V3R1.1	• OY08126 • OY08303 • VM28624
VTAM	LOCADDR	• MVS V3R1 • MVS/XA V3R1 • VM V3R1 • VSE V3R1	• OY01439 • OZ92295 • VM28356 • DY36768
VTAM	MAXLU	• MVS V3R1 • MVS/XA V3R1 • VM V3R1 • VSE V3R1	• OY11630 • OY11629 • VM28356 • DY36768
VTAM	Gateway-Capability	• MVS V2R2 • MVS V3R1 & V3R1.1 • MVS/XA V3R1 & V3R1.1 • VM V3R1.1	• OY06039 • OY06048 • OY06047 • VM27983
VTAM	MODIFY CDRM,I	• MVS V2R1& V2R2 • MVS V3R1 & V3R1.1 • MVS/XA V3R1 & V3R1.1 • VM V3R1 & V3R1.1 • VSE V2R1 • VSE V3R1	• OY07559 • OY07558 • OY07557 • VM28638 • DY36867 • DY36868
VTAM	V3R2 Assembled Tables	• MVS V2R1 & V2R2 • MVS V3R1 & V3R1.1 • MVS/XA V3R1 & V3R1.1 • VM V3R1 & V3R1.1 • VSE V2R1 • VSE V3R1	• OY10716 • OY10714 • OY10570 • VM29925 • DY37218 • DY37219

Table 14-3. APAR LIST--Part 2

PRODUCT	DESCRIPTION	VERSION	APAR NUMBER
DFP/XA	V3R2 Table Load	• V1R2.1, V2R1 & V2R3	• OY04556
NCP	VTAM V3R2 Control Vector	• V3 • V2 3705 • V2 3725	• IR73425 • IR76632 • IR76633
NCP	VTAM V3R2 XID Changes	• V4 • Subsets • V3 • V2 3705 • V2 3725	• IR76227 • IR76325 • IR76328 • IR76326 • IR76327
IMS	IMS Assembly Error	• V2R2 • V1R3	• PL21802 • PL25343
NPM	NCP Compatibility	• V1R3 • V1R3	• OY11594 • OY11595
Netview	VTAM V3R2 Compatibility	• V1R2/XA • V1R2/370	• OY11799 • OY11956
VTAM	VTAM V3R2 Compatibility	• OS/VS1 V2R1	• OX31372
S/36	VTAM V3R2 Compatibility	• APPN	• S337080
Series/1	VTAM V3R2 Compatibility	• RPS R7.1 & R7.2 • RPS R7.1 & R7.2	• S110206 • S110359
CICS	VTAM V3R2 Compatibility	• V1R7.1	• PL18043
APPC/PC	VTAM V3R2 Compatibility		• IR76161
S/38	VTAM V3R2 Compatibility	• R 8.0	• MT45081

14.3 Software Compatibility Matrices

The following five charts list compatible software levels between VTAM and NCP, NCP and NCP, and SSP and NCP. There is a separate chart for MVS/VTAM, for VM/VTAM, and for VSE/VTAM.

LOAD, ACTIVATE, DUMP CAPABILITIES

MVS VTAM

NCP	V2R1	V2R2	V3R1	V3R1.1	V3R1.2	V3R2
V2/3705	YES	YES	YES	YES	N/A	YES
V2/3725	YES	YES	YES	YES	N/A	YES
V3/3705	YES	YES	YES	YES	N/A	YES
V3/3725	YES	YES	YES	YES	N/A	YES
V4R1	YES	YES	YES	YES	N/A	YES
VSE SUBSET	YES#	YES#	YES#	YES#	N/A	YES#
V4R2	YES	YES	YES	YES	N/A	YES
MVS/VM SUBSET	YES	YES	YES	PTF	N/A	YES
V4R2 FEATURE	YES	YES	YES	PTF	N/A	YES
SUBSET FEATURE	YES	YES	YES	PTF	N/A	YES
V4R3	LU/LU	LU/LU	PTF	PTF	N/A	YES
V4R3.1	LU/LU	LU/LU	PTF	PTF	N/A	PTF
V5R1	LU/LU	LU/LU	PTF+	PTF	N/A	YES
V5R2	LU/LU	LU/LU	PTF+	PTF	N/A	YES
V5R2.1	LU/LU	LU/LU	PTF+	PTF	N/A	PTF

#ACTIVATION ONLY: the NCP cannot be generated or loaded from the operating system for this VTAM. However, the NCP RRT and a copy of the NCP source may be moved to this operating system and VTAM is then capable of activating this NCP.

+3720 ONLY; FOR 3745, LU/LU

LU/LU: These VTAM levels do not support channel attachment with this level of NCP. LU-to-LU sessions can be established with NCP boundary resources over INN links to an NCP channel attached to these VTAM levels.

Figure 14-9. MVS VTAM / NCP Support Matrix

LOAD, ACTIVATE, DUMP CAPABILITIES

VM VTAM

NCP	V2R1	V2R2	V3R1	V3R1.1	V3R1.2	V3R2
V2/3705	N/A	N/A	YES#	YES#	YES#	YES#
V2/3725	N/A	N/A	YES#	YES#	YES#	YES#
V3/3705	N/A	N/A	YES	YES	YES	YES
V3/3725	N/A	N/A	YES	YES	YES	YES
V4R1	N/A	N/A	YES#	YES#	YES#	YES
VSE SUBSET	N/A	N/A	YES#	YES#	YES#	YES#
V4R2	N/A	N/A	YES	YES	YES	YES
MVS/VM SUBSET	N/A	N/A	YES	PTF	YES	YES
V4R2 FEATURE	N/A	N/A	YES#	YES#	YES#	YES#
SUBSET FEATURE	N/A	N/A	YES#	YES#	YES#	YES#
V4R3	N/A	N/A	PTF#	PTF	YES	YES
V4R3.1	N/A	N/A	PTF#	YES#	YES#	PTF#
V5R1	N/A	N/A	YES-	PTF+	PTF#	YES#
V5R2	N/A	N/A	PTF-	PTF+	PTF	YES
V5R2.1	N/A	N/A	PTF-	PTF-	PTF#	PTF#

#ACTIVATION ONLY: the NCP cannot be generated or loaded from the operating system for this VTAM. However, the NCP RRT and a copy of the NCP source may be moved to this operating system and VTAM is then capable of activating this NCP.

+3720 ONLY; FOR 3745, DATA HOST ONLY

-3720 ACTIVATION ONLY; for 3745, LU-to-LU sessions only: these VTAM levels do not support a channel attached 3745. LU-to-LU sessions can be established with NCP boundary resources over INN links to an NCP channel attached to these VTAM levels.

Figure 14-10. VM VTAM / NCP Support Matrix

LOAD, ACTIVATE, DUMP CAPABILITIES

	VSE VTAM					
NCP	V2R1	V2R2	V3R1	V3R1.1	V3R1.2	V3R2
V2/3705	YES	N/A	YES	N/A	YES	YES
V2/3725	NO	N/A	YES	N/A	YES	YES
V3/3705	YES	N/A	YES	N/A	YES	YES
V3/3725	NO	N/A	YES#	N/A	YES#	YES#
V4R1	YES	N/A	YES	N/A	YES	YES
VSE SUBSET	YES	N/A	YES	N/A	YES	YES
V4R2	NO	N/A	YES#	N/A	YES#	YES#
MVS/VM SUBSET	YES#	N/A	YES#	N/A	YES#	YES#
V4R2 FEATURE	NO	N/A	YES#	N/A	YES#	YES#
SUBSET FEATURE	YES#	N/A	YES#	N/A	YES#	YES#
V4R3	LU/LU	N/A	PTF	N/A	YES	YES
V4R3.1	LU/LU	N/A	PTF#	N/A	YES#	YES#
V5R1	LU/LU	N/A	PTF-	N/A	PTF-	YES#
V5R2	LU/LU	N/A	PTF=	N/A	PTF=	YES
V5R2.1	LU/LU	N/A	PTF-	N/A	PTF-	YES#

#ACTIVATION ONLY: the NCP cannot be generated or loaded from the operating system for this VTAM. However, the NCP RRT and a copy of the NCP source may be moved to this operating system and VTAM is then capable of activating this NCP.

=3720 ONLY; for 3745, DATA HOST ONLY

-3720 ACTIVATION ONLY; for 3745, DATA HOST ONLY

LU/LU: these VTAM levels do not support channel attachment with this level of NCP. LU-to-LU sessions can be established with NCP boundary resources over INN links to an NCP channel attached to these VTAM levels.

Figure 14-11. VSE VTAM / NCP Support Matrix

	NCP						
NCP	V4R2*	SUBSET*	V4R3	V4R3.1	V5R1	V5R2	V5R2.1
V2	YES	YES	YES	YES	YES	YES	YES
V3	YES	YES	YES	YES	YES	YES	YES
V4R1/R2	YES	YES	YES	YES	YES	YES	YES
SUBSET	YES	YES	YES	YES	YES	YES	YES
V4R2*	YES	YES	YES	YES	YES	YES	YES
SUBSET*	YES	YES	YES	YES	YES	YES	YES
V4R3	YES	YES	YES	YES	YES	YES	YES
V4R3.1	YES	YES	YES	YES	YES	YES	YES
V5R1	YES	YES	YES	YES	YES	YES	YES
V5R2	YES	YES	YES	YES	YES	YES	YES
V5R2.1	YES	YES	YES	YES	YES	YES	YES

*FEATURE

Figure 14-12. NCP / NCP Compatibility Matrix

SSP / NCP COMPATIBILITY MATRIX

	SSP				
NCP	V3R2	V3R2*	V3R3	V3R4	V3R4.1
V2/3705	LOAD	LOAD	LOAD	LOAD	LOAD
V2/3725	LOAD	LOAD	LOAD	LOAD	LOAD
V3/3705	YES	YES	YES	YES	YES
V3/3725	YES	YES	YES	YES	YES
V4R1	YES	YES	YES	YES	YES
V4R2	YES	YES	YES	YES	YES
SUBSET	YES	YES	YES	YES	YES
V4R2*		YES		YES	YES
SUBSET*		YES		YES	YES
V4R3				YES	YES
V4R3.1					YES
V5R1			YES	YES	YES
V5R2				YES	YES
V5R2.1					YES

*FEATURE
YES : Can generate, load, and dump
LOAD : Can load and dump (cannot generate)

Figure 14-13. SSP / NCP Compatibility Matrix

14.4 Compatibility Examples

14.4.1 Network with Mixed Levels of VTAM and NCP

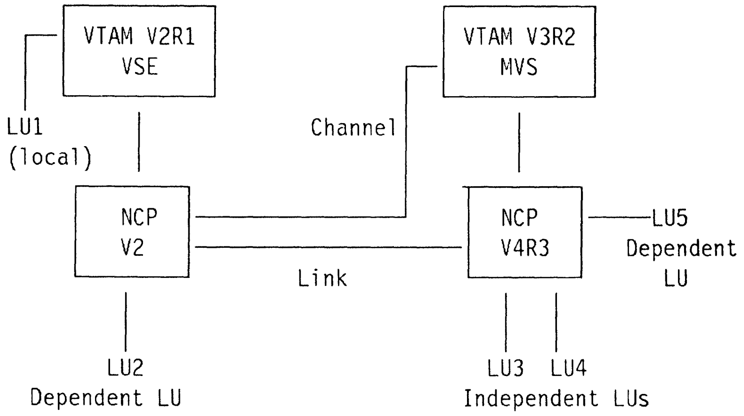


Figure 14-14. NCP V2 in a VTAM V3R2 Network

Using Figure 14-14 as an example, and the preceeding compatibility charts as a reference, MVS/VTAM V3R2 supports NCP V2; the maintenance detailed in Section 1.1.9 would be needed on NCP V2. If the NCP V2 system were to be generated, loaded, and dumped from the MVS/VTAM V3R2 system, two different SSP libraries would be needed. The level of SSP, V4R3 or V4R3.1, required for generating, loading, and dumping NCP V4R3, cannot be used to generate NCP V2; it can be used to load and dump NCP V2.

The independent LUs, LU3 and LU4, must be owned by VTAM V3R2 and attached to NCP V4R3 or NCP V5R2 in order to be supported as independent LUs. The independent LUs can have multiple and/or parallel sessions with each other, with an application on VTAM V3R2 or an application on VTAM V2R1. Multiple sessions have always been allowed between VTAM application programs so VTAM V3R2 is not necessary in order for the VSE system to support multiple sessions between its application programs and other application programs or independent logical units.

The dependent LU, LU5, can have sessions with applications at both VTAMs (LU-to-LU sessions are supported between VSE VTAM V2R1 and NCP V4R3). VSE VTAM V2R1 cannot activate NCP V4R3.

LU1 (locally attached to VTAM V2R1) can be in session with applications on the MVS VTAM V3R2 host. VTAM-to-VTAM communication is supported between VTAM V3R2 and any VTAM V2 or VTAM V3 system.

14.4.2 9370 VSE VTAM and NCP

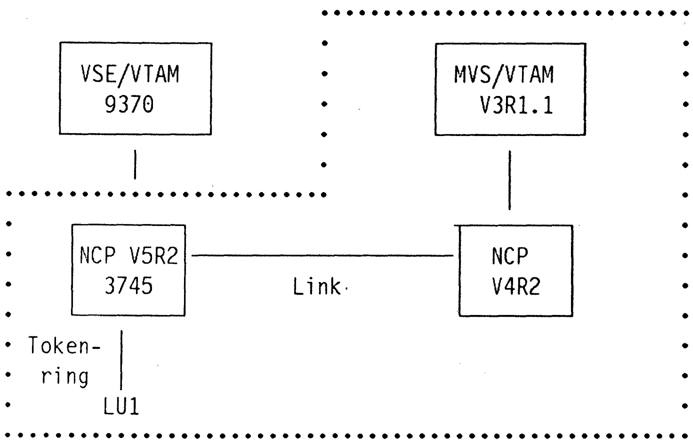


Figure 14-15. 9370 VSE VTAM and NCP

Using Figure 14-15 as an example, the compatibility charts can be used to analyze what levels of NCP and VTAM are needed in order for the 9370 to join the existing network. The dotted line signifies that MVS/VTAM V3R1.1 owns both NCPs. The 3745 is owned and loaded from the MVS VTAM V3R1.1 system. The 9370 could run VSE/VTAM V3R1, V3R1.2, or V3R2 and connect to the 3745 as a Data Host; running VTAM V3R2, the 9370 could own the NCP. Reference the preceding compatibility chart labeled VSE/VTAM where the Columns for V3R1, V3R1.2, and V3R2 intersect with the row for NCP V5R2. The "PTF" indicates that VSE/VTAM V3R1 and V3R1.2 need maintenance to support the NCP. The "=" indicates that VSE/VTAM V3R1 and V3R1.2 can channel attach to a 3745 running NCP V5R2 as a Data Host (activation of the NCP is **not** supported). This means that the VSE/VTAM can only define the NCP as a channel attached major node and cannot own any resources attached to the NCP. In order for VSE/VTAM to be an owner of the NCP and its resources, VTAM V3R2 must be installed.

NOTE: VSE VTAM only supports communication with token-ring attached devices at the level of VTAM V3R2. VM VTAM and MVS VTAM support communication with token-ring attached devices at the level of VTAM V3R1.1 or higher in conjunction with NCP V4R2 or higher.

NOTE: If the 9370 shown in Figure 14-15 is link attached to the NCP V5R2 using the 9370 Telecommunications Subsystem Controller, it cannot own (activate) the NCP V5R2. Ownership of an NCP through a Telecommunication Subsystem Controller (or 4300 ICA) is not supported by VTAM. In order for a 9370 or 4361 to own an NCP, the NCP must be channel attached or must be remotely attached to a channel attached NCP.

14.4.3 VM VTAM and Guest Systems

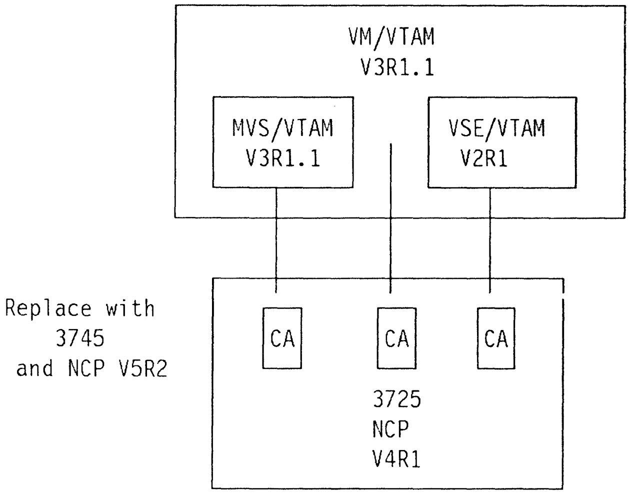


Figure 14-16. VM/VTAM and Guest Systems

Using Figure 14-16 as an example, all of the VTAM levels shown above support communication with NCP V4R1. Note that each different VTAM access method requires a separate channel adapter on the 3725 in order for VTAM to communicate with NCP either as an owner of the NCP or as a Data Host. If the 3725 were to be replaced with a 3745, the compatibility charts shown in Section 12.3 should be used to analyze compatible levels of VTAM/NCP and SSP. If each VTAM were upgraded to VTAM V3R2, then each VTAM system could own the NCP and its resources and support other new functions such as Multiple Load Modules on the 3745 Hard Disk, independent LUs, Dynamic Path Update, etc..

If the VSE/VTAM system were not upgraded, communication between it and the NCP could not be supported. (The Compatibility Charts indicate LU-LU sessions only). If the VSE/VTAM system were upgraded to VTAM V3R1, it could attach to the NCP as a Data Host or could communicate with NCP attached resources using a Virtual Channel-to-Channel (CTC) connection with one of the other VTAMs. CTC is not supported by VSE/VTAM V2.

If the VM/VTAM system were not upgraded, it could attach to the 3745 only as a Data Host.

If the MVS/VTAM system were not upgraded, it could generate, load, and dump the NCP V5R2 with the proper maintenance supplied.

NOTE: Refer to other Chapters of this bulletin for considerations for supporting new functions such as Dynamic Path Update, Dynamic Reconfiguration, etc.. in a mixed VTAM/NCP software environment.

Index

VTAM V3R2 AND NCP V4R3/V5R2
INSTALLATION CONSIDERATIONS

READER'S COMMENT FORM

GG66-3102-01

Please use this form to write down your comments about this manual, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, is deemed appropriate.

Note: IBM publications are not stocked at the location to which this form is addressed.

	Yes	No
■ Does the publication meet your needs?	□	□
■ Did you find the information:		
Accurate?	□	□
Easy to read and understand?	□	□
Organized for convenient use?	□	□
Complete?	□	□
■ What is your occupation?		

Comments:

If you want a reply, please give your name, mailing address, and date:

____

____

____

____

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

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

VTAM V3R2 AND NCP V4R3/V5R2
INSTALLATION CONSIDERATIONS

Reader's Comment Form

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
N. W. Gates
Washington Systems Center
800 North Frederick Avenue
Gaithersburg, MD 20879

Fold and tape | Please Do Not Staple | Fold and tape

IBM®

Printed in U.S.A.

VTAM V3R2 AND NCP V4R3/V5R2
INSTALLATION CONSIDERATIONS

READER'S COMMENT FORM

GG66-3102-01

Please use this form to write down your comments about this manual, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, is deemed appropriate.

Note: IBM publications are not stocked at the location to which this form is addressed.

	Yes	No
■ Does the publication meet your needs?	□	□
■ Did you find the information:		
Accurate?	□	□
Easy to read and understand?	□	□
Organized for convenient use?	□	□
Complete?	□	□
■ What is your occupation?		

Comments:

If you want a reply, please give your name, mailing address, and date:

____

____

____

____

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

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

Reader's Comment Form

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
N. W. Gates
Washington Systems Center
800 North Frederick Avenue
Gaithersburg, MD 20879

Fold and tape | Please Do Not Staple | Fold and tape

IBM®

Printed in U.S.A.

VTAM V3R2 AND NCP V4R3/V5R2 INSTALLATION CONSIDERATIONS

READER'S COMMENT FORM

GG66-3102-01

Please use this form to write down your comments about this manual, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, is deemed appropriate.

Note: IBM publications are not stocked at the location to which this form is addressed.

	Yes	No
■ Does the publication meet your needs?	□	□
■ Did you find the information:		
Accurate?	□	□
Easy to read and understand?	□	□
Organized for convenient use?	□	□
Complete?	□	□
■ What is your occupation?		

Comments:

If you want a reply, please give your name, mailing address, and date:

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

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

Reader's Comment Form

Fold and tape Please Do Not Staple Fold and tape

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
N. W. Gates
Washington Systems Center
800 North Frederick Avenue
Gaithersburg, MD 20879

Fold and tape Please Do Not Staple Fold and tape

IBM®

VTAM V3R2 AND NCP V4R3/V5R2
INSTALLATION CONSIDERATIONS

READER'S COMMENT FORM

GG66-3102-01

Please use this form to write down your comments about this manual, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, is deemed appropriate.

Note: IBM publications are not stocked at the location to which this form is addressed.

	Yes	No
■ Does the publication meet your needs?	□	□
■ Did you find the information:		
Accurate?	□	□
Easy to read and understand?	□	□
Organized for convenient use?	□	□
Complete?	□	□
■ What is your occupation?		

Comments:

If you want a reply, please give your name, mailing address, and date:

____

____

____

____

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

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

Reader's Comment Form

Fold and tape   **Please Do Not Staple**   Fold and tape

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS   PERMIT NO. 40   ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
N. W. Gates
Washington Systems Center
800 North Frederick Avenue
Gaithersburg, MD 20879

Fold and tape   **Please Do Not Staple**   Fold and tape

IBM®

Printed in U.S.A.

IBM

GG66-3102-01