

APPN Dependent LU Requester Architecture
=== > Draft document < ===

Document Number DLUR-3c

February 3, 1994

Edmond R. Tremblay

APPN Architecture
Dept. C74 / Bldg. 673
IBM Corporation
Research Triangle Park, NC
Phone: 919-254-5515
E-Mail: etremblay@vnet.ibm.com

Unclassified

Preface

This is the third draft of the Dependent LU Requester (DLUR) Architecture document. Any questions or comments relative to the contents of this document should be sent to the following Internet address: etremblay@vnet.ibm.com.

Acknowledgements

!
!
!
!
!

This document was initially authored by Dave Bryant (formerly IBM Corporation, currently 3Com Corporation) and Jim Fletcher (IBM Corporation). The multi-subnet chapter was initially authored by Doyle Horne (IBM Corporation). Many thanks to them for their efforts in providing much of the fundamental framework for this architecture.

Trademarks

The following terms, denoted by an asterisk (*) on their first occurrence in this document, are trademarks of the IBM Corporation in the United States and/or other countries:

- Advanced Peer-to-Peer Networking
- APPN

Revisions

*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*

The following revisions were made based on 7/93 review comments to Document Number DLUR-1. These changes are indicated by an asterisk (*) throughout the text.

- (global) more consistent usage of the terms CP-SVR pipe and CPSVRMGR session; also replacement of the term association with session or pipe - these changes are **NOT** marked by revision asterisks
- (2.2, 5.2) changed references to a single LU 6.2 session to a pair of LU 6.2 sessions
- (3.2.2, 6.2, 6.2.2) changed "non-genned terminal" to "Configuration Services XID Exit"
- (5.1, new 5.4.3) added dynamic DLUR determination
- (5.3) changed "byte 8 bit 8" to "bytes 8-9 bit 8" - these changes are **NOT** marked by revision asterisks
- (5.3) clarified DLUR option of initiating its conwinner CPSVRMGR session prior to receiving any requests for SSCP services from its PUs
- (5.4.1) clarified that whatever signaling information is sent by the DLUR will be echoed back by the DLUS
- (5.5.2.1, 6.1.2, 8.2.2, 13.1.1.1) added an XID CV to the GDS X'1500' encapsulating a REQACTPU for an internal PU

- * • (5.5.2.1, 5.5.2.3, 6.2.2, 6.2.3.1, 6.2.3.2) added internal activation signal to the stimuli triggering DLUR-initiated CP-SVR pipe activation
- * • (5.6.3, new 5.6.3.1, new 5.6.3.2) split DLUR-initiated abnormal SSCP-PU/SSCP-LU session deactivation into already existing downstream PU outage case and new receipt of REQDISCONT from downstream PU case
- * • (5.7.2) replaced description of suspending an active LU-LU session with description of the inability of that session to obtain SSCP services
- * • (new 5.8, new 6.5) added sections for additional error cases
- * • (new 6.1.3) added section on addressing per stage
- * • (6.2) clarified when the ACTPU is sent over the pipe (once the DLUS conwinner CPSVRMGR session has been established)
- * • (6.2.2) corrected statement that REQACTPU RU carries the XID image - the FID2 Encapsulation GDS variable which encapsulates the REQACTPU RU will carry the XID image
- * • (8.1.4) changed forwarding BIND to a dependent LU without an associated active CP-SVR pipe from a DLUR requirement to a DLUR product option
- * • (9.4) corrected in step 25 the reference to step 7 - should be step 10
- * • (13.1.3, D.1.2) removed parenthetical comments in the description of the REQACTPU Format field (PU identification is carried in the XID for both formats)

% The following revisions were made based on 9/93 review comments to Document
 % Number DLUR-2. These changes are indicated by an percent sign (%) throughout
 % the text.

- % • (5.6.3, 13.1.4, D.1.3) corrected DLUR processing upon receipt of a
 % REQDISCONT RU from its serviced PU
- % – postponed DLC deactivation to allow SSCP-PU and SSCP-LU deacti-
 % vation flows to be sent to and from the serviced PU
- % – created two new REQDACTPU cause values to indicate receipt of
 % REQDISCONT(normal) and REQDISCONT(immediate) so that DLUS
 % can proceed with orderly or forced PU deactivation as appropriate
- % • (8.1.5) corrected adaptive session pacing processing for locally attached
 % dependent LUs

! The following revisions were made based on review comments to Document
 ! Number DLUR-3. These changes are indicated by an exclamation point (!)
 ! throughout the text.

- ! • (5.5.2) correct statement that DLUR can deactivate CP-SVR pipe upon
 ! SSCP-PU/LU session rejection - DLUS always deactivates the pipe in this situ-
 ! ation
- ! • (5.5.2, 5.5.2.3, 5.6.2, new 5.6.2.2, 13.4.5, now D.2.6) separated CP-SVR pipe &
 ! SSCP-PU session rejection processing

- ! – created new section for processing DLUR/S capabilities mismatches,
! including cases for:
- ! – DLUR-DLUS pipe / DLUS detected mismatch
- ! – DLUR-DLUS pipe / DLUR detected mismatch
- ! – DLUS-DLUS pipe
- ! – DLUR-DLUR pipe
- ! – updated UNBIND sense codes for DLUR/S capabilities mismatches
- ! • (5.5.2.3) made use of LU 6.2 BIS protocols optional for CP-SVR pipe
- ! • (5.6.2.1.2, new 6.4.1, 13.1.7, now D.1.18) added limited DLUR ANS support
! (only ANS= STOP supported)
- ! • (5.6.3.2, 6.4, 6.4.2) clarified forced DACTPU processing (DLUS doesn't wait to
! send RSP(DACTPU) to SSCP), correcting two existing flows, one where the
! CP-SVR pipe remains active and the other where the CP-SVR pipe is deacti-
! vated
- ! • (5.7.3.2) added SSCP-PU session deactivation race (DACTPU/REQDACTPU)
! to the Single DLUS Race Conditions section
- ! • (new 5.8.3, 13.4.5, new D.2.5) added sense code for DLUR/S over TCP/IP
- ! • (6.1.2) added CPname as an identifier for a single DLUR-serviced internal PU
- ! • (6.4.2) specified sense codes associated with SSCP-PU and SSCP-LU traffic
! arriving at the DLUR when the session with the SSCP is inactive
- ! • (new 6.4.3) added section describing PU reactivation after deactivation,
! including effects on active LU-LU sessions
- ! • (8.1.2, 8.1.4, 8.1.5, now D.1.9) clarified DLUR CV X'0C' processing
- ! • (8.1.4) corrected DLUR BIND name substitution processing
- ! • (8.2, 11.1, 11.2.4, 11.3.2, 11.3.3, 11.3.5, 11.3.6, 12.1.6, 13.1.7, now D.1.18) added
! limited DLUR multi-subnet support; related impacts include:
- ! – requiring all EN DLURs to register their tail vectors with their DLUS, even
! if they are not in the same subnet
- ! – added a configuration restriction for the DLUS-DLUR path
- ! • (new 11.3.4, now 11.3.5.1, 12.1.6) added EN DLUR multi-subnet impact
- ! • (13.1.6, now D.1.17) removed subfield X'80' from CV X'46' when it is included
! in a GDS X'1500'
- ! • (now D.1.17) updated CV X'46' subfields:
- ! – removed format field from subfield X'91'
- ! – renamed DLC type SDLCLE (SDLC leased) to SDL CNS (SDLC non-
! switched) in subfield X'91'
- ! – added subfields X'92' and X'93' for DLC type INTPU
- ! • (appendix D) revised formats appendix

- !
 - !
 - !
 - !
- removed licensed material unrelated to DLUR/S from formats already in appendix
 - added unchanged formats related to DLUR/S which currently do not appear in open SNA formats document

Contents

	Chapter 1. Introduction	1-1
	1.1 Existing APPN Dependent LU Support Architecture	1-1
	1.2 Alternative Approach: Dependent LU Requester/Server Architecture	1-2
	Chapter 2. Requirements	2-1
	2.1 Objectives	2-1
	Chapter 3. Functional Overview	3-1
	3.1 Overview	3-1
	3.2 Assumptions And Dependencies	3-4
	3.2.1 DLUR Base/Option Set Dependencies	3-4
*	3.2.2 Configuration Services XID Exit Support	3-4
	3.2.3 Self-Defining Dependent LU Support	3-4
	3.2.4 Network Management	3-5
	3.3 Structure	3-6
	Chapter 4. Network Management Concerns	4-1
	4.1 Naming Of DLUS-Supported Resources	4-1
	Chapter 5. CP-SVR Pipe	5-1
	5.1 Overview	5-1
	5.2 CPSVRMGR Mode	5-1
	5.3 DLUR-Initiated CP-SVR Pipe Activation	5-2
	5.3.1 Pre-Defined DLUS Node Determination	5-2
	5.3.2 Dynamic DLUS Node Determination	5-3
	5.4 DLUS-Initiated CP-SVR Pipe Activation	5-4
	5.4.1 New PU Definition Parameters	5-4
	5.4.2 DLUS-Driven PU Activation	5-4
*	5.4.3 Dynamic DLUR Node Determination	5-5
	5.5 CP-SVR Pipe Activation	5-5
	5.5.1 CPSVRMGR Session Usage	5-5
	5.5.2 Accepting Or Rejecting A CPSVRMGR Session	5-6
	5.6 CP-SVR Pipe Deactivation	5-16
	5.6.1 Normal CP-SVR Pipe Deactivation	5-16
	5.6.2 Abnormal CP-SVR Pipe Deactivation	5-25
	5.6.3 Abnormal SSCP-PU/SSCP-LU Session Deactivation (DLUR-Initiated)	5-41
	5.7 Pipe And Session Failure Logic	5-51
	5.7.1 CPSVRMGR Session Recovery	5-51
	5.7.2 SSCP-PU And SSCP-LU Session Recovery	5-51
	5.7.3 Pipe And Session Activation/Recovery Race Conditions	5-51
*	5.8 Additional Error Cases	5-67
*	5.8.1 CP-SVR Pipe Limit Exceeded	5-68
*	5.8.2 GDS Other Than X'1500' Received After Receive_Encap Attach	5-68
!	5.8.3 CPSVRMGR Session Over Non-TCP/IP LEN Connection	5-68
	Chapter 6. SSCP-PU/SSCP-LU Session Encapsulation	6-1
	6.1 Encapsulation Addressing	6-1

!
*

!
!
!
*
*
*
*
*

6.1.1 Multiple PU Images	6-1
6.1.2 Internal PU Identification	6-2
6.1.3 Addressing Per Stage	6-3
6.2 SSCP-PU Session Activation	6-6
6.2.1 Pre-Defined PU Activation	6-7
6.2.2 Dynamic PU Activation	6-10
6.2.3 SSCP-PU Session Activation Race Conditions	6-13
6.3 SSCP-LU Session Activation	6-22
6.3.1 Activation Of Pre-Defined Dependent LUs	6-23
6.3.2 Dynamic Registration Of Dependent LUs	6-25
6.3.3 Dynamic Registration And Activation Of Dependent LUs	6-26
6.4 SSCP-PU/SSCP-LU Session Deactivation (DLUS-Initiated)	6-28
6.4.1 DLUR ANS Support	6-28
6.4.2 SSCP-PU/SSCP-LU Session Deactivation Flows	6-29
6.4.3 PU Activation After Deactivation	6-38
6.5 Additional Error Cases	6-39
6.5.1 Received GDS X'1500' Without CV X'60'	6-39
6.5.2 Received GDS X'1500' (ACTPU Encapsulated) With Missing Or Incomplete CV X'46'	6-39
6.5.3 Received GDS X'1500' (ACTPU Encapsulated) Without CV X'51'	6-39
Chapter 7. Unformatted Session Services Management	7-1
Chapter 8. LU-LU Session Routing	8-1
8.1 BIND Routing	8-1
8.1.1 ACTLU	8-1
8.1.2 +RSP(ACTLU)	8-2
8.1.3 INIT-SELF	8-2
8.1.4 BIND	8-2
8.1.5 +RSP(BIND)	8-3
8.1.6 LU-LU Session Deactivation	8-3
8.2 DLUR EN Tail Vector Registration	8-4
8.2.1 DLUR/S Capabilities Exchange	8-4
8.2.2 Initial EN Tail Vector Registration	8-5
8.2.3 EN Tail Vector Registration Updates	8-5
Chapter 9. Dependent LU Session Establishment Flows	9-1
9.1 USS Flows For SLU Init	9-2
9.2 SLU Init To APPN PLU	9-4
9.3 APPN PLU Init To A Dependent APPN SLU	9-7
9.4 USS Flows For SLU Init To A Subarea PLU	9-9
9.5 USS Flows For LU-LU Session Termination	9-14
Chapter 10. Dependent LU Session Awareness Management	10-1
10.1 Functional Overview	10-2
10.1.1 Pre-DLUR/S Dependent LU Session Awareness	10-2
10.1.2 DLUR/S Dependent LU Session Awareness	10-2
10.2 Dependent LU Session Awareness Flows	10-4
10.2.1 LU-LU Session Activation Flows	10-4
10.2.2 SSCP Takeover & Giveback Flows	10-11
10.2.3 Normal LU-LU Session Deactivation Flows	10-30

10.2.4	Abnormal LU-LU Session Deactivation Flows	10-37
Chapter 11.	Multi-Subnet DLUR/S	11-1
11.1	Overview	11-1
11.2	Multiple Subnet DLUR/S Configurations	11-1
11.2.1	Case 1: DLUS And DLUR In Different Subnets	11-1
11.2.2	Case 2: DLUS And PLU In Different Subnets	11-2
11.2.3	Case 3: PLU And DLUR In Different Subnets	11-2
11.2.4	Configuration Restrictions	11-3
11.3	Functional Description	11-4
11.3.1	Border Node Terminology	11-4
11.3.2	Multi-Subnet DLUR/S Indicators	11-4
11.3.3	Multi-Subnet CP-SVR Pipe Activation	11-5
11.3.4	Multi-Subnet EN DLUR Capabilities	11-5
11.3.5	Multi-Subnet DLUR/S Locate Searches	11-6
11.3.6	Multi-Subnet Functional Impact Summary	11-7
11.4	Multi-Subnet DLUR/S Flows	11-7
11.4.1	Multi-Subnet PLU-Initiated Searches	11-7
11.4.2	Multi-Subnet SLU-Initiated Searches	11-12
Chapter 12.	DLUR Option Set Content	12-1
12.1	DLUR Tower Requirements	12-1
12.1.1	CP-SVR Pipe	12-1
12.1.2	SSCP-PU & SSCP-LU Session Activation/Encapsulation	12-1
12.1.3	EN TGV Reporting	12-2
12.1.4	LU-LU BIND Processing	12-2
12.1.5	Session Awareness Reporting	12-2
12.1.6	Multi-Subnet Support	12-3
Chapter 13.	Format Changes	13-1
13.1	CP-SVR Pipe Format Changes	13-1
13.1.1	FID2 Encapsulation (X'1500') GDS Variable	13-1
13.1.2	ACTPU (Activate Physical Unit)	13-2
13.1.3	REQACTPU (Request ACTPU)	13-2
13.1.4	REQDACTPU (Request DACTPU)	13-3
13.1.5	Network Name (X'0E') Control Vector	13-3
13.1.6	TG Descriptor (X'46') Control Vector	13-3
13.1.7	DLUR/S Capabilities (X'51') Control Vector	13-4
13.2	Multi-Subnet Session Format Changes	13-5
13.2.1	Locate (X'12C4') GDS Variable	13-5
13.2.2	Find Resource (X'12CA') GDS Variable	13-5
13.2.3	Found Resource (X'12CB') GDS Variable	13-5
13.3	Dependent LU Session Awareness Format Changes	13-6
13.3.1	SESSST (Session Started)	13-6
13.3.2	SESEND (Session Ended)	13-6
13.3.3	RSP(ACTLU)	13-6
13.3.4	Session Information (X'2A') Control Vector	13-7
13.4	Sense Codes	13-8
13.4.1	ACTPU	13-8
13.4.2	LOCATE	13-8
13.4.3	REQACTPU	13-8

13.4.4 REQDACTPU	13-8
13.4.5 UNBIND	13-8
Appendix A. Definition of Terms	A-1
Appendix B. Alternate Solutions	B-1
B.1 LU-LU Session Route Calculation	B-1
Appendix C. Sample Network Configurations	C-1
C.1 Sample Network Configuration Number 1	C-1
C.1.1 Sample Network Interpretation	C-2
C.2 Sample Network Configuration Number 2	C-3
Appendix D. Formats For DLUR/S Architecture	D-1
D.1 Formats	D-1
D.1.1 ACTLU	D-1
D.1.2 ACTPU	D-1
D.1.3 REQACTPU	D-3
D.1.4 REQDACTPU	D-3
D.1.5 REQDISCONT	D-4
D.1.6 SESSEND	D-4
D.1.7 SESSST	D-5
D.1.8 RSP(ACTLU)	D-6
D.1.9 LU-LU Session Services Capabilities (X'0C') Control Vector	D-6
D.1.10 Network Name (X'0E') Control Vector	D-8
D.1.11 Local-Form Session Identifier (X'23') Control Vector	D-9
D.1.12 Security ID Control (X'25') Control Vector	D-10
D.1.13 Session Information (X'2A') Control Vector	D-10
D.1.14 Assign LU Characteristics (X'30') Control Vector	D-11
D.1.15 BIND Image (X'31') Control Vector	D-12
D.1.16 Extended SDLC Secondary Station (X'43') Control Vector	D-12
D.1.17 TG Descriptor (X'46') Control Vector	D-15
D.1.18 DLUR/S Capabilities (X'51') Control Vector	D-17
D.1.19 Locate (X'12C4') GDS Variable	D-18
D.1.20 Find Resource (X'12CA') GDS Variable	D-20
D.1.21 Found Resource (X'12CB') GDS Variable	D-23
D.1.22 FID2 Encapsulation (X'1500') GDS Variable	D-25
D.2 Sense Codes	D-26
D.2.1 0801	D-26
D.2.2 0806	D-26
D.2.3 082C	D-26
D.2.4 0852	D-26
D.2.5 0877	D-26
D.2.6 088E	D-31
D.2.7 0890	D-31
D.2.8 08A0	D-31
D.2.9 8019	D-31

Figures

	3-1.	Dependent LU Requester/Server Architecture	3-2
*	5-1.	DLUR-initiated CP-SVR pipe activation (DLUS known)	5-7
	5-2.	DLUS-initiated CP-SVR pipe activation	5-11
!	5-3.	DLUR-initiated, DLUS-rejected CP-SVR pipe activation	5-13
	5-4.	CP-SVR pipe deactivation - second UNBIND arrives before first is processed	5-17
	5-5.	CP-SVR pipe deactivation - UNBIND(conloser) arrives after UNBIND(conwinner) is processed	5-19
	5-6.	CP-SVR pipe deactivation - UNBIND(conwinner) arrives after UNBIND(conloser) is processed	5-22
!	5-7.	DLUR-initiated, DLUS-rejected CP-SVR pipe activation	5-27
!	5-8.	DLUS-initiated, DLUS-rejected CP-SVR pipe activation	5-29
!	5-9.	DLUR-initiated, DLUR-rejected CP-SVR pipe activation	5-32
!	5-10.	DLUS-initiated, DLUR-rejected CP-SVR pipe activation	5-36
!	5-11.	DLUS-initiated, DLUS-rejected CP-SVR pipe activation (DLUS-DLUS pipe)	5-38
!	5-12.	DLUR-initiated, DLUR-rejected CP-SVR pipe activation (DLUR-DLUR pipe)	5-40
!	5-13.	Downstream PU outage	5-43
%	5-14.	Receipt of REQDISCONT(normal) from downstream PU	5-46
!	5-15.	Receipt of REQDISCONT(immediate) from downstream PU	5-49
	5-16.	CP-SVR pipe override (rejected)	5-53
	5-17.	CP-SVR pipe deactivation/SSCP-PU session activation race - redrive pipe activation	5-56
	5-18.	CP-SVR pipe deactivation/SSCP-PU session activation race - do not redrive pipe activation	5-59
	5-19.	"Surprise" CP-SVR pipe activation	5-61
!	5-20.	SSCP-PU session deactivation race - DACTPU/REQDACTPU	5-65
*	6-1.	Addressing For SSCP-PU And SSCP-LU Sessions	6-4
*	6-2.	Addressing For LU-LU Sessions	6-5
	6-3.	Pre-defined PU activation	6-8
*	6-4.	Dynamic PU activation	6-10
*	6-5.	SSCP-PU session activation race - DLUR- and DLUS-initiated requests on same CP-SVR pipe	6-14
*	6-6.	SSCP-PU session activation race - DLUR- and DLUS-initiated requests on different CP-SVR pipes	6-17
*	6-7.	SSCP-PU session activation race - two DLUS-initiated requests	6-20
	6-8.	Activation of pre-defined LUs	6-23
	6-9.	Dynamic dependent LU activation	6-26
	6-10.	Normal SSCP-PU/SSCP-LU session deactivation	6-30
!	6-11.	Forced SSCP-PU/SSCP-LU session deactivation - CP-SVR pipe remains active	6-33
!	6-12.	Giveback SSCP-PU/SSCP-LU session deactivation - ANS = CONT	6-35
!	6-13.	Giveback SSCP-PU/SSCP-LU session deactivation - ANS = STOP	6-37
	7-1.	Unformatted Session Services Management	7-2
	9-1.	Unformatted Session Services SLU-initiated LU-LU session	9-2
	9-2.	Unformatted Session Services SLU-initiated LU-LU session to APPN PLU	9-5

9-3.	APPN PLU-initiated LU-LU session to a dependent SLU	9-7
9-4.	Unformatted Session Services SLU-initiated LU-LU session to subarea PLU	9-10
9-5.	Unformatted Session Services LU-LU session termination	9-14
10-1.	LU-LU session activation - SSCP-LU, LU-LU sessions routed through same BF-NCP	10-5
10-2.	LU-LU session activation - SSCP-LU, LU-LU sessions routed through different BF-NCPs (same SSCP)	10-6
10-3.	LU-LU session activation - SSCP-LU, LU-LU sessions routed through different BF-NCPs (different SSCPs)	10-8
10-4.	LU-LU session activation - LU-LU session not routed through a BF-NCP	10-10
10-5.	SSCP takeover - SSCP-LU, LU-LU sessions routed through same BF-NCP	10-13
10-6.	SSCP takeover & giveback - SSCP-LU, LU-LU sessions routed through same BF-NCP	10-17
10-7.	SSCP takeover - SSCP-LU, LU-LU sessions routed through different BF-NCPs / same takeover SSCP	10-23
10-8.	SSCP takeover - SSCP-LU, LU-LU sessions routed through different BF-NCPs / different takeover SSCPs	10-27
10-9.	LU-LU session deactivation - SSCP-LU, LU-LU sessions routed through same BF-NCP	10-31
10-10.	LU-LU session deactivation - SSCP-LU, LU-LU sessions routed through different BF-NCPs (same SSCP)	10-32
10-11.	LU-LU session deactivation - SSCP-LU, LU-LU sessions routed through different BF-NCPs (different SSCPs)	10-34
10-12.	LU-LU session deactivation - LU-LU session not routed through a BF-NCP	10-36
10-13.	LU-LU session deactivation (abnormal)	10-38
11-1.	DLUS, DLUR, PLU in different subnets / one BN per subnet	11-8
11-2.	PLU-initiated search for dependent LU when DLUS, DLUR and PLU are all in different subnets.	11-9
11-3.	PLU-initiated search for dependent LU support when DLUS, DLUR and PLU are all in different subnets (cont.).	11-10
11-4.	PLU-initiated search for dependent LU support when DLUS, DLUR and PLU are all in different subnets (cont.).	11-11
11-5.	DLUS, DLUR, PLU in different subnets / more than one BN per subnet	11-12
11-6.	SLU-initiated search when DLUS, DLUR and PLU are all in different subnets.	11-13
11-7.	SLU-initiated search when DLUS, DLUR and PLU are all in different subnets (continued).	11-14
C-1.	Sample network configuration number 1	C-1
C-2.	Sample Network Configuration Number 2	C-3

Chapter 1. Introduction

It is well known that while Advanced Peer-to-Peer Networking* (APPN*) protocols provide enhanced networking services for independent LUs, APPN protocols do not currently provide services for dependent LUs. As SNA products move from the current Subarea-based protocols to APPN-based protocols, it is increasingly more important to provide dependent LU services in APPN. This is particularly true when one considers the enormous customer investment in dependent LU types and the anticipated long migration period toward independent LU communications.

The following paper describes a method by which dependent LU support can be provided in APPN networks in a timely fashion with minimal implementation cost and minimal customer impact. It leverages the existing investment in dependent LU architecture while taking advantage of the dynamics available in APPN networks. The architecture described in this document is intended to provide the strategic method of dependent LU support in APPN networks.

Comments and suggestions on the architecture contained herein should be directed to the authors of this paper.

1.1 Existing APPN Dependent LU Support Architecture

The existing architecture extensions to APPN that provide support for dependent LUs are in several option sets (please refer to "SNA APPN Architecture Reference," SC30-3422, for more information):

- 1060 - Prerequisites for Session Services Extensions
- 1062 - Session Services Extensions CP Support
- 1063 - Session Services Extensions NNS Support
- 1064 - Session Services Extensions PLU Node Support
- 1065 - Session Services Extensions CP(SLU)(SSCP) Support

These option sets cover extensions to APPN to handle the additional session initiation types used by dependent LUs (i.e., Initiate or Queue, Initiate or Notify, Queue only, Autologon, and Third-party Initiate). The basic design premise of these option sets is that current SSCP support for dependent LUs will be moved out of the large mainframe host and distributed throughout the network in the peripheral T2.1 nodes. This approach moves dependent LU ownership and management out of the mainframe and into the peripheral T2.1 node. In addition, this approach requires the SSCP code to be duplicated in peripheral T2.1 nodes. And lastly, this approach moves the system definition responsibilities for dependent LUs to peripheral T2.1 nodes.

This is a fundamental change to the way customers design, define, and manage their existing dependent LUs. In addition, the cost of implementing SSCP function in the peripheral T2.1 nodes is prohibitive to most products implementing such architecture.

1.2 Alternative Approach: Dependent LU Requester/Server Architecture

Since SSCP function already resides in the existing VTAM Subarea product, this AWP describes an alternative solution that leverages this support to enable dependent LUs in an APPN network. In this solution, an LU6.2 session pipe is established between the T2.1 Node supporting dependent SLUs and the T2.1 Node providing SSCP function (initially either VTAM as pure NN or as Interchange Node). In this scenario, the T2.1 node supporting dependent SLUs is called the Dependent LU Requester (*DLUR* - option set 1067) and the T2.1 node providing SSCP function is known as the Dependent LU Server (*DLUS* - option set 1066). Once a pair of LU6.2 sessions have been brought up between the DLUR and DLUS, dependent LU flows (SSCP-PU and SSCP-LU sessions) are encapsulated over the LU6.2 pipe (known as the CP-SVR pipe) between the DLUR and DLUS SSCP. In this way, SSCP services are provided from VTAM without requiring the distribution of SSCP code or definition. The remainder of this document provides more detail on the architecture to provide such SSCP services to remote APPN dependent LUs.

Chapter 2. Requirements

2.1 Objectives

It is the objective of this architecture to:

- Leverage the current investments in dependent LU support to the advantage of both the implementer and its customers.
- Reduce the impact to customers in supporting dependent LUs in an APPN environment.
- Develop a solution to supporting dependent LUs in APPN that can and will be readily implemented by APPN products.

In addition to the objectives listed above, it is the intent of this architecture to be compatible with existing dependent LU support solutions. This will enable full support of dependent LUs including: USS Messages, Interpret Function, Autolog Support, and Session Queuing/Notification, SLU initiation, PLU initiation, third party initiation, etc. It is also a requirement to provide both Unformatted and Formatted Session Services support. Another important requirement will be the ability to handle SSCP takeover scenarios. Lastly, it is seen as a requirement to provide session routing independent of the SSCP providing dependent LU services. In this way, more optimal routes can be achieved between the PLU and SLU.

Chapter 3. Functional Overview

3.1 Overview

Dependent LU Requester/Server architecture consists of two separate APPN option sets:

- Dependent LU Server (*DLUS*) option set
- Dependent LU Requester (*DLUR*) option set

The intent of these two option sets is to provide dependent SLU support by establishing an LU6.2 session pipe (known as the CP-SVR pipe) between the requester and its server. Once this pipe has been established, the requester can obtain SSCP services for its dependent LUs from the server. This is done by multiplexing the necessary SSCP-PU and SSCP-LU sessions over the CP-SVR pipe and sending the dependent LU session initiation and management flows directly between the SSCP server and the PLU node.

In this way, the dependent LU appears to be located within the domain of the serving SSCP. Session initiation flows will be emulated from the SSCP server, but session BIND and data paths will be calculated directly between the dependent LU and its session partner. This path may or may not traverse the serving SSCP.

Since the serving SSCP will be a VTAM APPN Network Node, it will be able to provide both the standard SSCP-SSCP Subarea and Session Services Extensions APPN session initiation flows on behalf of the served dependent SLUs. When the PLU exists in the Subarea network, the server will use SSCP-SSCP flows to establish a session between the PLU and served SLU. When the PLU exists in the APPN network, the server will use Session Services Extensions flows to establish a session between the PLU and served SLU.

In this sense, DLUR/S architecture will not eliminate the need for Session Services Extensions within an APPN network. For many products, however, DLUR/S architecture will eliminate the need to implement Session Services Extensions. Products that only require dependent SLU support can implement the DLUR option set of this architecture and receive the Session Services Extensions support from their SSCP server via the CP-SVR pipe. As such, most products should only be required to provide the DLUR option set, while VTAM will provide both the DLUS option set and full Session Services Extensions support.

DLUR/S architecture is illustrated in the following figure:

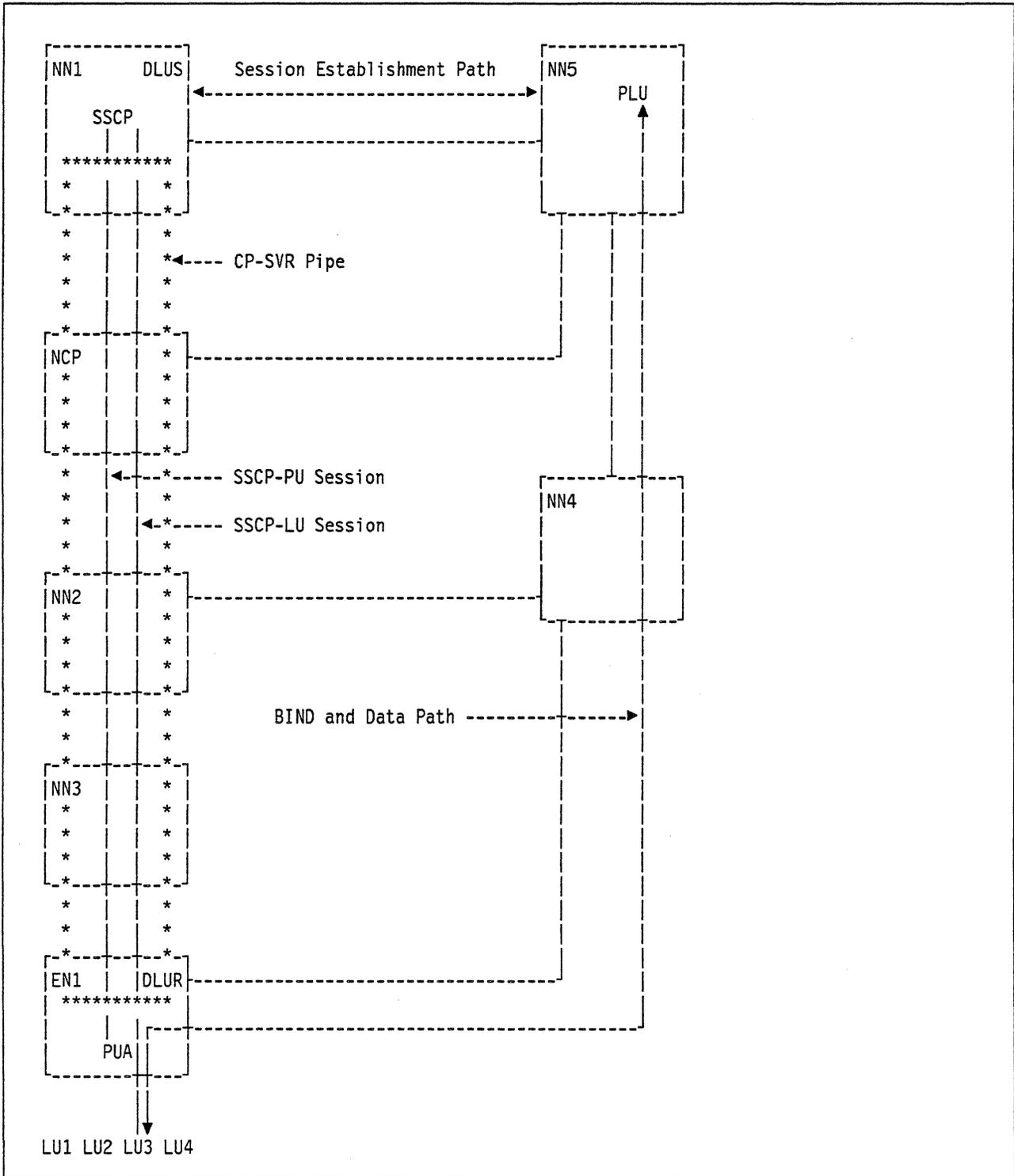


Figure 3-1. Dependent LU Requester/Server Architecture

In this figure, EN1 is supporting dependent SLUs: LU1, LU2, LU3, and LU4. EN1 supports these dependent SLUs with an internal PU2 known as PUA. A CP-SVR pipe is established between EN1 and

the Interchange Node, NN1. The SSCP-PU and SSCP-LU sessions are then encapsulated and multiplexed over the CP-SVR pipe. Session initiation flows for the dependent SLUs on EN1 are sent over the SSCP-LU session to NN1. NN1 then uses Session Services Extensions flows to initiate a session with the PLU located in the APPN network on NN5. NN5 must support the Session Services Extensions PLU functions for the specified session initiation type. Once the PLU has been located on NN5 and the necessary resource reservations, session queuing management, etc. have taken place, a BIND is issued in the APPN network directly from NN5 to EN1. In this scenario, the DLUS option set is located on NN1, the DLUR option set is located on EN1, and Session Services Extensions function is located at both NN1 and NN5.

3.2 Assumptions And Dependencies

This architecture assumes that the DLUS architecture will be implemented by VTAM in their APPN NN implementation. VTAM APPN NNs will have the capability to provide both APPN T2.1 and SSCP control point services.

Other products are assumed to implement the DLUR function to support either local or downstream (i.e., LAN-attached DOS workstations, directly attached PU2s, etc.) dependent SLUs.

Products that wish to provide dependent PLU support or dependent SLU support in absence of the DLUR/S architecture must use Session Services Extensions.

APPN nodes implementing the DLUR function for *local* dependent LUs must implement PU2.0 images. Because these PU2.0 images will be part of the logical network, they must be uniquely named in flows or network management displays that concern them. This implies that they will not be the CPname of the DLUR node or any other node in the network for that matter.

DLUS-served resources will be represented as switched LUs which are coupled to the CP-SVR pipe. In essence, the CP-SVR pipe is the LINE for the session, although internally it cannot be represented as a LINE, since the Control Point name will already be represented to DLUS as an application.

3.2.1 DLUR Base/Option Set Dependencies

The DLUR option set (1067) can be implemented on either Version 1 or Version 2 Base APPN, and on either NN or EN.

* 3.2.2 Configuration Services XID Exit Support

- * In order to dynamically create a PU definition for APPN dependent LUs, the DLUS node must provide the Configuration Services XID Exit support provided as part of the Dynamic Network Access function of VTAM Version 3 Release 4. This function will allow the DLUS node to dynamically create a PU definition for the DLUR PUs based upon information sent on the CP-SVR pipe.

LUs can also be dynamically defined using this facility, but the newer Self-Defining Dependent LU Support (see 3.2.3, "Self-Defining Dependent LU Support") should be used.

As a customer option, dependent LUs may also be pre-defined at the DLUS node; however, this will require that the corresponding PU also be pre-defined.

3.2.3 Self-Defining Dependent LU Support

VTAM Version 3 Release 4.1 introduces Self-Defining Dependent LU (SDDL) Support, which uses the Product Set Identifier (PSID) information sent on SSCP-PU sessions to dynamically create LU definitions for downstream dependent LUs. DLUR/S architecture will require this support in both the DLUS and the PU node if dynamic LU definition is desired. The PU node must provide PSID information to the DLUS, who will interpret it and dynamically create LU definitions for the T2.1 APPN dependent SLUs at or downstream from the DLUR node. If this function does not exist at one of the nodes, then pre-definition of dependent LUs will be required.

3.2.4 Network Management

Most of the network management concerns for dependent LU supported via the DLUR/S architecture will be met by existing NMVT support. Since the SSCP-PU flows remain intact and are simply encapsulated on the CP-SVR pipe, all dependent LU management of this type will be preserved. DLUR/S architecture, however, does present a unique management concern with respect to the CP-SVR pipe. Management Services components will be required to understand the relationship between SSCP-PU and SSCP-LU sessions, the CP-SVR pipe, and the real resources that support them. This architecture is being defined and will appear in a future draft of this document.

3.3 Structure

The DLUS and DLUR option sets described in this document constitute new components to the Meta-implementation reference model. These new components, DLUR and DLUS, are part of the Control Point. They have various interfaces to other components in the node which will be described in later versions of this document. These components also contain two architected LU 6.2 TPs for the purpose of encapsulating and de-encapsulating messages which flow on the SSCP-PU and SSCP-LU sessions within the CP-SVR pipe. These TPs are responsible for encapsulating and de-encapsulating messages as well as multiplexing and de-multiplexing the SSCP-PU/LU sessions to the proper node components.¹

¹ The description of node structure, DLUS and DLUR components are not an implementation requirement. They serve as a reference model through which the intent of the architecture can be more clearly explained and understood.

Chapter 4. Network Management Concerns

Dependent LUs supported through this architecture should be able to obtain the same level of network management support that currently exists today via the SSCP-PU session. The T2.0 PU node (either the DLUR node itself or some downstream PU) should be able to flow the existing NMVT support of any existing PU today. One special consideration will be required to address this SSCP-PU network management, i.e., including the PU name in the ACTPU request.

There are, also, special considerations with respect to managing the CP-SVR pipe and SSCP-PU and SSCP-LU session encapsulation. Network operators will need to know the relationship between the encapsulated sessions, CP-SVR pipes, and real network resources. The management architecture is being defined and will be included in a future draft of this document.

4.1 Naming Of DLUS-Supported Resources

The DLUR node learns the names of the PU and LU resources when the DLUS node activates them via ACTPU and ACTLU over the CP-SVR pipe. Currently, the LU name is included in the Network Name (X'0E') control vector type X'F3' (LU name) which is carried in the ACTLU request. However, the corresponding ACTPU request does not include a Network Name control vector. Therefore, ACTPU will be modified to optionally include a Network Name control vector type X'F1' (PU name), and type X'F1' will be modified to allow the PU name to be network-qualified.

The DLUS will always include a CV0E type F1 within the ACTPU RU when this RU is sent encapsulated in a GDS X'1500' to a DLUR-serviced PU. The PU name will be network-qualified when the PU NETID is different from the SSCP's NETID.

The DLUR will always strip the CV0E from the de-encapsulated ACTPU before passing it on to its PU, and it will save the CV0E information for network management.

Chapter 5. CP-SVR Pipe

5.1 Overview

CP-SVR pipe initiation is completely tied to PU activation. A DLUS or DLUR node will only activate a CP-SVR pipe when there exists some PU requiring activation and no CP-SVR pipe between the DLUR and the DLUS (or SSCP) supporting the PU already exists. A PU would require such activation under several circumstances including:

1. Receipt of an XID requesting ACTPU
2. Take over from loss of an active CP-SVR pipe
3. Explicit node operator command

In addition, the CP-SVR pipe may be initiated in one of two directions:

1. From the requester to the server
2. From the server to the requester

It is the intent of this architecture to support dynamic establishment of the CP-SVR pipe with an option to explicitly define the requester/server relationships if so desired. Due to the complexities of such dynamics, however, this version of the document requires that the DLUR and DLUS relationships be pre-defined.

- * Dynamic determination of a DLUS by a DLUR node as well as dynamic determination of a DLUR by a DLUS node will be addressed by a separate architecture extension.

The following sections will describe how the CP-SVR pipe is established in both the DLUS to DLUR and DLUR to DLUS direction.

5.2 CPSVRMGR Mode

- * The pair of LU 6.2 sessions which make up a CP-SVR pipe will use a new mode known as CPSVRMGR. CPSVCMG will not be used due to architecture assumptions regarding adjacency of CPSVCMG session partners. SNASVCMG will not be used to avoid contention for the session with other node functions such as network management and CNOS negotiation. An additional benefit of using a new mode (CPSVRMGR) is that these sessions can be uniquely identified by the CPSVRMGR mode name, thus allowing the node to provide special session management logic to these sessions as described in 5.5, "CP-SVR Pipe Activation" on page 5-5.

The CPSVRMGR mode will use the SNASVCMG COS. By using this default COS definition, DLUR nodes will not have to worry about matching COS definitions at the NNS. In this way, a DLUR EN can be supported by an existing base APPN NN.

5.3 DLUR-Initiated CP-SVR Pipe Activation

By definition, all dependent LUs require the services of an SSCP. This is done via SSCP-PU and SSCP-LU sessions. The SSCP-PU session flows between the SSCP and PU2 or T2.1 node providing local support for the dependent LU. This session is created when the SSCP sends an ACTPU RU to the PU2/T2.1 node. The SSCP-LU session is created when the SSCP sends an ACTLU over the SSCP-PU session to the dependent LU itself. Normally a T2.1 or PU2 device supporting dependent LUs is Boundary Function-attached to a Subarea node and is either provided the SSCP services upon activation or (in the case to T2.1 nodes) requests them via XID3 (this is reflected in the ACTPU suppression indicator: bytes 8-9 bit 8 of XID3).

DLUR nodes are APPN T2.1 nodes (EN or NN) which provide both local and downstream support for dependent LUs. Since DLUR nodes provide Boundary Function-like support for these LUs, each DLUR node must signal to the SSCP (or DLUS node in this case) when a PU (either local or downstream) requires SSCP services. For locally supported dependent LUs, the DLUR node requests SSCP services upon node activation. For downstream nodes supporting dependent LUs, the DLUR node waits for the downstream node to request them via XID. When a T2.1 or T2.0 PU exchanges XIDs (or equivalent) with a DLUR node, it signals via XID that it requests SSCP services (this is done by setting the ACTPU suppression indicator, bytes 8-9 bit 8 of XID3, to 0; for XID0 or SNRM without XID, ACTPU is always sent).

DLUR nodes request SSCP services from a DLUS node via the CP-SVR pipe and the REQACTPU RU. When a DLUR node detects the need for SSCP services (as described above), it activates a conwinner CPSVRMGR session (assuming one does not already exist) to a DLUS node and sends a REQACTPU RU. The DLUS node then responds by sending an ACTPU to the T2.0 PU/T2.1 supporting the dependent LUs via a conwinner CPSVRMGR session to the DLUR node. This creates an SSCP-PU session encapsulated in a CP-SVR pipe, enabling the DLUS node to provide SSCP services to the dependent LUs.

- * A DLUR optionally may establish a conwinner CPSVRMGR session with a DLUS prior to receiving any
- * requests for SSCP services from its PUs.

In order to establish a CP-SVR pipe with a DLUS node, the DLUR must either *know* or *determine* the DLUS node to be contacted. The following sections describe how these two scenarios are handled:

5.3.1 Pre-Defined DLUS Node Determination

A DLUS node is said to be *known* when this information is pre-defined (sysdef'd) at the DLUR node. Since the relationship is between PUs and SSCPs, this means that the DLUR would have to have some product-specific definition that gives the CPname of a DLUS node to be contacted when a given PU requires activation. This can be done in any fashion suitable to the DLUR implementing product (defaults, operator defined, etc.).

Once defined, the DLUR component will use this information (when necessary) to bring up the CP-SVR pipe to a DLUS node. When a given T2.0 PU/T2.1 node requires an ACTPU, the DLUR node will use the pre-defined information to determine the DLUS node to be contacted. It must then check to see if a conwinner CPSVRMGR session already exists with that DLUS node. If such a session does not already exist, the DLUR uses the Send_Encap_Msg_TP to issue an ALLOCATE verb with the DLUS's CPname as the partner LU name. This will trigger the normal APPN session initiation flows to occur (i.e., a search being issued, either broadcast or directed, for the DLUS node) with an eventual BIND message flowing between the DLUR and DLUS node. This session will use the CPSVRMGR mode.

If the CPSVRMGR session to the DLUS node fails for any reason (DLUS not active, routes not available, etc.), the dependent LUs will not be activated. Appropriate notification should be signalled at the DLUR node. If driven by the XID of a downstream dependent LU device, the link between the DLUR and the downstream device will remain active, but the DLUR must either redrive the DLUS request via some product-specific mechanism (backup DLUS definition, timer-driven retry, event-driven retry, operator-driven retry, etc.), or wait for a DLUS-driven PU activation as described in 5.4, “DLUS-Initiated CP-SVR Pipe Activation” on page 5-4.

5.3.2 Dynamic DLUS Node Determination

If no DLUS name is defined at the requester node, then the DLUR node must first find an appropriate DLUS node and then initiate a session to it as described in 5.3.1, “Pre-Defined DLUS Node Determination” on page 5-2. Dynamic determination of a DLUS node by a DLUR node will be addressed at a later date by a separate architecture extension.

5.4 DLUS-Initiated CP-SVR Pipe Activation

- In the normal case, dependent LUs are activated from the SSCP. In order to simulate the scenario when VTAM activates a PU, the DLUR/S architecture will also support dependent LU activation from a DLUS node. In this scenario, the DLUS node must first determine the DLUR node associated with a given dependent LU. If no conwinner CPSVRMGR session already exists with this node, it will activate one. It may then send an ACTPU RU over the conwinner CPSVRMGR session to the DLUR node, who will in
- * turn send the ACTPU to the appropriate node (downstream T2.0 PU/T2.1 or DLUR node itself). The
 - * DLUR will activate a conwinner CPSVRMGR session to the DLUS (if one is not already active) and send
 - * the RSP(ACTPU) to the DLUS. After this, the ACTLU RUs may flow over the SSCP-PU session
 - * encapsulated in the CP-SVR pipe to activate the dependent LUs.

When activating a pre-defined PU, however, the DLUS node must know where to send the ACTPU RU. If the dependent LU is located on or downstream from some DLUR node, the DLUS must know this. As such, DLUR-supported dependent LUs will require new system-defined parameters in the DLUS node. These new parameters will be as follows:

5.4.1 New PU Definition Parameters

- *DLUR node*

This parameter will identify the CPname of the DLUR node that supports the PU either locally or via some downstream connection. This implies that system administrators must coordinate the dependent LU definitions in a DLUS node with the CPname of the DLUR node supporting them.

- *PU TG Descriptor*

This parameter will identify the DLUR-to-PU connectivity. This parameter must also be coordinated between the DLUR and DLUS nodes. When the PU is locally supported on the DLUR node, this parameter will identify which PU image is associated with the DLUS PU definition. When the PU is located downstream from the DLUR node, this parameter will identify the TG providing connectivity between the DLUR and PU node. In the case of leased line connections, this description must provide the port and, potentially, the polling address of the downstream PU. In the case of a connection via some Shared Access Transport Facility (SATF), this parameter must provide the Port, MAC and SAP address of the downstream PU. In the case of switched callout line connections, the description will specify IDBLK and IDNUM values as well as dial digits.

A new group of subfields (X'91'-X'F0') are being defined for both subarea Extended DIALNO (to be included in CV X'69') and DLUR/S (to be included in CV X'46') in order to carry the descriptions (a.k.a. signaling information). These subfields are currently being architected and will be included in a future draft of this document.

- * Whatever signaling information is sent by the DLUR will be echoed back by the DLUS.

5.4.2 DLUS-Driven PU Activation

When a DLUR PU is activated at a DLUS node, the DLUS node will use the DLUR node specification in the PU definition to activate a CP-SVR pipe to the DLUR node. If no such pipe already exists, the DLUS node will initiate one by using the Send_Encap_Msg_TP to issue an ALLOCATE to the Receive_Encap_Msg_TP on the CP_LU of the DLUR node. This will result in a conwinner session being activated between the DLUS and DLUR node.

If the DLUR node is not available at the time the DLUS attempts to initiate a session to it, the PU activation request will fail. At some later time, when the DLUR becomes available, it may initiate a session to a server as described in 5.3, “DLUR-Initiated CP-SVR Pipe Activation” on page 5-2. Otherwise, some other product-specific mechanism will be required to re-attempt PU activation (i.e., Operator-driven retry, timer-based retry, Network Mgmt exits, etc.).

* 5.4.3 Dynamic DLUR Node Determination

- * If no DLUR name is defined at the server node, then the DLUS node must first find the appropriate DLUR
- * node and then initiate a session to it as described in 5.4.2, “DLUS-Driven PU Activation” on page 5-4.
- * Dynamic determination of a DLUR node by a DLUS node will be addressed at a later date by a separate
- * architecture extension.

5.5 CP-SVR Pipe Activation

The CP-SVR pipe will actually consist of two LU6.2 sessions between the CPs of the server and requester nodes. These sessions will be very similar to the CP-CP sessions between adjacent APPN Control Points. There will be one conwinner and one conloser session between the control points. CP-SVR sessions will, however, use a new mode name, CPSVRMGR, for reasons cited in 5.2, “CPSVRMGR Mode” on page 5-1. CP-SVR pipe activation will be triggered by the need to activate an SSCP-PU session. This can occur as the result of node activation, error recovery scenarios, XID exchange, or explicit Node Operator Facility interfaces.

The CP-SVR pipes between DLUS and DLUR nodes will exist between the Control Points of these two nodes. They are established when either the DLUS or DLUR node has something to send to the other node. This causes the DLUS or DLUR component to invoke the `Send_Encap_Msg_TP` to its partner. This TP will allocate a conwinner session to the partner CPname using the CPSVRMGR mode and will attach the `Receive_Encap_Msg_TP` at the other side.

Because the receipt of flows from one node will necessitate some response to the originator, any DLUS or DLUR node that receives an incoming conloser CPSVRMGR session must bring up a conwinner CPSVRMGR session to its partner. At this point, the CPSVRMGR (actually the encapsulated SSCP-PU) session may be accepted or rejected as described below.

5.5.1 CPSVRMGR Session Usage

Like the SNASVCMG and CPSVCMG modes already architected within the APPN specification, the DLUS and DLUR components will employ a one-way bracket conversation technique to communicate messages between the `Send_Encap_Msg_TP` and `Receive_Encap_Msg_TP`. This technique often facilitates better performance within a given implementation and will reduce the storage requirements for a DLUS node that serves many DLUR nodes. Since all products already use this technique on CPSVCMG sessions, it is assumed that the associated Attach/Detach overhead problems have already been internally resolved.

In addition, DLUS and DLUR nodes will not UNBIND CPSVRMGR sessions when no conversation is currently allocated over the session. In general, it is not recommended to BIND CPSVRMGR sessions over limited resources. DLUS and DLUR nodes should not UNBIND a CPSVRMGR session (across a limited resource or not) unless one of the session deactivation conditions specified within 5.6, “CP-SVR Pipe Deactivation” on page 5-16 is encountered.

5.5.2 Accepting Or Rejecting A CPSVRMGR Session

When a DLUR or DLUS node receives an incoming CPSVRMGR BIND from its partner node, it must determine if it accepts the session. Since CPSVRMGR sessions are established for the purpose of carrying SSCP-PU (and SSCP-LU) session traffic, nodes should accept or reject them on the basis of the SSCP to PU session. This session is initiated when either the REQACTPU or ACTPU (see 6.2, "SSCP-PU Session Activation" on page 6-6) RU flows over the CP-SVR pipe. For this reason, a DLUS or DLUR node receiving a CPSVRMGR BIND (assuming no other problems) should provide a positive BIND RSP. After which, an incoming Attach for the Receive_Encap_MSG_TP will be received and either a REQACTPU or ACTPU RU will flow. At this point the DLUS or DLUR node can determine if it accepts the SSCP to PU session. Since a given PU can only be owned by one SSCP, the session may be rejected for this reason. System-defined parameters or other product-specific restrictions may also be used to make this determination.

One new factor in making that determination will be a DLUR/S capabilities control vector. In this initial release the control vector will carry an indication that this is a first release DLUS or DLUR (it is expected that future releases will use this CV in migration support processing). The CV (referred to as DL CAP in the flows) will be included in the FID2 Encapsulation GDS variable which carries the REQACTPU or ACTPU which triggers the activation of the CPSVRMGR session. The DLUS or DLUR receiving the CV will send back its own copy of the CV in the GDS variable carrying the corresponding RSP(REQACTPU) or RSP(ACTPU). If either the DLUS or DLUR detects a capabilities mismatch, it will reject the SSCP-PU session, and the CP-SVR pipe will be deactivated (see 5.6.2, "Abnormal CP-SVR Pipe Deactivation" on page 5-25 for more information on deactivation of the CP-SVR pipe due to DLUR/S capabilities mismatches).

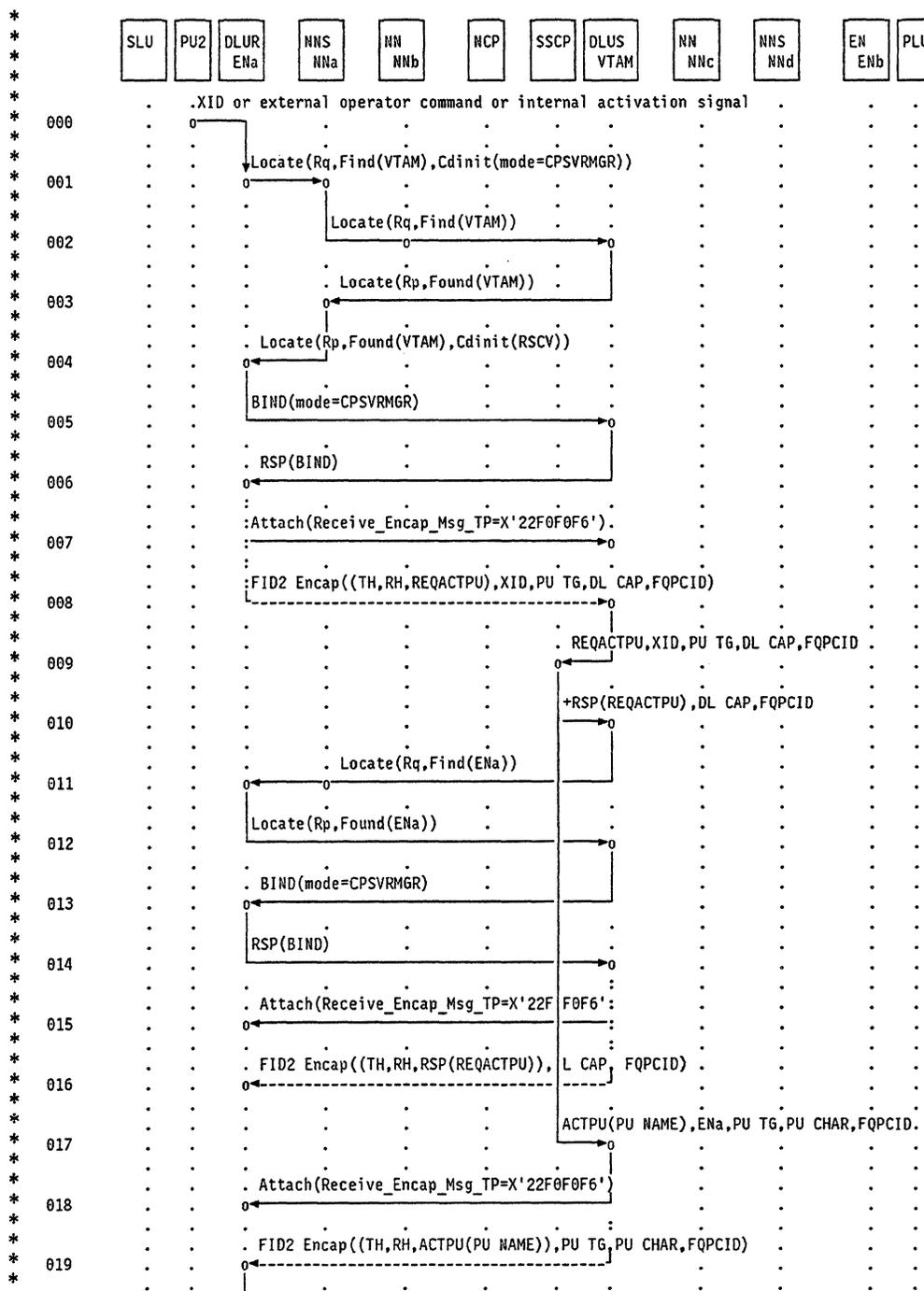
If the SSCP-PU session is accepted, the accepting node must send a +RSP to the message received. This is done by issuing an Allocate from the accepting node, bringing up the parallel conwinner session to the partner, and sending the +RSP on the session.

! If the SSCP-PU session is rejected, the rejecting node will send a -RSP (to the REQACTPU or ACTPU ! RU) in the same manner. This terminates the SSCP-PU session activation. After a node has sent the ! -RSP, the DLUS node must check for any other active SSCP-PU/LU sessions or pending requests on the ! CP-SVR pipe. If no such encapsulated sessions exist, then the DLUS node must follow by sending ! UNBINDs for its conwinner and conloser CPSVRMGR sessions to its partner. This will cause the partner ! DLUR node to send RSP(UNBIND)s for these sessions which completes termination of the CP-SVR pipe ! (see 5.6, "CP-SVR Pipe Deactivation" on page 5-16 for more information on deactivation of the CP-SVR ! pipe).

The following figures illustrate the CP-SVR pipe activation scenarios described thus far.

Note: The reader should note that the flows described below contain more detail than has yet been described. This is because of the tight relationship between SSCP-PU and CPSVRMGR session activation. For a more complete understanding of the SSCP-PU related session initiation information, please refer to 6.2, "SSCP-PU Session Activation" on page 6-6.

5.5.2.1 DLUR-Initiated CP-SVR Pipe Activation



* Figure 5-1. DLUR-initiated CP-SVR pipe activation (DLUS known)

1. Upon receipt of some indication that a DLUR-supported PU requires activation (represented in flow 0), the DLUR node determines the DLUS node to be contacted on behalf of this PU. (See 5.3, "DLUR-Initiated CP-SVR Pipe Activation" on page 5-2) Once the DLUS node has been determined, the DLUR node Allocates a conwinner conversation between its Send_Encap_Msg_TP and the Receive_Encap_Msg_TP (X'22F0F0F6') at the DLUS node. Since no session using the CPSVRMGR mode between this DLUR and the DLUS already exists, the DLUR node initiates the normal APPN flow sequence to bring up such a session. This begins with the DLUR node sending a Locate Find Cdinit request to its network node server. The LU target of the session is the CPname of the DLUS node = VTAM in this case.
 2. The NNS(DLUR) issues either a broadcast or directed search for the CP_LU of the DLUS = VTAM.
 3. The DLUS node responds to the search request with a Locate Found reply.
 4. The NNS(DLUR) uses the information in the Locate reply to calculate a route to the DLUS node and passes this back to the DLUR in the form of an RSCV.
 5. The DLUR node issues a CPSVRMGR BIND to the DLUS node.
 6. The DLUS node provides a BIND Response.
 7. Following the BIND RSP, the DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
 8. Once the Attach has been sent, the DLUR must send a REQACTPU RU up to the DLUS node. This will indicate to the DLUS node that an ACTPU is requested on behalf of some local or downstream PU. Before the REQACTPU RU is sent, however, the DLUR node must generate an FQPCID for this PU. This CV X'60' will be used to uniquely identify the SSCP-PU sessions when more than one exists on a given CPSVRMGR session pipe. A FID2 Encapsulation GDS variable (X'1500') is built, including within it
 - the FID2 PIU (TH, RH, REQACTPU RU)
 - * • an XID image (CV X'81') carrying IDBLK/IDNUM
 - * – an actual XID image if the PU is externally attached to the DLUR
 - * – a DLUR-generated XID format 0 for an internal PU
 - the PU TG (CV X'46') control vector, which identifies the connection between the PU and the DLUR
 - a DL CAP (CV X'51') control vector, which identifies the DLUR's capabilities
 - the FQPCID (CV X'60')
- The FID2 Encapsulation GDS variable X'1500' is then sent up the CPSVRMGR session to the DLUS node.
9. When the DLUS component of the DLUS node receives the FID2 Encapsulation GDS variable, it removes the encapsulation headers and passes the information within the variable to the SSCP component of the DLUS node.
 10. At this point, the DLUS node makes a determination whether it accepts the SSCP-PU session. In this case, the session is accepted and a +RSP to the REQACTPU RU is generated and sent back to the DLUS component of the DLUS node. Also included is another DL CAP control vector, this one identifying the DLUS's capabilities.
 11. The DLUS must now send the +RSP to the REQACTPU RU to the DLUR node. To do this, the DLUS Send_Encap_Msg_TP allocates a conwinner conversation to the Receive_Encap_Msg_TP at the

DLUR node. This causes the normal APPN session activation flows to occur between the DLUS and DLUR node. This begins with the DLUS node initiating a Locate Find Cdinit for the CPname of the DLUR (= ENa in this case).

12. The DLUR responds to the Locate request with a Locate Found reply.
13. The DLUS node uses the information contained in the Locate reply to calculate a route to the DLUR node. It then issues a BIND to the DLUR node to establish its conwinner session with the DLUR node.
14. The DLUR node provides a BIND Response.
15. The DLUS node then sends an FMH5 Attach to initiate the Receive_Encap_Msg_TP at the DLUR node.
16. The DLUS node can now send its REQACTPU RSP to the DLUR. It encapsulates in the FID2 Encapsulation GDS variable the FID2 PIU, the DL CAP control vector, and the FQPCID (to identify the related request to the DLUR). The DLUR examines the information in the variable and also agrees to proceed.
17. Meanwhile, the DLUS SSCP has issued an ACTPU RU in response to the REQACTPU received from the DLUR. The ACTPU RU includes the name of the PU.
18. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
19. The DLUS component of the DLUS node then encapsulates the ACTPU RU FID2 PIU along with the PU TG control vector, PU CHAR control vector (used to define such PU characteristics as ANS=STOP|CONT), and the FQPCID of this SSCP-PU relationship with a FID2 Encapsulation GDS variable and sends the flow on its conwinner CPSVRMGR session to the DLUR.
20. The Receive_Encap_Msg_TP at the DLUR receives the encapsulated ACTPU message and removes the encapsulation headers. It uses the FQPCID information and PU TG control vector to identify the PU and its location in order to send the ACTPU to the PU itself. The DLUR removes the PU name from the ACTPU RU and stores it for network management.
21. When the PU receives the ACTPU RU, it will respond to the DLUR component with an ACTPU Response RU.
22. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
23. The DLUR component then encapsulates the RSP(ACTPU) FID2 PIU along with the FQPCID SSCP-PU session identifier in a FID2 Encapsulation GDS variable and sends it over the DLUR conwinner CPSVRMGR session to the DLUS.
24. The Receive_Encap_Msg_TP at the DLUS receives the message, removes the encapsulation headers, and passes the RSP(ACTPU) RU to the SSCP component of the DLUS node. The CP-SVR pipe is now up and an associated SSCP-PU session has been established.

5.5.2.2 DLUS-Initiated CP-SVR Pipe Activation

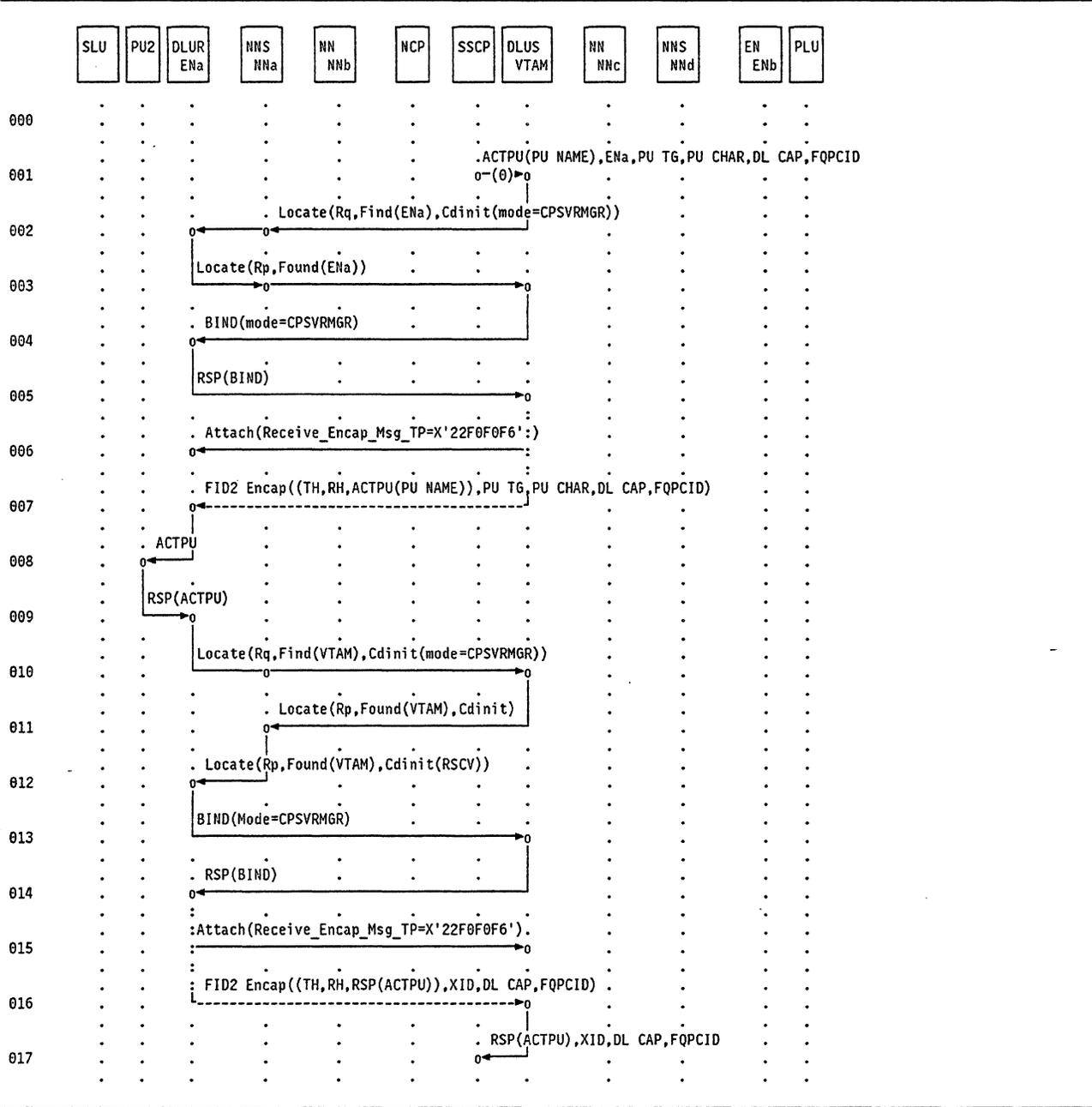
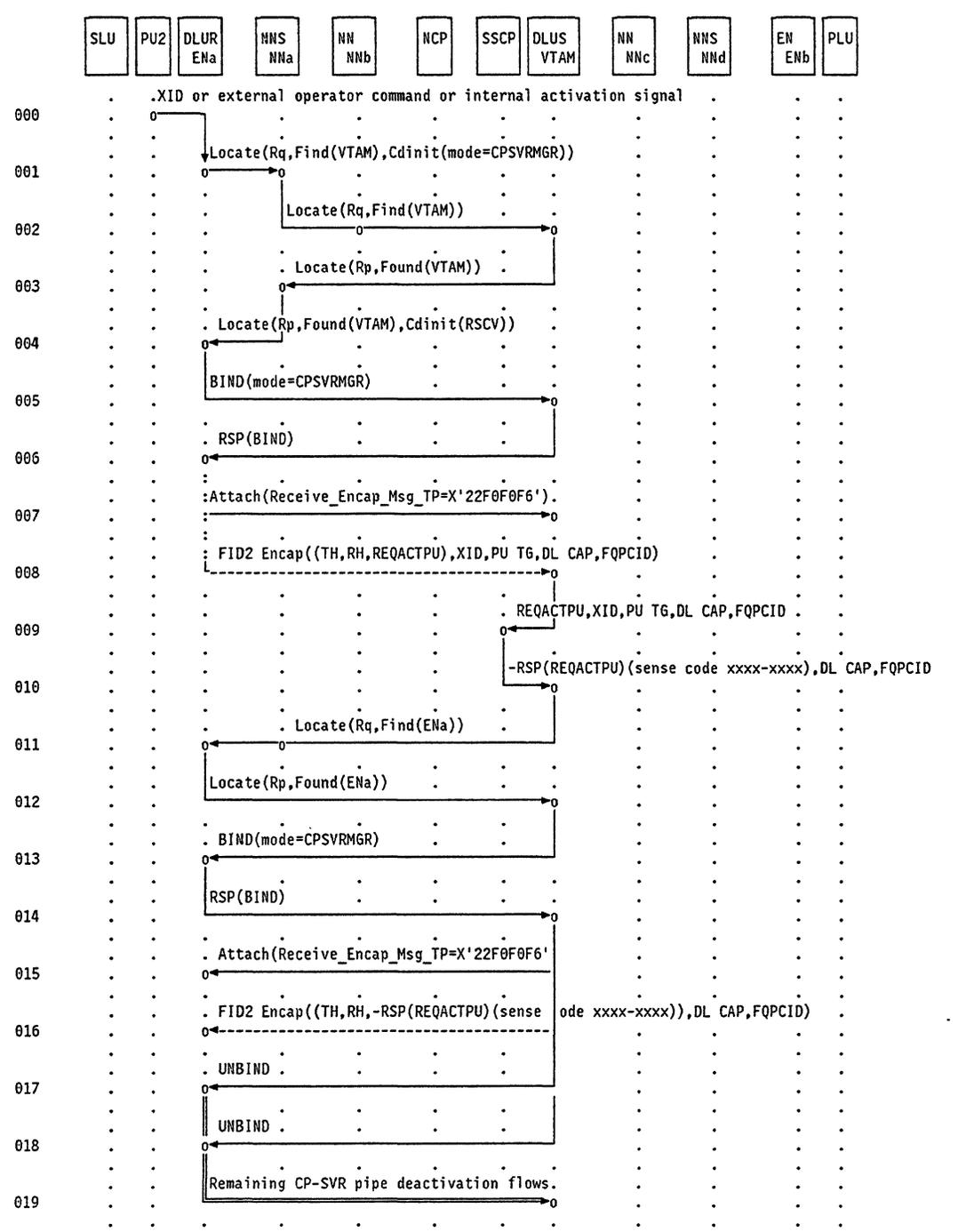


Figure 5-2. DLUS-initiated CP-SVR pipe activation. Footnotes:

0) Internal signal

1. When a DLUS node wishes to activate a PU, it uses its internal information (pre-defined or dynamically generated) to send an ACTPU signal to the DLUS component. This includes information about the PU, its connection to the DLUR, the DLUS's capabilities, and an FQPCID.
2. The DLUS component uses this same information to determine the DLUR node associated with this PU. (See 5.4.1, "New PU Definition Parameters" on page 5-4 for more details.) It then allocates a conversation to the Receive_Encap_Msg_TP at the DLUR node. Since no session already exists to carry this conversation, the DLUS node initiates one starting with a Locate Find Cdinit for the CP_LU of the DLUR node = ENa.
3. The DLUR node provides a Locate reply.
4. The DLUS node uses the information in the Locate reply to calculate a route to the DLUR node and send a BIND using the CPSVRMGR mode.
5. The DLUR node provides a BIND response.
6. Once the BIND response has been received, the DLUS node sends an FMH5 to attach the Receive_Encap_Msg_TP at the DLUR node.
7. The DLUS now follows the Attach with an encapsulated ACTPU FID2 PIU. This FID2 Encapsulation GDS variable carries with it an FQPCID that has been generated by the DLUS node to be used as an SSCP-PU session identifier for the DLUR and DLUS nodes, as well as the PU TG, PU CHAR, and DL CAP control vectors.
8. When the DLUR node receives the encapsulated message, it removes the headers, validates the capabilities, uses information with the ACTPU (such as the PU TG, and PU CHAR control vectors) to determine the receiving PU, removes the PU name (storing it for network management), and sends the ACTPU to this node.
9. The PU responds to the ACTPU with an ACTPU response message to the DLUR.
10. The DLUR must now send the RSP(ACTPU) to the DLUS. In order to do this, it needs to allocate a conwinner conversation between its Send_Encap_Msg_TP and the Receive_Encap_Msg_TP at the DLUS node. Since no DLUR conwinner session already exists, the DLUR node uses the normal APPN flows to create such a session beginning with a Locate Find Cdinit being issued to the DLUS CP_LU=VTAM.
11. The DLUS node responds to the Locate with a Locate reply.
12. The NNS(DLUR) uses the information in the Locate reply to calculate a route to the DLUS node and sends this to the DLUR in the form of an RSCV attached to the Locate reply.
13. The DLUR node now issues a CPSVRMGR BIND to the DLUS node.
14. The DLUS node provides a BIND response.
15. After receiving the RSP(BIND), the DLUR node now sends an Attach for the Receive_Encap_Msg_TP at the DLUS.
16. Following the Attach, the DLUR node can send the encapsulated RSP(ACTPU) to the DLUS. It also includes the XID control vector as well as a DL CAP control vector which identifies the DLUR's capabilities.
17. The DLUS component of the DLUS node removes the encapsulation headers, validates the capabilities, and passes the RSP(ACTPU) RU to the SSCP component of the node. The CP-SVR pipe and associated SSCP-PU session have now been activated.

5.5.2.3 SSCP-PU Session Rejection



! Figure 5-3. DLUR-initiated, DLUS-rejected CP-SVR pipe activation. Footnotes:

- ! 0) To avoid UNBIND overtaking data (flow 18 and 19) BIS protocols can be used, but have been omitted from the diagram.
- !

1. Following some stimulus (flow 0), the DLUR node desires to activate a PU. To do this, it needs to request an ACTPU from some DLUS node. It uses pre-defined information to determine the DLUS node and allocates a conversation to the Receive_Encap_Msg_TP at that DLUS node. Since no conwinner session to that DLUS already exists, the DLUR node uses normal APPN flows to initiate such a session. This begins with a Locate Find Cdinit for the DLUS node.
2. The NNS(DLUR) issues a broadcast or directed search for the DLUS.
3. The DLUS provides a locate reply.
4. The NNS(DLUR) uses information in the Locate reply to calculate a route to the DLUS and appends this to the Locate reply in the form of an RSCV.
5. The DLUR node then issues a conwinner BIND to the DLUS node.
6. The DLUS node provides a RSP(BIND).
7. Once the DLUR receives a RSP(BIND), it issues an Attach to initiate the Receive_Encap_Msg_TP at the DLUS.
8. The DLUR follows the Attach with an encapsulated REQACTPU RU FID2 PIU for the DLUS node.
9. The Receive_Encap_Msg_TP at the DLUS node removes the encapsulation headers and passes the REQACTPU RU to the SSCP component.
10. The SSCP component looks at the information contained with the REQACTPU RU. It then determines that it will not accept the SSCP-PU session. This can be for a number of reasons. For example:
 - The DLUR-generated FQPCID is not unique (sense code X'083B 0002').
 - The DLUS is out of necessary resources (X'0812 0000').
 - The DLUS node does not recognize this PU and does not support dynamic PU activation (X'0806 0000').
 - The IDBLK/IDNUM and/or CP name supplied with the REQACTPU RU are not unique (X'0806 0000').
 - The Configuration Services XID Exit rejects the XID and the CV X'46' which were included with the REQACTPU RU (X'0809 0000').
 - The DLUS node does not have a network address pair to assign to the PU (X'0812 0007').
 - ! • etc.
- ! So the SSCP generates a negative response to the REQACTPU RU with an appropriate sense code.
11. The DLUS component must now send the RSP(REQACTPU) to the DLUR node. It, therefore, allocates a conwinner conversation to the Receive_Encap_Msg_TP at the DLUR node. Since no such session already exists to carry the conversation, the DLUS node initiates one. This begins with a Locate Find Cdinit being issued for the DLUR node.
12. The DLUR node provides a Locate reply.
13. The DLUS node uses information in the Locate reply to calculate a route to the DLUR and send a CPSVRMGR BIND to the DLUR.
14. The DLUR provides a BIND response.
15. Once the DLUS receives the RSP(BIND), it issues an FMH5 Attach for the Receive_Encap_Msg_TP at the DLUR.
16. The DLUS then sends the encapsulated RSP(REQACTPU) RU to the DLUR.

17. Since there are no SSCP-PU sessions and none outstanding on the conwinner CPSVRMGR session to the DLUR, the DLUS will deactivate the CP-SVR pipe. This is begun by sending an UNBIND on the conwinner session to its partner.
Note: The UNBIND will not overtake the encapsulated RSP(REQACTPU) when LU6.2 BIS protocols (not shown here) are used to clear the pipe first. Use of LU 6.2 BIS protocols is optional.
18. The DLUS also sends an UNBIND on the conloser session to its partner.
19. The DLUR node sends a RSP(UNBIND) to the DLUS for the conwinner session and another RSP(UNBIND) for the conloser session. The SSCP-PU session and CP-SVR pipe are now deactivated. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.

5.6 CP-SVR Pipe Deactivation

There are several situations where a CP-SVR pipe will be brought down, including normal deactivation and loss of session partner. In general, the DLUS will normally bring down the pipe, but either the DLUR or DLUS may deactivate the pipe if error or outage conditions have been detected. In every case, whichever end initiates the deactivation will deactivate both its conwinner and conloser sessions. In addition, when a DLUS or DLUR is requested to deactivate its conloser session, it should also deactivate its own conwinner session. Reactivation of the CP-SVR pipe should not begin until both conwinner and conloser sessions are deactivated (see 5.7, "Pipe And Session Failure Logic" on page 5-51 for more information on recovery of the CP-SVR pipe).

5.6.1 Normal CP-SVR Pipe Deactivation

When the DLUS receives an encapsulated positive RSP(DACTPU) on the CP-SVR pipe, in addition to de-encapsulating the RU and passing it on to the SSCP, it will check to see if it can deactivate the CP-SVR pipe. The DLUS will deactivate the CP-SVR pipe if:

- there are no active SSCP-PU sessions using the pipe; and
 - there are pending sessions but a DLUR/S capabilities mismatch was detected; or
 - there are no pending SSCP-PU sessions, e.g.,
 - REQACTPU was received from the DLUR, but no ACTPU has been sent by the SSCP yet
 - ACTPU was sent by the SSCP, but no RSP(ACTPU) has been received by the DLUS yet

If these conditions exist, the DLUS will deactivate its conwinner and conloser sessions by issuing an UNBIND on each session. Depending on the order and timing of the arrival of the UNBINDs, the DLUR will act as follows:

- *second UNBIND arrives before first is processed* - the DLUR will return a positive RSP(UNBIND) on each session, and the pipe will be deactivated;
- *UNBIND(DLUS conloser) arrives after UNBIND(DLUS conwinner) is processed* - the DLUR will return a positive RSP(UNBIND) on its conloser session and send an UNBIND on its own conwinner session. When the DLUS and DLUR receive the UNBINDs on this (DLUR conwinner) session, they will each return a positive RSP(UNBIND), and the pipe will be deactivated. The RSP(UNBIND)s will clean up the session connectors as they flow, so neither of them will actually arrive at their destination.
- *UNBIND(DLUS conwinner) arrives after UNBIND(DLUS conloser) is processed* - the DLUR will return a positive RSP(UNBIND) on its conwinner session and send an UNBIND on its own conloser session. When the DLUS and DLUR receive the UNBINDs on this (DLUR conloser) session, they will each return a positive RSP(UNBIND), and the pipe will be deactivated. The RSP(UNBIND)s will clean up the session connectors as they flow, so neither of them will actually arrive at their destination.

The following figures show these normal CP-SVR pipe deactivation scenarios:

- **Note** - In all other flows in this document, CP-SVR pipe deactivation will be represented by two UNBINDs being sent and a generic "Remaining CP-SVR pipe deactivation flows" to cover all possible UNBIND arrival cases. Each flow will be referred to this section for further details.

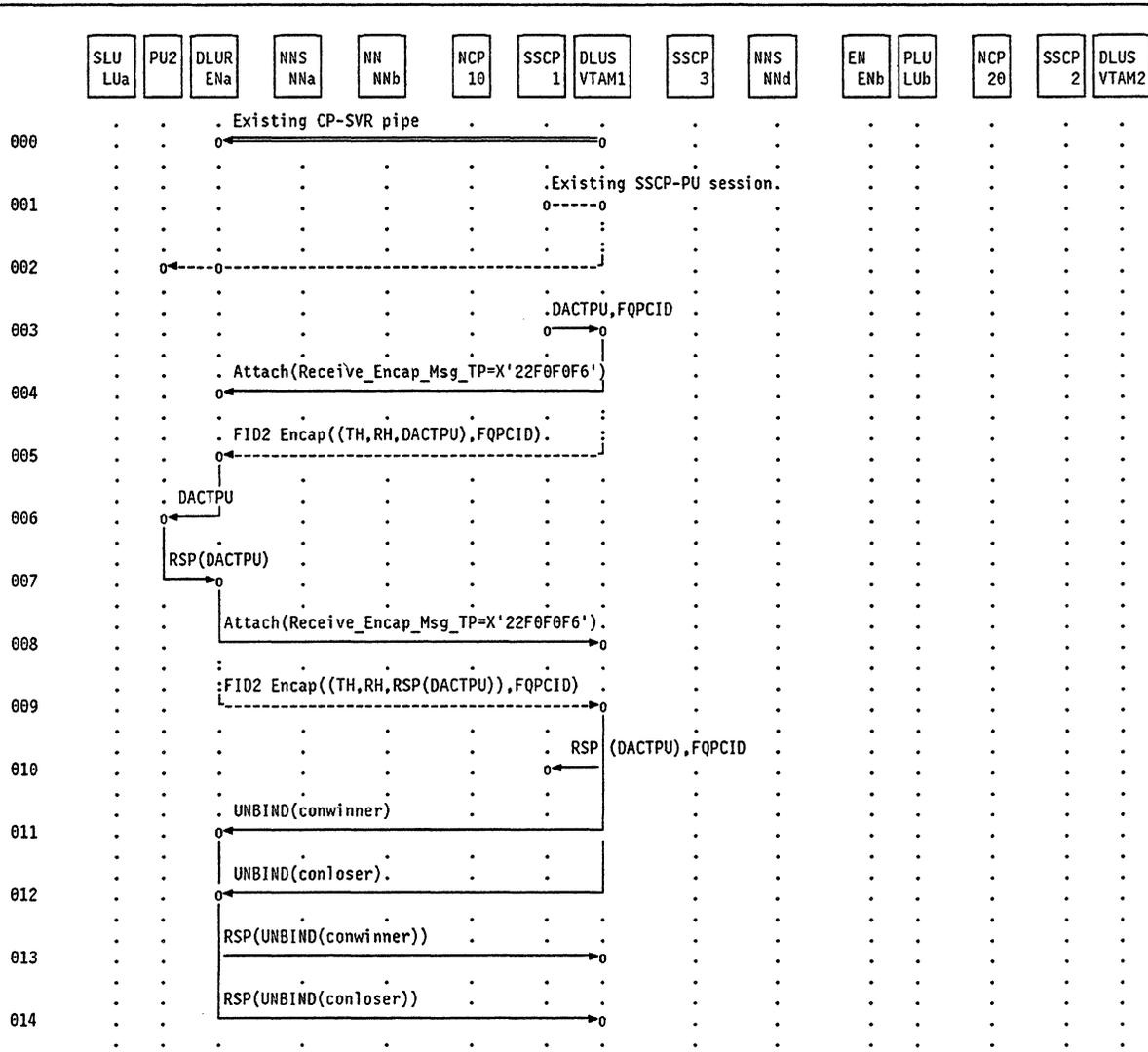


Figure 5-4. CP-SVR pipe deactivation - second UNBIND arrives before first is processed

1. There is an active session between SSCP1 and PU2. It is the only active SSCP-PU session using the CP-SVR pipe between VTAM1 and ENa.
- 2.
3. SSCP1 initiates normal deactivation of its session with PU2 by building a DACTPU and passing it on to VTAM1.
4. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
5. The DLUS VTAM1 encapsulates the DACTPU and forwards it to the DLUR ENa.
6. The DLUR ENa de-encapsulates the DACTPU and forwards it to PU2.
7. PU2 returns a positive RSP(DACTPU) to SSCP1 via ENa.
8. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
9. ENa encapsulates the RSP(DACTPU) and sends it to VTAM1.
10. VTAM1 de-encapsulates the RSP(DACTPU) and forwards it to SSCP1.
11. VTAM1 also determines that there are now no active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it sends an UNBIND to ENa on VTAM1's conwinner session to begin deactivation of the pipe.
12. VTAM1 also sends an UNBIND to ENa on VTAM1's conloser session.
13. ENa returns a positive RSP(UNBIND) and takes down its conloser session. Since it has already received an UNBIND on its conwinner session, ENa does not send an UNBIND of its own. Upon receipt of the RSP(UNBIND), VTAM1 takes down its conwinner session.
14. ENa returns a positive RSP(UNBIND) and takes down its conwinner session. Since it has already received an UNBIND on its conloser session, ENa does not send an UNBIND of its own. Upon receipt of the RSP(UNBIND), VTAM1 takes down its conloser session.

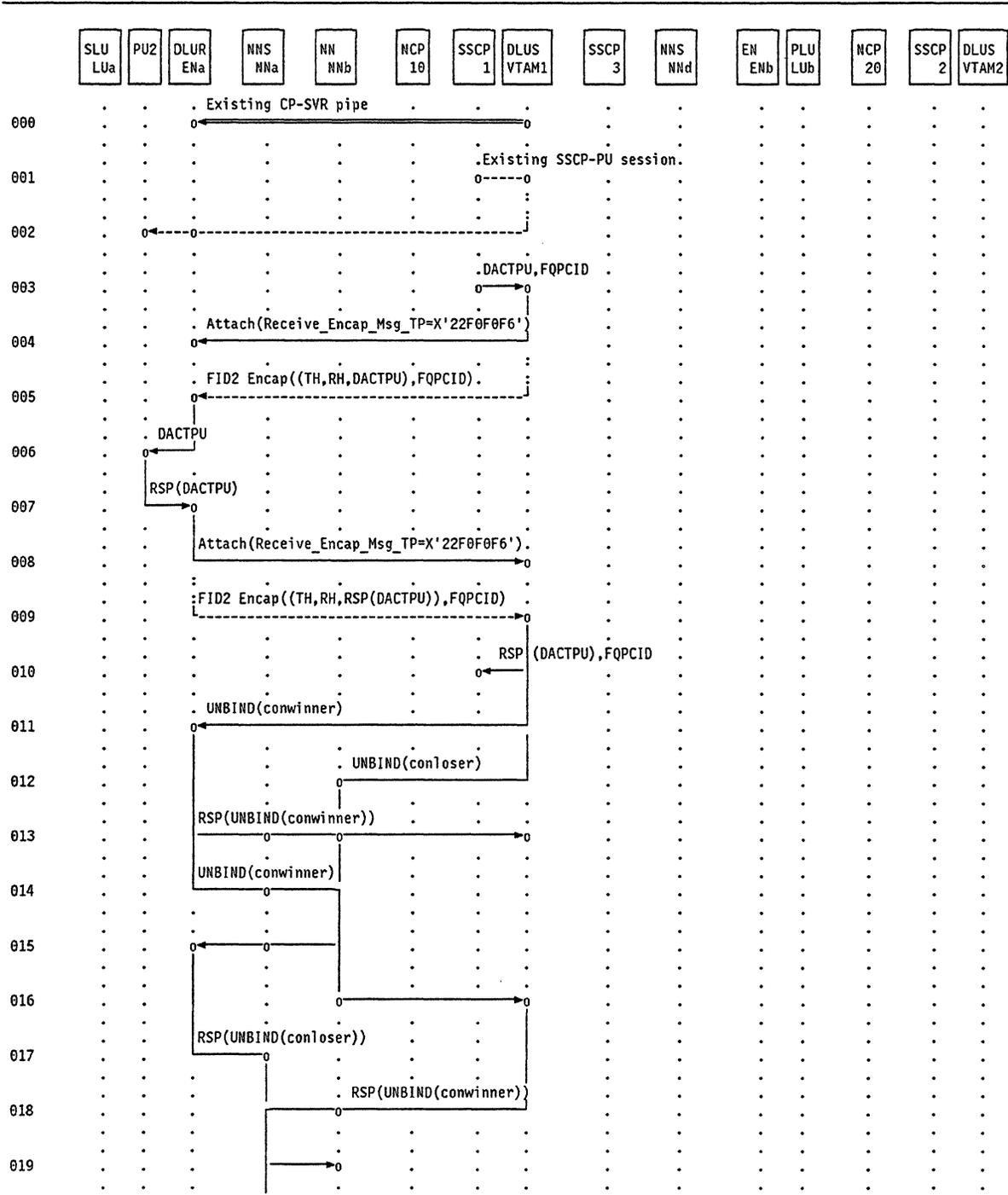


Figure 5-5. CP-SVR pipe deactivation - UNBIND(conloser) arrives after UNBIND(conwinner) is processed

	SLU LUa	PU2	DLUR ENa	NNS NNa	NN NNb	NCP 10	SSCP 1	DLUS VTAM1	SSCP 3	NNS NNd	EN ENb	PLU LUB	NCP 20	SSCP 2	DLUS VTAM2
020	:	:	:	↓ 0	:	:	:	:	:	:	:	:	:	:	:
	:	:	:

1. There is an active session between SSCP1 and PU2. It is the only active SSCP-PU session using the CP-SVR pipe between VTAM1 and ENa.
- 2.
3. SSCP1 initiates normal deactivation of its session with PU2 by building a DACTPU and passing it on to VTAM1.
4. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
5. The DLUS VTAM1 encapsulates the DACTPU and forwards it to the DLUR ENa.
6. The DLUR ENa de-encapsulates the DACTPU and forwards it to PU2.
7. PU2 returns a positive RSP(DACTPU) to SSCP1 via ENa.
8. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
9. ENa encapsulates the RSP(DACTPU) and sends it to VTAM1.
10. VTAM1 de-encapsulates the RSP(DACTPU) and forwards it to SSCP1.
11. VTAM1 also determines that there are now no active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it sends an UNBIND to ENa on VTAM1's conwinner session to begin deactivation of the pipe.
12. VTAM1 also sends an UNBIND to ENa on VTAM1's conloser session.
13. ENa returns a positive RSP(UNBIND) and takes down its conloser session. Upon receipt of the RSP(UNBIND), VTAM1 takes down its conwinner session.
14. Since it has not received an UNBIND on its conwinner session, ENa sends an UNBIND(conwinner) of its own.
15. ENa receives the UNBIND(conloser) from VTAM1.
16. VTAM1 receives the UNBIND(conwinner) from ENa.
17. ENa returns a positive RSP(UNBIND) and takes down its conwinner session.
18. VTAM1 returns a positive RSP(UNBIND) and takes down its conloser session.
19. NNb discards the RSP(UNBIND) from ENa since it tore down the session connector when it forwarded the RSP(UNBIND) from VTAM1.
20. NNa discards the RSP(UNBIND) from VTAM1 since it tore down the session connector when it forwarded the RSP(UNBIND) from ENa.

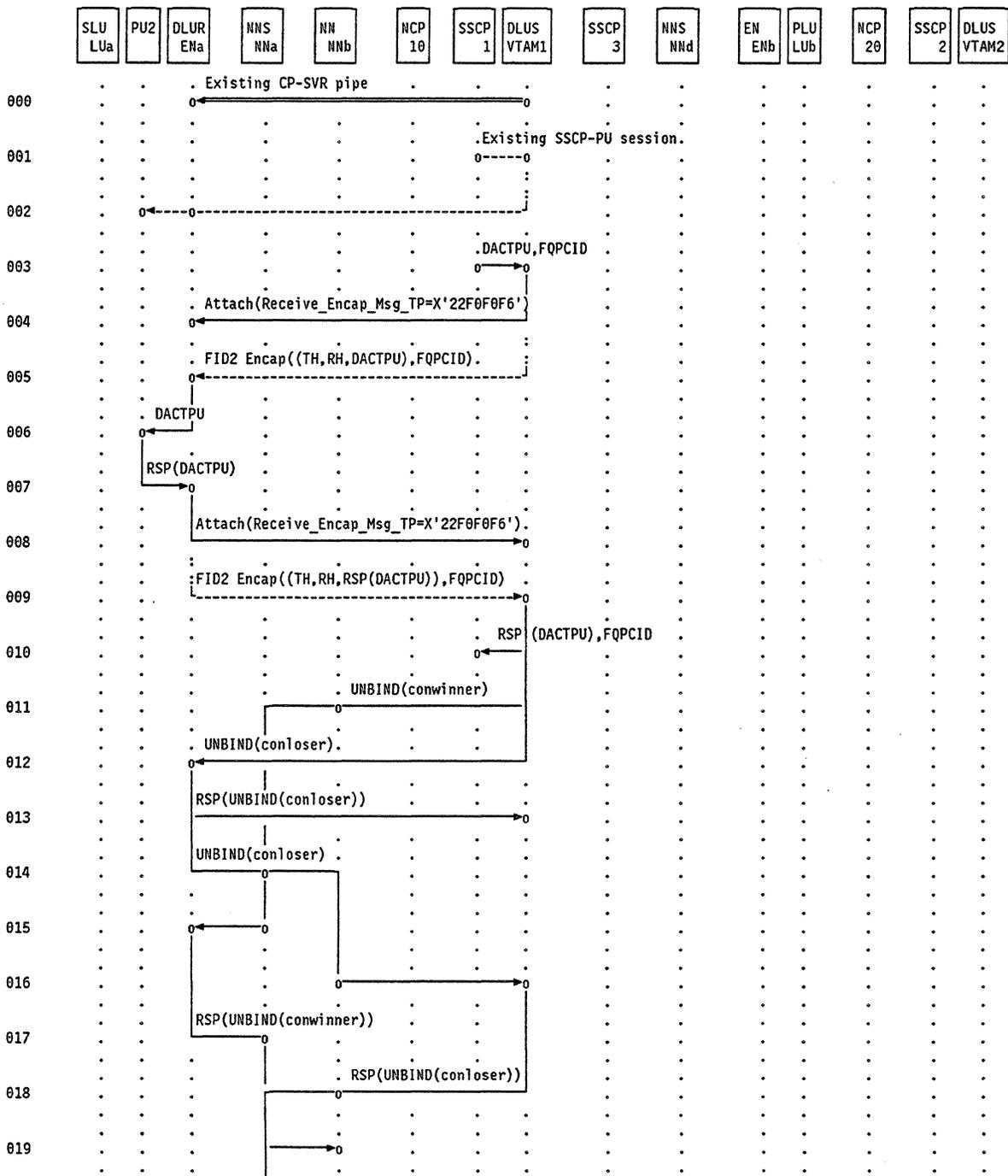


Figure 5-6. CP-SVR pipe deactivation - UNBIND(conwinner) arrives after UNBIND(conloser) is processed

	SLU LUa	PU2	DLUR ENa	NNS NNa ↓ 0	NN NNb	NCP 10	SSCP 1 DLUS VTAM1	SSCP 3	NNS NNd	EN ENb	PLU LUb	NCP 20	SSCP 2	DLUS VTAM2
020

1. There is an active session between SSCP1 and PU2. It is the only active SSCP-PU session using the CP-SVR pipe between VTAM1 and ENa.
- 2.
3. SSCP1 initiates normal deactivation of its session with PU2 by building a DACTPU and passing it on to VTAM1.
4. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
5. The DLUS VTAM1 encapsulates the DACTPU and forwards it to the DLUR ENa.
6. The DLUR ENa de-encapsulates the DACTPU and forwards it to PU2.
7. PU2 returns a positive RSP(DACTPU) to SSCP1 via ENa.
8. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
9. ENa encapsulates the RSP(DACTPU) and sends it to VTAM1.
10. VTAM1 de-encapsulates the RSP(DACTPU) and forwards it to SSCP1.
11. VTAM1 also determines that there are now no active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it sends an UNBIND to ENa on VTAM1's conwinner session to begin deactivation of the pipe.
12. VTAM1 also sends an UNBIND to ENa on VTAM1's conloser session.
13. ENa returns a positive RSP(UNBIND) and takes down its conwinner session. Upon receipt of the RSP(UNBIND), VTAM1 takes down its conwinner session.
14. Since it has not received an UNBIND on its conloser session, ENa sends an UNBIND(conloser) of its own.
15. ENa receives the UNBIND(conwinner) from VTAM1.
16. VTAM1 receives the UNBIND(conloser) from ENa.
17. ENa returns a positive RSP(UNBIND) and takes down its conloser session.
18. VTAM1 returns a positive RSP(UNBIND) and takes down its conwinner session.
19. NNb discards the RSP(UNBIND) from ENa since it tore down the session connector when it forwarded the RSP(UNBIND) from VTAM1.
20. NNa discards the RSP(UNBIND) from VTAM1 since it tore down the session connector when it forwarded the RSP(UNBIND) from ENa.

5.6.2 Abnormal CP-SVR Pipe Deactivation

There are other situations where the DLUS and DLUR should deactivate the CP-SVR pipe:

- If either the DLUR or DLUS should receive an UNBIND on its conloser session, it should return a positive RSP(UNBIND) on that session and, if its conwinner session is active or pending, send an UNBIND on the conwinner session.
 - If an outage is detected on the CP-SVR pipe, and an UNBIND cannot be successfully routed to the session partner, then the DLUS or DLUR should deactivate and clean up both its conwinner and conloser sessions.
- ! • Whenever a DLUR/S capabilities mismatch is detected, the CP-SVR pipe should be deactivated.

Reactivation of the CP-SVR pipe can be initiated from either the DLUS or DLUR, or both, upon completion of the deactivation of the current pipe (see 5.7, “Pipe And Session Failure Logic” on page 5-51 for more information on recovery of the CP-SVR pipe).

5.6.2.1 UNBIND-Initiated CP-SVR Pipe Deactivation

One of the DLUS requirements is to support forced deactivation of the DLUR-related sessions, i.e.,

- the CP-SVR pipe between the DLUS and the DLUR
- the SSCP-PU sessions between the SSCP in the DLUS and the PU serviced by the DLUR
- the SSCP-LU sessions between the SSCP in the DLUS and the LU serviced by the DLUR
- the LU-LU sessions involving the LU which has an SSCP-LU session with the SSCP in the DLUS

Normally, DACTLUs and DACTPUs would be sent followed by an UNBIND for the CPSVRMGR session. To speed things up, new sense codes have been created for the CP-SVR pipe UNBIND which will indicate to the DLUR to deactivate some or all of the session types listed above. There are two major sense code categories:

- disruptive - all sessions (CPSVRMGR, SSCP-PU, SSCP-LU, LU-LU) will be deactivated by the DLUR for its PUs and LUs serviced by this DLUS
- non-disruptive - all sessions except active LU-LU sessions will be deactivated

5.6.2.1.1 Disruptive UNBIND-Initiated CP-SVR Pipe Deactivation: There is one disruptive session deactivation sense code, X'08A0 0007'. If sense code X'08A0 0007' appears in CV X'35' in an UNBIND on the conloser CPSVRMGR session, the DLUR will:

- deactivate all SSCP-PU, SSCP-LU, and LU-LU sessions for its PUs and LUs serviced by this DLUS
- return a positive RSP(UNBIND) on the conloser session and, if its conwinner session is active or pending, send an UNBIND on the conwinner session

5.6.2.1.2 Non-Disruptive UNBIND-Initiated CP-SVR Pipe Deactivation: There are three non-disruptive session deactivation sense codes: X'08A0 0008', X'08A0 0009', and X'08A0 000A'. If one of these sense codes appears in CV X'35' in an UNBIND on the conloser CPSVRMGR session, the DLUR will:

- deactivate all SSCP-PU and SSCP-LU sessions for PUs and LUs serviced by this DLUS
- ! • deactivate all active and pending LU-LU sessions for LUs
- ! – serviced by this DLUS, and

- ! - defined with ANS = STOP
- return a positive RSP(UNBIND) on the conloser session and, if its conwinner session is active or pending, send an UNBIND on the conwinner session

The distinctions between each of these sense codes are:

- X'08A0 0008' - DLUR can immediately reinitiate CP-SVR pipe activation
- X'08A0 0009' - DLUR can immediately reinitiate CP-SVR pipe activation / this sense code is used when a protocol violation is detected, such as a DLUR/S capabilities mismatch
- X'08A0 000A' - DLUR should wait for a DLUS (need not be the same DLUS sending the UNBIND) to reinitiate CP-SVR pipe activation / DLUR-implementing products may choose to wait:
 - indefinitely, or
 - for a designated (either system- or user-defined) period of time for a CP-SVR pipe activation from a DLUS, after which the DLUR will go ahead with CP-SVR pipe activation, with either the same or a different DLUS.

5.6.2.1.3 Reactivation After UNBIND-Initiated CP-SVR Pipe Deactivation: With one exception, after the CP-SVR pipe deactivation has completed, both the DLUS and the DLUR can go right ahead and bring up a new CP-SVR pipe, with the same or a different DLUR/S partner. The exception is, upon receipt of a X'08A0 000A' sense code, the DLUR must wait for a DLUS to bring up the pipe; it is a product option as to whether the DLUR waits indefinitely or for a finite (system- or user-defined) period of time before bringing up a new CP-SVR pipe.

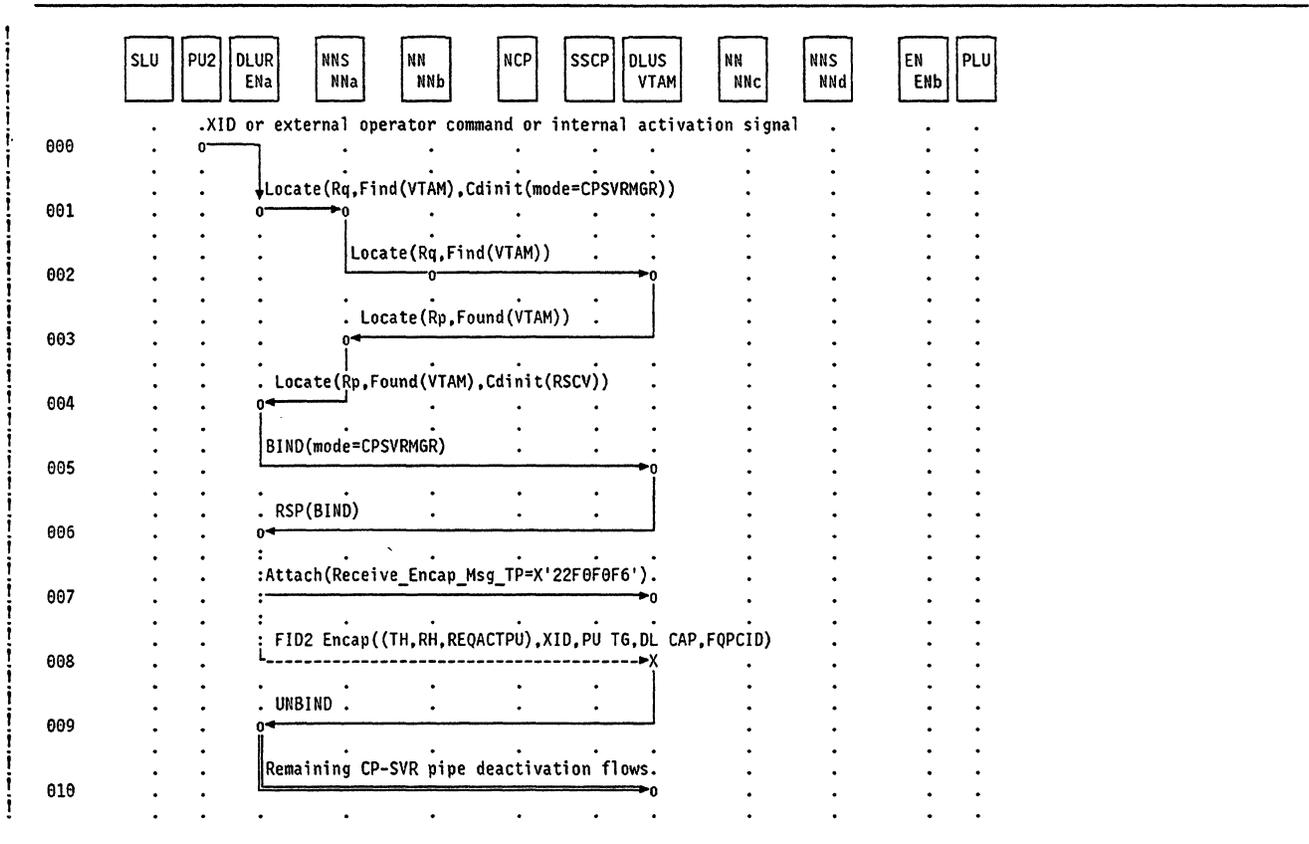
! 5.6.2.2 DLUR/S Capabilities Mismatches

! A CP-SVR pipe should only be established between a DLUS and a DLUR with compatible capabilities.
! Exchanging the DLUR/S Capabilities control vector (CV X'51') assists a node in determining whether to
! allow a CP-SVR pipe to be activated between itself and a partner node.

! • To ensure detection of a capabilities mismatch, the CV X'51' should be examined before the RU
! included in the GDS variable is processed.

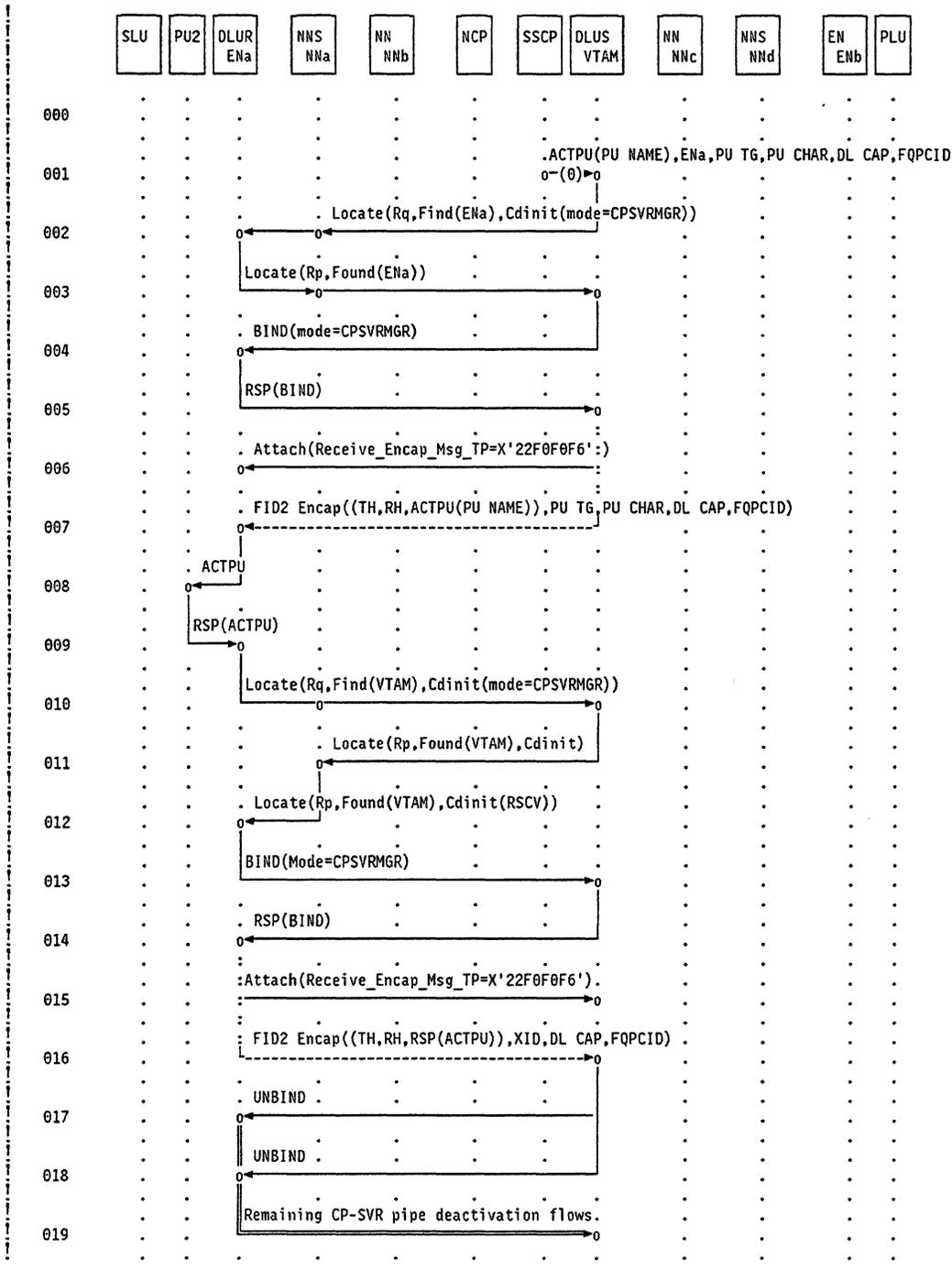
! Whenever a mismatch in DLUR/S capabilities is detected, the CP-SVR pipe should be deactivated. Which-
! ever node detects the mismatch will initiate the pipe deactivation.

! **5.6.2.2.1 DLUS-Detected Capabilities Mismatch (DLUS-DLUR Pipe):** The following figures show examples
! of DLUR/S capabilities mismatches detected by the DLUS during activation of a CP-SVR pipe between a
! DLUS and a DLUR.



! Figure 5-7. DLUR-initiated, DLUS-rejected CP-SVR pipe activation

- ! 1. Following some stimulus (flow 0), the DLUR node desires to activate a PU. To do this, it needs to request an ACTPU from some DLUS node. It uses pre-defined information to determine the DLUS node and allocates a conversation to the Receive_Encap_Msg_TP at that DLUS node. Since no conwinner session to that DLUS already exists, the DLUR node uses normal APPN flows to initiate such a session. This begins with a Locate Find Cdinit for the DLUS node.
- ! 2. The NNS(DLUR) issues a broadcast or directed search for the DLUS.
- ! 3. The DLUS provides a locate reply.
- ! 4. The NNS(DLUR) uses information in the Locate reply to calculate a route to the DLUS and appends this to the Locate reply in the form of an RSCV.
- ! 5. The DLUR node then issues a conwinner BIND to the DLUS node.
- ! 6. The DLUS node provides a RSP(BIND).
- ! 7. Once the DLUR receives a RSP(BIND), it issues an Attach to initiate the Receive_Encap_Msg_TP at the DLUS.
- ! 8. The DLUR follows the Attach with an encapsulated REQACTPU RU FID2 PIU for the DLUS node.
- ! 9. The Receive_Encap_Msg_TP at the DLUS node removes the encapsulation headers. The DLUS component looks at the DLUR's capabilities and determines that they are incompatible with those of the DLUS. Since a DLUR/S capabilities mismatch was detected, the DLUS will deactivate the CP-SVR pipe. This is begun by sending an UNBIND on the conloser session to its partner, including X'088E 0008' in the CV X'35'.
- ! 10. The DLUR node sends a RSP(UNBIND) to the DLUS for its conwinner session. The CP-SVR pipe is now deactivated. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.

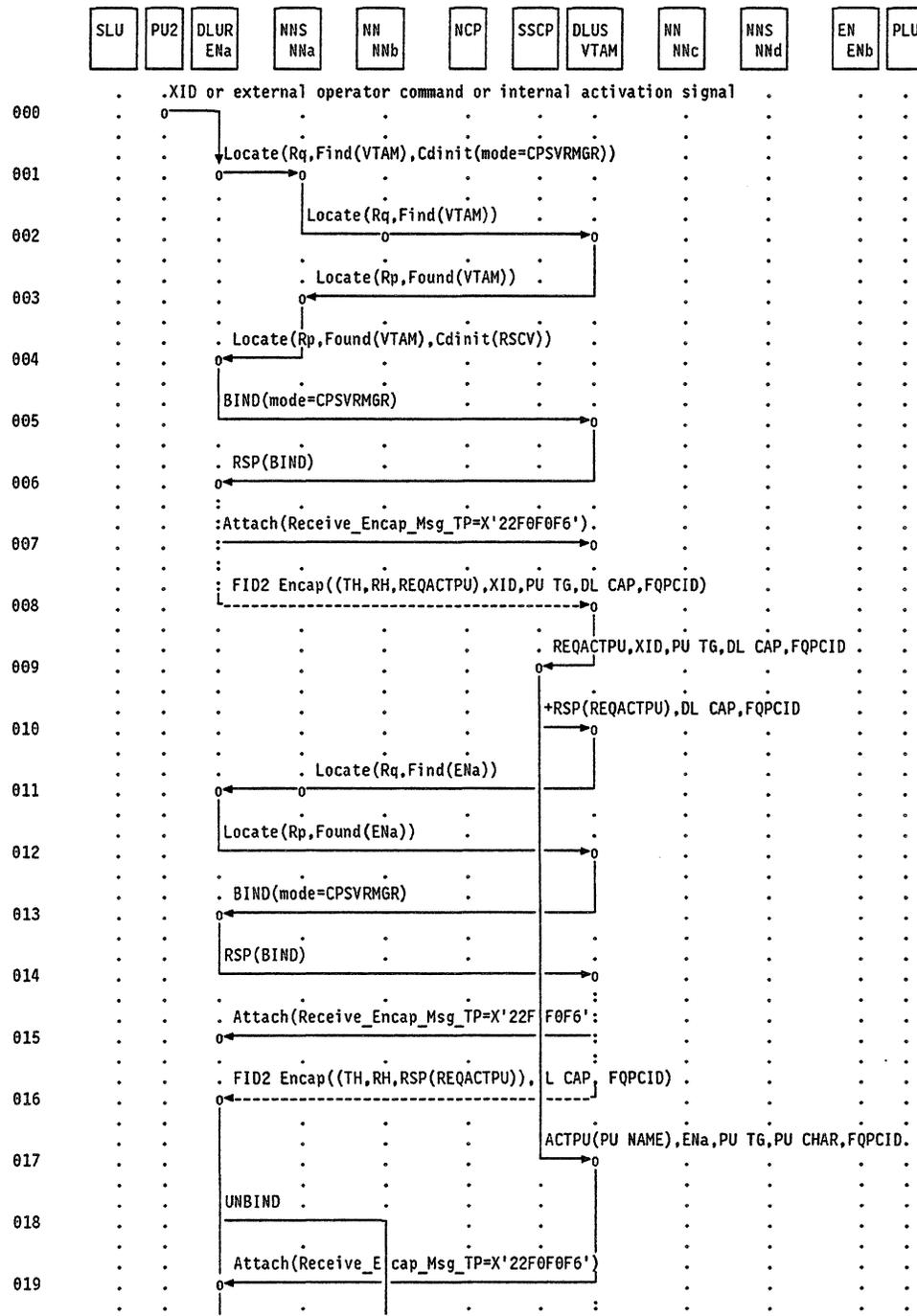


! Figure 5-8. DLUS-initiated, DLUS-rejected CP-SVR pipe activation. Footnotes:

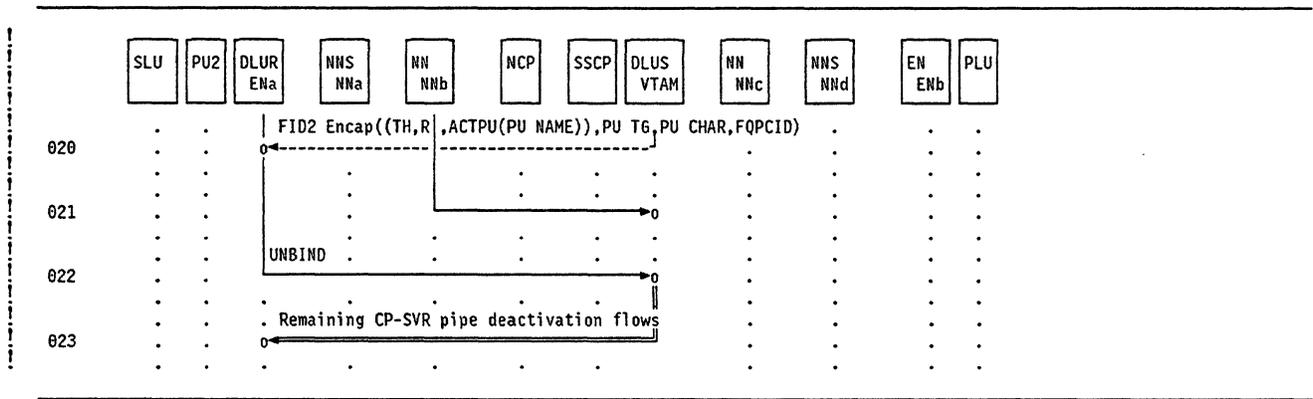
! 0) Internal signal

- ! 1. When a DLUS node wishes to activate a PU, it uses its internal information (pre-defined or dynamically generated) to send an ACTPU signal to the DLUS component. This includes information about the PU, its connection to the DLUR, the DLUS's capabilities, and an FQPCID.
- ! 2. The DLUS component uses this same information to determine the DLUR node associated with this PU. (See 5.4.1, "New PU Definition Parameters" on page 5-4 for more details.) It then allocates a conversation to the Receive_Encap_Msg_TP at the DLUR node. Since no session already exists to carry this conversation, the DLUS node initiates one starting with a Locate Find Cdinit for the CP_LU of the DLUR node = ENa.
- ! 3. The DLUR node provides a Locate reply.
- ! 4. The DLUS node uses the information in the Locate reply to calculate a route to the DLUR node and send a BIND using the CPSVRMGR mode.
- ! 5. The DLUR node provides a BIND response.
- ! 6. Once the BIND response has been received, the DLUS node sends an FMH5 to attach the Receive_Encap_Msg_TP at the DLUR node.
- ! 7. The DLUS now follows the attach with an encapsulated ACTPU FID2 PIU. This FID2 Encapsulation GDS variable carries with it an FQPCID that has been generated by the DLUS node to be used as an SSCP-PU session identifier for the DLUR and DLUS nodes, as well as the PU TG and DL CAP control vectors.
- ! 8. When the DLUR node receives the encapsulated message it removes the headers, validates the capabilities, uses information with the ACTPU (such as the PU TG and PU CHAR control vectors) to determine the receiving PU, removes the PU name (storing it for network management), and sends the ACTPU to this node.
- ! 9. The PU responds to the ACTPU with an ACTPU response message to the DLUR.
- ! 10. The DLUR must now send the RSP(ACTPU) to the DLUS. In order to do this, it needs to allocate a conwinner conversation between its Send_Encap_Msg_TP and the Receive_Encap_Msg_TP at the DLUS node. Since no DLUR conwinner session already exists, the DLUR node uses the normal APPN flows to create such a session beginning with a Locate Find Cdinit being issued to the DLUS CP_LU=VTAM.
- ! 11. The DLUS node responds to the Locate with a Locate reply.
- ! 12. The NNS(DLUR) uses the information in the Locate reply to calculate a route to the DLUS node and sends this to the DLUR in the form of an RSCV attached to the Locate reply.
- ! 13. The DLUR node now issues a CPSVRMGR BIND to the DLUS node.
- ! 14. The DLUS node provides a BIND response.
- ! 15. After receiving the RSP(BIND), the DLUR node now sends an Attach for the Receive_Encap_Msg_TP at the DLUS.
- ! 16. Following the Attach, the DLUR node can send the encapsulated RSP(ACTPU) to the DLUS. It also includes the XID control vector as well as a DL CAP control vector identifying the DLUR's capabilities.
- ! 17. The DLUS component of the DLUS node removes the encapsulation headers and validates the capabilities. In this case, the DLUS examines the DLUR's capabilities and finds a mismatch with its own. Therefore, the DLUS will deactivate the CP-SVR pipe. This is begun by sending an UNBIND, including sense code X'088E 0008' in the CV X'35', on the conwinner session to its partner.

- ! 18. The DLUS also sends an UNBIND, including sense code X'088E 0008' in the CV X'35', on the conloser session to its partner.
- ! 19. The DLUR node sends a RSP(UNBIND) to the DLUS for the conwinner session and another RSP(UNBIND) for the conloser session. The SSCP-PU session and CP-SVR pipe are now deactivated. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.
- ! **5.6.2.2.2 DLUR-Detected Capabilities Mismatch (DLUS-DLUR Pipe):** The following figures show examples of DLUR/S capabilities mismatches detected by the DLUR during activation of a CP-SVR pipe between a DLUS and a DLUR.

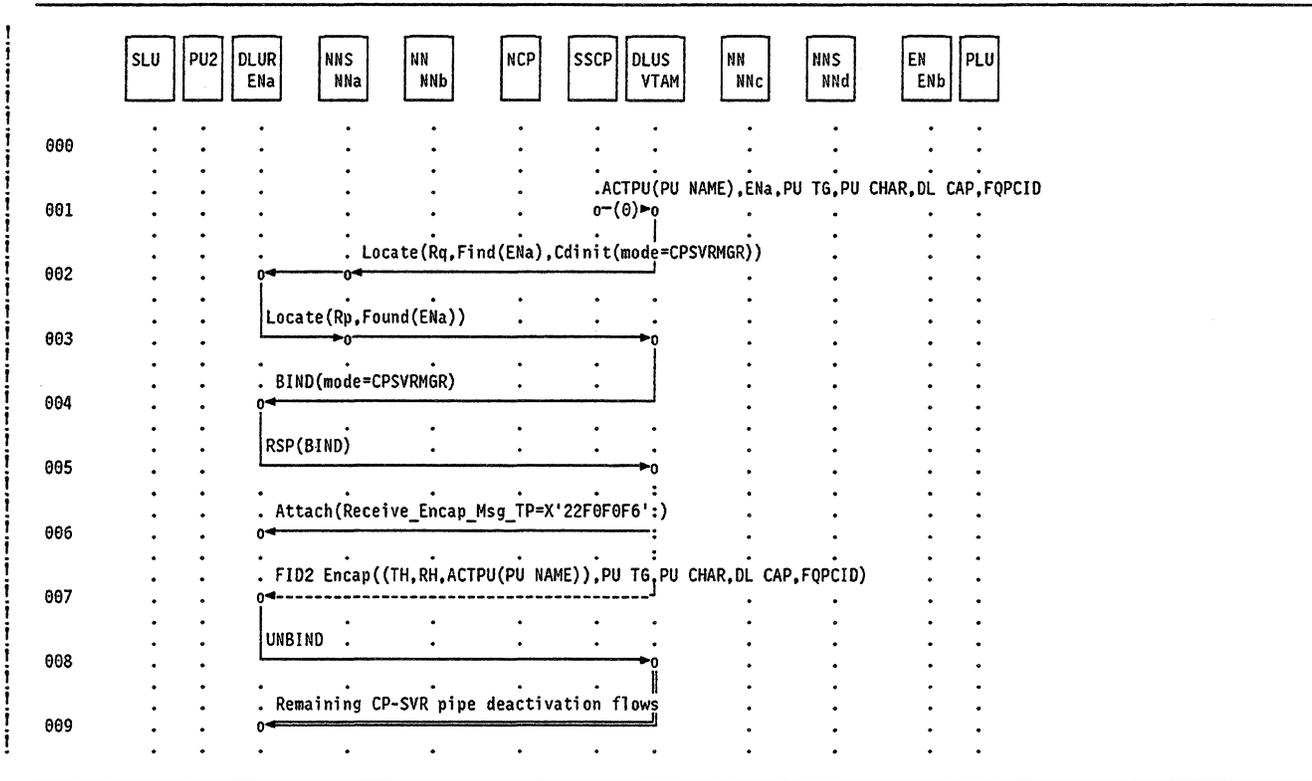


! Figure 5-9. DLUR-initiated, DLUR-rejected CP-SVR pipe activation



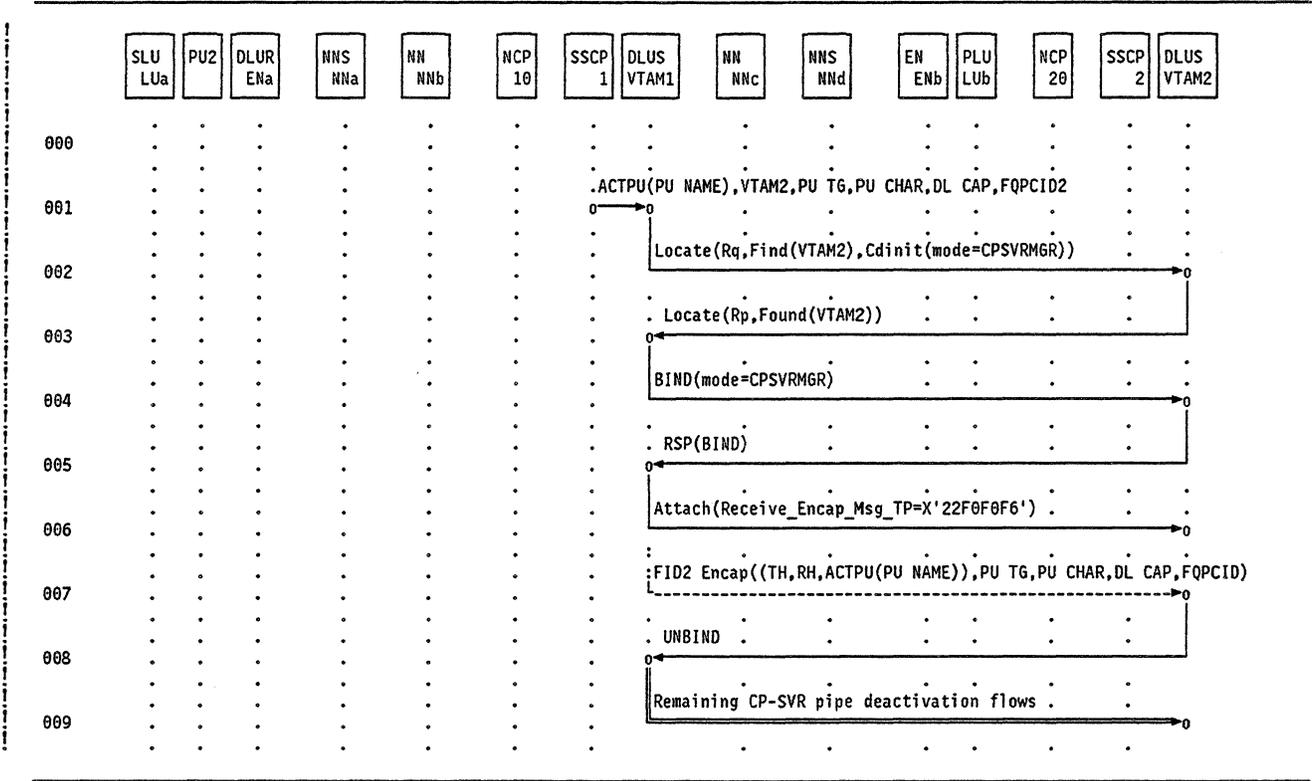
- ! 1. Upon receipt of some indication that a DLUR-supported PU requires activation (represented in flow 0), the DLUR node determines the DLUS node to be contacted on behalf of this PU. (See 5.3, "DLUR-Initiated CP-SVR Pipe Activation" on page 5-2) Once the DLUS node has been determined, the DLUR node Allocates a conwinner conversation between its Send_Encap_Msg_TP and the Receive_Encap_Msg_TP (X'22F0F0F6') at the DLUS node. Since no session using the CPSVRMGR mode between this DLUR and the DLUS already exists, the DLUR node initiates the normal APPN flow sequence to bring up such a session. This begins with the DLUR node sending a Locate Find Cdinit request to its network node server. The LU target of the session is the CPname of the DLUS node = VTAM in this case.
- ! 2. The NNS(DLUR) issues either a broadcast or directed search for the CP_LU of the DLUS = VTAM.
- ! 3. The DLUS node responds to the search request with a Locate Found reply.
- ! 4. The NNS(DLUR) uses the information in the Locate reply to calculate a route to the DLUS node and passes this back to the DLUR in the form of an RSCV.
- ! 5. The DLUR node issues a CPSVRMGR BIND to the DLUS node.
- ! 6. The DLUS node provides a BIND Response.
- ! 7. Following the BIND RSP, the DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
- ! 8. Once the Attach has been sent, the DLUR must send a REQACTPU RU up to the DLUS node. This will indicate to the DLUS node that an ACTPU is requested on behalf of some local or downstream PU. Before the REQACTPU RU is sent, however, the DLUR node must generate an FQPCID for this PU. This CV60, will be used to uniquely identify the SSCP-PU sessions when more than one exists on a given CP-SVR pipe. A FID2 Encapsulation GDS variable (X'1500') is built, including within it the FID2 PIU, XID image, PU TG CV, DL CAP CV, and FQPCID. The GDS variable is then sent up the conwinner CPSVRMGR session to the DLUS node.
- ! 9. When the DLUS component of the DLUS node receives the FID2 Encapsulation GDS variable, it removes the encapsulation headers and passes the information within the variable to the SSCP component of the DLUS node.
- ! 10. At this point, the DLUS node makes a determination whether it accepts the SSCP-PU session. In this case, the session is accepted and a +RSP to the REQACTPU RU is generated and sent back to the DLUS component of the DLUS node. Also included is another DL CAP control vector, this one identifying the DLUS's capabilities.
- ! 11. The DLUS must now send the +RSP to the REQACTPU RU to the DLUR node. To do this, the DLUS Send_Encap_Msg_TP allocates a conwinner conversation to the Receive_Encap_Msg_TP at the DLUR node. This causes the normal APPN session activation flows to occur between the DLUS and DLUR node. This begins with the DLUS node initiating a Locate Find Cdinit for the CPname of the DLUR = ENa in this case.
- ! 12. The DLUR responds to the Locate request with a Locate Found reply.
- ! 13. The DLUS node uses the information contained in the Locate reply to calculate a route to the DLUR node. It then issues a BIND to the DLUR node to establish its conwinner session with the DLUR node.
- ! 14. The DLUR node provides a BIND Response.
- ! 15. The DLUS node then sends an FMH5 Attach to initiate the Receive_Encap_Msg_TP at the DLUR node.

- ! 16. The DLUS node can now send its RSP(REQACTPU) to the DLUR. It encapsulates in the FID2 Encapsulation GDS variable the FID2 PIU, the DL CAP control vector, and the FQPCID (to identify the related request to the DLUR). In this case, it examines the DLUS's capabilities and finds a mismatch with its own.
- ! 17. Meanwhile, the DLUS SSCP has issued an ACTPU RU in response to the REQACTPU received from the DLUR.
- ! 18. Since a DLUR/S capabilities mismatch was detected, the DLUR will deactivate the CP-SVR pipe. This is begun by sending an UNBIND on the conwinner session to its partner, including X'088E 0009' in CV X'35'.
- ! 19. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
- ! 20. The DLUS component of the DLUS node then encapsulates the ACTPU RU FID2 PIU along with the PU TG control vector, PU CHAR control vector (used to define such PU characteristics as ANS=STOP|CONT), and the FQPCID of this SSCP-PU relationship with a FID2 Encapsulation GDS variable and sends the flow on its conwinner CPSVRMGR session to the DLUR.
- ! 21. The Receive_Encap_Msg_TP at the DLUR receives the encapsulated ACTPU message and removes the encapsulation headers. It uses the FQPCID information and PU TG control vector to identify the PU and its location. In this case, since it is in the process of deactivating the CP-SVR pipe due to a capabilities mismatch, it will discard the ACTPU.
- ! 22. The DLUR also sends an UNBIND on the conloser session to its partner, including X'088E 0009' in CV X'35'.
- ! 23. The DLUS node sends a RSP(UNBIND) to the DLUR for the conwinner session and another RSP(UNBIND) for the conloser session. The SSCP-PU session and CP-SVR pipe are now deactivated. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.



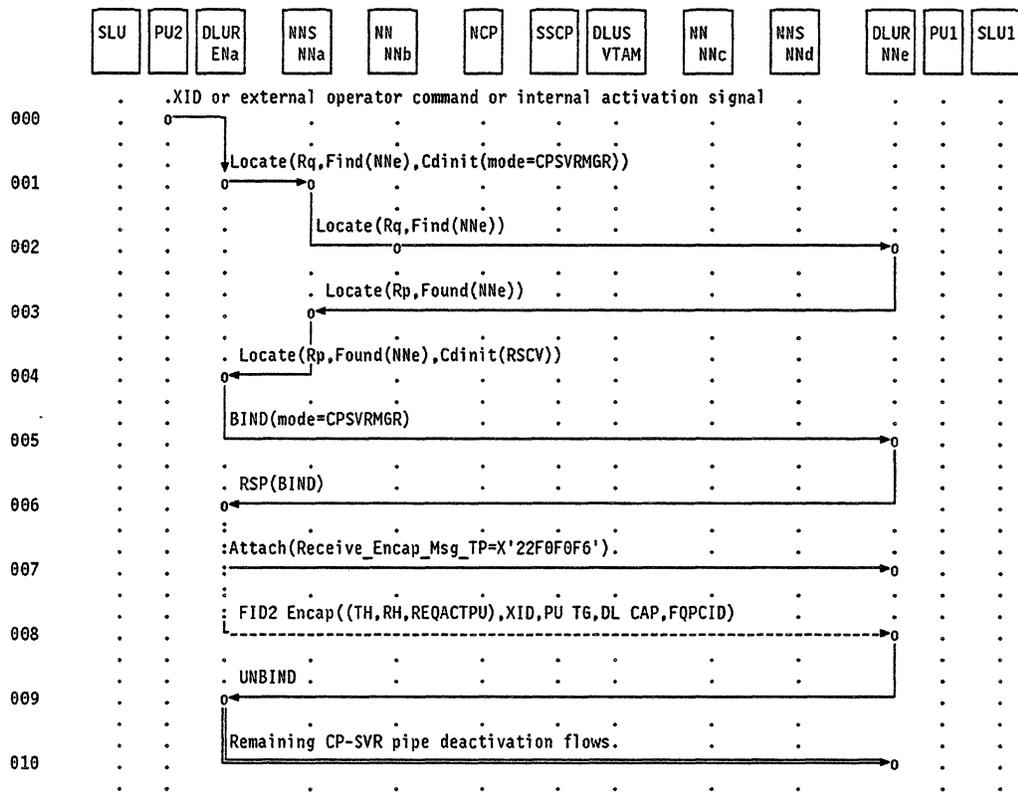
! Figure 5-10. DLUS-initiated, DLUR-rejected CP-SVR pipe activation. Footnotes:
 ! 0) Internal signal

- ! 1. When a DLUS node wishes to activate a PU, it uses its internal information (pre-defined or dynamically generated) to send an ACTPU signal to the DLUS component. This includes information about the PU, its connection to the DLUR, the DLUS's capabilities, and an FQPCID.
 - ! 2. The DLUS component uses this same information to determine the DLUR node associated with this PU. (See 5.4.1, "New PU Definition Parameters" on page 5-4 for more details.) It then allocates a conversation to the Receive_Encap_Msg_TP at the DLUR node. Since no session already exists to carry this conversation, the DLUS node initiates one starting with a Locate Find Cdinit for the CP_LU of the DLUR node = ENa.
 - ! 3. The DLUR node provides a Locate reply.
 - ! 4. The DLUS node uses the information in the Locate reply to calculate a route to the DLUR node and send a BIND using the CPSVRMGR mode.
 - ! 5. The DLUR node provides a BIND response.
 - ! 6. Once the BIND response has been received, the DLUS node sends an FMH5 to attach the Receive_Encap_Msg_TP at the DLUR node.
 - ! 7. The DLUS now follows the Attach with an encapsulated ACTPU FID2 PIU. This FID2 Encapsulation GDS variable carries with it an FQPCID that has been generated by the DLUS node to be used as an SSCP-PU session identifier for the DLUR and DLUS nodes, as well as the PU TG and DL CAP control vectors.
 - ! 8. When the DLUR node receives the encapsulated message, it removes the headers and validates the capabilities. In this case, it examines the DLUS's capabilities and finds a mismatch with its own. The DLUR must deactivate the CP-SVR pipe. This is begun by sending an UNBIND on the conloser session to its partner, including X'088E 0009' in CV X'35'.
 - ! 9. The DLUS node sends a RSP(UNBIND) to the DLUR for its conwinner session. The CP-SVR pipe is now deactivated. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.
- ! **5.6.2.2.3 DLUS-Detected Capabilities Mismatch (DLUS-DLUS Pipe):** The following figure shows an example of a DLUR/S capabilities mismatch detected by the DLUS during activation of a CP-SVR pipe between a DLUS and a DLUS.



! Figure 5-11. DLUS-initiated, DLUS-rejected CP-SVR pipe activation (DLUS-DLUS pipe)

- ! 1. When DLUS node VTAM1 wishes to activate a PU, it uses its internal information (pre-defined or dynamically generated) to send an ACTPU signal to the DLUS component. This includes information about the PU, its connection to the DLUR, the DLUS's capabilities, and an FQPCID.
 - ! 2. The DLUS component uses this same information to determine the DLUR node associated with this PU. (See 5.4.1, "New PU Definition Parameters" on page 5-4 for more details.) In this case the DLUR node was erroneously predefined as VTAM2. VTAM1 then allocates a conversation to the Receive_Encap_Msg_TP at VTAM2. Since no session already exists to carry this conversation, VTAM1 initiates one starting with a Locate Find Cdinit for the CP_LU of the DLUR node = VTAM2.
 - ! 3. VTAM2 provides a Locate reply.
 - ! 4. VTAM1 uses the information in the Locate reply to calculate a route to VTAM2 and send a BIND using the CPSVRMGR mode.
 - ! 5. VTAM2 provides a BIND response.
 - ! 6. Once the BIND response has been received, VTAM1 sends an FMH5 to attach the Receive_Encap_Msg_TP at VTAM2.
 - ! 7. VTAM1 now follows the Attach with an encapsulated ACTPU FID2 PIU. This FID2 Encapsulation GDS variable carries with it an FQPCID that has been generated by VTAM1 to be used as an SSCP-PU session identifier for the DLUR and DLUS nodes, as well as the PU TG, PU CHAR, and DL CAP control vectors.
 - ! 8. When VTAM2 receives the encapsulated message, it removes the headers and validates the capabilities. It recognizes a DLUR/S capabilities mismatch and deactivates the CP-SVR pipe. This is begun by sending an UNBIND on the conloser session to its partner, including X'088E 0008' in the CV X'35'.
 - ! 9. VTAM1 sends a RSP(UNBIND) to VTAM2 for its conwinner session. The CP-SVR pipe is now deactivated. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.
- ! **5.6.2.2.4 DLUR-Detected Capabilities Mismatch (DLUR-DLUR Pipe):** The following figure shows an example of a DLUR/S capabilities mismatch detected by the DLUR during activation of a CP-SVR pipe between a DLUR and a DLUR.



! Figure 5-12. DLUR-initiated, DLUR-rejected CP-SVR pipe activation (DLUR-DLUR pipe)

- ! 1. Upon receipt of some indication that a DLUR-supported PU requires activation (represented in flow 0), the DLUR node determines the DLUS node to be contacted on behalf of this PU. (See 5.3, "DLUR-Initiated CP-SVR Pipe Activation" on page 5-2) In this case, at ENa the DLUS name was erroneously predefined as NNe. ENa Allocates a conwinner conversation between its Send_Encap_Msg_TP and the Receive_Encap_Msg_TP (X'22F0F0F6') at NNe. Since no session using the CPSVRMGR mode between ENa and NNe already exists, ENa initiates the normal APPN flow sequence to bring up such a session. This begins with ENa sending a Locate Find Cdinit request to its network node server. The LU target of the session is the CPname of the DLUS node = NNe in this case.
- ! 2. The NNS(DLUR) issues either a broadcast or directed search for the CP_LU of the DLUS = NNe.
- ! 3. NNe responds to the search request with a Locate Found reply.
- ! 4. The NNS(DLUR) uses the information in the Locate reply to calculate a route to NNe and passes this back to ENa in the form of an RSCV.
- ! 5. ENa issues a CPSVRMGR BIND to NNe.
- ! 6. NNe provides a BIND Response.
- ! 7. Following the BIND RSP, ENa sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at NNe.
- ! 8. Once the Attach has been sent, ENa must send a REQACTPU RU up to NNe. This will indicate to NNe that an ACTPU is requested on behalf of some local or downstream PU. Before the REQACTPU RU is sent, however, ENa must generate an FQPCID for this PU. This CV60, will be used to uniquely identify the SSCP-PU sessions when more than one exists on a given CP-SVR pipe. A FID2 Encapsulation GDS variable (X'1500') is built, including within it the FID2 PIU, XID image, PU TG CV, DL CAP CV, and FQPCID. The GDS variable is then sent up the conwinner CPSVRMGR session to NNe.
- ! 9. When the DLUR component of NNe receives the FID2 Encapsulation GDS variable, it removes the encapsulation headers and examines the information within the variable. At this point, NNe finds a DLUR/S capabilities mismatch (DLUR-DLUR pipe) and deactivates the CP-SVR pipe. This is begun by sending an UNBIND on the conloser session to its partner ENa, including X'088E 0008' in CV X'35'.
- ! 10. ENa sends a RSP(UNBIND) to the DLUS for its conwinner session. The CP-SVR pipe is now deactivated. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.

5.6.3 Abnormal SSCP-PU/SSCP-LU Session Deactivation (DLUR-Initiated)

(For DLUS-initiated SSCP-PU and SSCP-LU session deactivation, see 6.4, "SSCP-PU/SSCP-LU Session Deactivation (DLUS-Initiated)" on page 6-28)

A DLUR with a downstream PU has a mechanism (REQACTPU) to communicate to its DLUS a request to start an SSCP-PU session to that downstream PU. There must be a similar mechanism available to the DLUR to signal to the SSCP that the DLUR has lost its connection with the PU, so that the SSCP can terminate the SSCP-PU session (and SSCP-LU session(s) if any).

This mechanism shall be a new RU, REQDACTPU, to be sent on the SSCP-PU session by the DLUR to request to the SSCP that it take down the SSCP-PU session as well as any associated SSCP-LU sessions. Since the DLUR will no longer have a connection to the downstream device, it will have to respond to the

DACTPU sent by the SSCP in response to the REQDACTPU. Once the SSCP-PU session has been deactivated, the DLUS can decide, using normal deactivation rules, whether or not to deactivate the CP-SVR % pipe. There are three different deactivation scenarios, each identified with different cause value in the % REQDACTPU RU:

- % • Cause X'00': downstream PU outage
- % • Cause X'01': receipt of REQDISCONT(normal) from downstream PU
- % • Cause X'02': receipt of REQDISCONT(immediate) from downstream PU

* **5.6.3.1 Downstream PU Outage**

- * In this scenario the DLUR BF detects a loss of connectivity with the downstream PU (there are several possible failures, e.g., polling timeouts and link failures to name two). This situation corresponds in the subarea BF to an INOPERATIVE condition.

Optionally, the DLUR can try to recover the downstream connection and, failing that, send the REQDACTPU, or it can send the REQDACTPU right away.

The following figure shows a downstream PU outage and the resulting SSCP-PU session and CP-SVR pipe deactivation:

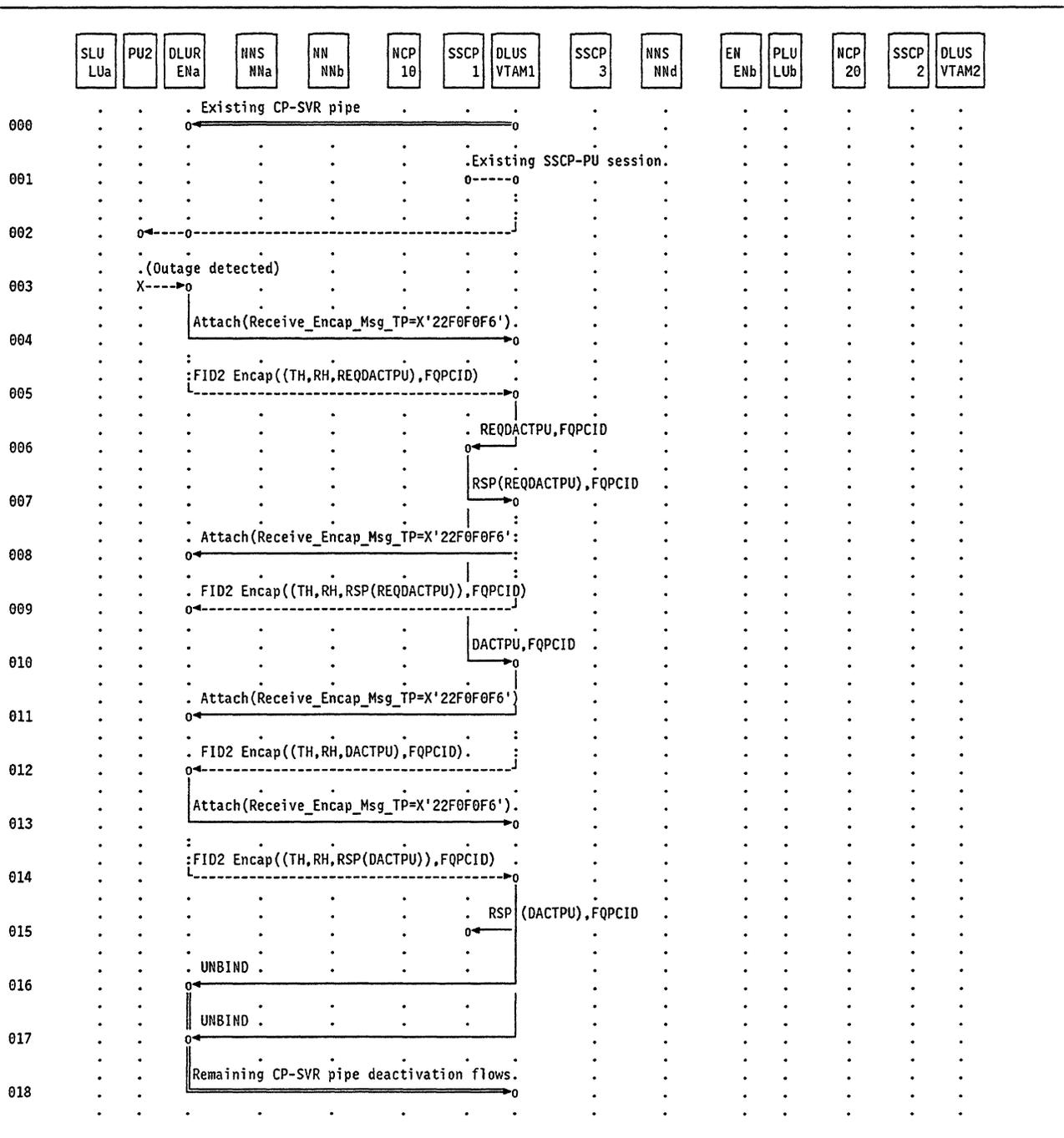


Figure 5-13. Downstream PU outage

1. There is an active session between SSCP1 and PU2. It is the only active SSCP-PU session using the CP-SVR pipe between VTAM1 and ENa.
- 2.
3. ENa detects an outage on its connection to PU2.
 - **Note** - If there had been an active LU-LU session involving LUa, ENa would build a Format 3 SESSEND RU, encapsulate it, and forward it to VTAM1 on the CP-SVR pipe. VTAM1 would then de-encapsulate the SESSEND and forward it to SSCP1.
4. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
- % 5. To signal to SSCP1 that its SSCP-PU session with PU2 is broken, ENa builds a REQDACTPU with a
% cause value of X'00', encapsulates it, and forwards it to VTAM1 on its CP-SVR pipe.
6. VTAM1 de-encapsulates the REQDACTPU and forwards it to SSCP1.
7. SSCP1 responds to the REQDACTPU positively, passing the RSP(REQDACTPU) on to VTAM1.
8. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
9. VTAM1 encapsulates the RSP(REQDACTPU) and forwards it to ENa.
10. SSCP1 builds a DACTPU and passes it on to VTAM1.
11. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
12. VTAM1 encapsulates the DACTPU and forwards it to ENa.
13. ENa de-encapsulates the DACTPU, and since it no longer has a connection to PU2, converts the DACTPU into a positive RSP(DACTPU). The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
14. ENa then encapsulates the RSP(DACTPU) and sends it to VTAM1.
15. VTAM1 de-encapsulates the RSP(DACTPU) and forwards it to SSCP1.
16. VTAM1 also determines that there are now no active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it sends an UNBIND to ENa on VTAM1's conwinner session to begin deactivation of the pipe.
17. VTAM1 also sends an UNBIND on its conloser session to ENa.
18. The DLUR node sends a RSP(UNBIND) to the DLUS for the conwinner session and another RSP(UNBIND) for the conloser session. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.

* 5.6.3.2 Receipt Of REQDISCONT From Downstream PU

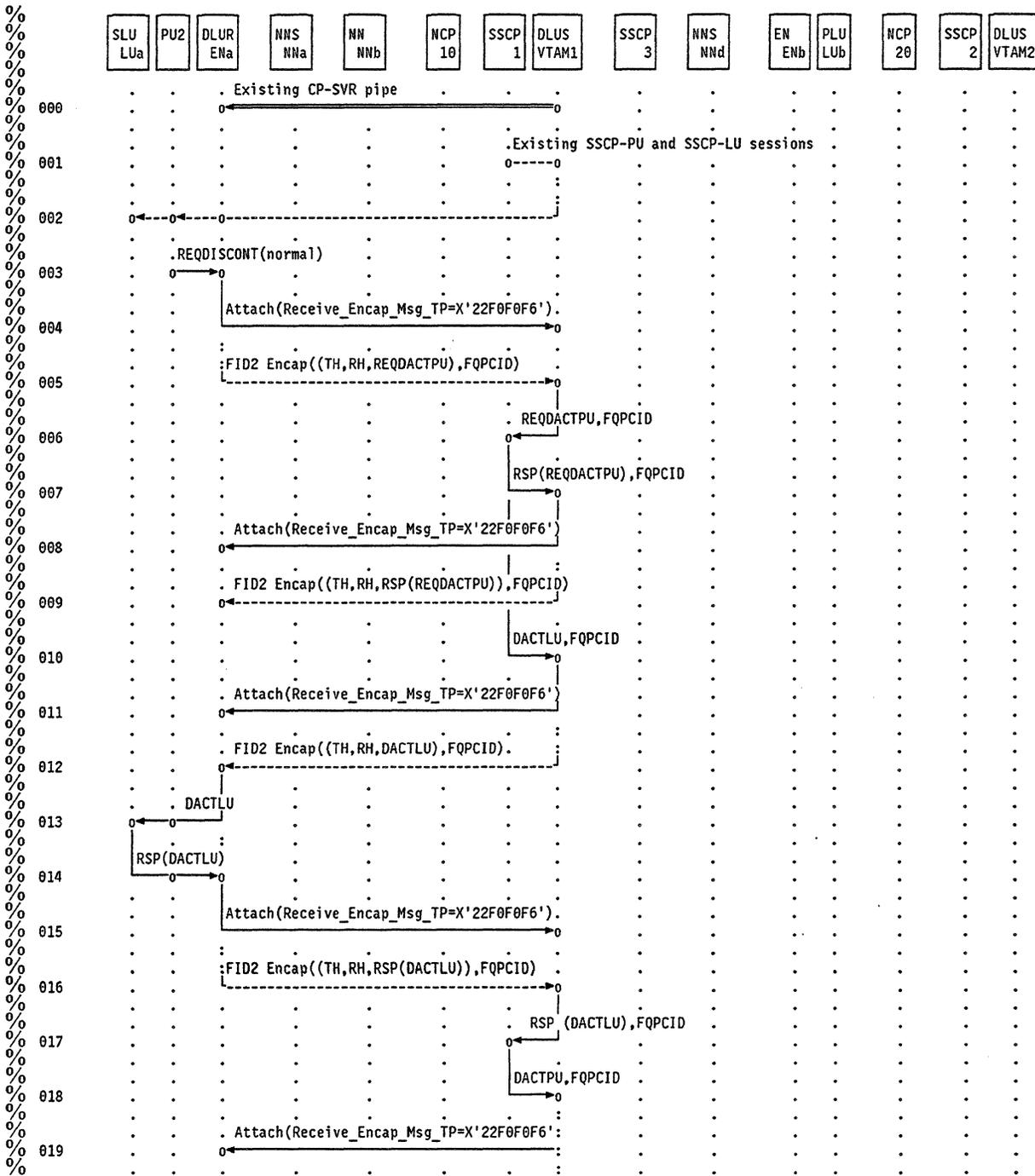
% In these scenarios the DLUR BF receives a REQDISCONT RU from the downstream PU. In this situation the DLUR will:

- % • convert the REQDISCONT to a REQDACTPU, indicating a cause value of X'01' if the
% REQDISCONT type is normal and X'02' if the type is immediate
- % • encapsulate the REQDACTPU and send it to the DLUS on the CP-SVR pipe
- ! • allow the DLUS to deactivate the PU gracefully if type = normal or forced if type = immediate (see
! 6.4, "SSCP-PU/SSCP-LU Session Deactivation (DLUS-Initiated)" on page 6-28 for more information
! on normal and forced SSCP-PU and SSCP-LU session deactivation), and then

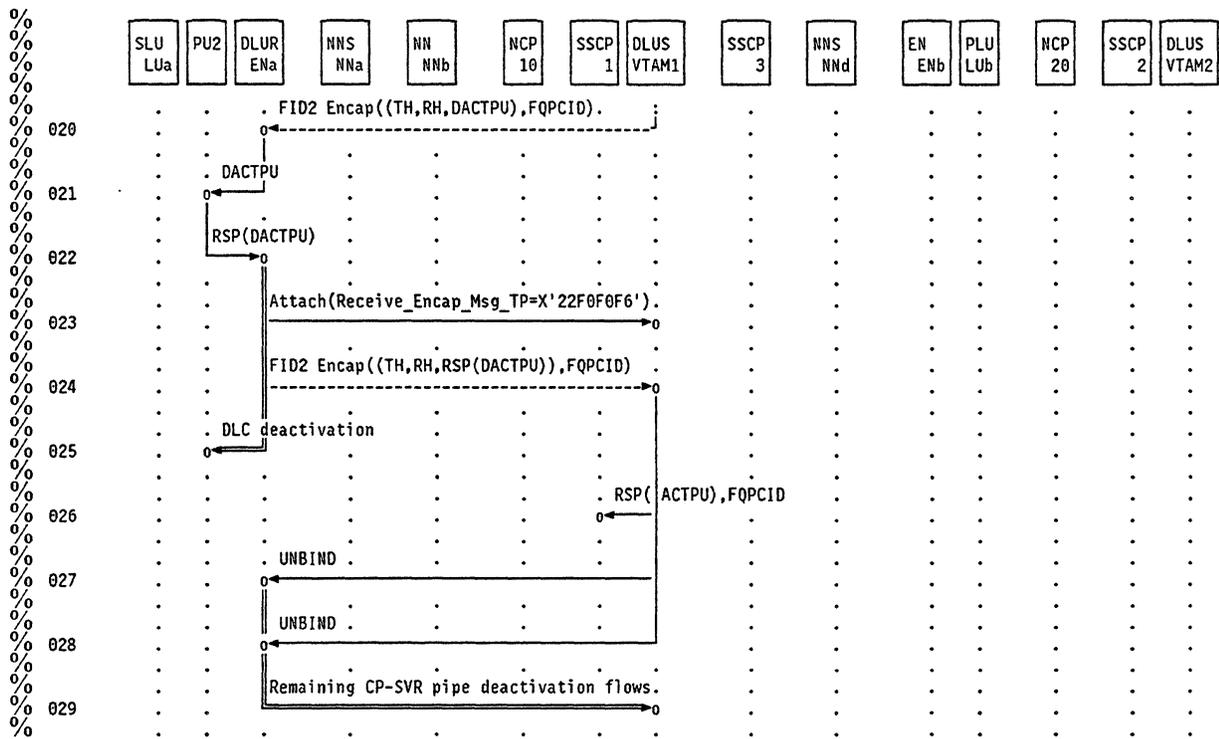
Unclassified

% • deactivate the DLC from the BF to the DSPU (emulating the receipt of a DISCONTACT)

% The following figure shows receipt of a REQDISCONT(normal) and the resulting SSCP-PU/LU session and CP-SVR pipe deactivation:

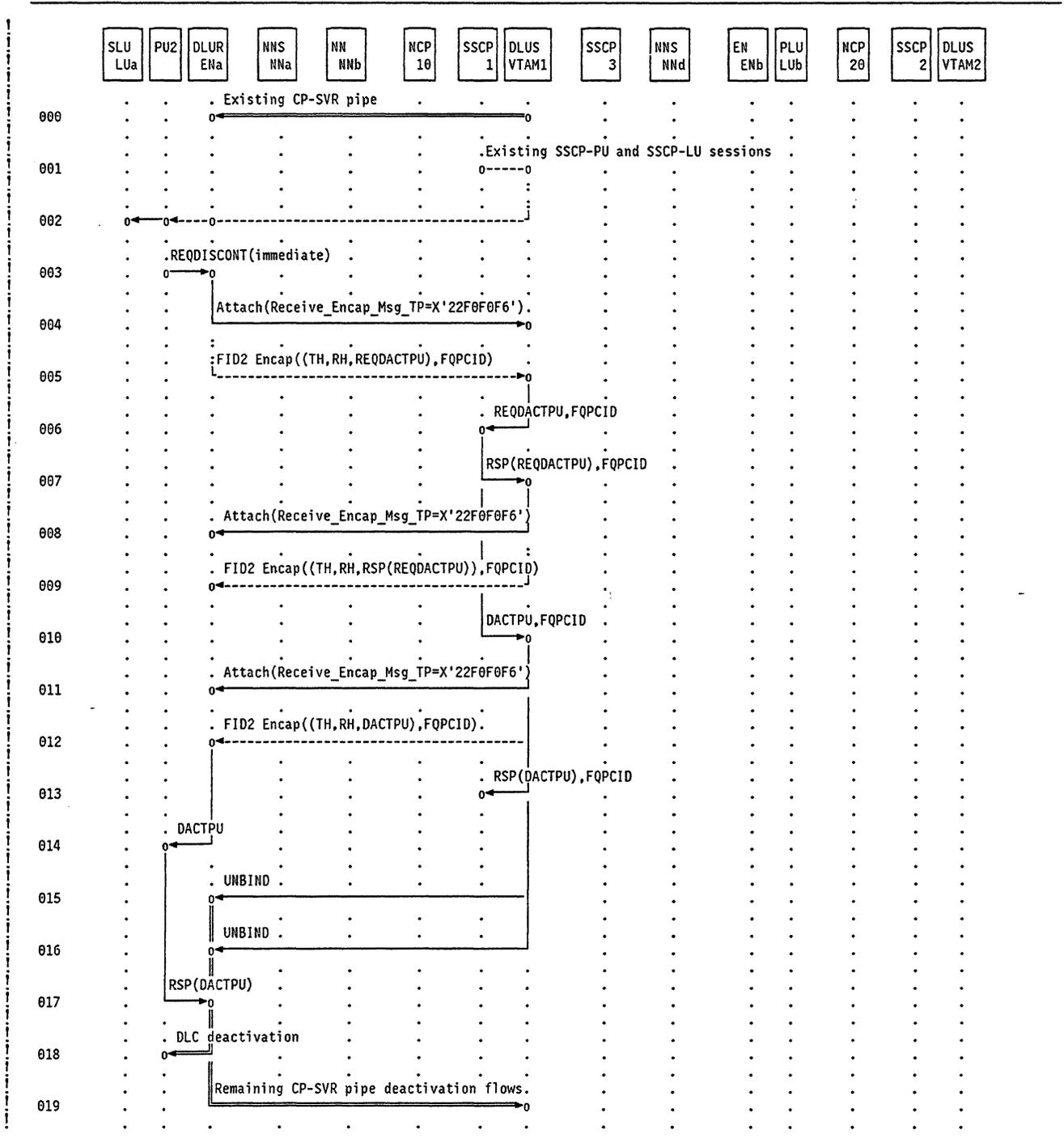


% Figure 5-14. Receipt of REQDISCONT(normal) from downstream PU



- % 1. There are active sessions between SSCP1 and PU2 and SSCP1 and LUa. These are the only active SSCP-PU and SSCP-LU sessions using the CP-SVR pipe between VTAM1 and ENa.
- * 2.
- * 3. ENa receives a REQDISCONT RU from PU2.
- * 4. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
- % 5. To signal to SSCP1 that its SSCP-PU session with PU2 is broken, ENa builds a REQDACTPU with a cause value of X'01', encapsulates it, and forwards it to VTAM1 on its CP-SVR pipe.
- * 6. VTAM1 de-encapsulates the REQDACTPU and forwards it to SSCP1.
- * 7. SSCP1 responds to the REQDACTPU positively, passing the RSP(REQDACTPU) on to VTAM1.
- * 8. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
- * 9. VTAM1 encapsulates the RSP(REQDACTPU) and forwards it to ENa.
- % 10. SSCP1 builds a DACTLU and passes it on to VTAM1.
- % 11. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
- % 12. VTAM1 encapsulates the DACTLU and forwards it to ENa.
- % 13. ENa de-encapsulates the DACTLU and forwards it to PU2 and then to LUa.
- % 14. LUa deactivates the SSCP-LU session, converts the DACTLU into a positive RSP(DACTLU), and forwards it to PU2 and then to ENa.
- % 15. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
- % 16. ENa then encapsulates the RSP(DACTLU) and sends it to VTAM1.
- % 17. VTAM1 de-encapsulates the RSP(DACTLU) and forwards it to SSCP1.
- * 18. SSCP1 builds a DACTPU and passes it on to VTAM1.
- * 19. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
- * 20. VTAM1 encapsulates the DACTPU and forwards it to ENa.
- % 21. ENa de-encapsulates the DACTPU and forwards it to PU2.
- % 22. PU2 deactivates the SSCP-PU session, converts the DACTPU into a positive RSP(DACTPU), and forwards it to ENa.
- % 23. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
- % 24. ENa then encapsulates the RSP(DACTPU) and sends it to VTAM1.
- % 25. ENa initiates DLC deactivation to PU2.
- * 26. VTAM1 de-encapsulates the RSP(DACTPU) and forwards it to SSCP1.
- * 27. VTAM1 also determines that there are now no active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it sends an UNBIND to ENa on VTAM1's conwinner session to begin deactivation of the pipe.
- * 28. VTAM1 also sends an UNBIND on its conloser session to ENa.
- * 29. The DLUR node sends a RSP(UNBIND) to the DLUS for the conwinner session and another RSP(UNBIND) for the conloser session. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.

% The following figure shows receipt of a REQDISCONT(immediate) and the resulting SSCP-PU/LU session and CP-SVR pipe deactivation:



! Figure 5-15. Receipt of REQDISCONT(immediate) from downstream PU

- % 1. There are active sessions between SSCP1 and PU2 and SSCP1 and LUa. These are the only active SSCP-PU and SSCP-LU sessions using the CP-SVR pipe between VTAM1 and ENa.
- % 2.
- % 3. ENa receives a REQDISCONT RU from PU2.
- % 4. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
- % 5. To signal to SSCP1 that its SSCP-PU session with PU2 is broken, ENa builds a REQDACTPU with a cause value of X'02', encapsulates it, and forwards it to VTAM1 on its CP-SVR pipe.
- % 6. VTAM1 de-encapsulates the REQDACTPU and forwards it to SSCP1.
- % 7. SSCP1 responds to the REQDACTPU positively, passing the RSP(REQDACTPU) on to VTAM1.
- % 8. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
- % 9. VTAM1 encapsulates the RSP(REQDACTPU) and forwards it to ENa.
- % 10. SSCP1 builds a DACTPU and passes it on to VTAM1.
- % 11. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
- % 12. VTAM1 encapsulates the DACTPU and forwards it to ENa.
- ! 13. VTAM1 builds a RSP(DACTPU) and forwards it to SSCP1, not waiting for the response to return from the PU.
- ! 14. ENa de-encapsulates the DACTPU and forwards it to PU2.
- ! 15. VTAM1 determines that there are now no active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it sends an UNBIND to ENa on VTAM1's conwinner session to begin deactivation of the pipe.
- ! 16. VTAM1 also sends an UNBIND on its conloser session to ENa.
- ! 17. PU2 deactivates the SSCP-PU and associated SSCP-LU session. It converts the DACTPU into a positive RSP(DACTPU) and returns it to ENa.
- ! 18. ENa initiates DLC deactivation to PU2, and discards the RSP(DACTPU) since the CP-SVR pipe is in the process of being deactivated.
- % 19. The DLUR node sends a RSP(UNBIND) to the DLUS for the conwinner session and another RSP(UNBIND) for the conloser session. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.

5.7 Pipe And Session Failure Logic

As mentioned above, CP-SVR pipes will actually consist of two CPSVRMGR sessions between the DLUR and DLUS CPs. The SSCP-PU and SSCP-LU sessions will be encapsulated on the two CPSVRMGR sessions such that flows from the SSCP to the PU/LU will flow on the conwinner session from DLUS to DLUR node, and flows from the PU/LU to the SSCP will flow on the conwinner session from DLUR to DLUS node. Moreover, each node (DLUR and DLUS) will compute the session route independently. It is, therefore, possible that the two CPSVRMGR sessions will not traverse the same physical path. Consequently, one direction of the SSCP-PU and SSCP-LU sessions may fail and not the other. Even if the same physical path is used, it is still possible for one session to fail and not the other. This will necessitate special handling on the part of DLUS and DLUR nodes.

5.7.1 CPSVRMGR Session Recovery

If a CPSVRMGR session fails, it is the responsibility of both the DLUS and the DLUR to bring down its conwinner and conloser sessions with a non-retryable sense code. At this point, either or both nodes may attempt to reinstate the CP-SVR pipe and recover the SSCP-PU/LU sessions.

The DLUR node may chose to attempt to reactivate its dependent LUs by activating a CP-SVR pipe (and subsequent SSCP-PU and SSCP-LU sessions) with the same or some other DLUS node. The DLUR node may only attempt this after both the conwinner and conloser sessions have been successfully deactivated. Otherwise, the DLUR node may activate a conwinner session with one DLUS node while a conloser session exists with some other DLUS node. Alternative DLUS nodes may be either dynamically discovered (as described in 5.3.2, "Dynamic DLUS Node Determination" on page 5-3) or pre-defined (as described in 5.3.1, "Pre-Defined DLUS Node Determination" on page 5-2).

As previously described in 5.6.2.1, "UNBIND-Initiated CP-SVR Pipe Deactivation" on page 5-25, the DLUR may reactivate immediately after completion of the CP-SVR pipe deactivation, with one exception (receipt of X'08A0 000A' sense code with UNBIND requires the DLUR to wait for a DLUS to activate the CP-SVR pipe; DLUR can wait indefinitely or set a timeout, after which the DLUR can activate a CP-SVR pipe with the same or a different DLUS).

It is also possible that the DLUS node may try to re-activate the dependent LUs by re-establishing a CP-SVR pipe to the DLUR node as described in 5.4, "DLUS-Initiated CP-SVR Pipe Activation" on page 5-4.

5.7.2 SSCP-PU And SSCP-LU Session Recovery

When both of the CPSVRMGR sessions are lost, the DLUS and DLUR nodes must recognize this situation. Ongoing LU-LU sessions may remain active but will not be able to obtain SSCP services when required. This is functionally equivalent to existing Subarea operation. The SSCP-PU and SSCP-LU sessions can be recovered through SSCP takeover (i.e., DLUS takeover) flows which reset the SSCP-PU and SSCP-LU sessions as described in 10.2.2, "SSCP Takeover & Giveback Flows" on page 10-11. When the DLUR attempts to reactivate the SSCP-PU and SSCP-LU sessions, it can choose the same or another DLUS node. If the reason for session deactivation was the DLUR receiving a DACTPU with a "giveback" reason code, the DLUR should try to get in session with a new DLUS.

5.7.3 Pipe And Session Activation/Recovery Race Conditions

5.7.3.1 Multi-DLUS Race Conditions

Race conditions can exist where more than one DLUS may be involved in CP-SVR pipe activation for the same dependent LU. For example:

- *initial CP-SVR pipe activation* - a pipe is being activated and an activation request for a second pipe is received
- *CP-SVR pipe override* - a pipe to a backup DLUS is active and the pre-defined DLUS requests activation of a pipe
- *CP-SVR pipe recovery* - a pipe to a backup DLUS is being activated and the pre-defined DLUS requests activation of a pipe

In these situations, the DLUR node is responsible for deciding with which DLUS it will establish the CP-SVR pipe for a given PU. In doing so, the DLUR will use the following rules:

- If the DLUR has a choice between an active pipe and a request to activate a pipe, it should choose the DLUS with the active pipe.
- If the DLUR has a choice between two requests for activation, the implementing product can decide on its DLUS selection criteria:
 - "known" (pre-defined) vs. "determined" (found) - select "known"
 - "determined" vs. "determined" - product option to choose one

Once the race condition is resolved, the DLUR node is responsible for rejecting the CP-SVR pipe with one of the two DLUS nodes. This is done by sending -RSP to one of the ACTPU messages. The DLUS receiving the -RSP(ACTPU) will then decide whether to take down the CP-SVR pipe using the same criteria as normal CP-SVR pipe deactivation, i.e., if there are no other active or pending SSCP-PU sessions using the CP-SVR pipe, the DLUS will UNBIND its CPSVRMGR conwinner and conloser sessions.

The following figure shows an unsuccessful attempt at overriding an existing CP-SVR pipe:

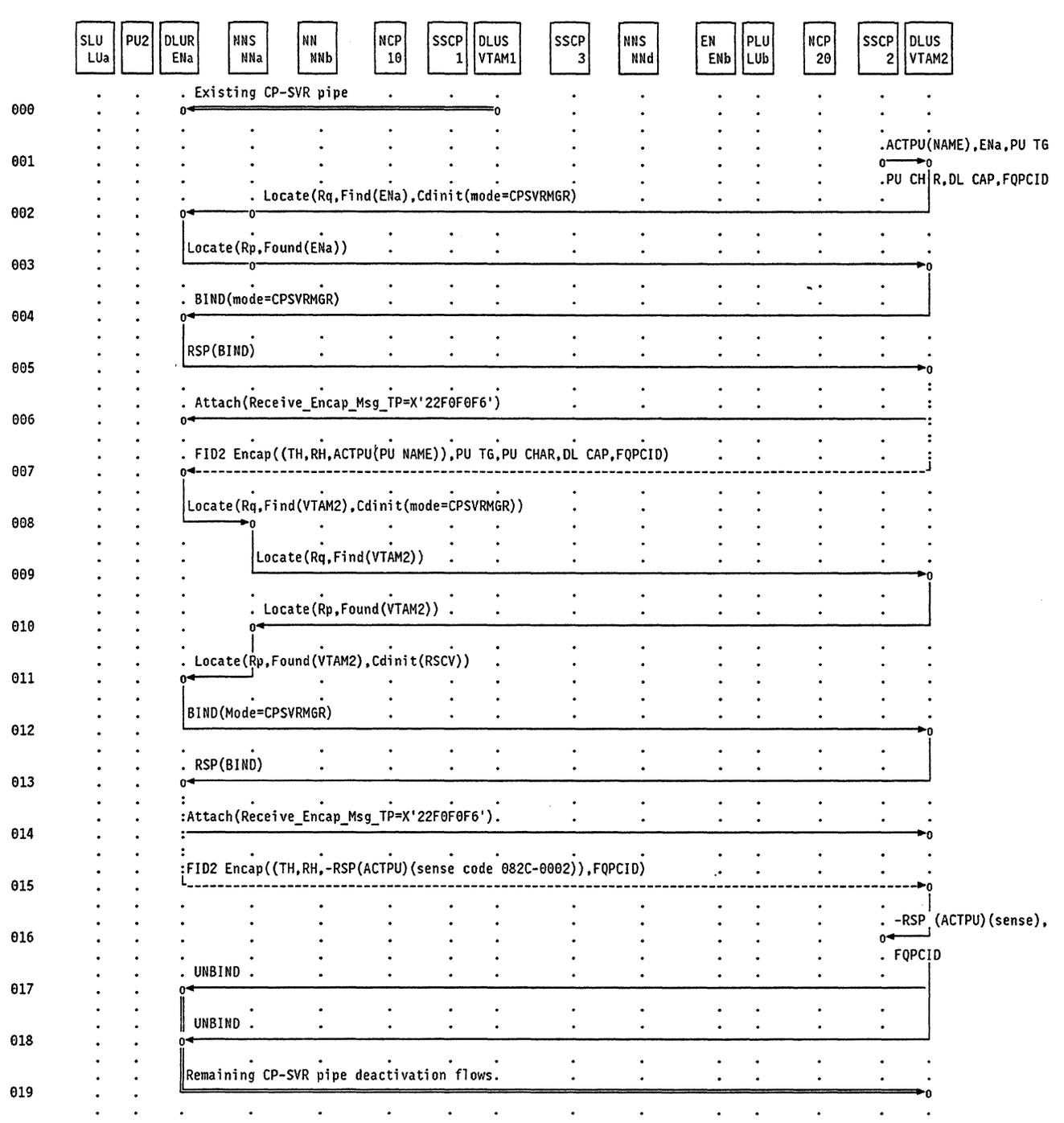


Figure 5-16. CP-SVR pipe override (rejected)

1. There is an active CP-SVR pipe between VTAM1 and ENa, where VTAM1 is a "known" DLUS. SSCP1 and PU2 have an active SSCP-PU session. SSCP2 builds an ACTPU to start an SSCP-PU session with PU2 and forwards the RU to its DLUS, VTAM2.
2. VTAM2 locates the DLUR ENa.
- 3.
4. VTAM2 activates its CPSVRMGR conwinner session with ENa.
- 5.
- 6.
7. VTAM2 encapsulates the ACTPU and sends it to ENa. This ACTPU is from SSCP2 via VTAM2, which to ENa is a "determined" DLUS. Since there is already an active SSCP-PU session for PU2 (doesn't matter whether it is with a "known" or "determined" DLUS), the new ACTPU will be rejected with a sense code of X'082C 0002'.
8. ENa locates the DLUS VTAM2.
- 9.
- 10.
- 11.
12. ENa activates its CPSVRMGR conwinner session with VTAM2.
- 13.
- 14.
15. ENa sends a negative RSP(ACTPU) to terminate the pending SSCP-PU session.
- 16.
17. VTAM2 also determines that there are now no active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it sends an UNBIND to ENa on VTAM2's conwinner session to begin deactivation of the pipe.
18. VTAM2 also sends an UNBIND on its conloser session to ENa.
19. The DLUR node sends a RSP(UNBIND) to the DLUS for the conwinner session and another RSP(UNBIND) for the conloser session. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.

5.7.3.2 Single DLUS Race Conditions

Race conditions can exist between a DLUR and one DLUS. For example:

- *CP-SVR pipe deactivation/SSCP-PU session activation race* - a DLUR has just sent a REQACTPU to the DLUS, and it receives an UNBIND for the CP-SVR pipe with that DLUS
 - the DLUR has two options, equivalent to the Auto Network Shutdown function (user preference could be optionally be pre-defined per PU):
 - the DLUR could deactivate the CP-SVR pipe and subsequently redrive reactivation of that pipe for the pending SSCP-PU session, retransmitting the REQACTPU on the new pipe (similar to ANS=CONT)
 - the DLUR could deactivate the CP-SVR pipe and the connection to the PU (similar to ANS=STOP)
- *"surprise" CP-SVR pipe activation* - a pipe is active and an activation request for the same pipe is received from the same DLUS
 - the activation request should come from a DLUS attempting to recover a CP-SVR pipe it has determined has gone down; since the DLUR has not yet determined that the pipe is down, the new activation request should be rejected (no available conwinner or conloser sessions to that DLUS) - eventually the DLUR will determine that the pipe needs to be deactivated (for instance, receipt of UNBIND from DLUS) and will complete the pipe deactivation and then drive pipe reactivation
- ! • *SSCP-PU session deactivation race (DACTPU/REQDACTPU)* - a node (either requester or server) is processing or has sent an SSCP-PU session deactivation request (DACTPU from the DLUS; REQDACTPU from the DLUR), and it receives an SSCP-PU session deactivation request for the same PU from the other node
 - ! – DLUR processing
 - ! – if the requester is in an awaiting RSP(REQDACTPU) state and it receives a DACTPU from the DLUS for that PU, it should accept and process that DACTPU request
 - ! – if while processing the DACTPU, the +RSP(REQDACTPU) should arrive, the requester should accept the response and continue DACTPU processing

The following figures show examples of these race conditions:

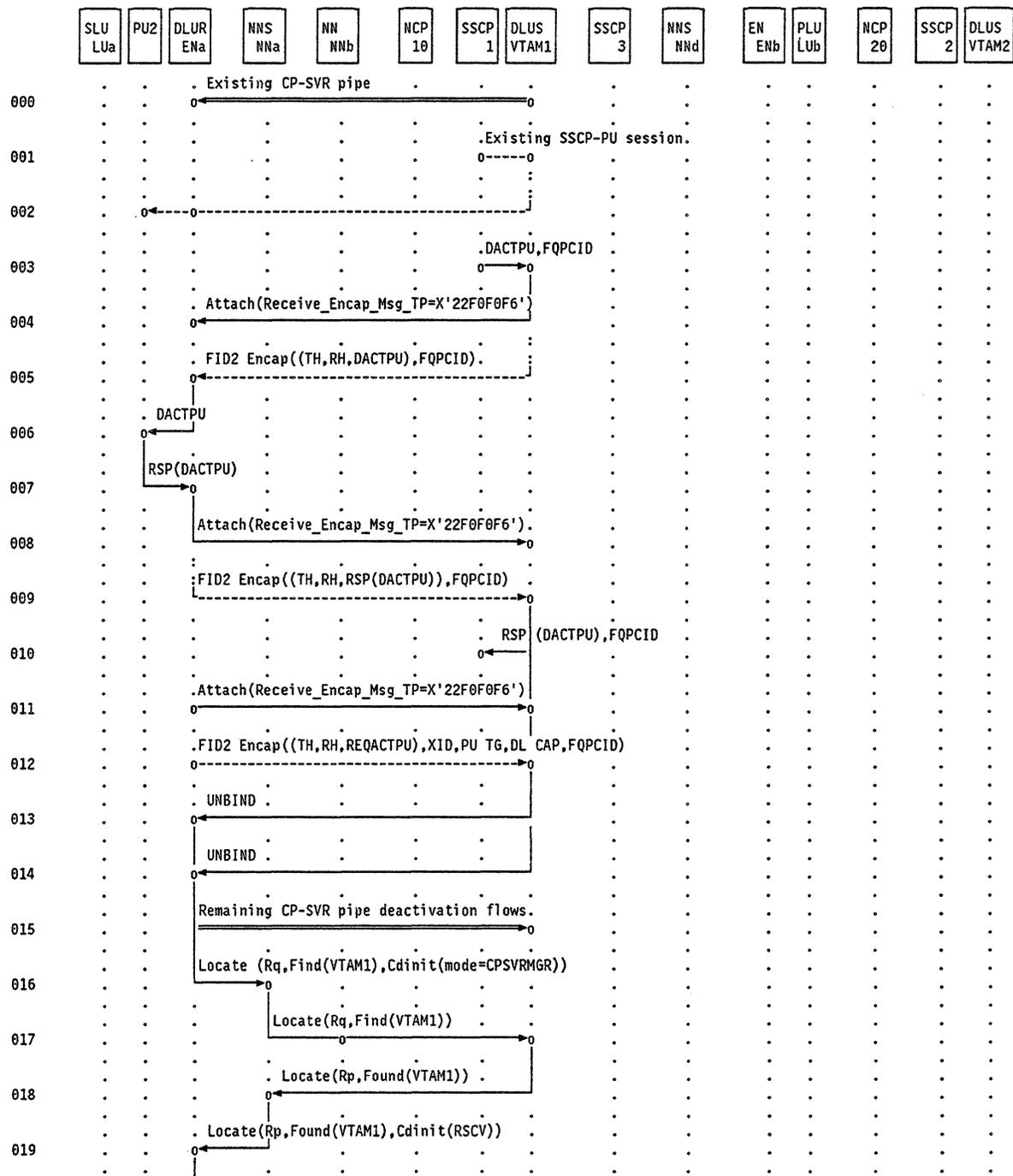
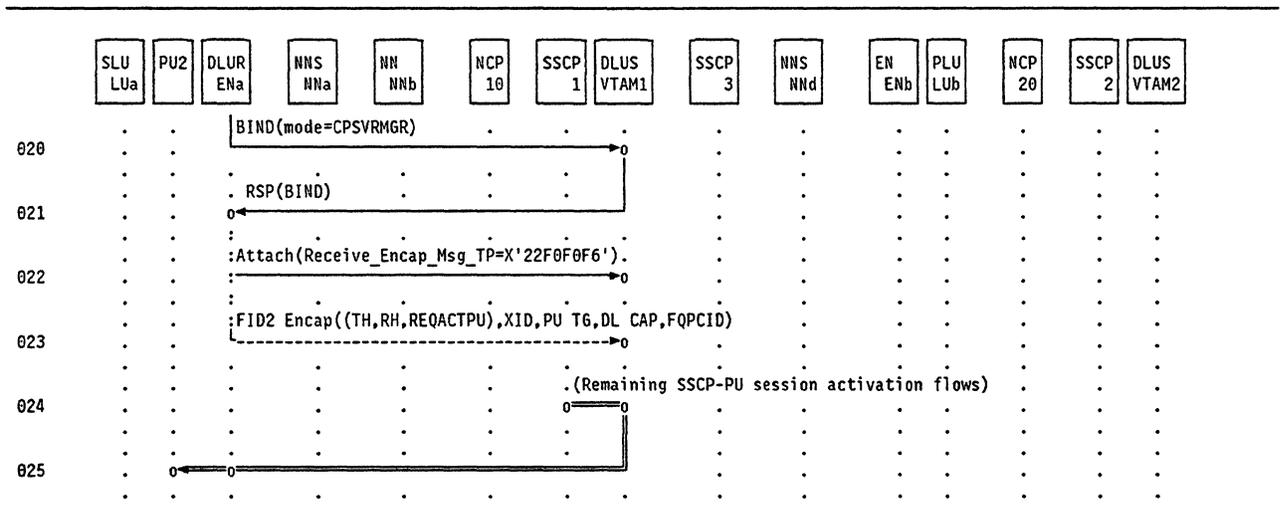


Figure 5-17. CP-SVR pipe deactivation/SSCP-PU session activation race - redrive pipe activation



1. There is an active CP-SVR pipe between VTAM1 and ENa, where VTAM1 is a "known" DLUS. SSCP1 and PU2 have an active SSCP-PU session. It is the only active SSCP-PU session using the CP-SVR pipe between VTAM1 and ENa. It has been indicated that PU2 session with SSCP1 should be restarted in case of pipe failure.
- 2.
3. SSCP1 initiates normal deactivation of its session with PU2 by building a DACTPU and passing it on to VTAM1.
4. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
5. The DLUS VTAM1 encapsulates the DACTPU and forwards it to the DLUR ENa.
6. The DLUR ENa de-encapsulates the DACTPU and forwards it to PU2.
7. PU2 returns a positive RSP(DACTPU) to SSCP1 via ENa.
8. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
9. ENa encapsulates the RSP(DACTPU) and sends it to VTAM1.
10. VTAM1 de-encapsulates the RSP(DACTPU) and forwards it to SSCP1.
11. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
12. ENa encapsulates and sends to VTAM1 a REQACTPU RU.
13. Meanwhile VTAM1 determines that there are now no active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it sends an UNBIND to ENa on VTAM1's conwinner session to begin deactivation of the pipe.
14. VTAM1 also sends an UNBIND on its conloser session to ENa.
15. The DLUR node sends a RSP(UNBIND) to the DLUS for the conwinner session and another RSP(UNBIND) for the conloser session. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.
16. ENa determines that there was a pending SSCP-PU session on the pipe, and it restarts activation of the CP-SVR pipe to VTAM1.
- 17.
- 18.
- 19.
- 20.
- 21.
22. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
23. ENa encapsulates and sends to VTAM1 the REQACTPU RU sent originally on the previous pipe.
24. SSCP-PU session reactivation proceeds normally.
- 25.

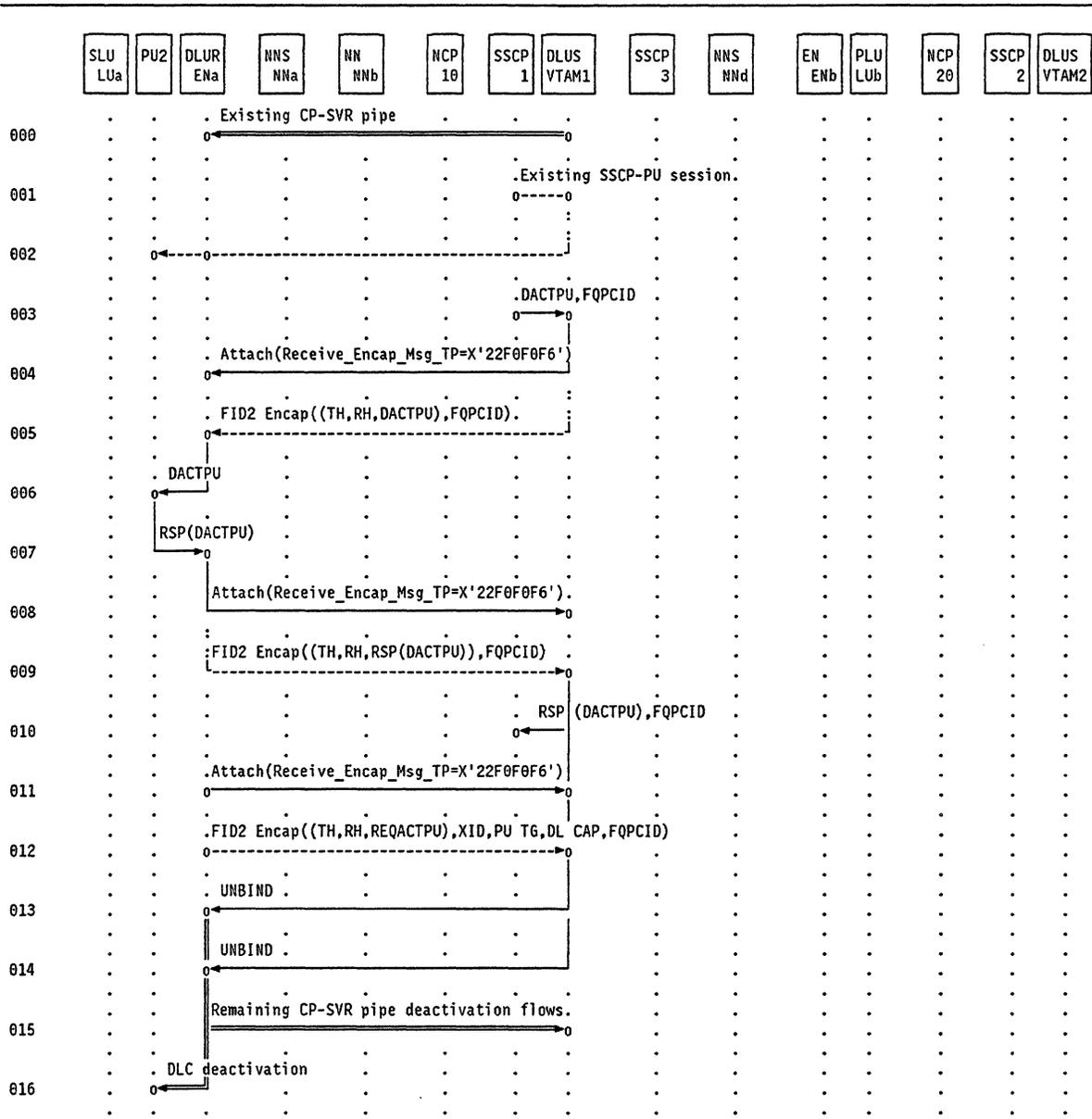


Figure 5-18. CP-SVR pipe deactivation/SSCP-PU session activation race - do not redrive pipe activation

1. There is an active CP-SVR pipe between VTAM1 and ENa, where VTAM1 is a "known" DLUS. SSCP1 and PU2 have an active SSCP-PU session. It is the only active SSCP-PU session using the CP-SVR pipe between VTAM1 and ENa. It has been indicated that PU2 session with SSCP1 should not be restarted in case of pipe failure.
- 2.
3. SSCP1 initiates normal deactivation of its session with PU2 by building a DACTPU and passing it on to VTAM1.
4. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
5. The DLUS VTAM1 encapsulates the DACTPU and forwards it to the DLUR ENa.
6. The DLUR ENa de-encapsulates the DACTPU and forwards it to PU2.
7. PU2 returns a positive RSP(DACTPU) to SSCP1 via ENa.
8. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
9. ENa encapsulates the RSP(DACTPU) and sends it to VTAM1.
10. VTAM1 de-encapsulates the RSP(DACTPU) and forwards it to SSCP1.
11. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
12. ENa encapsulates and sends to VTAM1 a REQACTPU RU.
13. Meanwhile VTAM1 determines that there are now no active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it sends an UNBIND to ENa on VTAM1's conwinner session to begin deactivation of the pipe.
14. VTAM1 also sends an UNBIND on its conloser session to ENa.
15. The DLUR node sends a RSP(UNBIND) to the DLUS for the conwinner session and another RSP(UNBIND) for the conloser session. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.
16. ENa determines that there was a pending SSCP-PU session on the pipe, but it does not attempt to redrive the activation request. Instead it terminates the downstream PU connection.

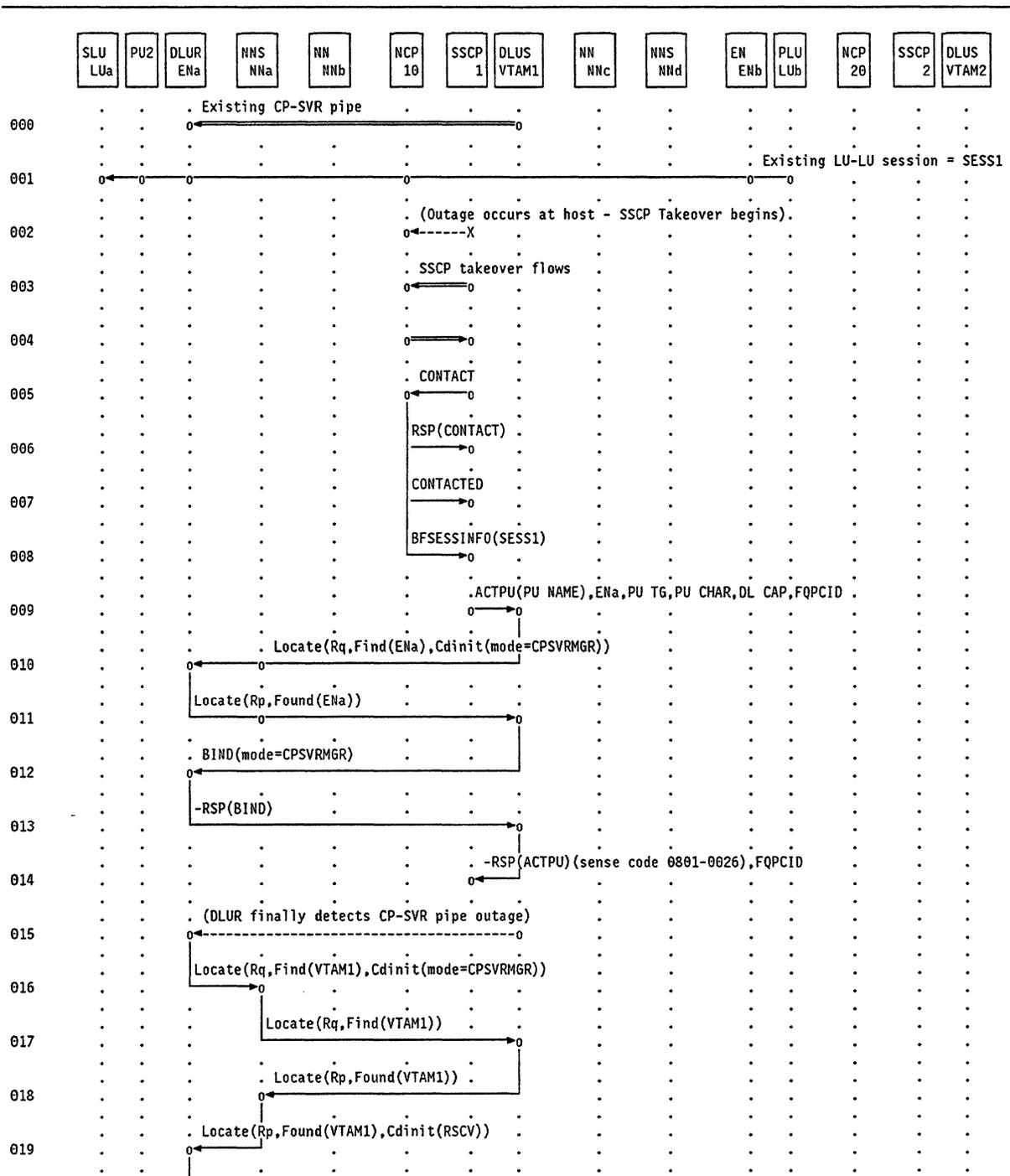
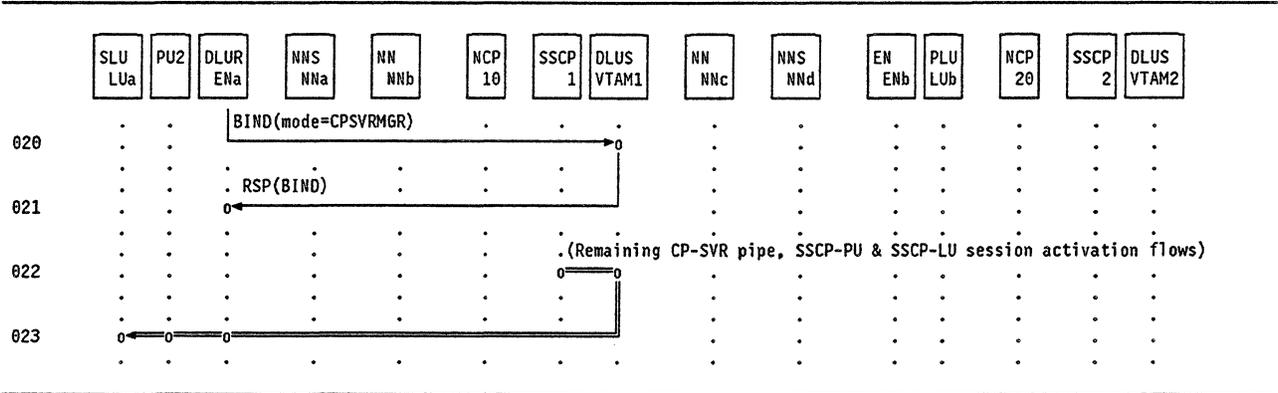
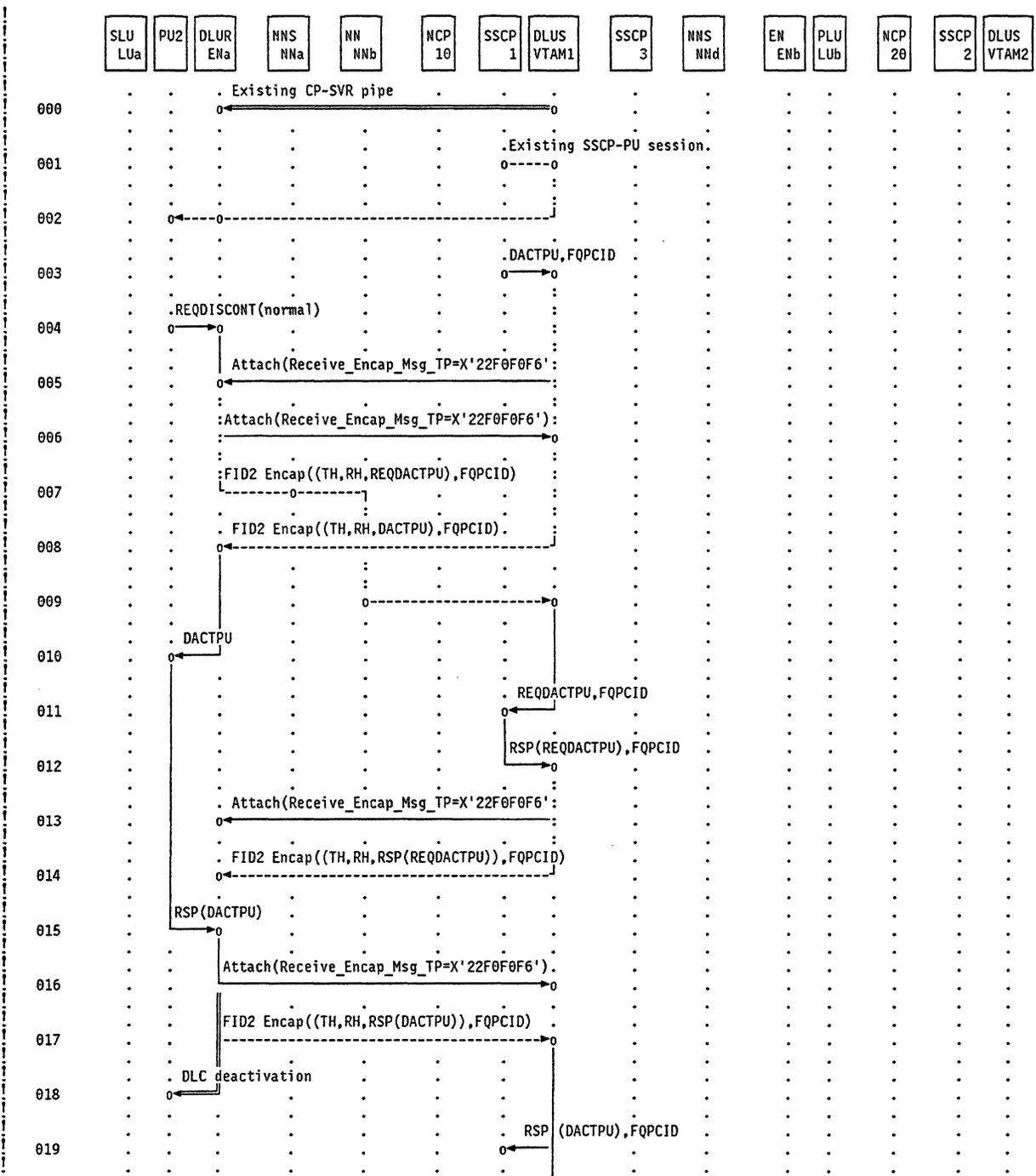


Figure 5-19. "Surprise" CP-SVR pipe activation

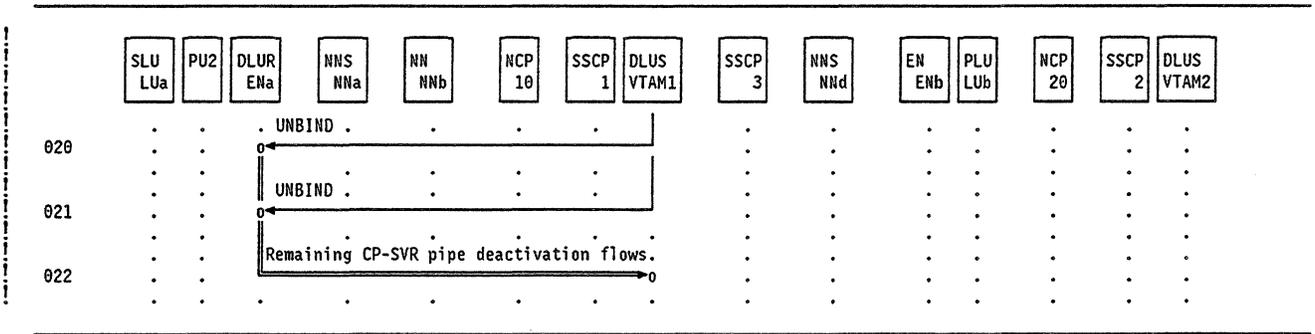


1. The current LU-LU session between LUa and LUb routes through PU2, ENa, NNa, NNb, NCP10, NNc, NNd, and ENb.
2. The SSCP-NCP session between SSCP1 and NCP10 ends due to a host outage.
3. SSCP1 reacquires ownership of NCP10 and its attached resources.
- 4.
- 5.
- 6.
- 7.
8. When SSCP1 has acquired ownership of the link station through which the LU-LU session is routed, NCP10 builds and sends to SSCP1 a BFSESSINFO RU. This RU indicates that a session exists between LUa (which it believes to be an independent LU) and LUb.
9. SSCP1 issues ACTPU to PU2 to restart its SSCP-PU session. SSCP1 passes the ACTPU to its DLUS component, VTAM1.
10. VTAM1 determines that ENa is the DLUR associated with PU2. VTAM1 then allocates a conwinner conversation between its Send_Encap_Msg_TP and the Receive_Encap_Msg_TP at the DLUR node. Since no session already exists to carry this conversation, VTAM1 starts one by sending out a Locate Find Cdinit for the CP_LU of the DLUR, mode = CPSVRMGR.
11. The DLUR ENa returns a Locate reply.
12. The DLUS VTAM1 uses the information in the Locate reply to calculate a route to the DLUR node and then builds and sends a BIND using the CPSVRMGR mode.
13. Since ENa has not yet received indication of the loss of the current CP-SVR pipe with VTAM1, it has no available conloser session to provide for the new CP-SVR pipe activation request. Therefore, ENa sends back a negative response to the BIND indicating there is no available conloser session.
14. Upon receipt of the negative response to the BIND, VTAM1 converts the ACTPU request into a negative response (sense code X'0801 0026') and returns it to SSCP1.
15. ENa finally receives indication of the failure the CP-SVR pipe with VTAM1. Having an attached LU with an active LU-LU session, ENa will attempt to activate a CP-SVR pipe with a DLUS.
16. The DLUR node determines the DLUS node to be contacted on behalf of PU2. Once the DLUS node has been determined, the DLUR node Allocates a conwinner conversation between its Send_Encap_Msg_TP and the Receive_Encap_Msg_TP (X'22F0F0F6') at the DLUS node. Since no session using the CPSVRMGR mode between this DLUR and the DLUS already exists, the DLUR node initiates the normal APPN flow sequence to bring up such a session. This begins with the DLUR node sending a Locate Find Cdinit request to its network node server. The LU target of the session is the CPname of the DLUS node = VTAM1 in this case.
17. The NNS(ENa) issues either a broadcast or directed search for the CP_LU of the DLUS = VTAM1.
18. The DLUS node responds to the search request with a Locate Found reply.
19. The NNS(ENa) uses the information in the Locate reply to calculate a route to the DLUS node and passes this back to the DLUR in the form of an RSCV.
20. The DLUR node ENa issues a CPSVRMGR BIND request to the DLUS node VTAM1.
21. The DLUS node VTAM1 returns a positive response to the BIND.
22. The activation of the CP-SVR pipe and the SSCP-PU and SSCP-LU sessions proceeds normally.

23.



! Figure 5-20. SSCP-PU session deactivation race - DACTPU/REQDACTPU



- ! 1. There are active sessions between SSCP1 and PU2 and SSCP1 and LUa. These are the only active SSCP-PU and SSCP-LU sessions using the CP-SVR pipe between VTAM1 and ENa.
- ! 2.
- ! 3. SSCP1 initiates normal deactivation of its session with PU2 by building a DACTPU and passing it on to VTAM1.
- ! 4. ENa receives a REQDISCONT RU from PU2.
- ! 5. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
- ! 6. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
- ! 7. To signal to SSCP1 that its SSCP-PU session with PU2 is broken, ENa builds a REQDACTPU with a cause value of X'01', encapsulates it, and forwards it to VTAM1 on its CP-SVR pipe.
- ! 8. The DLUS VTAM1 encapsulates the DACTPU and forwards it to the DLUR ENa.
- ! 9.
- ! 10. ENa de-encapsulates the DACTPU and forwards it to PU2; this is done as part of the SSCP-PU session deactivation race condition processing - since the requester is in an awaiting RSP(REQDACTPU) state and it receives a DACTPU from the DLUS for that PU, it should accept and process that DACTPU request.
- ! 11. VTAM1 de-encapsulates the REQDACTPU and forwards it to SSCP1, having also recognized the SSCP-PU session deactivation race condition.
- ! 12. SSCP1 responds to the REQDACTPU positively, passing the RSP(REQDACTPU) on to VTAM1 and remaining in a DACTPU processing state.
- ! 13. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
- ! 14. VTAM1 encapsulates the RSP(REQDACTPU) and forwards it to ENa; since it is still processing the DACTPU, the +RSP(REQDACTPU) will be accepted and DACTPU processing will continue.
- ! 15. PU2 deactivates the SSCP-PU session, converts the DACTPU into a positive RSP(DACTPU), and forwards it to ENa.
- ! 16. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
- ! 17. ENa then encapsulates the RSP(DACTPU) and sends it to VTAM1.
- ! 18. ENa initiates DLC deactivation to PU2.
- ! 19. VTAM1 de-encapsulates the RSP(DACTPU) and forwards it to SSCP1.
- ! 20. VTAM1 also determines that there are now no active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it sends an UNBIND to ENa on VTAM1's conwinner session to begin deactivation of the pipe.
- ! 21. VTAM1 also sends an UNBIND on its conloser session to ENa.
- ! 22. The DLUR node sends a RSP(UNBIND) to the DLUS for the conwinner session and another RSP(UNBIND) for the conloser session. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.

* **5.8 Additional Error Cases**

*** 5.8.1 CP-SVR Pipe Limit Exceeded**

- * DLUR products will have the option to limit the number of concurrently active CP-SVR pipes (what
- * number will be selected as the limit will also be a product option). If a limit is implemented and reached,
- * any attempts to activate additional CP-SVR pipes will be rejected by sending an UNBIND request with
- * sense code X'0812 0000'.

*** 5.8.2 GDS Other Than X'1500' Received After Receive_Encap Attach**

- * If the Receive_Encap_Msg_TP is attached and the GDS variable received is not X'1500', the DLUR will
- * send an UNBIND request with sense code X'1001 0000'.

! 5.8.3 CPSVRMGR Session Over Non-TCP/IP LEN Connection

- ! If a CPSVRMGR session is established over a LEN connection which is not a connection into a TCP/IP
- ! network, the DLUS will send an UNBIND request with sense code X'0877 0056'.

Chapter 6. SSCP-PU/SSCP-LU Session Encapsulation

Once the CP-SVR pipe has been established between server and requester nodes, SSCP-PU and SSCP-LU sessions must be established to provide SSCP services to the dependent LUs. These sessions will be encapsulated on the CP-SVR pipe. This will require a special encapsulation and de-encapsulation transaction program on either side of the conwinner and conloser CPSVRMGR sessions. These TPs are known as the `Send_Encap_Msg_TP` and `Receive_Encap_Msg_TP` respectively. Since the `Receive_Encap_Msg_TP` must be attached by the `Send_Encap_Msg_TP`, it has been assigned X'22F0F0F6' as its architected TP name.

6.1 Encapsulation Addressing

The information carried on the SSCP-PU and SSCP-LU sessions will be encapsulated in a new FID2 Encapsulation GDS variable X'1500'. This GDS variable will contain the entire FID2 message that is normally carried between the SSCP Boundary Function and peripheral node. This includes the Transmission Header (TH), Request/Response Header (RH), and Request Unit (RU) fields. The TH field will provide the appropriate PU and LU addressing as it does today, and the RH and RU fields will enable the PU to interpret the RUs and function the same way in both the DLUR/S and SSCP Subarea environments. Because a given CP-SVR pipe may support multiple SSCP-PU sessions, it will be important to distinguish which SSCP-PU/LU flows go with which PU image. This will be done by carrying a Fully Qualified Procedure Correlator Identifier (FQPCID) control vector (CV X'60') on each FID2 Encapsulation GDS variable. The FQPCID will be generated by the node creating the first encapsulated flow. This will be the DLUR node when `REQACTPU` is used, and it will be the DLUS node when an unsolicited `ACTPU` is sent. Once assigned, the CV X'60' will flow on each ensuing encapsulated flow having to do with that particular PU image (or its dependent LUs). This FQPCID essentially extends the ability for a single node to support more than 255 Dependent LUs by providing addressing to multiple PU images. In addition, the FQPCID will provide the DLUS with a mechanism to quickly index resource-related information for control flows. Because the FQPCID is being used in this manner, it is essential that it is unique for each DLUS node. Since the FQPCID is fully qualified, this implies that each node must ensure that it does not generate duplicate active FQPCIDs.

An FQPCID includes a PCID, a network-qualified CP name, and an optional PCID modifier. The network-qualified CP name will be defined as follows:

- DLUS-generated - the network-qualified SSCP name
- DLUR-generated - the network-qualified CP(DLUR) name

6.1.1 Multiple PU Images

As mentioned above, it is possible for a DLUR node to support more than 255 dependent LUs. In this case, it will be necessary for DLUR nodes to emulate multiple PU images. PU images that are being served by the same SSCP (DLUS node) will share the CP-SVR pipe. It is possible, however, for different PU images to be served by different SSCPs (DLUS nodes). In this case, a given DLUR node may have multiple CP-SVR pipes to multiple DLUS nodes. In either case, a given SSCP-PU and its subsequent SSCP-LU sessions will all be encapsulated on the same CP-SVR pipe. When a DLUR node is supporting multiple CP-SVR pipes to multiple DLUS nodes, the DLUR node may (or may not) run multiple instances of the `Send_Encap_Msg_TP` and the `Receive_Encap_Msg_TP`; it must keep track of which PU images are using which CP-SVR pipes.

! 6.1.2 Internal PU Identification

! Whether a DLUR node has a single PU or multiple PU images, if any of its PUs are internal (not down-stream PUs), each internal PU will be uniquely identified by either the DLUR's CPname or the internal PU's own IDBLK/IDNUM field. A unique identifier is necessary to identify which internal PU is being activated; therefore, at most one internal PU per DLUR can be optionally identified by the DLUR's CPname instead of IDBLK/IDNUM in the PU TG field.

! SV X'92' will carry IDBLK/IDNUM and SV X'93' will carry CPname. The SV combinations that will be permitted when SV X'91' = INTPU are:

! Table 6-1. Internal PU Identification		
! Option	! SV X'92'	! SV X'93'
! 1	! internal PU's IDBLK/IDNUM	! (not included)
! 2a	! (not included)	! CPname
! 2b	! CP's IDBLK/IDNUM	! CPname

! 6.1.2.1 Internal PU Identification Option 1

! The internal PU's IDBLK/IDNUM will be carried in the PU TG field (CV X'4691' = INTPU, CV X'4692' = IDBLK/IDNUM) in the GDS X'1500' for:

- ! • REQACTPU & ACTPU when PU activation is DLUR-initiated
- ! • ACTPU when PU activation is DLUS-initiated

! This IDBLK/IDNUM will also be carried in the XID field (CV X'81') in the GDS X'1500' for:

- ! • REQACTPU when PU activation is DLUR-initiated
- ! • RSP(ACTPU) when PU activation is DLUS-initiated

! No CV X'4693' will be included.

! 6.1.2.2 Internal PU Identification Option 2a

! The DLUR's CPname will be carried in the PU TG field (CV X'4691' = INTPU, CV X'4693' = CPname) in the GDS X'1500' for:

- ! • REQACTPU & ACTPU when PU activation is DLUR-initiated
- ! • ACTPU when PU activation is DLUS-initiated

! When a CPname is carried in the PU TG field, the DLUR CP's IDBLK/IDNUM will be carried in the XID field (CV X'81') in the GDS X'1500' for:

- ! • REQACTPU when PU activation is DLUR-initiated
- ! • RSP(ACTPU) when PU activation is DLUS-initiated

! No CV X'4692' will be included.

! 6.1.2.3 Internal PU Identification Option 2b

! The DLUR CP's IDBLK/IDNUM and CPname will be carried in the PU TG field (CV X'4691' = INTPU, CV X'4692' = IDBLK/IDNUM, CV X'4693' = CPname) in the GDS X'1500' for:

- ! • REACTPU & ACTPU when PU activation is DLUR-initiated
- ! • ACTPU when PU activation is DLUS-initiated

! When a CPname is carried in the PU TG field, the DLUR CP's IDBLK/IDNUM will be carried in the XID field (CV X'81') in the GDS X'1500' for:

- ! • REQACTPU when PU activation is DLUR-initiated
- ! • RSP(ACTPU) when PU activation is DLUS-initiated

! When this PU TG field is received, it will be compared against defined PUs for a match. DLUR will check both SV X'92' and SV X'93'; both will have to match the corresponding values for a given PU.

* 6.1.3 Addressing Per Stage

* Much of this is derived from the ASM (Address Space Manager) chapter in the APPN Architecture Reference (pages 6-5 through 6-7).

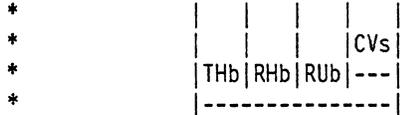
* LFSIDs are the addresses carried in FID2 THs; these addresses have three parts: SIDH (1 byte), SIDL (1 byte), ODAI (1 bit).

*** 6.1.3.2 LU-LU Sessions**

```

* *****
* * PLU *-----* NN *-----* NN *-----* DLUR *-----*PU/LU*
* ***** stage ***** stage ***** stage ***** stage *****
*          1          2          3          4

```



<-----> flows on stages 1, 2, 3, 4

```

* LU-LU
* THb  stages 1, 2, 3  stage 4
* -----
* SIDH  X'02'-X'FE'    X'01'
* SIDL  X'00'-X'FF'    X'01'-X'FF'
* ODAI  negotiated    0
*      during XID
*      exchange

```

* Figure 6-2. Addressing For LU-LU Sessions

* For stages 1, 2, and 3, LFSIDs for THa are independently set for each stage and for each direction; in the configuration shown, there will be six different THb LFSIDs for a message sent from PLU to DLUR and back to PLU. For stage 4, SIDL should be the same value used for the SIDL in THb for that LU's SSCP-LU session.

6.2 SSCP-PU Session Activation

To support dependent SLUs in or downstream from the DLUR node, the DLUS node must initiate SSCP-PU and SSCP-LU sessions over the CP-SVR pipe to the T2.0 PU image supporting the dependent SLUs.

- * This begins with the establishment of an SSCP-PU session. Once the DLUS conwinner CPSVRMGR
- * session has been established, an ACTPU message must be sent over the CP-SVR pipe to the T2.0 PU to establish an SSCP-PU session. This can be done in two ways:

1. *The DLUS node may activate a pre-defined PU.*

In this case, the SSCP (DLUS node) has a pre-defined PU image associated with the DLUR node. When the DLUS node explicitly activates the PU, it will know the DLUR name and downstream addressing associated with the PU as defined in 5.4.1, "New PU Definition Parameters" on page 5-4. This enables the DLUS to establish a CP-SVR pipe to the DLUR. Once the CP-SVR pipe has been activated, the SSCP will use the system-defined information and knowledge that the CP-SVR pipe has been activated, to generate an ACTPU RU and send it to the downstream PU.

2. *The DLUR node may request dynamic PU activation via the REQACTPU RU.*

- This is a new RU that identifies to the DLUS node that the DLUR node is requesting an ACTPU be sent for a previously defined or undefined downstream PU. The DLUS node will use information contained in the REQACTPU (BLKID, IDNUM, and optional CPname in the case of a T2.1 LEN node supporting
- * dependent LUs) to search for a pre-defined PU definition or use the Configuration Services XID exit support to create an ACTPU and send it to the downstream PU.

In either case, the FID2 Encapsulation GDS variable will be used to transport the REQACTPU|ACTPU messages over the CP-SVR pipe. In order to do this, the FQPCID and FID2 TH, RH, and RU fields must be properly assigned. The flows in this section describe how these fields are assigned and the PU images are activated in both cases.

DLC activation must precede forwarding the ACTPU to the PU. In a subarea network, DLC activation may require several RU flows, e.g., ACTLINK, CONNOUT, CONTACT, CONTACTED. The DLUR will perform DLC activation function, but without the receipt of these DLC activation commands. This activation will either be triggered by operator command, external contact from the PU, or by the receipt of an ACTPU from the DLUS. So that the DLUR has the necessary information to perform DLC activation, the REQACTPU and the ACTPU will be encapsulated with XID (X'81') and TG Descriptor (X'46') control vectors (both of which the DLUR will process before passing the ACTPU on to the PU). The XID control vector is identified in the flows as XID, and the TG Descriptor control vector as PU TG.

The DLUS will include the PU name with the ACTPU RU (the name will be defined in a Network Name (X'0E') control vector type X'F1'). This name will be network-qualified if the PU's NETID is different than the SSCP's NETID. The DLUR will strip off this control vector before passing the ACTPU on to the PU and will store the PU name for use in network management.

In a subarea network, activation of an SSCP-PU session may require several RU flows, e.g., RNAA, SETCV, and ACTPU. For a DLUR-attached PU, these RUs will be combined into one, ACTPU. Therefore, ACTPU will also be encapsulated with an Extended SDLC Station (X'43') control vector, which the DLUR will process before passing the ACTPU on to the PU. The control vector is identified in the flows as PU CHAR.

6.2.1 Pre-Defined PU Activation

In this scenario, the DLUS node has a pre-defined PU definition that it wishes to activate. It is also possible that this PU definition has been dynamically created and the DLUS operator has issued a command to activate/reactivate the PU. In this case, the PU definition must have the new DLUR and PU TG parameters as described in 5.4.1, “New PU Definition Parameters” on page 5-4.

Note: This diagram assumes the CP-SVR pipe between DLUS and DLUR is already active and has been activated by a previous SSCP-PU session activation sequence. Reference Figure 5-2 on page 5-11 for flows to describe the CP-SVR pipe activation associated with pre-defined PU activation.

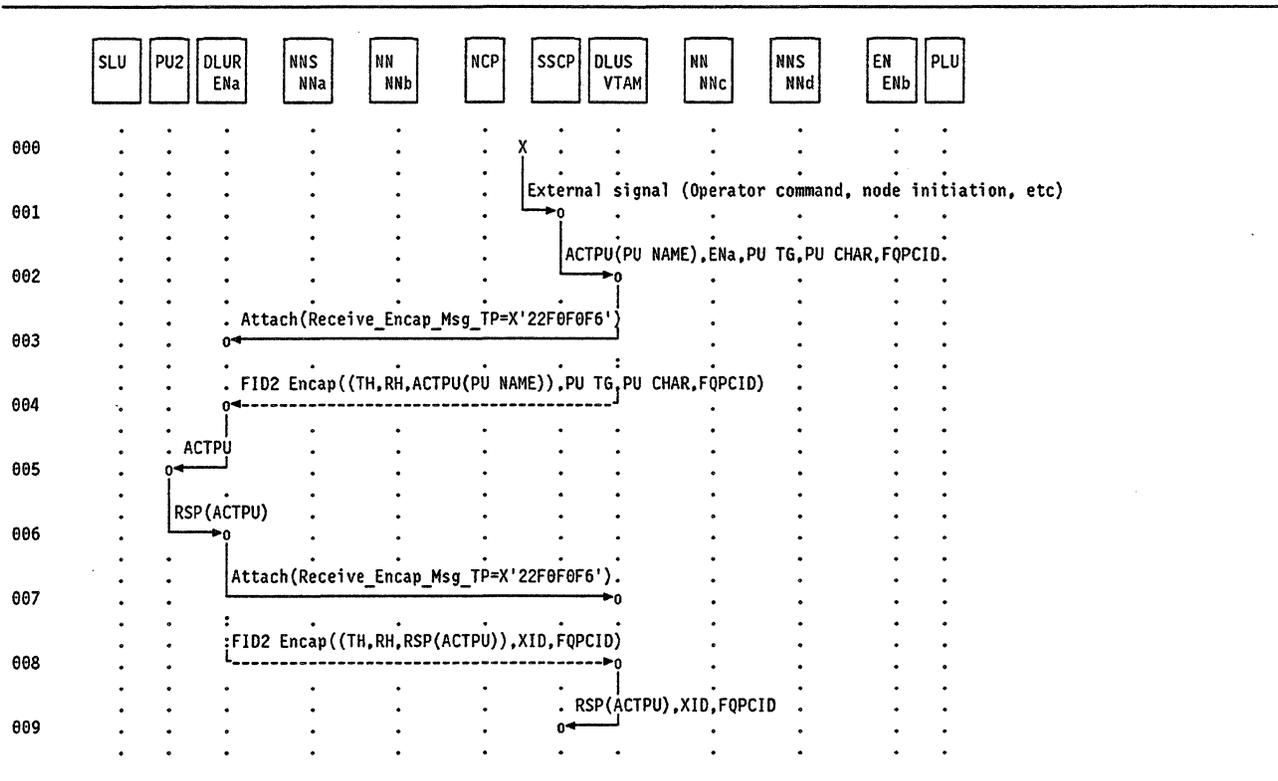
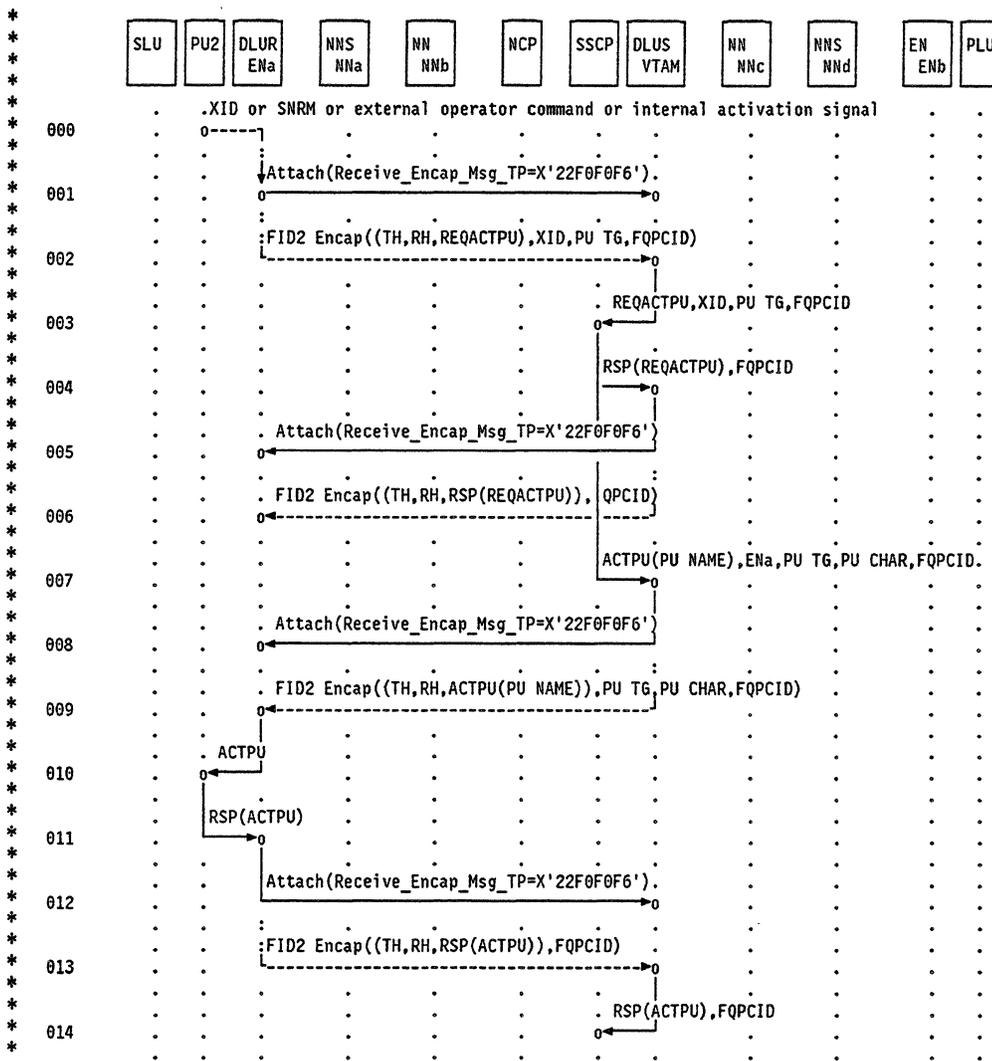


Figure 6-3. Pre-defined PU activation. Footnotes:

- 0) Encapsulated session traffic is indicated with dotted lines.

1. When an external operator command or internal process signals to the SSCP that a PU should be activated, a signal is sent to the SSCP to generate the associated ACTPU RU.
2. The ACTPU is sent from the SSCP to DLUS component of the NN. The ACTPU includes the name of the PU, as well as the FQPCID generated by the SSCP to identify and index the PU control block.
3. The DLUS component uses the DLUR node specification in the PU definition to determine which CP-SVR pipe the encapsulated ACTPU should flow on. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
4. The ACTPU FID2 is sent on the CP-SVR pipe to the DLUR node. The FQPCID generated by the SSCP to identify the PU is included in the FID2 Encapsulation GDS variable. The TH of the FID2 will address both the SSCP and PU in the OAF and DAF fields respectively. The DLUS node includes the PU CHAR field with the ACTPU; this field carries PU information such as data mode and retry limits. The DLUS node also includes the PU TG identifier with the ACTPU RU; this will be used by the DLUR node to identify which PU is being activated.
5. The DLUR node receives the encapsulated ACTPU command. It must look in the FID2 Encapsulation GDS variable for the PU TG vector to identify the PU being activated. If the PU is downstream, the PU TG will identify a physical link to the downstream node. The DLUR node may have to initiate DLC activation on the link to the PU; the PU TG and PU CHAR fields carry information needed to activate the DLC. If the PU is internal to the DLUR, the PU TG is used as a local correlator to the DLUR to identify the PU image. It must then store the FQPCID associated with this PU so that subsequent data flows on the SSCP-PU session are forwarded to the correct PU. The DLUR strips off the GDS variable, including the PU TG, PU CHAR, and FQPCID fields, removes the PU name (storing it for network management), and forwards the ACTPU to the appropriate PU.
6. The PU generates an ACTPU RSP and sends it to the DLUR component.
7. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
8. The DLUR component uses the previously stored FQPCID to return the ACTPU RSP in a FID2 Encapsulation GDS variable on the CP-SVR pipe. This FQPCID is supplied to allow the DLUS to quickly access the PU control block.
9. The DLUS node receives the encapsulated ACTPU RSP and strips off the GDS variable. The ACTPU RSP is forwarded from the DLUS to the SSCP component. The FQPCID is now used on all subsequent SSCP-PU data flows to identify the PU to the DLUR. Both the DLUS and DLUR node will set this field to the same value when referring to the SSCP-PU session for this PU.

6.2.2 Dynamic PU Activation



* Figure 6-4. Dynamic PU activation. Footnotes:

* 0) Encapsulated messages are indicated with dotted lines

1. Sometime after activation of the CP-SVR pipe between DLUS and DLUR nodes, the DLUR node indicates that it wishes to activate a PU. This could be in response to an operator command, internal signal, or external contact from a downstream PU (for cases where the CP-SVR pipe has not already been established, see Figure 5-1 on page 5-7). The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
2. The DLUR signals the desire to activate a PU by sending an encapsulated REQACTPU RU on the CP-SVR pipe. The DLUR node will generate an FQPCID to identify the PU to the SSCP. It will also remember this FQPCID to identify the specific PU instance (and associated PU TG information if applicable) to itself. The TH DAF network address field is set to 0 identifying the SSCP, and the OAF address is set to 0 identifying the PU. The FID2 Encapsulation GDS variable carrying the REQACTPU PU contains an XID0 or XID3 that will be used by the SSCP to search for a pre-defined PU image. In the case where a downstream PU has exchanged XID's with the DLUR, this will be the true XID image. In the case where this has not taken place, the DLUR will generate an XID0 image as a means to supply a IDBLK/IDNUM value to the SSCP. The FID2 Encapsulation GDS variable carrying the REQACTPU PU also carries a PU TG identifier to describe the link attaching the downstream PU to the DLUR.
3. The DLUS component receives the encapsulated REQACTPU RU and passes it and the DLUR-generated FQPCID to the SSCP after stripping off the FID2 Encapsulation GDS variable.
4. The SSCP must verify that the FQPCID generated by the DLUR is unique and the PU can be activated. A RSP(REQACTPU) is then generated and sent to the DLUR node via the DLUS component. Before sending a positive RSP(REQACTPU), the SSCP must verify that the FQPCID is unique for this node and either find a pre-defined PU definition or generate a PU definition via the Configuration Services XID Exit support mentioned in 3.2.2, "Configuration Services XID Exit Support" on page 3-4. The SSCP may also chose to reject the REQACTPU regardless of Configuration Services XID Exit support, for local security or other product/customer-specific reasons. If the above conditions are met, the PU can be activated, and a positive response is generated. If the FQPCID is not unique or the PU cannot be activated, a negative response and sense code is generated. The RSP(REQACTPU) is sent along with the FQPCID to the DLUS component of the DLUS node.
5. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
6. The DLUS component encapsulates the RSP(REQACTPU) PIU and uses the FQPCID generated by the DLUR component to identify the PU to the DLUR node. This encapsulated message is sent over the CP-SVR pipe to the DLUR node.
7. The SSCP follows a positive RSP(REQACTPU) with an ACTPU for the PU. This ACTPU includes the name of the PU.
8. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
9. The ACTPU PIU is encapsulated along with the FQPCID to identify the PU to the DLUR node. It is sent to the DLUR node.
10. The DLUR node uses the FQPCID field to identify the PU to receive the encapsulated flow. The FID2 Encapsulation GDS variable, and its PU TG and PU CHAR fields are stripped by the DLUR. The DLUR then removes the PU name (storing it for network management) from the ACTPU and forwards the ACTPU to the appropriate PU.
11. The PU generates a RSP(ACTPU) and sends it to the DLUR node.
12. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
13. The DLUR component encapsulates the RSP(ACTPU) PIU along with the FQPCID and sends it on the CP-SVR pipe to the DLUS node.

14. The DLUS component removes the FID2 Encapsulation GDS variable and forwards it to the SSCP. Both the DLUS and DLUR nodes will now use the FQPCID to uniquely identify this SSCP-PU session.

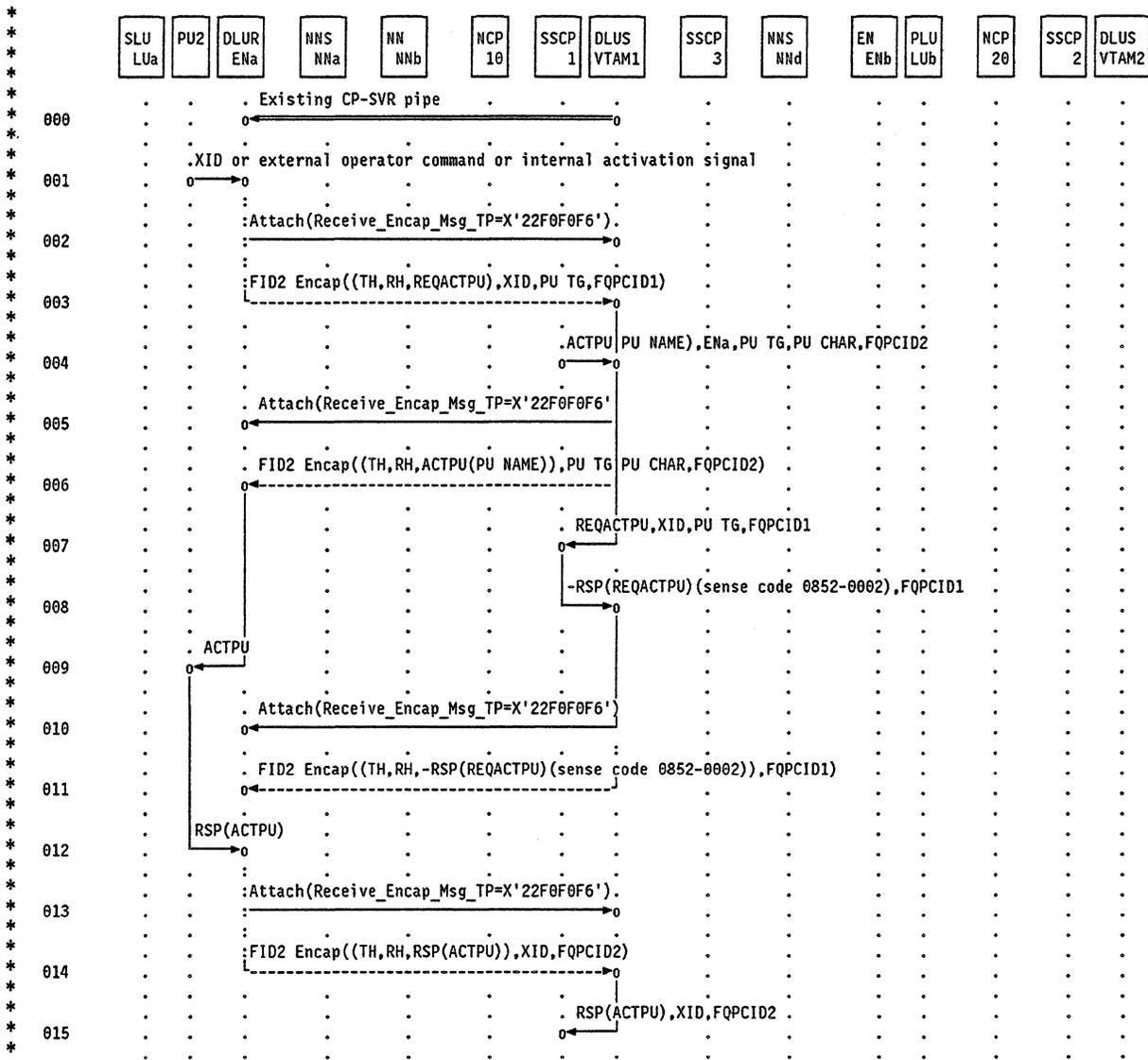
6.2.3 SSCP-PU Session Activation Race Conditions

When a DLUS and a DLUR attempt to initiate SSCP-PU sessions with the same PU, the DLUS-initiated session will be started and the DLUR-initiated session will be rejected. When two DLUS's attempt to initiate SSCP-PU sessions with the same PU, the first activation request received by the DLUR will be accepted and the second rejected.

6.2.3.1 Single DLUS Race Conditions

Race conditions can exist between a DLUR and one DLUS, when both are trying to start an SSCP-PU session for the same PU at the same time. The DLUR will send a REQACTPU to the DLUS while the DLUS is sending an ACTPU to the DLUR. The resolution of this situation is for the DLUR to process the ACTPU and for the DLUS to reject the REQACTPU.

The following figure shows an example of this race condition:



* Figure 6-5. SSCP-PU session activation race - DLUR- and DLUS-initiated requests on same CP-SVR pipe

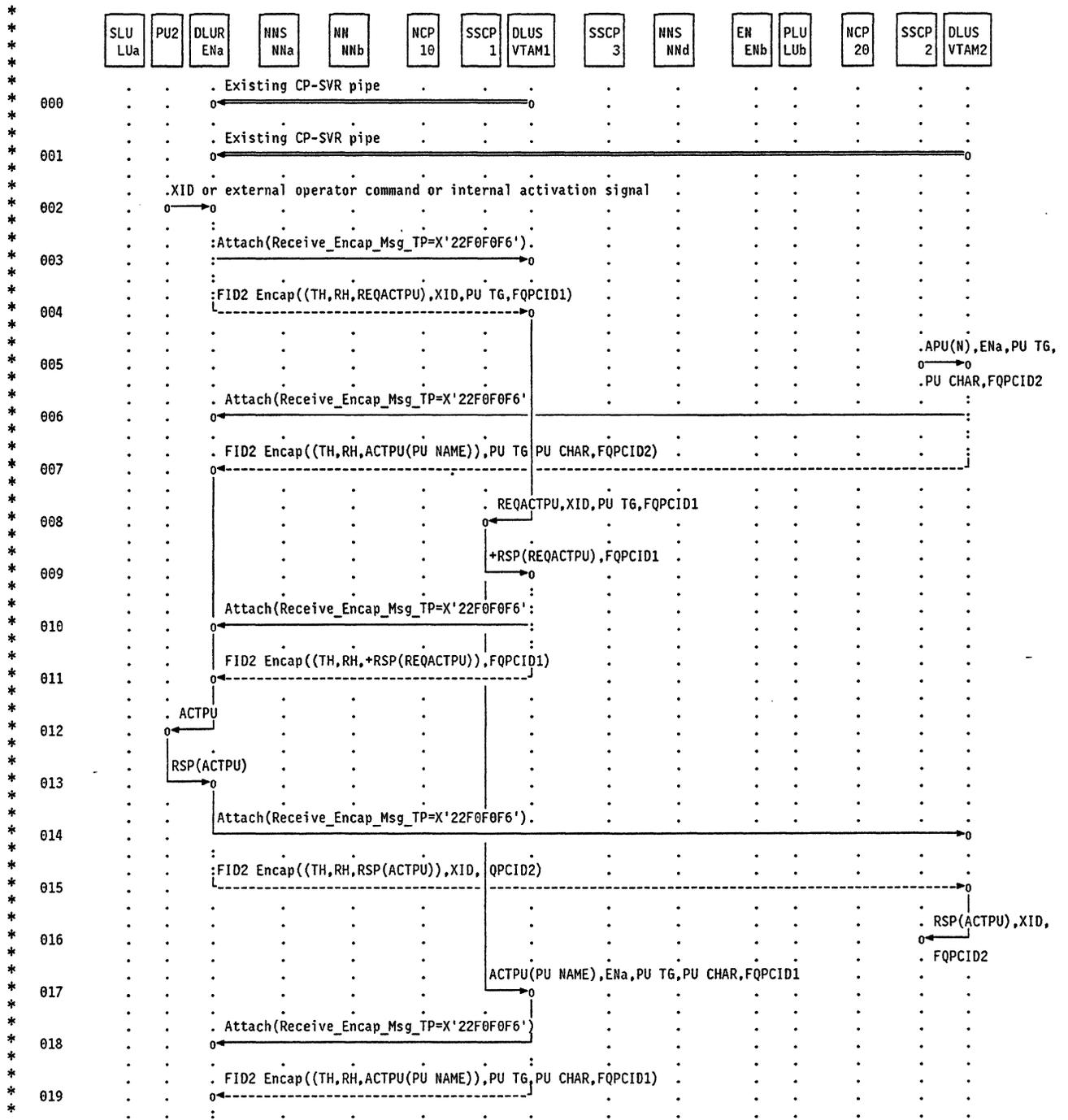
1. There is an active CP-SVR pipe between VTAM1 and ENa. ENa receives some indication that PU2 requires activation.
2. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
3. ENa encapsulates and sends to VTAM1 a REQACTPU RU.
4. Meanwhile SSCP1 also receives some indication to activate PU2, so it builds an ACTPU to be sent to PU2.
5. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
6. VTAM1 encapsulates the ACTPU and sends it to ENa.
7. VTAM1 de-encapsulates the REQACTPU and forwards it to SSCP1.
8. SSCP1 recognizes that it already has a pending SSCP-PU session with PU2, so it negatively responds to the REQACTPU with sense code X'0852 0002'.
9. ENa recognizes that it already has a pending SSCP-PU session with PU2. Knowing that SSCP1 will reject the REQACTPU, ENa de-encapsulates the ACTPU and forwards it to PU2.
10. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
11. VTAM1 encapsulates the -RSP(REQACTPU) and sends it to ENa.
12. PU2 returns a positive RSP(ACTPU) to SSCP1 via ENa.
13. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
14. ENa encapsulates the RSP(ACTPU) and sends it to VTAM1.
15. VTAM1 de-encapsulates the RSP(ACTPU) and passes it on to SSCP1.

6.2.3.2 Multi-DLUS Race Conditions

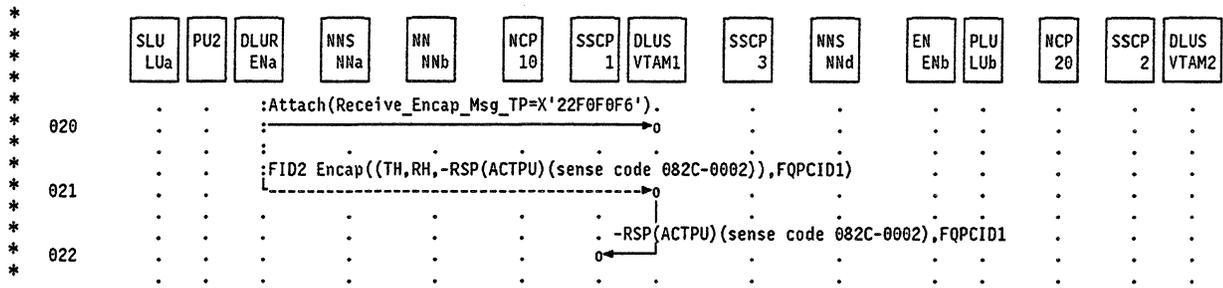
Race conditions can exist between a DLUR and two DLUS's; the rule the DLUR applies to resolve these races is the same: **always accept the first ACTPU received**. Here's how this rule is applied in different race conditions:

- *race between DLUS-initiated and DLUR-initiated session activations* - In this case the DLUR will send a REQACTPU to one DLUS while the other DLUS is sending an ACTPU to the DLUR for the same PU. The resolution of this situation is for the DLUR to process the DLUS-initiated ACTPU and reject the ACTPU received in response to the REQACTPU.
- *race between two DLUS-initiated session activations* - In this case both DLUS's will send ACTPUs to the DLUR for the same PU. The resolution of this situation is for the DLUR to both process the first received DLUS-initiated ACTPU and reject the second ACTPU.

The following figures show examples of these race conditions:



* Figure 6-6. SSCP-PU session activation race - DLUR- and DLUS-initiated requests on different CP-SVR pipes



1. There are active CP-SVR pipes between VTAM1 and ENa and between VTAM2 and ENa.
2. ENa receives some indication that PU2 requires activation.
3. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
4. ENa encapsulates and sends to VTAM1 a REQACTPU RU.
5. Meanwhile SSCP2 also receives some indication to activate PU2, so it builds an ACTPU to be sent to PU2.
6. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
7. VTAM2 encapsulates the ACTPU and sends it to ENa.
8. VTAM1 de-encapsulates the REQACTPU and forwards it to SSCP1.
9. SSCP1 recognizes that it has no active or pending SSCP-PU session with PU2, so it positively responds to the REQACTPU and passes the response to VTAM1.
10. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
11. VTAM1 encapsulates the +RSP(REQACTPU) and sends it to ENa.
12. ENa recognizes that it already has a pending SSCP-PU session with PU2. It must wait for the ACTPU from SSCP1 to reject that session activation attempt. Meanwhile, ENa de-encapsulates the ACTPU from SSCP2 and forwards it to PU2.
13. PU2 returns a positive RSP(ACTPU) to SSCP1 via ENa.
14. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
15. ENa encapsulates the RSP(ACTPU) and sends it to VTAM2.
16. VTAM2 de-encapsulates the RSP(ACTPU) and passes it on to SSCP2.
17. SSCP1 builds an ACTPU for PU2 and passes it to VTAM1.
18. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
19. VTAM1 encapsulates the ACTPU and sends it to ENa.
20. ENa recognizes that there already is an SSCP-PU session with PU2. It negatively responds to the ACTPU with sense code X'082C 0002'. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
21. ENa encapsulates the -RSP(ACTPU) and sends it to VTAM1.
22. VTAM1 de-encapsulates the -RSP(ACTPU) and passes it on to SSCP1.

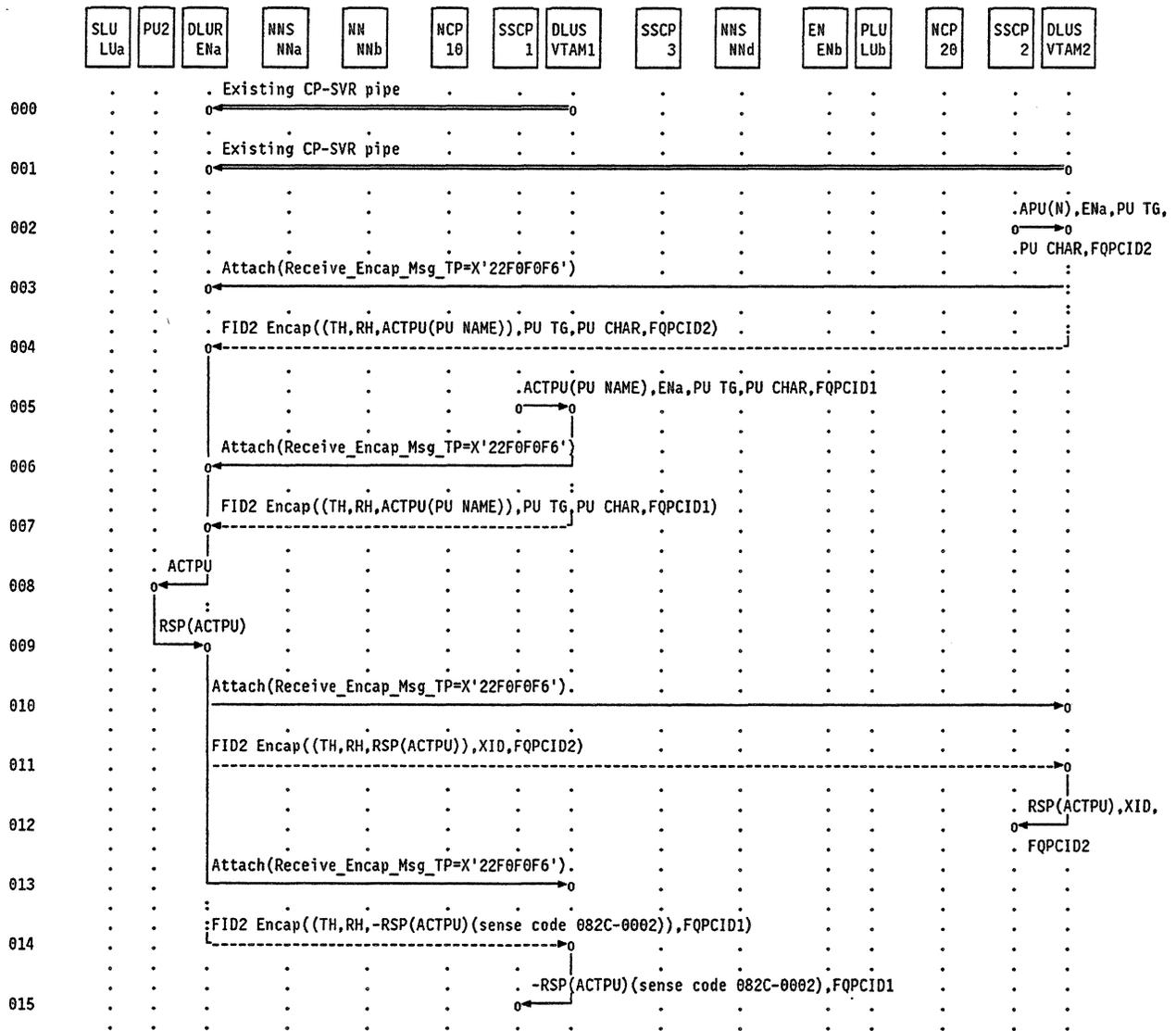


Figure 6-7. SSCP-PU session activation race - two DLUS-initiated requests

1. There are active CP-SVR pipes between VTAM1 and ENa and between VTAM2 and ENa.
2. SSCP2 receives some indication to activate PU2, so it builds an ACTPU to be sent to PU2.
3. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
4. VTAM2 encapsulates the ACTPU and sends it to ENa.
5. Meanwhile SSCP1 also receives some indication to activate PU2, so it builds an ACTPU to be sent to PU2.
6. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
7. VTAM1 encapsulates the ACTPU and sends it to ENa.
8. ENa will process the ACTPU from SSCP2 (since it arrived first) reject the ACTPU from SSCP1. ENa de-encapsulates the ACTPU from SSCP2 and forwards it to PU2.
9. PU2 returns a positive RSP(ACTPU) to SSCP1 via ENa.
10. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
11. ENa encapsulates the RSP(ACTPU) and sends it to VTAM2.
12. VTAM2 de-encapsulates the RSP(ACTPU) and passes it on to SSCP2.
13. ENa recognizes that there already is an SSCP-PU session with PU2. It negatively responds to the ACTPU with sense code X'082C 0002'. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
14. ENa de-encapsulates the -RSP(ACTPU) and sends it to VTAM1.
15. VTAM1 de-encapsulates the -RSP(ACTPU) and passes it on to SSCP1.

6.3 SSCP-LU Session Activation

Once the ACTPU and ACTPU RSP have flowed, the DLUS node will activate the dependent LUs. This is done by sending an ACTLU over the SSCP-PU session from the DLUS to DLUR node. Since dependent LUs may be either pre-defined or dynamically created, there are two ways in which the ACTLU messages can be sent:

1. *The DLUS node may activate pre-defined LUs.*

In this case, the DLUS node will use system-defined information to send ACTLU messages over the SSCP-PU session to the DLUR node.

2. *The DLUR node may request dynamic activation of its dependent LUs.*

In this case, the DLUR node will use the Self-Defining Dependent LU (SDDL) support in VTAM to request activation of its dependent LUs (see 3.2.3, "Self-Defining Dependent LU Support" on page 3-4).

In either case, the DLUR node must also process the ACTLUs it receives to maintain a table associating dependent LU names with their supporting PU and PU connectivity information. This must be done because BINDs for dependent LUs must be routed by SLU name. The RSCV for the extended BIND will terminate in the DLUR node and if the dependent LU is actually located on a downstream PU, the DLUR node will have to know how to send the BIND to the downstream PU/LU.

The information the DLUR needs to do this is the LU name, included by the DLUS in the ACTLU RU. The LU name will be defined in a Network Name (X'0E') control vector type X'F3'. This name will be network-qualified if the PU's NETID is different than the SSCP's NETID. The DLUR will strip off this control vector before passing the ACTLU on to the PU and will store the LU name for several uses, including PU/LU correlation, BIND routing, and network management.

In a subarea network, activation of an SSCP-LU session may require several RU flows, i.e., RNAA, SETCV, and ACTLU. For a DLUR-attached LU, these RUs will be combined into one, ACTLU. Therefore, ACTLU will now be encapsulated with an Assign LU Characteristics (X'30') control vector, which the DLUR will process and strip from the RU before passing the ACTLU on to the LU. The control vector is identified in the flows as LU CHAR.

6.3.1 Activation Of Pre-Defined Dependent LUs



Figure 6-8. Activation of pre-defined LUs

1. When dependent LUs are pre-defined at the SSCP, the DLUS node will activate each of the LUs following activation of the SSCP-PU session.
2. The SSCP signals an ACTLU to the DLUS component. The DLUS node encapsulates the LU CHAR field with the ACTLU; this field carries LU information such as pacing parameters.
3. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
4. The DLUS component will encapsulate the ACTLU PIU with the FQPCID of the PU, and send it over the CP-SVR pipe to the DLUR node.
5. The DLUR node strips off the FID2 Encapsulation GDS variable (including the LU CHAR and FQPCID CVs), strips the Network Name (X'0E') CV (which includes the LU name) from the ACTLU RU, and sends the ACTLU to the PU which forwards it to the LU.
6. The LU generates an RSP(ACTLU), sends it to the PU, which in turn sends the response to the DLUR component.
7. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
8. The DLUR component then uses the FQPCID of the PU and encapsulates the RSP(ACTLU) PIU. This is sent over the CP-SVR pipe to the DLUS node.
9. The DLUS node strips the FID2 Encapsulation GDS variable and sends the RSP(ACTLU) to the SSCP. Both nodes now use the FQPCID and FID2 TH addressing to reference the SSCP-LU session.

6.3.2 Dynamic Registration Of Dependent LUs

Since VTAM will provide SDDL support, DLUR nodes will be able to dynamically register and obtain SSCP services for its dependent LUs. This is done by sending an NMVT with Product Set Identifier (PSID) for each dependent LU on the SSCP-PU session from the DLUR to DLUS node.

Note: In order to utilize this function, downstream or local PUs supported by DLUR nodes must provide the PSID and SDDL support required.

Based upon information received in the PSID subvector, the SSCP will use model definitions to dynamically create the necessary resource definitions and control blocks. A subarea address will be assigned and the SSCP will respond with an ACTLU to the DLUR node. The DLUS node will use the FQPCID of the SSCP-PU session and the addressing in the TH of the FID2 to build an appropriate FID2 Encapsulation GDS variable and send the ACTLU RU to the DLUR node via the CP-SVR pipe. When the DLUR node receives the ACTLU RU, it will strip the FID2 Encapsulation GDS variable and deliver it to the appropriate LU. The LU will respond with an RSP(ACTLU) which the DLUR node will encapsulate and send to the DLUS node.

The following flow describes dynamic registration and activation of dependent LUs.

6.3.3 Dynamic Registration And Activation Of Dependent LUs

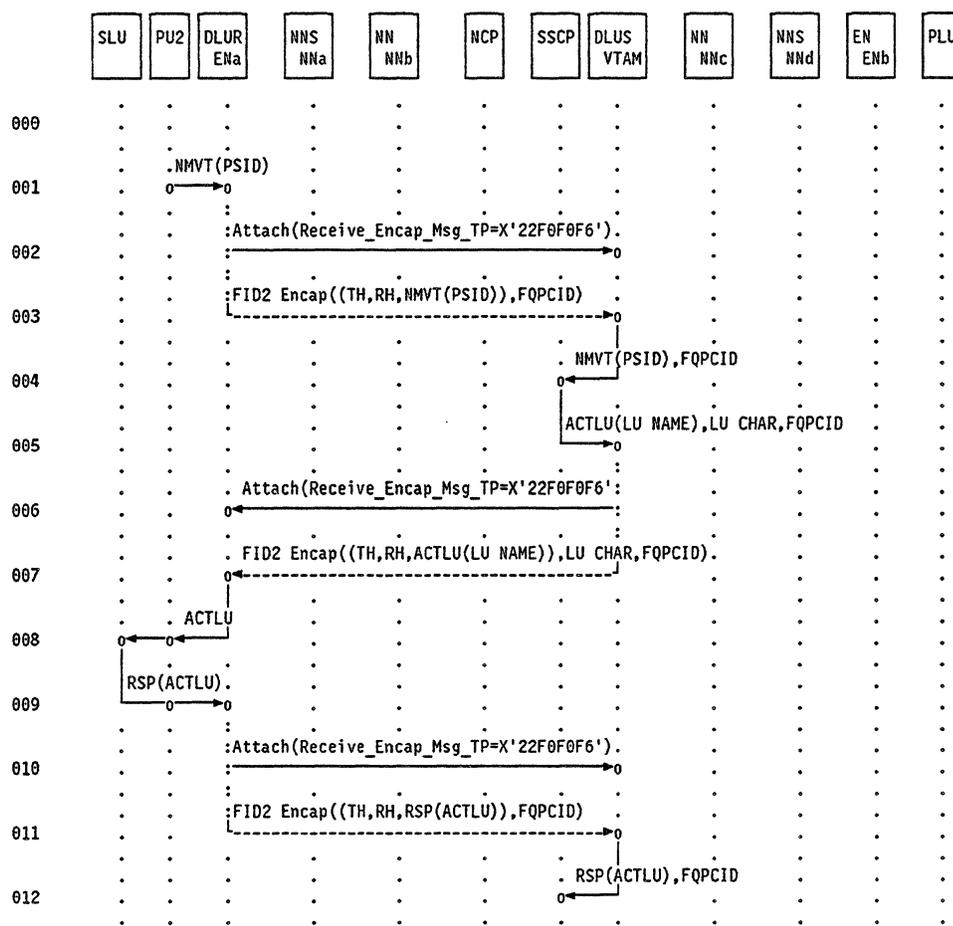


Figure 6-9. Dynamic dependent LU activation

1. PUs that have the necessary Self-Defining Dependent LU (SDDL) support will generate an NMVT with the Product Set ID (PSID) information specific to a given LU and send it on the SSCP-PU session up to the host. PUs with this support in a DLUR node will send this NMVT to the DLUR component.
2. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
3. The DLUR component will use the FQPCID of the SSCP-PU session to encapsulate the NMVT on the SSCP-PU session and send it up the CP-SVR pipe to the DLUS node.
4. The DLUS node will strip the FID2 Encapsulation GDS variable and send the NMVT to the SSCP.
5. The SSCP will use the SDDL support to generate an LUsername, internal control blocks, and an ACTLU RU.
6. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
7. The DLUS component will then use FQPCID of the SSCP-PU session to identify the PU to the DLUR node. The LU address will be in the DAF field of the TH of the ACTLU PIU.
Note: The NMVT carries a SNA Address List Subvector that identifies the local address of the LU to be activated. This, in turn, is included in the DAF field of the ACTLU PIU sent to the PU.
8. The DLUR node removes the FID2 Encapsulation GDS variable (including the LU CHAR and FQPCID CVs) and strips the Network Name (X'0E') CV (which includes the LU name) from the ACTLU RU. The DLUR uses the FQPCID to pass the ACTLU command to the appropriate PU. The DLUR node uses the LU name to build a table associating LUs with PUs. This enables the DLUR to route subsequent BINDs (by SLU name) to the proper PU. The ACTLU is forwarded from PU to LU.
9. The RSP(ACTLU) is generated by the LU, passed to the PU, and forwarded to the DLUR component.
10. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
11. The DLUR component uses the FQPCID of the PU and encapsulates the RSP(ACTLU) PIU on the CP-SVR pipe to the DLUS node.
12. The DLUS node strips the FID2 Encapsulation GDS variable and passes the RSP(ACTLU) to the SSCP. Both the DLUS and DLUR nodes will now use the FQPCID of SSCP-PU session and the FID2 TH addressing of the PIUs to reference the SSCP-LU session for this LU.

6.4 SSCP-PU/SSCP-LU Session Deactivation (DLUS-Initiated)

(For DLUR-initiated SSCP-PU and SSCP-LU session deactivation, see 5.6.3, "Abnormal SSCP-PU/SSCP-LU Session Deactivation (DLUR-Initiated)" on page 5-41)

The DLUS has several mechanisms to deactivate existing SSCP-PU and SSCP-LU sessions with DLUR-attached resources:

- *normal* - SSCP-LU sessions are deactivated first followed by the corresponding SSCP-PU session; any active LU-LU sessions are deactivated during the corresponding SSCP-LU session deactivation.

Usually LU-LU sessions are deactivated first before deactivating the SSCP-LU session. Optionally the deactivation of the last LU-LU session can trigger the deactivation of the SSCP-LU session, and if it is the last active SSCP-LU session, this can trigger the deactivation of the SSCP-PU session, and if it is the last active SSCP-PU session, this can trigger the deactivation of the CP-SVR pipe. Another option is to keep the pipe active even when the last SSCP-PU session using it has been deactivated.

- *forced* - deactivation of the SSCP-PU session forces deactivation of any active SSCP-LU sessions associated with that PU; any active LU-LU sessions are deactivated during the SSCP-PU session deactivation

! The DLUS in this case will build and forward to its SSCP a RSP(DACTPU) right after sending the
! DACTPU request to the DLUR, not waiting for the PU-generated RSP(DACTPU). This means:

- ! - if the DLUS receives the RSP(DACTPU) from the DLUR, it should discard the response (see
! Figure 6-11 on page 6-33) - this could occur when other active SSCP-PU session(s) will keep the
! DLUS from deactivating the pipe - and if the DLUS receives a SESSEND from the DLUR, it
! should discard the request
- ! - if the DLUR receives an UNBIND for the CP-SVR pipe before it can send the RSP(DACTPU)
! from the PU back on the pipe, it should discard the response (see Figure 5-15 on page 5-49) - this
! could occur when the DLUS decides to deactivate the pipe before the DLUR has completed handling
! the DACTPU request and response

- *giveback* - deactivation of the SSCP-PU session forces deactivation of any active SSCP-LU sessions associated with that PU; any active LU-LU sessions are left active during the SSCP-PU session deactivation if the PU is defined with ANS=CONT - if the PU is defined with ANS=STOP, the active LU-LU sessions are also deactivated (the ANS value is carried in the PU CHAR (CV X'43') field on ACTPU).

! 6.4.1 DLUR ANS Support

! It is strongly recommended that DLUR implementations support both ANS=CONT and ANS=STOP.
! However, for those DLURs that can only support ANS=STOP, the following architecture is included so
! that the DLUS will know the DLUR's ANS support level:

- ! • A support indicator in the DLUR/S Capabilities (X'51') control vector has been defined to indicate the
! level of ANS support provided by a DLUR:
 - ! - 0 = full ANS support (both ANS = CONT and ANS = STOP supported)
 - ! - 1 = limited ANS support (only ANS = STOP supported)
- ! • When building a CV X'51', the DLUR will set this indicator based on its level of ANS support.

! **6.4.2 SSCP-PU/SSCP-LU Session Deactivation Flows**

! The following figures show examples of the types of deactivation described in 6.4, “SSCP-PU/SSCP-LU Session Deactivation (DLUS-Initiated)”:

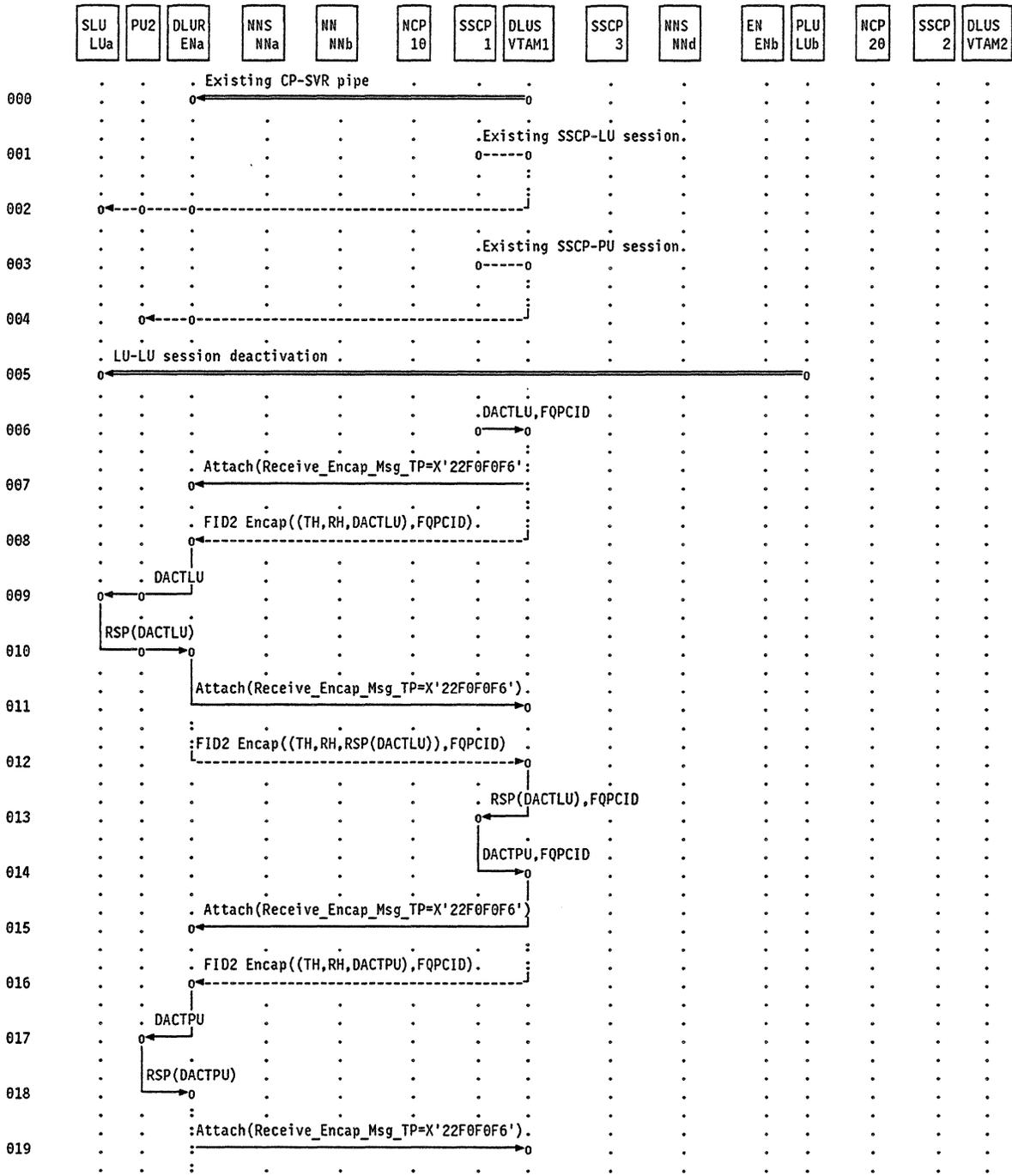
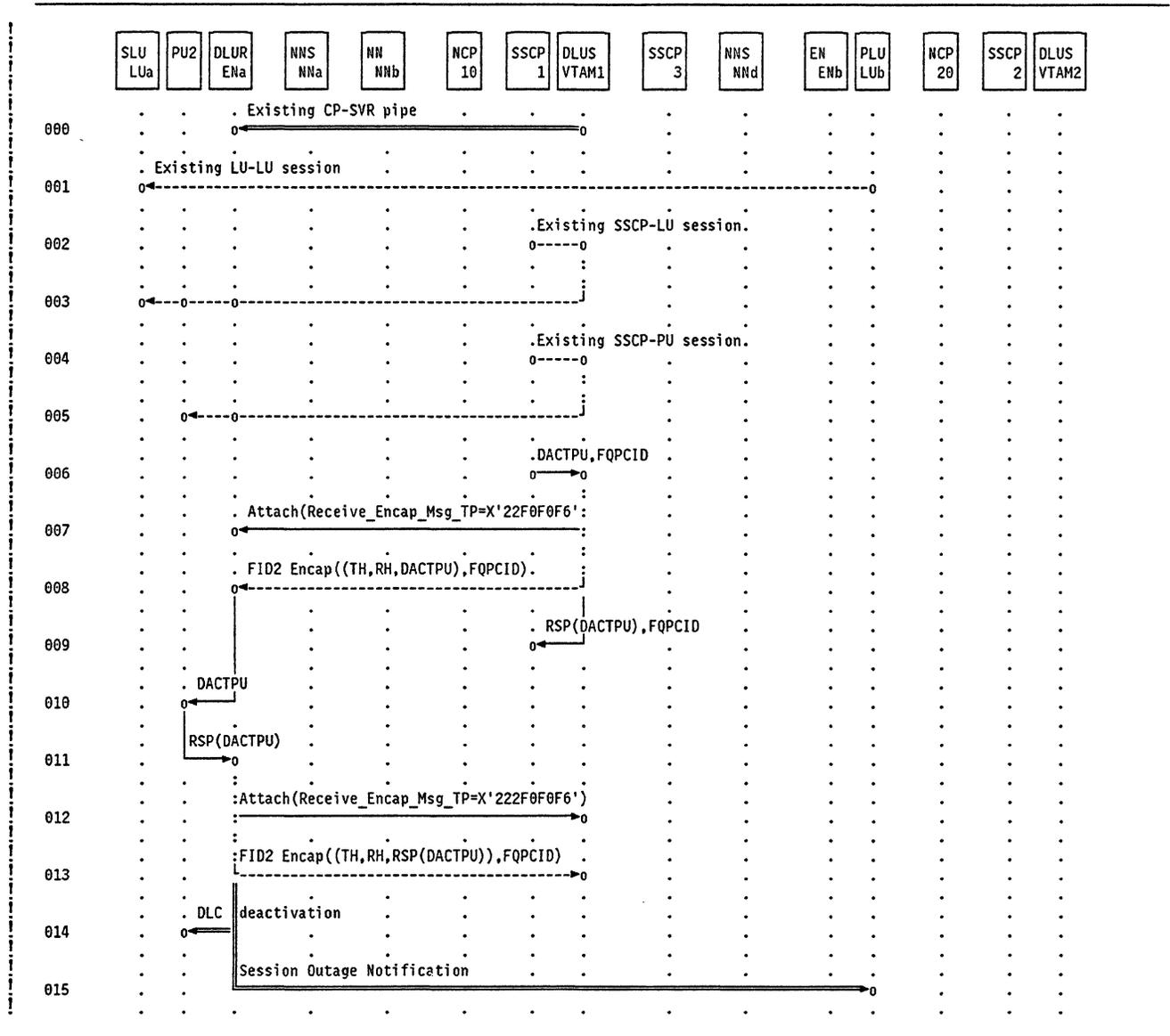


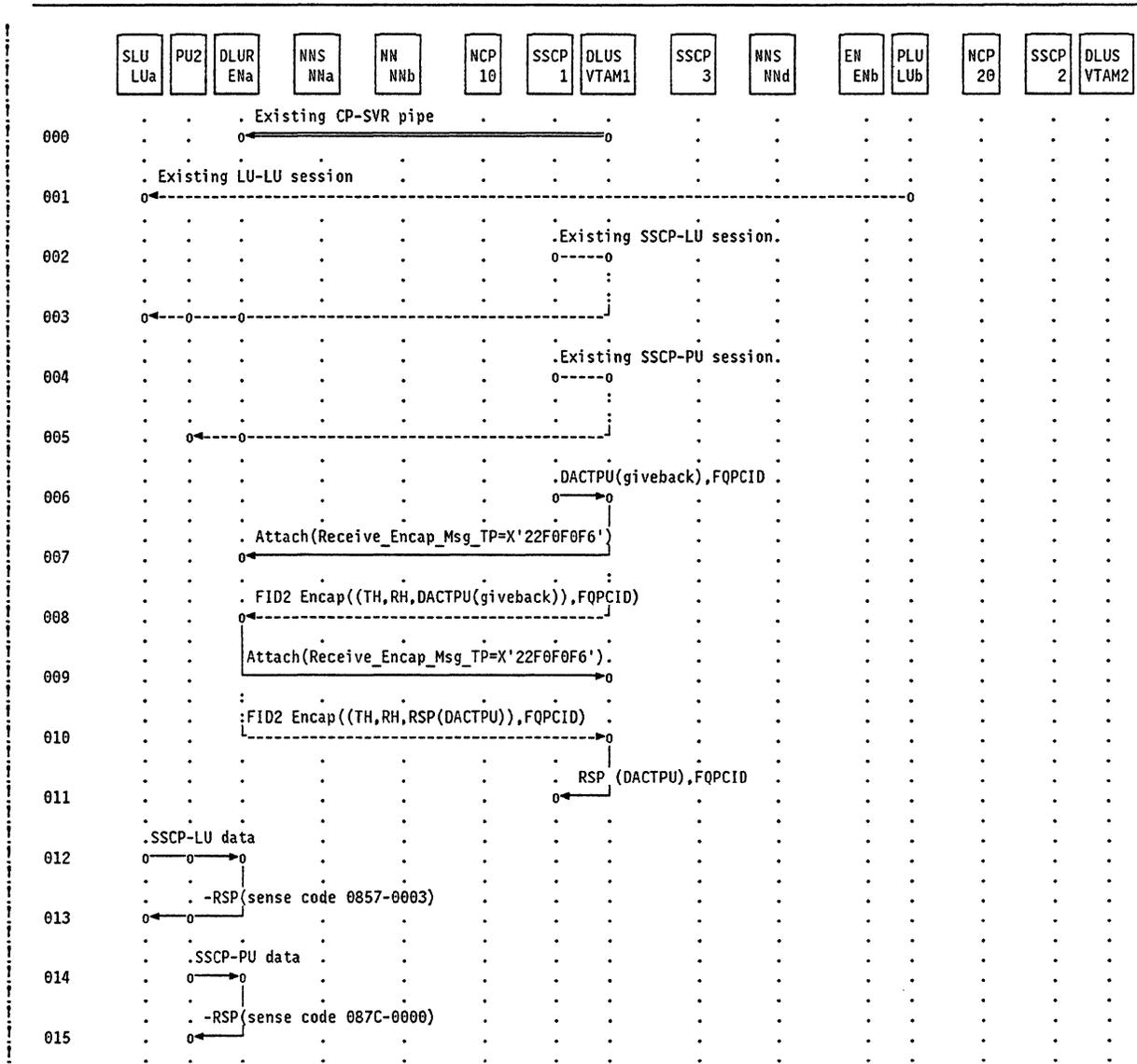
Figure 6-10. Normal SSCP-PU/SSCP-LU session deactivation

1. There is an active session between SSCP1 and LUa. It is the only active SSCP-LU session associated with PU2. There is also an active session between LUb and LUa.
- 2.
3. There is an active session between SSCP1 and PU2, but it is not the only active SSCP-PU session using the CP-SVR pipe between ENa and VTAM1. PU2 is the only active PU on the connection between PU2 and ENa.
- 4.
5. LUb terminates its session with LUa, triggering SSCP1 to begin normal deactivation of its session with LUa.
6. SSCP1 initiates normal deactivation of its session with LUa by building a DACTLU and passing it on to VTAM1.
7. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
8. The DLUS VTAM1 encapsulates the DACTLU and forwards it to the DLUR ENa.
9. The DLUR ENa de-encapsulates the DACTLU and forwards it to LUa.
10. LUa returns a positive RSP(DACTLU) to SSCP1 via ENa.
11. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
12. ENa encapsulates the RSP(DACTLU) and sends it to VTAM1.
13. VTAM1 de-encapsulates the RSP(DACTLU) and forwards it to SSCP1.
14. SSCP1 initiates normal deactivation of its session with PU2 by building a DACTPU and passing it on to VTAM1.
15. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
16. The DLUS VTAM1 encapsulates the DACTPU and forwards it to the DLUR ENa.
17. The DLUR ENa de-encapsulates the DACTPU and forwards it to PU2.
18. PU2 returns a positive RSP(DACTPU) to SSCP1 via ENa.
19. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
20. ENa encapsulates the RSP(DACTPU) and sends it to VTAM1.
21. VTAM1 de-encapsulates the RSP(DACTPU) and forwards it to SSCP1. VTAM1 also determines that there are other active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it does not take down the pipe.
22. ENa initiates DLC deactivation if PU2 is a downstream PU.



! Figure 6-11. Forced SSCP-PU/SSCP-LU session deactivation - CP-SVR pipe remains active

1. There is an active session between LUb and LUa.
2. There is an active session between SSCP1 and LUa. It is the only active SSCP-LU session associated with PU2.
- 3.
4. There is an active session between SSCP1 and PU2, but it is not the only active SSCP-PU session using the CP-SVR pipe between ENa and VTAM1. PU2 is the only active PU on the connection between PU2 and ENa.
- 5.
6. SSCP1 initiates forced deactivation of its session with PU2 by building a DACTPU and passing it on to VTAM1. SSCP1 also resets its SSCP-LU session with LUa.
7. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
8. The DLUS VTAM1 encapsulates the DACTPU and forwards it to the DLUR ENa.
- ! 9. VTAM1 builds a RSP(DACTPU) and forwards it to SSCP1, not waiting for the response to return
! from the PU. VTAM1 also determines that there are other active or pending SSCP-PU sessions on its
! CP-SVR pipe to ENa, so it does not take down the pipe.
10. The DLUR ENa de-encapsulates the DACTPU and forwards it to PU2.
11. PU2 returns a positive RSP(DACTPU) to SSCP1 via ENa. PU2 also resets the SSCP-LU session between SSCP1 and LUa.
12. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
- ! 13. ENa encapsulates the RSP(DACTPU) and sends it to VTAM1. VTAM1 de-encapsulates the
! RSP(DACTPU) and discards it, since VTAM1 has already sent a RSP(DACTPU) to SSCP1.
14. ENa initiates DLC deactivation if PU2 is a downstream PU.
15. ENa cleans up its status about the SSCP-LU session between SSCP1 and LUa and the LU-LU session between LUb and LUa. ENa also initiates Session Outage Notification to LUb, so that it too will reset the LU-LU session. During this process, if ENa builds and sends a SESSEND to VTAM1, the DLUS will discard it.



! Figure 6-12. Giveback SSCP-PU/SSCP-LU session deactivation - ANS = CONT

1. There is an active session between LUb and LUa.
2. There is an active session between SSCP1 and LUa. It is the only active SSCP-LU session associated with PU2.
- 3.
4. There is an active session between SSCP1 and PU2, but it is not the only active SSCP-PU session using the CP-SVR pipe between ENa and VTAM1. PU2 is the only active PU on the connection between PU2 and ENa, and PU2 has been defined with ANS = CONT.
- 5.
6. SSCP1 initiates giveback deactivation of its session with PU2 by building a DACTPU(giveback) and passing it on to VTAM1. SSCP1 also resets its SSCP-LU session with LUa.
7. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
8. The DLUS VTAM1 encapsulates the DACTPU(giveback) and forwards it to the DLUR ENa.
9. The DLUR ENa de-encapsulates the DACTPU(giveback); since PU2 is defined as ANS = CONT, the LU-LU sessions will remain active while the SSCP-PU and SSCP-LU sessions are reset. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
10. ENa converts the DACTPU(giveback) into a RSP(DACTPU), encapsulates it, and sends it to VTAM1. ENa must remember that the SSCP-PU and SSCP-LU sessions are reset in SSCP1. ENa also cleans up its status about the SSCP-LU session between SSCP1 and LUa. ENa leaves the LU-LU session and the associated DLC active.
11. VTAM1 de-encapsulates the RSP(DACTPU) and forwards it to SSCP1. VTAM1 also determines that there are other active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it does not take down the pipe.
12. Later on LUa sends a status message to SSCP1, believing the SSCP-LU session is active.
13. ENa, knowing that the SSCP-LU session actually is reset, converts the command into a -RSP with sense code X'0857 0003' (the SSCP-SLU session is inactive) and returns it to LUa.
- ! 14. PU2 sends a status message to SSCP1, believing the SSCP-PU session is active.
- ! 15. ENa, knowing that the SSCP-PU session actually is reset, converts the command into a -RSP with sense code X'087C 0000' (SSCP-PU session not active) and returns it to PU2.

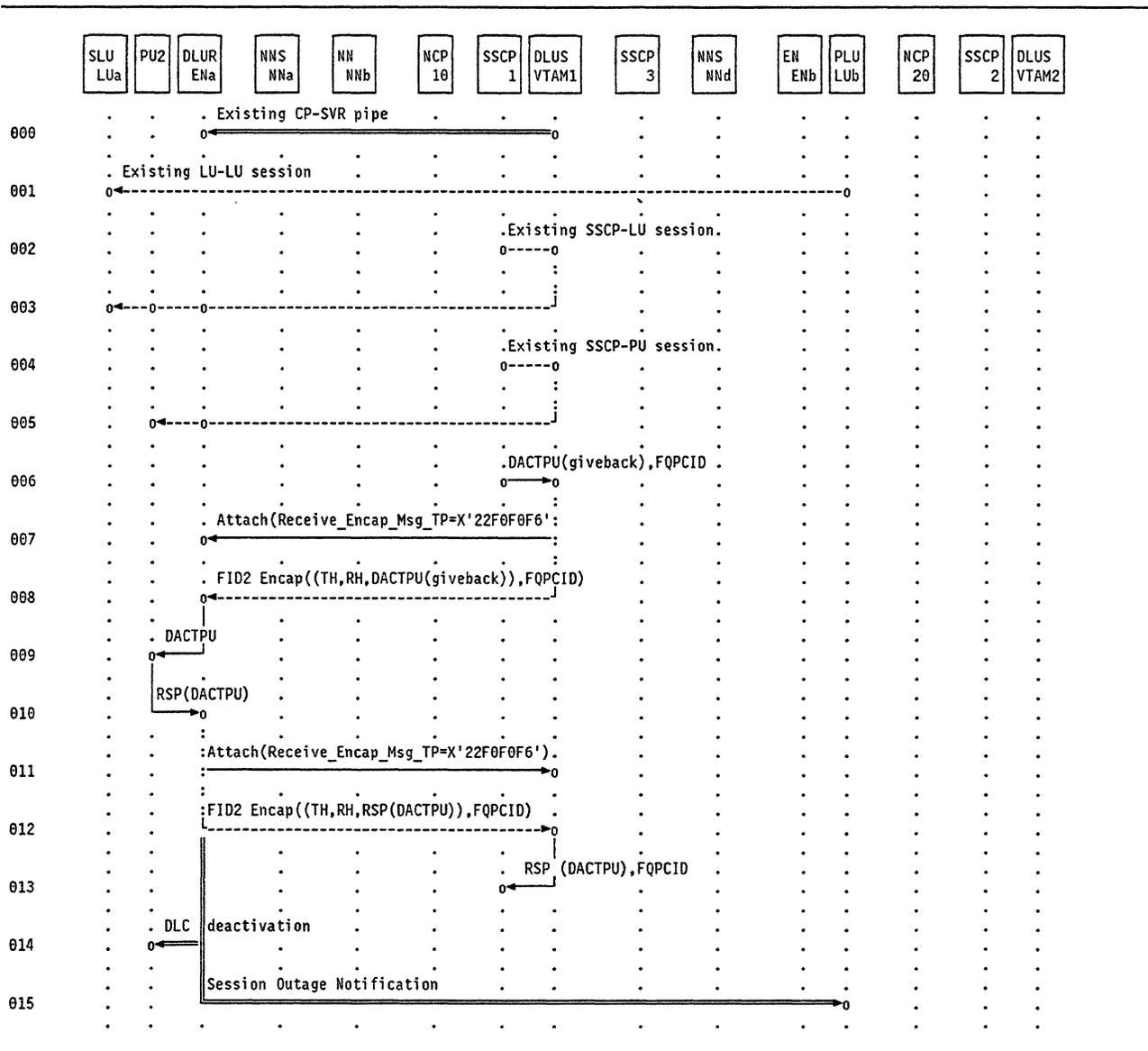


Figure 6-13. Giveback SSCP-PU/SSCP-LU session deactivation - ANS = STOP

1. There is an active session between LUb and LUa.
2. There is an active session between SSCP1 and LUa. It is the only active SSCP-LU session associated with PU2.
- 3.
4. There is an active session between SSCP1 and PU2, but it is not the only active SSCP-PU session using the CP-SVR pipe between ENa and VTAM1. PU2 is the only active PU on the connection between PU2 and ENa, and PU2 has been defined with ANS = STOP.
- 5.
6. SSCP1 initiates giveback deactivation of its session with PU2 by building a DACTPU(giveback) and passing it on to VTAM1. SSCP1 also resets its SSCP-LU session with LUa.
7. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
8. The DLUS VTAM1 encapsulates the DACTPU(giveback) and forwards it to the DLUR ENa.
9. The DLUR ENa de-encapsulates the DACTPU(giveback); since PU2 is defined as ANS = STOP, the LU-LU sessions will be deactivated along with the SSCP-PU and SSCP-LU sessions. This can be accomplished by normal deactivation procedures. So ENa converts the DACTPU(giveback) into a normal DACTPU and forwards it to PU2.
10. PU2 returns a positive RSP(DACTPU) to SSCP1 via ENa. PU2 also resets the SSCP-LU session between SSCP1 and LUa.
11. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
12. ENa encapsulates the RSP(DACTPU) and sends it to VTAM1.
13. VTAM1 de-encapsulates the RSP(DACTPU) and forwards it to SSCP1. VTAM1 also determines that there are other active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it does not take down the pipe.
14. ENa initiates DLC deactivation if PU2 is a downstream PU.
15. ENa cleans up its status about the SSCP-LU session between SSCP1 and LUa and the LU-LU session between LUb and LUa. ENa also initiates Session Outage Notification to LUb, so that it too will reset the LU-LU session.

! 6.4.3 PU Activation After Deactivation

! If a PU was initially activated with ANS = STOP, the deactivation process will deactivate any active LU-LU sessions associated with the PU. When the PU is reactivated, whether with ANS = STOP or ANS = CONT, there are no associated sessions to handle.

! If, however, the PU is initially activated with ANS = CONT, the deactivation process will not tear down the active LU-LU sessions. If the PU is reactivated with ANS = CONT, the LU-LU sessions stay up and will be identified to the new SSCP in a CV X'2A' included in RSP(ACTLU) by the DLUR (see 10.1.2, "DLUR/S Dependent LU Session Awareness" on page 10-2 for more information). If the PU is reactivated with ANS = STOP, the DLUR will deactivate any active LU-LU sessions as part of the PU activation process.

! To summarize:

Table 6-2. PU Reactivation Effects On Active LU-LU Sessions		
PU Activation/ANS Type	Second/STOP	Second/CONT
First/STOP	no LU-LU sessions to handle	no LU-LU sessions to handle
First/CONT	deactivate active LU-LU sessions	leave active LU-LU sessions alone

* 6.5 Additional Error Cases

* 6.5.1 Received GDS X'1500' Without CV X'60'

* Any PIU request encapsulated in a GDS X'1500' which does not include a CV X'60' will be discarded.

* 6.5.2 Received GDS X'1500' (ACTPU Encapsulated) With Missing Or Incomplete CV X'46'

* Any ACTPU request encapsulated in a GDS X'1500' which does not include a complete CV X'46' (e.g., CV missing its signalling information or CV missing completely) will be rejected with a X'0806 0000' sense code.

* 6.5.3 Received GDS X'1500' (ACTPU Encapsulated) Without CV X'51'

* Any ACTPU request encapsulated in a GDS X'1500' which does not include a CV X'51' when one is expected (first ACTPU after the CP-SVR pipe is activated) will be rejected with a X'086C 5100' sense code.

Chapter 7. Unformatted Session Services Management

Once the SSCP-LU session has been activated, the SSCP will respond by sending the appropriate USSMSG10 ("Good Morning" message) to the dependent LU via the CP-SVR pipe. If the RSP(ACTLU) indicates that the LU is enabled, then the SSCP will respond immediately with the USSMSG10. If the RSP(ACTLU) indicates that the LU is not enabled, the SSCP (DLUS node) will wait to receive a NOTIFY(CV0C) indicating that the LU has been enabled before sending the USSMSG10.

The following flow illustrates USSMSG10 processing:

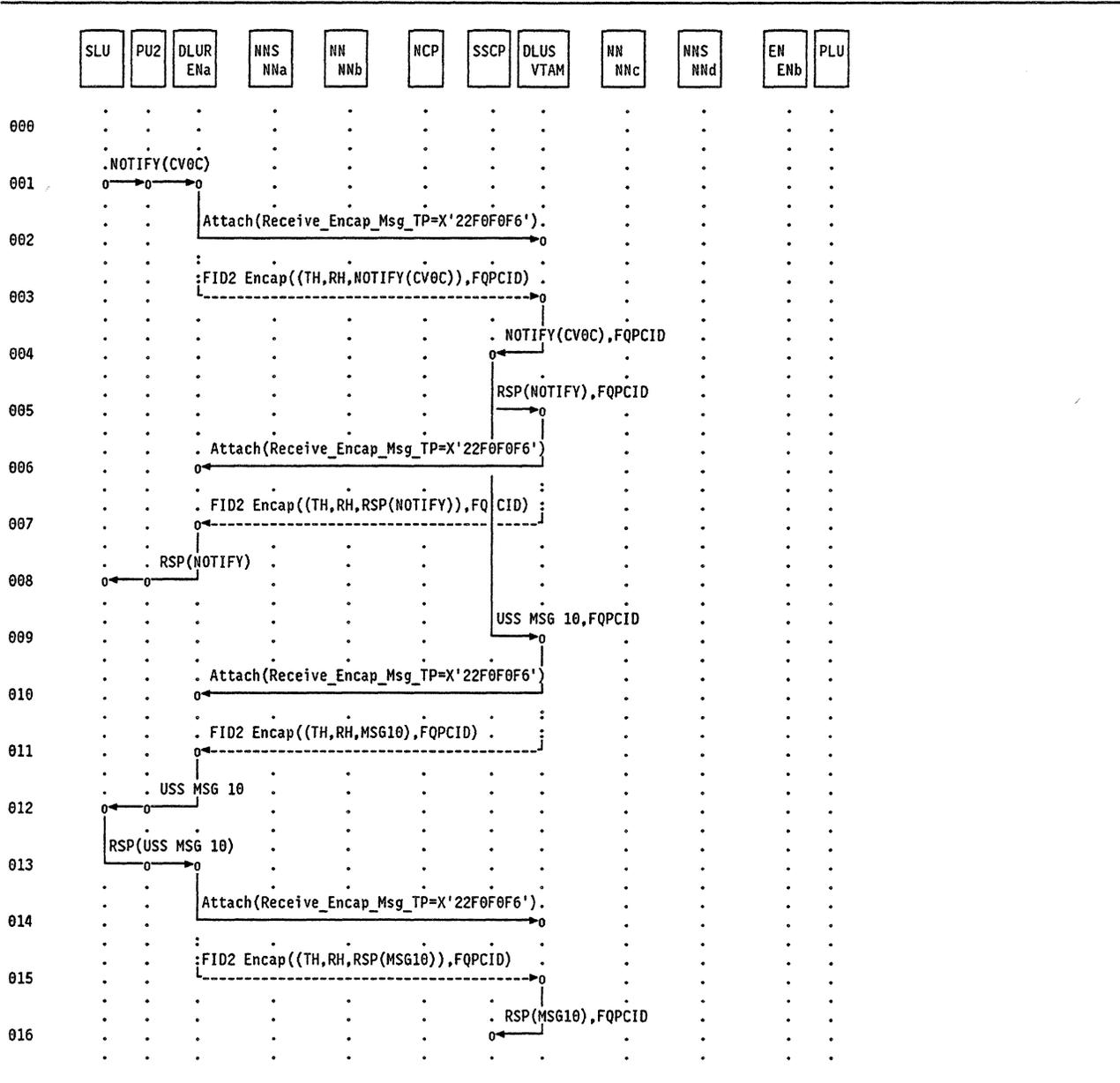


Figure 7-1. Unformatted Session Services Management. Footnotes:

- 0) Encapsulated SSCP-PU and SSCP-LU session traffic is indicated with dotted lines

1. In the case where the RSP(ACTLU) indicates that the LU is not currently enabled, MSG10 is not sent to the LU until it is enabled. The LU indicates that it is enabled by sending a NOTIFY(CV0C).
2. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
3. The DLUR component uses the FQPCID of the PU to encapsulate the NOTIFY(CV0C) PIU and sends it up the CP-SVR pipe to the DLUS node.
4. The DLUS node removes the FID2 Encapsulation GDS variable and sends the NOTIFY to the SSCP.
5. The SSCP sends a NOTIFY RSP.
6. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
7. The DLUS node uses the SSCP-LU FQPCID identifier in the FID2 Encapsulation GDS variable and sends the NOTIFY RSP to the DLUR node.
8. The DLUR node removes the FID2 Encapsulation GDS variable and forwards the NOTIFY RSP to the PU and LU.
9. After detecting the LU is enabled (either via ACTLU enabled or NOTIFY(CV0C)), the SSCP sends a MSG10 to the LU. This is done by sending the MSG10 format from the SSCP to the DLUS component of the DLUS node.
10. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
11. The DLUS component of the node uses the SSCP-LU session identifiers to encapsulate the MSG10 and send it on the CP-SVR pipe to the DLUR node.
12. The DLUR node removes the FID2 Encapsulation GDS variable and forwards the MSG10 message to the PU and LU.
13. The LU generates an acknowledgment to the MSG10 and sends the response to the PU and DLUR component.
14. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
15. The DLUR component will use the SSCP-LU session identifier information to encapsulate the positive response and send it over the CP-SVR pipe to the DLUS node.
16. The DLUS node receives the response, strips the FID2 Encapsulation GDS variable, and passes the response to the SSCP.

In addition to the USSMSG10 flows described above, the SSCP component of the DLUS node will also be required to support other session services message flows for both formatted and unformatted session initiation requests. The DLUS node will also be responsible for session limit management of its dependent (DLUR-attached) LUs. In order for the SSCP component to provide these services, SLU and PLU nodes must send Session Start (SESSST) and Session End (SESSEND) messages to the SSCP. This will be done over the CP-SVR pipe, Session Services Extensions Locate flows, or SSCP-PU4 flows depending upon the location of the PLU node. These message flows are indicated throughout the following sections that describe various session initiation/termination flows. They are also described in Chapter 10, "Dependent LU Session Awareness Management" on page 10-1.

Chapter 8. LU-LU Session Routing

SSCP services (or Session Services) will be provided to the Dependent SLUs by the DLUS node, but session routing will be directly between the PLU and SLU nodes. The DLUS node is the owner of the Dependent SLUs. As such, it is the CP responsible for responding to searches for the SLUs it serves. In this sense, it acts as the CP(SLU). Since the SLU is actually located on or directly attached to the DLUR node, the DLUS will appear in the associated resource hierarchy of Locate Find and Locate Found flows as if it were the NNS(SLU) and the DLUR node as the CP(SLU). In this way, other nodes will cache the DLUS node as the NNS of the SLU and will, therefore, send any ensuing session initiation requests to the DLUR node via the DLUS node. In order to get session routing directly between the PLU and SLU nodes, the associated resource hierarchy will also carry the tail vectors associated with the DLUR node. In the case where the DLUR is a T2.1 EN, these tail vectors will be the real DLUR node TGVs. In the case where the DLUR node is a T2.1 NN, the Cdinit will not carry any tail vectors. ¹

In the first case (when the DLUR node is an EN), TRS at the NNS(PLU) will compute a route from the origin node to one of the partner NNs specified in the DLUR TGVs. The final hop is constructed using the TG number specified in the associated TGV and the CPname of the DLUR node.

In the second case (when the DLUR node is an NN), SS at the NNS(PLU) should pass a REQUEST_ROUTE message to TRS indicating the DLUR (NN) as the target node with no tail vectors. Since the DLUR is actually a NN (and therefore part of the topology database), TRS should respond with the appropriate route.

8.1 BIND Routing

The DLUR must examine RUs on the SSCP-PU and SSCP-LU sessions and save information from these RUs for BIND processing:

8.1.1 ACTLU

Since the RSCV for incoming BINDs will terminate at the DLUR node, the DLUR node must maintain a knowledge of dependent LU locations. This is so that the DLUR node can route incoming BIND messages by SLU name to the dependent SLU. In order to do this, DLUR nodes will maintain a relationship between dependent LU names, their supporting PU, and the physical connectivity to the PU. This is done by examining the ACTLU commands as they are received at the DLUR node (as mentioned in 6.3, "SSCP-LU Session Activation" on page 6-22).

Part of the information maintained by the DLUR will be a session count field, initialized to 0. When the DLUR receives a PLU-initiated BIND, it does not know whether the DLUS has been involved in the LU-LU session activation process or rather that the RU is a surprise BIND. To keep from passing a BIND to an LU already in an LU-LU session, the DLUR will manage the session count field and only pass a

¹ This assumes that the NNS of APPN PLU nodes will support receipt of a Locate_Found indicating an NNCP, ENCP hierarchy with a Cdinit that contains no TGVs. Product polls to date indicate this is not a problem. See B.1, "LU-LU Session Route Calculation" on page B-1 for a description of an alternate approach to provide direct routing between the CP(PLU) and DLUR node, if TGVs are required.

BIND to an LU whose current session count is zero (maximum session count value for a dependent LU is one).

8.1.2 +RSP(ACTLU)

The DLUR must also examine the +RSP(ACTLU), since it carries the X'0C' (LU-LU Session Services Capabilities) control vector.

- ! • In the CV X'0C' is an indicator as to whether or not the dependent LU supports SESSSTs (byte 7, bit 2). The DLUR will set this indicator on, regardless of the value set by the dependent LU for this indicator. This support includes all aspects of session awareness, i.e., SESEND and CV X'2A' as well as SESSST (see 10.1.2, "DLUR/S Dependent LU Session Awareness" on page 10-2 for more information on session awareness).
- ! • In the CV X'0C' are two indicators as to whether or not the dependent/subarea LU supports extended BINDs (byte 7, bits 4 and 6). The DLUR will remember these indicators' settings with its other information about the dependent LU. Both bits off will indicate that the LU does not support extended BINDs, while either bit on will indicate the LU does support extended BINDs.
- ! • In the CV X'0C' is an indicator as to whether or not the dependent LU supports network-qualified names (byte 7, bit 5). The DLUR will remember this indicator's setting with its other information about the dependent LU.

If there already is an active LU-LU session when the +RSP(ACTLU) is received, the active LU-LU session awareness information will be sent to the SSCP in the DLUS by way of the CV X'2A' added to the +RSP(ACTLU) by the DLUR (see 10.1.2, "DLUR/S Dependent LU Session Awareness" on page 10-2 for more information on session awareness).

8.1.3 INIT-SELF

When the DLUR node receives an INIT-SELF from a DLUR-attached dependent LU, it will save the PLU name and URC from the RU and associate this information with the LU.

8.1.4 BIND

When the DLUR node receives a BIND whose RSCV terminates at the DLUR, the DLUR must check its LU table to determine the location of the SLU name specified in the NS header. Once located,

- ! • *SLU session count processing* - the LU's session count will be checked:
 - if the count is zero, it will be incremented to one, and processing will continue;
 - if the count is already one, the BIND will be rejected with a X'0805'000A' sense code;
- ! • *extended BIND processing* - the BIND will be unextended if necessary, based on the extended BIND support indicators in the CV X'0C' in the +RSP(ACTLU) - this involves removing all keyed control vectors and turning off the CVs included indicator;
- ! • *network-qualified name processing* - the NETIDs (if any) will be stripped from the LU names in the BIND if necessary, based on the network-qualified names support indicator in the CV X'0C' in the +RSP(ACTLU);
- ! • *name substitution processing* - the PLU name will be replaced if necessary, based on whether or not the saved and BIND's URCs match;

- ! – if the URCs do not match, BIND processing will continue;
- ! – if the URCs match, the BIND PLU name will be replaced with the saved PLU name, and the
- ! DLUR will discard the saved URC and the saved PLU name;
- *adaptive session pacing processing* - the Adaptive Session-Level Pacing support indicator (byte 9, bit 0 in the BIND) value for the APPN stage (from the DLUR into the APPN network) will be remembered by the DLUR, which will then set the indicator for the Route Extension (REX) stage (from the DLUR to the downstream PU) as follows:
 - if the SLU supports extended BINDs, set the ASP indicator to 1
 - if the SLU does not support extended BINDs, set the ASP indicator to 0

and the BIND (if not rejected) will be forwarded to the appropriate internal or external component.

When a DLUR receives a BIND for a dependent LU and there is no active CP-SVR pipe servicing that dependent LU, the DLUR as a product option can forward the BIND to the LU.

8.1.5 +RSP(BIND)

When the DLUR node receives a +RSP(BIND),

- ! • *extended BIND processing* - the +RSP(BIND) will be reextended if necessary, based on the extended
- ! BIND support indicators in the CV X'0C' in the +RSP(ACTLU) - this involves reinserting the
- FQPCID control vector and turning on the CVs included indicator;
- *adaptive session pacing processing* - the Adaptive Session-Level Pacing support indicator value will be
- % restored to its remembered value, unless the SLU is locally attached and does not support adaptive
- % session pacing, in which case the DLUR can set the ASP indicator to 0 regardless of the remembered
- % value;
- *session awareness processing* - SESSST will be built and sent to the SSCP on the CP-SVR pipe
 - When a DLUR receives a +RSP(BIND) for a dependent LU and there is no active CP-SVR pipe servicing that dependent LU, the DLUR should try to activate a CP-SVR pipe. When the CP-SVR pipe is activated, the SESSST should not be sent on the new CP-SVR pipe. The active LU-LU session awareness information will be sent to the SSCP in the DLUS by way of the CV X'2A' added to the +RSP(ACTLU) by the DLUR (see 10.1.2, “DLUR/S Dependent LU Session Awareness” on page 10-2 for more information on session awareness).

and the +RSP(BIND) will be forwarded to the appropriate internal or external component.

8.1.6 LU-LU Session Deactivation

When the DLUR node receives a -RSP(INIT-SELF) or detects a link failure affecting an LU-LU session, it will discard the saved URC and the saved PLU name associated with the dependent LU.

When the DLUR node receives a -RSP(BIND), UNBIND, or detects a link failure affecting an LU-LU session, it will reset the session count for the associated dependent LU to zero.

8.2 DLUR EN Tail Vector Registration

In order for the DLUS node to send Locate Find/Found flows with the appropriate associate resource hierarchy as described above, it will have to know the DLUR tail vectors. In the case where the DLUR node is a NN this will not be necessary. The EN DLUR node will be required to register its tail vectors with the DLUS node; this registration will be done when the DLUS and DLUR are in the same subnet as well as when they are in different subnets.

DLUR tail vector registration will involve sending TOPOLOGY_DATABASE_UPDATES (TDUs) to the DLUS node:

- if the DLUR and the DLUS are adjacent, and if the DLUS supports receipt of TDUs (indicated in the CP Capabilities (X'12C1') GDS variable, then the TDUs will be sent over the CP-CP session between the two nodes;
- otherwise the TDUs will be sent over the CP-SVR pipe to the DLUS node.

The DLUS node will receive, but not propagate, these TDUs.

This will require the DLUR and DLUS CPs to be modified such that the TDU send and receive TPs can also run over the CP-SVR pipe using the CPSVRMGR mode/COS. No special CP-SVR pipe should be brought up for this purpose. The existing CP-SVR pipe should be used. Since a DLUR may have CP-SVR pipes with multiple DLUS nodes, these TDUs must be sent over each CP-SVR pipe to each partner DLUS node.

8.2.1 DLUR/S Capabilities Exchange

TDUs can be sent on the pipe any time after CP-SVR pipe activation completes. Activation includes the exchange of DLUR/S Capabilities control vectors, so TDUs can be sent by the DLUR after:

- DLUR-initiated pipe activation: receipt of +RSP(REQACTPU) from DLUS
- DLUS-initiated pipe activation: transmission of +RSP(ACTPU) to DLUS

Included within the DLUR/S Capabilities control vector are fields relevant to EN tail vector registration:

- a Flow Reduction Sequence Number (FRSN) -
 - the DLUR will always set this field to zero in the DLUR/S Capabilities control vector; this will tell the DLUS to throw away any tail vectors previously registered by this DLUR;
 - the DLUS will set this field to the last FRSN value received from the DLUR;
- an indicator for support of the RECEIVE_TDU_TP (X'22F0F0F4') service transaction program -
 - when set, the sending node supports receipt of TDUs on this session
 - the DLUS should set this indicator on;
 - the DLUR should set this indicator off.

8.2.2 Initial EN Tail Vector Registration

After CP-SVR pipe activation, the DLUR will send all of its tail vectors to the DLUS (for more information, see "Processing CP_STATUS Signals" in the SNA APPN Architecture Reference).

The DLUS will process the received TDUs as described in "TDU Processing" in the SNA APPN Architecture Reference, with one exception: these TDUs should not be propagated to other NNs.

8.2.3 EN Tail Vector Registration Updates

When there is a change to a previously registered tail vector, the DLUR will send an update TDU to the DLUS (for more information, see "Processing Resource Updates" in the SNA APPN Architecture Reference).

The DLUS will process the received TDUs as described in "TDU Processing" in the SNA APPN Architecture Reference, with one exception: these TDUs should not be propagated to other NNs.

Chapter 9. Dependent LU Session Establishment Flows

The following flow diagrams illustrate how LU-LU sessions are established using the DLUR/S option sets.

9.1 USS Flows For SLU Init

The following flows represent the USS management flows for an SLU initiated session regardless of the PLU location.

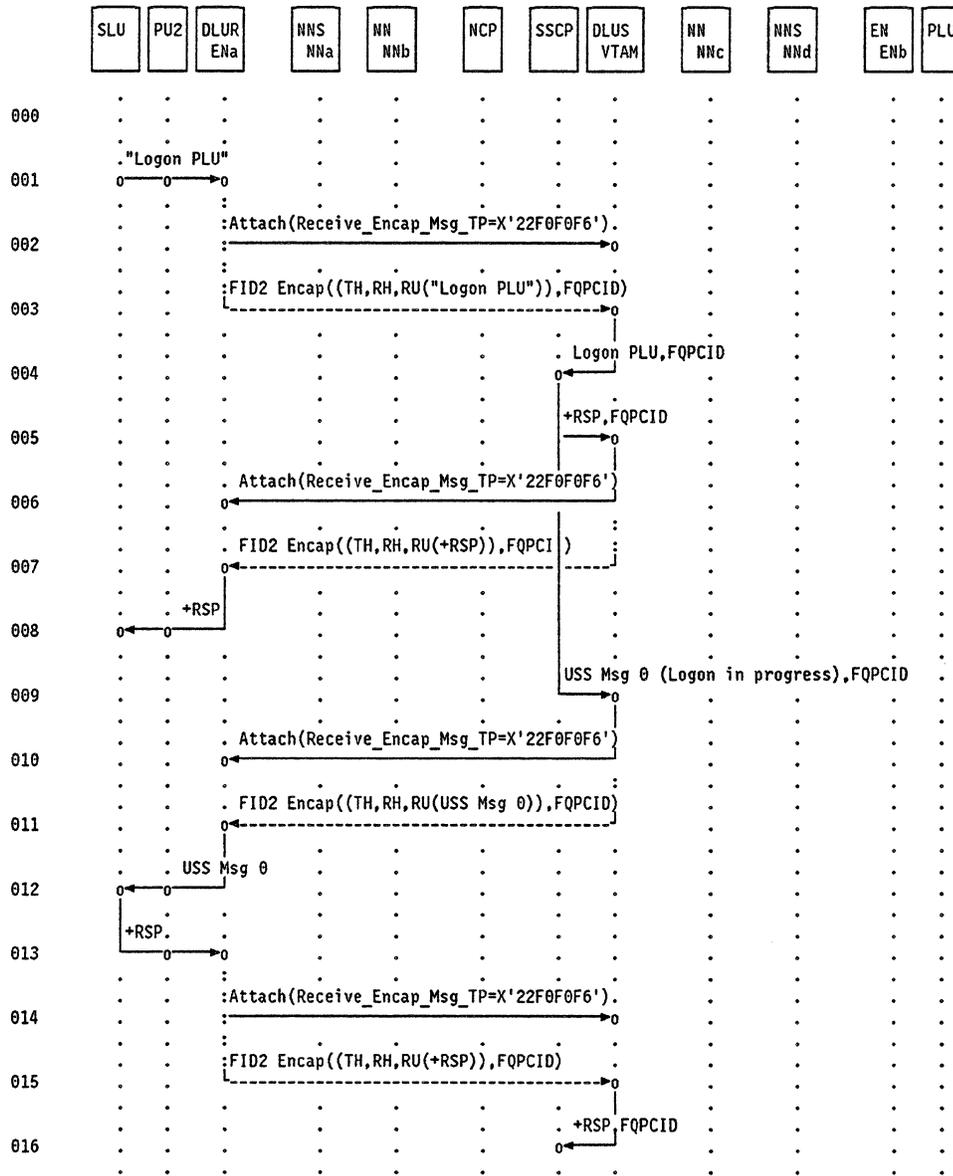


Figure 9-1. Unformatted Session Services SLU-initiated LU-LU session

1. The Logon PLU command is sent from the SLU through the PU to the DLUR component of the EN.
2. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
3. The DLUR component encapsulates the Logon command using the SSCP-LU session identifiers and sends it up to the DLUS node via the CP-SVR pipe.
4. The DLUS node removes the FID2 Encapsulation GDS variable and sends the Logon PLU command to the SSCP.
5. The SSCP receives the Logon command and issues a positive response to the LU via the DLUS component.
6. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
7. The DLUS component uses the SSCP-LU session identifiers to encapsulate the +RSP and send it to the DLUR node via the CP-SVR pipe.
8. The DLUR node removes the FID2 Encapsulation GDS variable and forwards the +RSP to the LU via the PU.
9. The SSCP also follows the +RSP to the Logon command with a USS Msg 0 (logon in progress) message.
10. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
11. The DLUS component encapsulates the USS Msg 0 and sends it over the CP-SVR pipe to the DLUR node.
12. The DLUR node removes the FID2 Encapsulation GDS variable and sends the USS Msg 0 to the LU via the PU.
13. The LU responds to receipt of the USS Msg 0 with a +RSP.
14. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
15. The DLUR component encapsulates the +RSP and sends it up the pipe to the DLUS node.
16. The DLUS node removes the FID2 Encapsulation GDS variable and sends the +RSP to the SSCP. At this point, the USS MSG handling is complete, and the session initiation proceeds.

9.2 SLU Init To APPN PLU

The following diagram illustrates the flows that occur between the DLUS node and an APPN PLU for a SLU-initiated session that eventually result in a BIND flowing directly between the APPN PLU and DLUR (SLU) nodes. These flows have been added to describe the relationship between DLUR/S architecture and Session Services Extensions.

Prior to the flows illustrated in the diagram, some SLU supported by a DLUR node has requested a session with the PLU. Flows of this type are described in Figure 9-1 on page 9-2. Flows in this diagram begin after the DLUS has been notified of the SLU session initiation request. In this case, the PLU is in the APPN portion of the network, so Session Services Extensions SLU Initiation session flows are used between the NN side of the DLUS Interchange node and the APPN CP(PLU).

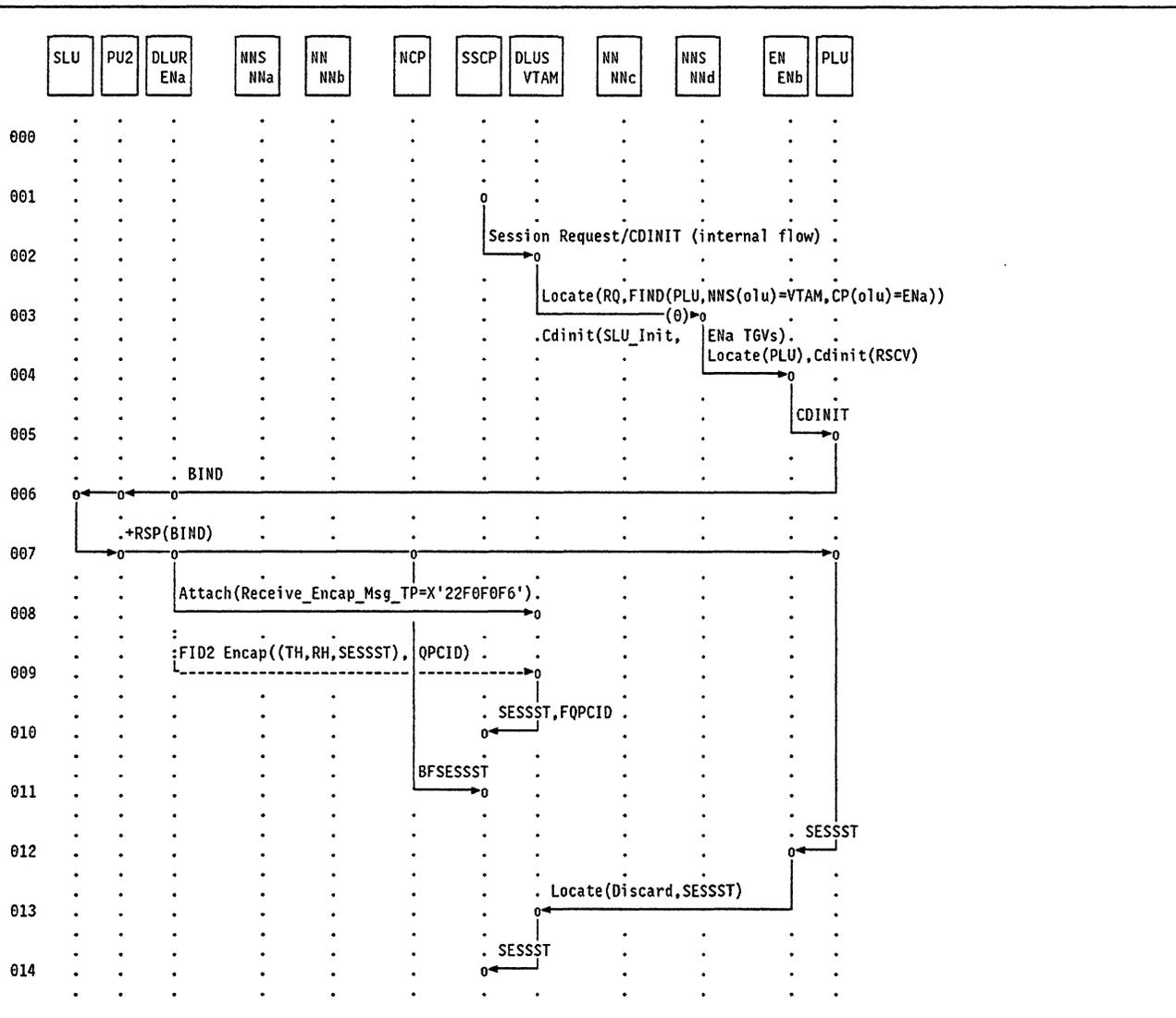


Figure 9-2. Unformatted Session Services SLU-initiated LU-LU session to APPN PLU. Footnotes:
 0) EN DLUR TGVs are provided from previous TGV registration over the CP-SVR pipe

1. The session initiation request is passed from the SSCP portion of the interchange node to the NN portion. This signals to the NN that the APPN portion of the network should be searched for the PLU for an SLU initiated session.
2. The NN sends a Locate Find Cdinit message into the APPN network to locate the PLU. In this case, the location of the PLU is already known, so a directed locate is issued directly to the CP(PLU) via the NNS(PLU). The Locate hierarchy indicates the NNS(OLU)=VTAM and the CP(OLU)=ENa. This will cause the NNS(PLU) to cache the DLUS node as the network node server (NNS) of the CP(OLU). The Cdinit specifies the OLU=SLU, initiation type of SLU Init, and the tail vectors of the DLUR node. The DLUR tail vectors are known from previous registration flows between the DLUR and DLUS nodes. See Chapter 8, "LU-LU Session Routing" on page 8-1. The DLUS-served LU indicator is turned on since the SLU is a dependent LU served by this DLUS.
3. When the NNS(PLU) receives the Locate Find Cdinit from the DLUS node (since it knows the location of the PLU from previous cached entries), it calculates a route between the CP(PLU) and CP(SLU)=DLUR node. This route is computed because the DLUR TGVs have been provided. See Chapter 8, "LU-LU Session Routing" on page 8-1 for further details. The Locate Find Cdinit with RSCV is then passed to the CP(PLU).
4. The CP(PLU) passes the CDINIT with SLU Session Characteristics (CV31 and CV65) to the PLU.
5. The PLU and CP(PLU) use the session characteristics passed in the Cdinit and the RSCV to construct a BIND image and send it to the CP(SLU). The FQPCID generated by the CP(SLU) is used to identify the BIND. If for any reason the CP(PLU) is incapable of sending a BIND to the CP(SLU) (i.e., PLU at session limit), an appropriate sense code is generated and returned to the DLUS node as a negative Cdinit reply.
6. The DLUR node receives the extended BIND image sent from the CP(PLU). Recognizing that this BIND is destined for a dependent LU, the DLUR node processes the BIND (optionally unextending it, removing NETIDs, performing name substitution, setting pacing indicator). This BIND is then forwarded to the SLU via the PU2. In the case where the PU is physically downstream from the DLUR node, the DLUR node will have to check the SLU name in the NS field of the BIND and compare this to its LU name table described in 6.3, "SSCP-LU Session Activation" on page 6-22. This information can then be used to determine which link to send the BIND over and what DAF to put in the TH field.
7. A +RSP(BIND) is generated by the SLU and flows along the session path to the PLU. The DLUR processes the +RSP(BIND) (optionally reextending it, setting pacing indicator).
8. Upon receipt of the +RSP(BIND), the DLUR node issues a SESSST RU to notify the SSCP that the session has started. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
9. The DLUR node encapsulates the SESSST RU with the SSCP-LU session identifiers and sends it up the CP-SVR pipe to the DLUS node.
10. The DLUS node receives the encapsulated SESSST command, de-encapsulates it and forwards it to the SSCP.
11. Upon receipt of the +RSP(BIND), NCP creates a BFSESSST RU and sends it to the SSCP.
12. When the PLU receives the +RSP(BIND), it generates a SESSST RU.
13. The APPN PLU node must then follow the Locate chain to the DLUS node with a Locate(discard) carrying the SESSST notification to the SSCP.
14. The SESSST signal is then passed from the DLUS component to the SSCP.

9.3 APPN PLU Init To A Dependent APPN SLU

The following flows illustrate an APPN PLU-initiated session to a dependent APPN SLU node being supported by DLUR/S architecture.

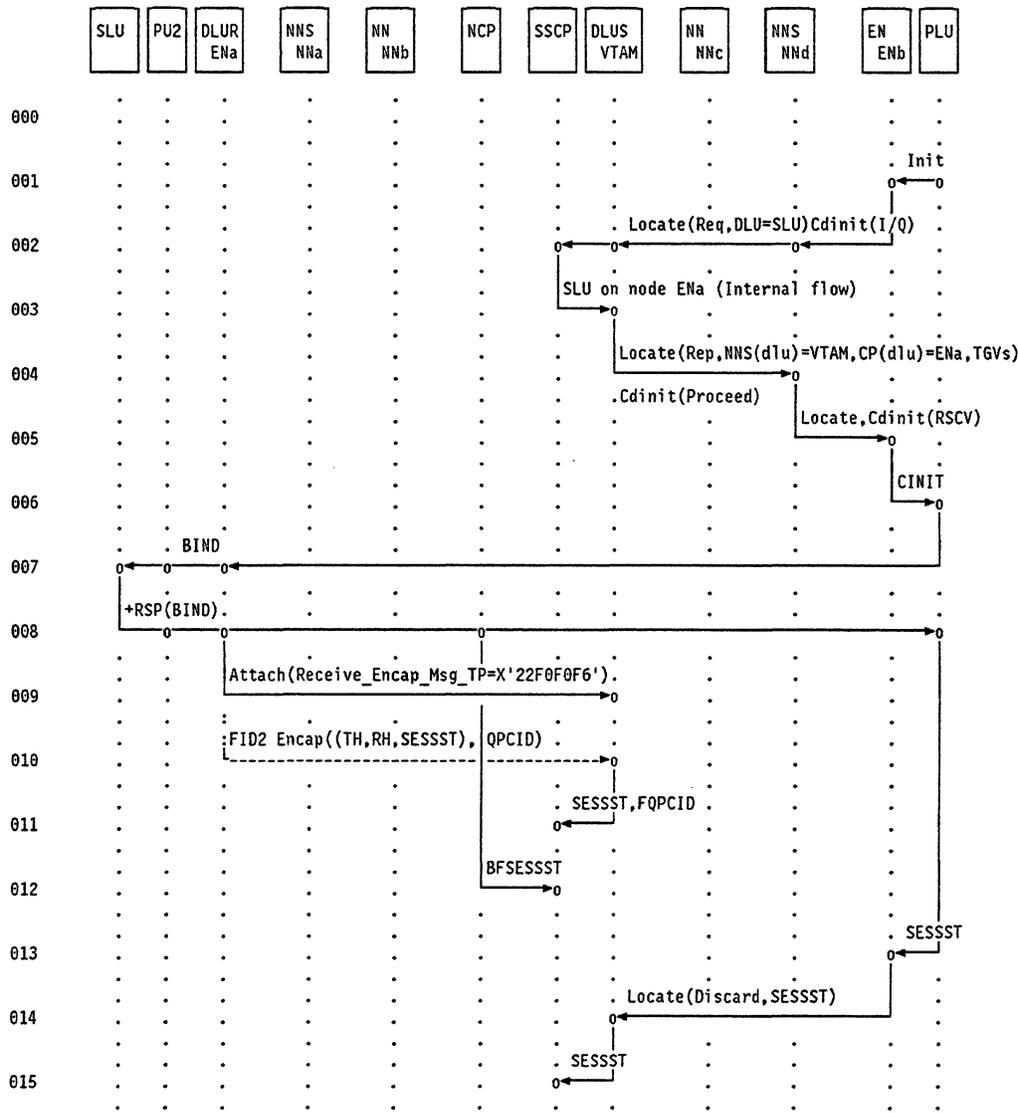


Figure 9-3. APPN PLU-initiated LU-LU session to a dependent SLU. Footnotes:

- 0) EN DLUR TGVs are provided from previous TGV registration over the CP-SVR pipe

1. The PLU issues an Init for the SLU.
2. The APPN CP(PLU) issues a Locate Find Cdinit for the SLU. This illustration assumes that the NNS(PLU) has already cached information regarding the location of the CP(SLU) and, therefore, sends a directed locate to the NNS(SLU) or DLUS node. If this were not the case, the NNS(PLU) would issue a broadcast search. When the broadcast search request reached the CP(SLU) (or DLUS), it would respond Locate Found indicating that the CP(PLU) should 'resubmit a directed' search to the DLUS node. *It is important to note that the DLUS node is the CP(SLU). The DLUR node would not respond positively to such a search as it is not the owner or CP of the SLU.* The SLU hierarchy would again indicate that the SLU exists on the CP(SLU) which is served by the DLUS node and has the real CP(SLU) tail vectors. Once the DLUS node receives the directed search request for the SLU, it will pass the SLU search request to the SSCP component.
3. The SSCP component indicates to the DLUS component that the SLU is located on the DLUR node.
4. The DLUS responds to the NNS(PLU) that the SLU has been located on the CP(SLU) which is served by the DLUS node (this is so that the NNS(PLU) will cache the DLUS node as the NNS of the dependent SLU) and has the *real TGVs of the CP(SLU)*. The real TGVs of the CP(SLU) will cause the NNS(PLU) to calculate a route directly from the CP(PLU) to the CP(SLU) (see Chapter 8, "LU-LU Session Routing" on page 8-1 for further details). Since the original request was an Initiate or Queue request, the DLUS node responds with a Cdinit(Proceed). The DLUS-served LU indicator is turned on since the SLU is a dependent LU served by this DLUS.
5. When the NNS(PLU) receives the Locate Found, it uses the CP(SLU) TGVs to calculate a route from the CP(PLU) to the CP(SLU). This RSCV is returned to the CP(PLU).
6. The CP(PLU) receives the Locate Cdinit with SLU session characteristics and RSCV. It passes the CINIT signal to the PLU which generates a BIND image to be sent to the CP(SLU).
7. The BIND image is then generated and passed to the APPN PLU node which appends an RSCV and FQPCID to extend the BIND. The BIND then flows through the APPN network to the DLUR node. When the DLUR node receives the BIND, it processes it (optionally unextending it, removing NETIDs, performing name substitution, setting pacing indicator) before passing it to the SLU via its PU2. In the case of downstream PUs, the DLUR uses the SLU name specified in the NS field of the BIND and its LU table (see 6.3, "SSCP-LU Session Activation" on page 6-22) to determine the target PU and link (if applicable).
8. A +RSP(BIND) is generated by the SLU and flows along the session path to the PLU. The DLUR processes the +RSP(BIND) (optionally reextending it, setting pacing indicator).
9. Upon receipt of the +RSP(BIND), the DLUR node generates a SESSST RU to notify the SSCP that the session has started. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
10. The DLUR node encapsulates the SESSST RU with the SSCP-LU session identifiers and sends it up the CP-SVR pipe to the DLUS node.
11. The DLUS node receives the encapsulated SESSST command, de-encapsulates it and forwards it to the SSCP.
12. Upon receipt of the +RSP(BIND), NCP creates a BFSESSST RU and sends it to the SSCP.
13. When the PLU receives the +RSP(BIND) it generates a SESSST signal to the CP(PLU).
14. The CP(PLU) then uses a Locate(discard) to follow the Locate chain back to the DLUS node and carry the SESSST signal to the SSCP.
15. The SESSST signal is then passed from the DLUS component to the SSCP.

9.4 USS Flows For SLU Init To A Subarea PLU

The following diagram illustrates the flows for a DLUR-supported SLU that issues an USS SLU Init request and the PLU is located in the subarea portion of the network. Note that the USS MSG 0 (Logon in Progress) message is not issued until the PLU has been located and the CDINIT RU has been processed by the SSCP(PLU).

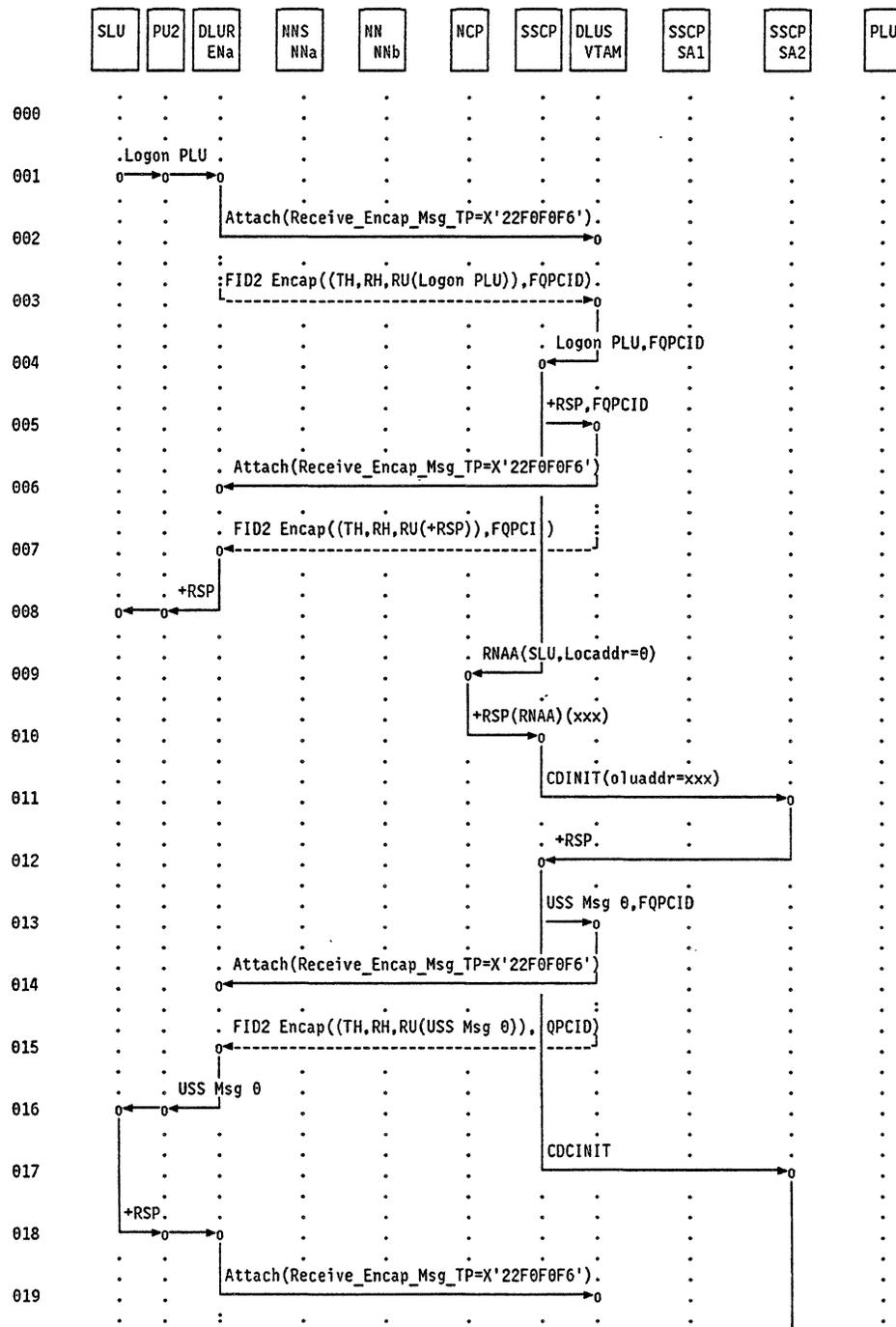
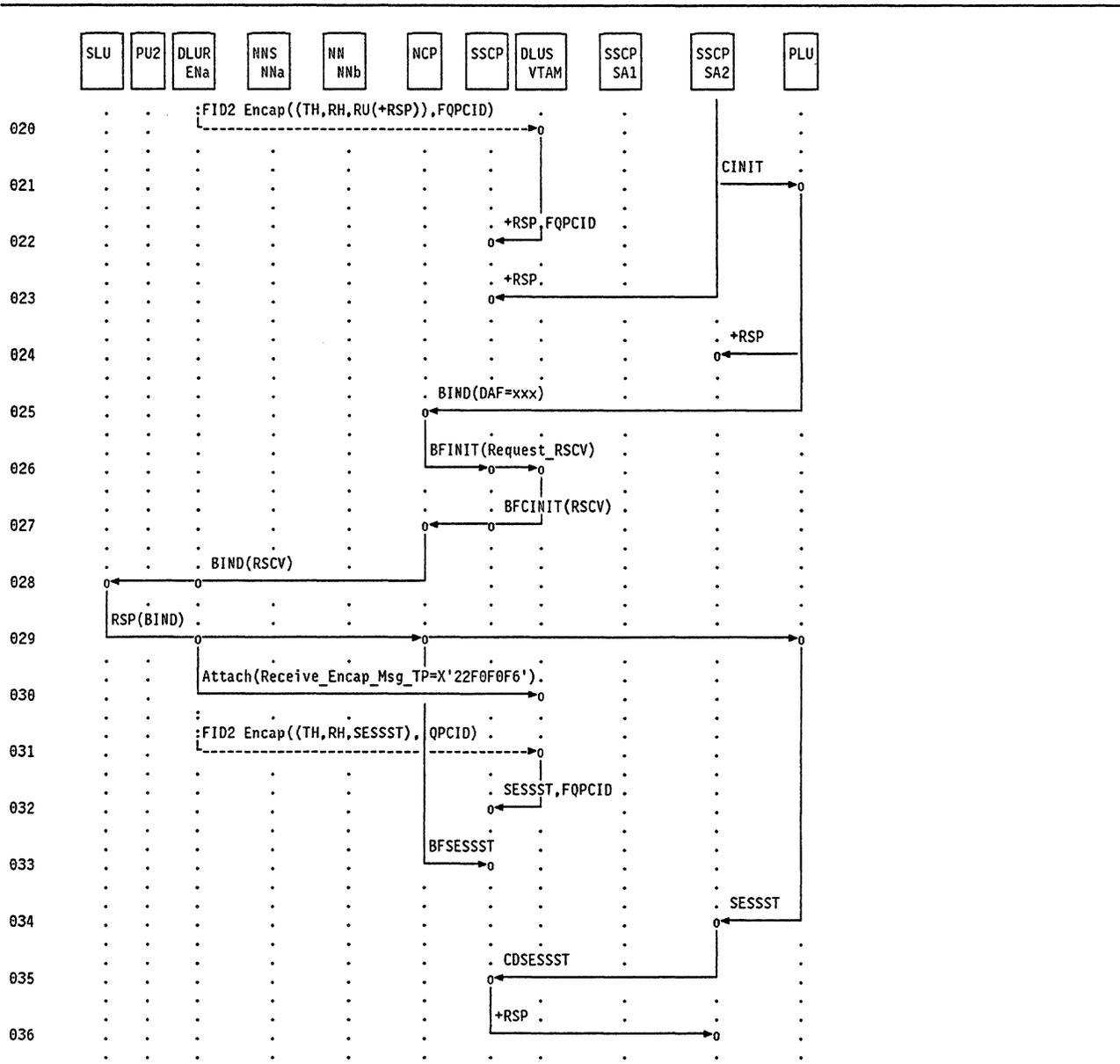


Figure 9-4. Unformatted Session Services SLU-initiated LU-LU session to subarea PLU



Footnotes:

- 0) Dotted lines indicate SSCP-PU or SSCP-LU flows over the CP-SVR pipe

1. The SLU sends a "Logon PLU" command to the CP (DLUR node) via its PU.
2. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
3. The DLUR component encapsulates the Logon command on the SSCP-LU session and sends it up the CP-SVR pipe to the DLUS node.
4. The DLUS component removes the FID2 Encapsulation GDS variable and passes the Logon command to the SSCP.
5. The SSCP issues a positive response (+ RSP) to receipt of the Logon command.
6. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
7. The DLUS component encapsulates the + RSP and sends it on the encapsulated SSCP-LU session back to the SLU via the DLUR node.
8. The DLUR node removes the FID2 Encapsulation GDS variable and forwards the + RSP to the LU via the PU.
9. In response to the Logon PLU command received from the SLU, the SSCP issues an RNAA to the Boundary Function NCP. This establishes a subarea address for the SLU.
10. The boundary function responds with address xxx.
11. The SSCP then issues a CDINIT to SSCP SA2 specifying xxx as the subarea SLU address.
12. The SSCP SA2 issues a CDINIT RSP back to the DLUS SSCP.
13. Receipt of the + RSP to the CDINIT indicates to the SSCP that the PLU has been located. This causes the SSCP to send USS MSG 0 (Logon in progress) to the SLU. This is passed from the SSCP to the DLUS component.
14. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
15. The DLUS component encapsulates the USS MSG 0 on the SSCP-LU session and sends it to the DLUR node.
16. The DLUR node removes the FID2 Encapsulation GDS variable and sends the USS MSG 0 to the LU via the PU2.
17. Meanwhile, the DLUS SSCP issues a CDCINIT to SA2 in response to the CDINIT RSP.
18. The LU issues a + RSP to the MSG 0 and sends it to the DLUR component.
19. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
20. The DLUR node encapsulates the message on the SSCP-LU session and sends it to the DLUS node.
21. Meanwhile after receiving the CDCINIT, SA2 forwards the CINIT signal to the PLU.
22. The DLUS component removes the FID2 Encapsulation GDS variable and forwards the USS MSG 0 + RSP to the SSCP.
23. SA2 provides a + RSP to the CDCINIT from the DLUS SSCP.
24. The PLU issues a + RSP to the CINIT from its SSCP, SA2.
25. The PLU follows the CINIT with a BIND specifying a destination address (DAF) of xxx which was assigned by the SLU boundary function in step 10.
- * 26. When the BIND reaches the boundary function, it must request an RSCV to send the BIND into the APPN portion of the composite network. A BFINIT(request_rscv) is issued back to the interchange node.

27. The interchange node responds with a BFCINIT and an RSCV computed to the DLUR node. This RSCV is computed using information resident in the SSCP portion of the DLUS node that specifies the DLUR node supporting the SLU. The DLUR tail vector information is present from previous TGV registration activities (see Chapter 8, "LU-LU Session Routing" on page 8-1 for further information).
28. The boundary function re-issues the BIND in extended format with the FQPCID and RSCV into the APPN portion of the network. When the DLUR node receives the BIND, it forwards to the LU. If the LU is on a downstream PU, the DLUR must use the LU table to find the appropriate target PU (see 6.3, "SSCP-LU Session Activation" on page 6-22).
29. Meanwhile, the SLU responds to the BIND with a +RSP(BIND) over the data path to the PLU.
30. Upon receipt of the +RSP(BIND), the DLUR node generates a SESSST RU to notify the SSCP that the session has started. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
31. The DLUR node encapsulates the SESSST RU with the SSCP-LU session identifiers and sends it up the CP-SVR pipe to the DLUS node.
32. The DLUS node removes the FID2 Encapsulation GDS variable and sends the SESSST RU to the SSCP.
33. When the subarea boundary function receives the +RSP(BIND), it forwards a BFSESSST to the DLUS SSCP.
34. When the PLU receives the +RSP(BIND), it sends a SESSST RU to its SSCP, SA2.
35. SA2 then sends a CDESSST RU to the SSCP of the SLU (the DLUS node).
36. The DLUS node issues a +RSP to the CDESSST RU.

9.5 USS Flows For LU-LU Session Termination

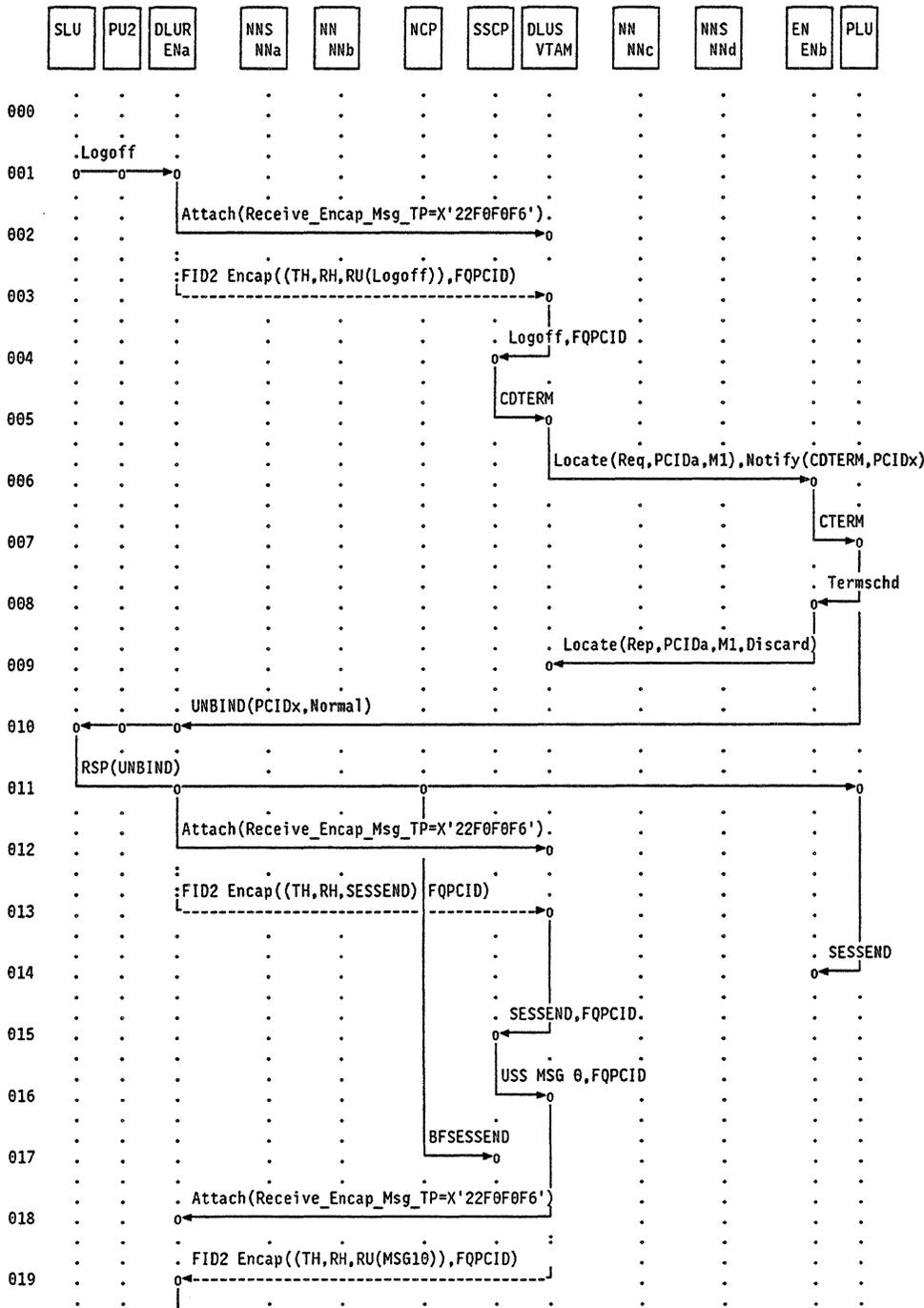
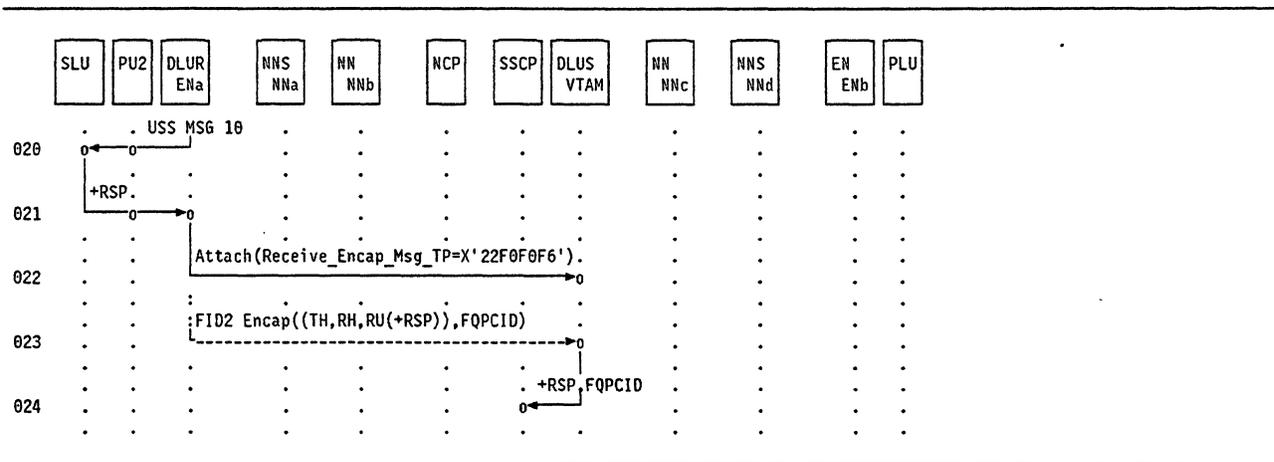


Figure 9-5. Unformatted Session Services LU-LU session termination



1. The SLU sends a "Logoff PLU" command to the CP (DLUR node) via its PU.
2. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
3. The DLUR component encapsulates the Logoff command on the SSCP-LU session and sends it up the CP-SVR pipe to the DLUS node.
4. The DLUS component removes the FID2 Encapsulation GDS variable and passes the Logoff command to the SSCP.
5. The SSCP then issues a CDTERM to its DLUS component.
6. The DLUS issues a Locate Notify Cdterm for the PLU. The Locate request is sent with a new FQPCID (PCIDa) generated by the DLUS node. The Notify Cdterm indicates that the termination of the session is requested and contains the FQPCID (PCIDx) of the session to be terminated. Note that both of these FQPCIDs are different from the FQPCID included in the FID2 Encapsulation GDS variable which correlates encapsulated messages with the correct dependent LU's PU.
7. When the Locate request is received by the CP(PLU), it passes a CDTERM signal to the PLU.
8. Once termination has been scheduled, the PLU notifies the CP(PLU).
9. The CP(PLU) sends a Locate reply, with the 'discard' indicator set, back to the DLUS node to indicate the status of the terminate request.
10. After the session has quiesced, the PLU generates an UNBIND image which flows along the session path to the SLU.
11. The SLU responds to the UNBIND with a RSP(UNBIND) over the data path to the PLU.
12. Upon receipt of the RSP(UNBIND), the DLUR node generates a SESSEND RU to notify the SSCP that the session has ended. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
13. The DLUR node encapsulates the SESSEND RU with the SSCP-LU session identifiers and sends it up the CP-SVR pipe to the DLUS node.
14. Upon receipt of the RSP(UNBIND), the PLU generates a SESSEND signal to the CP(PLU).
15. The DLUS node removes the FID2 Encapsulation GDS variable and sends the SESSEND RU to the SSCP.
16. The SSCP then sends a MSG10 ("Good Morning" message) to the DLUS component of the DLUS node.
17. Upon receipt of the RSP(UNBIND), the subarea boundary function creates a BFSESSST RU and sends it to its SSCP.
18. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
19. The DLUS component of the node uses the SSCP-LU session identifiers to encapsulate the MSG10 and send it on the CP-SVR pipe to the DLUR node.
20. The DLUR node removes the FID2 Encapsulation GDS variable and forwards the MSG10 message to the PU and LU.
21. The LU generates an acknowledgment to the MSG10 and sends the response to the PU and DLUR component.
22. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.

23. The DLUR component will use the SSCP-LU session identifier information to encapsulate the positive response and send it over the CP-SVR pipe to the DLUS node.
24. The DLUS node receives the response, strips the FID2 Encapsulation GDS variable, and passes the response to the SSCP.

Chapter 10. Dependent LU Session Awareness Management

Session awareness (SAW) involves an SSCP being informed of the start and end of LU-LU sessions. For a given LU-LU session, session awareness RUs are sent for both the PLU and the SLU. Host application LUs send their own Session Started (SESSST) and Session Ended (SESSEND) RUs. The Boundary Function (BF) builds session awareness RUs for the other LUs, SESSST and SESSEND for dependent LUs, and BFSESSST and BFSESSEND for independent LUs.

Session awareness also includes informing a takeover SSCP of the current active LU-LU sessions. This information is carried in RSP(ACTLU) for dependent LUs, and in BFSESSINFO for independent LUs.

Since a DLUR will provide a BF (Boundary Function) for a dependent LU, the DLUR will provide session awareness information to the SSCP on the SSCP-LU session via the DLUR's CP-SVR pipe with its DLUS.

A DLUR-attached dependent LU can only be a SLU; session awareness function for PLUs remains unchanged. Session awareness function for independent LUs also remains unchanged.

10.1 Functional Overview

10.1.1 Pre-DLUR/S Dependent LU Session Awareness

Prior to DLUR/S, the dependent LU's SSCP-LU and LU-LU sessions must route through the same BF-NCP. For dependent LU session awareness, this means that:

- LU-LU Session Activation
 - Upon LU-LU session activation, the BF-NCP builds a SESSST RU and sends it to the SSCP.
- SSCP Takeover (Takeover & Giveback)
 - During SSCP takeover, the BF-NCP intercepts the RSP(ACTLU) RU built by the dependent LU and, if the dependent LU has an active LU-LU session, inserts several control vectors, including a Session Information (X'2A') control vector in the RU before sending it on to the SSCP (the same also occurs during SSCP giveback).
- LU-LU Session Deactivation (Normal & Abnormal)
 - Upon LU-LU session deactivation, the BF-NCP builds a SESSEND RU and sends it to the SSCP.

10.1.2 DLUR/S Dependent LU Session Awareness

With the DLUR/S function, the DLUR will assume BF responsibilities previously handled by the BF-NCP for dependent LUs associated with it. The session awareness responsibilities will include building the SESSST and SESSEND; including in these RUs, as well as the RSP(ACTLU), the proper dependent LU session information; and sending all of these RUs to the DLUS's SSCP on the SSCP-LU session encapsulated in the CP-SVR pipe.

As part of the shift of function to the DLUR, the BF-NCP will now consider a DLUR-attached dependent LU to be an independent LU. This means that there will be two sources of session awareness for the same dependent LU:

1. the DLUR will build and send dependent LU session awareness RUs to the DLUS's SSCP;
2. the BF-NCP through which the dependent LU's LU-LU session flows will build and send independent LU session awareness RUs to its SSCP.

The RUs built/updated by the DLUR and the BF-NCP for DLUR-attached dependent LU session awareness are listed in the following table:

Table 10-1. DLUR/S Dependent LU Session Awareness RUs		
	DLUR (dependent LU SAW)	BF-NCP (independent LU SAW)
LU-LU Session Activation	SESSST	BFSESSST
SSCP Takeover	RSP(ACTLU)	BFSESSINFO
LU-LU Session Deactivation	SESEND	BFSESEND

There are several considerations which are introduced with having two sources of session awareness for the same dependent LU:

- The contents of the dependent LU and independent LU SAW RUs will not be identical. The DLUR will not be aware of the network address of the dependent LU's LU-LU session partner; this information will only appear in the independent LU SAW RUs .
- The SSCP receiving a dependent LU SAW RU may not be the same SSCP which will receive the corresponding independent LU SAW RU. This may happen since, with DLUS, the dependent LU's SSCP-LU and LU-LU sessions may now route through different BF-NCPs, which may be owned by different SSCPs. The SSCP receiving the independent LU SAW RU need not be in an Interchange Node with a DLUS; the SSCP receiving the dependent LU SAW RU must have a DLUS associated with it.
- The LU-LU session's route may not include a BF-NCP at all, in which case no independent LU SAW RU will be built.
- Even if both the dependent LU and independent LU SAW RUs are sent to the same SSCP, there is no guarantee as to which will arrive first.

So the DLUS's SSCP may receive both RUs and correlate them, or it may only receive one of the RUs. Correlation will be done via the FQPCID (X'60') control vector which is included in each of these RUs.

Not only will the contents of the dependent LU and independent LU SAW RUs be different, so also will the contents of the dependent LU SAW RUs built by a DLUR and a BF-NCP be different. New formats of the dependent LU SAW RUs will be defined for the DLUR to build and send to the DLUS's SSCP; these formats will have several items excluded (see 13.3, "Dependent LU Session Awareness Format Changes" on page 13-6 for details):

- The DLUR will not be aware of the network address of the dependent LU's LU-LU session partner; this information will only appear in the corresponding independent LU SAW RUs. The new formats of some of the dependent LU SAW RUs (SESSST, SESSEND) will not include a session key, which is where the PLU and SLU network addresses are carried today.
- Control vectors included in some of the dependent LU SAW RUs (SESSST, RSP(ACTLU)) by the BF-NCP, such as VR-ER Mapping Data (X'1E') and XRF/Cryptography (X'68'), will not be included by the DLUR in its dependent LU SAW RUs, since it has no VR, ER, or XRF information.

10.2 Dependent LU Session Awareness Flows

Note: Readers should reference Figure C-2 on page C-3 for an illustration of the sample network configuration used in the flows described in this section of the document.

10.2.1 LU-LU Session Activation Flows

For LU-LU session activation, there will be the following different dependent LU session awareness flow scenarios in the DLUR/S environment:

1. the dependent LU's SSCP-LU and LU-LU sessions route through the same BF-NCP
 - the SESSST and BFSESSST are received by the DLUS's SSCP
2. the dependent LU's SSCP-LU and LU-LU sessions route through different BF-NCPs owned by the same SSCP (the DLUS's SSCP)
 - the SESSST and BFSESSST are received by the DLUS's SSCP
3. the dependent LU's SSCP-LU and LU-LU sessions route through different BF-NCPs owned by different SSCPs
 - only the SESSST is received by the DLUS's SSCP; the BFSESSST is received by a different SSCP
4. the dependent LU's LU-LU session does not route through a BF-NCP
 - no BFSESSST is built; only the SESSST is received by the DLUS's SSCP

The DLUR will build a new format of SESSST, Format 3; it differs from the current Format 1 in that:

- it will not carry several control vectors found in Format 1
- it will include the BIND image control vector

(see 13.3, "Dependent LU Session Awareness Format Changes" on page 13-6 for the new SESSST format).

10.2.1.1 SSCP-LU, LU-LU Sessions Routed Through Same BF-NCP

In this example, SSCP-PU and SSCP-LU sessions have been established from SSCP1 to PU2 and LUa, respectively, over the CP-SVR pipe between ENa and VTAM1. This pipe routes through the BF in NCP10, as will the LU-LU session between LUa and LUb. NCP10 is owned by SSCP1, which will receive all of the session awareness RUs concerning LUa.

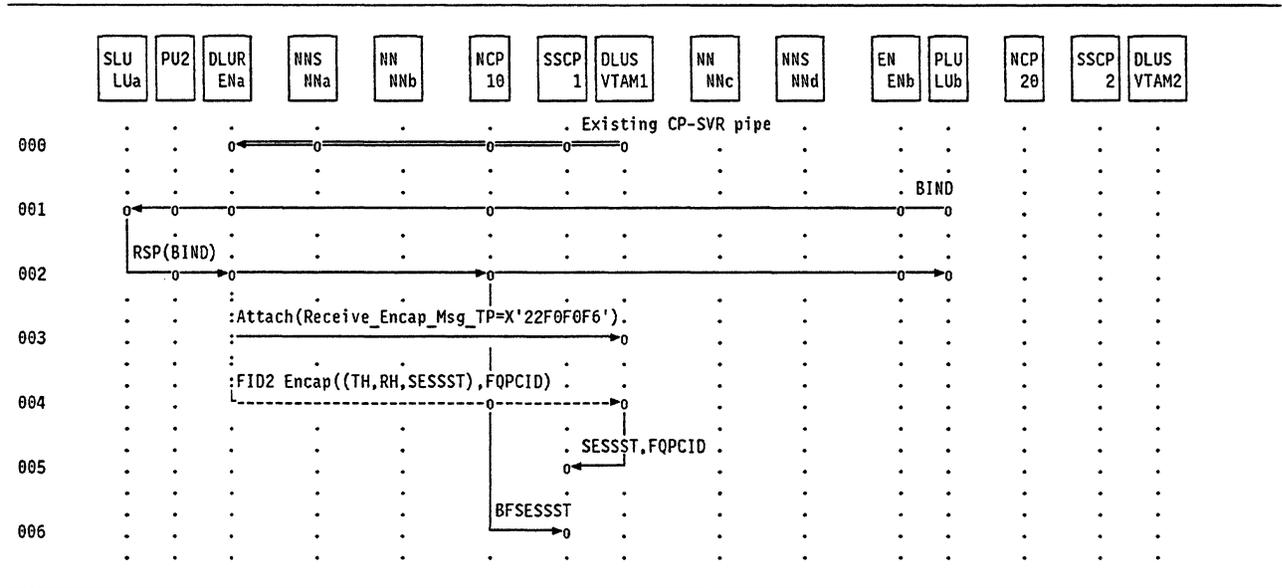


Figure 10-1. LU-LU session activation - SSCP-LU, LU-LU sessions routed through same BF-NCP

1. BIND is sent by LUb to LUa, routing through ENb, NNd, NNc, NCP10, NNb, NNa, ENa, and PU2.
2. RSP(BIND) is sent by LUa to LUb over the same route.
3. Upon receipt of RSP(BIND), the DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
4. The DLUR in ENa creates a Format 3 SESSST RU and encapsulates it, sending it on the CP-SVR pipe to the DLUS in VTAM1.
5. DLUS de-encapsulates the SESSST and forwards it to SSCP1.
6. Upon receipt of RSP(BIND), NCP10 creates a BFSESSST and sends it to SSCP1. SSCP1 correlates the BFSESSST with the SESSST by the FQPCID contained in each RU.

10.2.1.2 SSCP-LU, LU-LU Sessions Routed Through Different BF-NCPs (Same SSCP)

In this example, SSCP-PU and SSCP-LU sessions have been established from SSCP1 to PU2 and LUa, respectively, over the CP-SVR pipe between ENa and VTAM1. This pipe routes through the BF in NCP10. The LU-LU session between LUa and LUb will route, however, through NCP20. Both NCPs are owned by SSCP1, so all session awareness RUs concerning LUa will be sent to SSCP1.

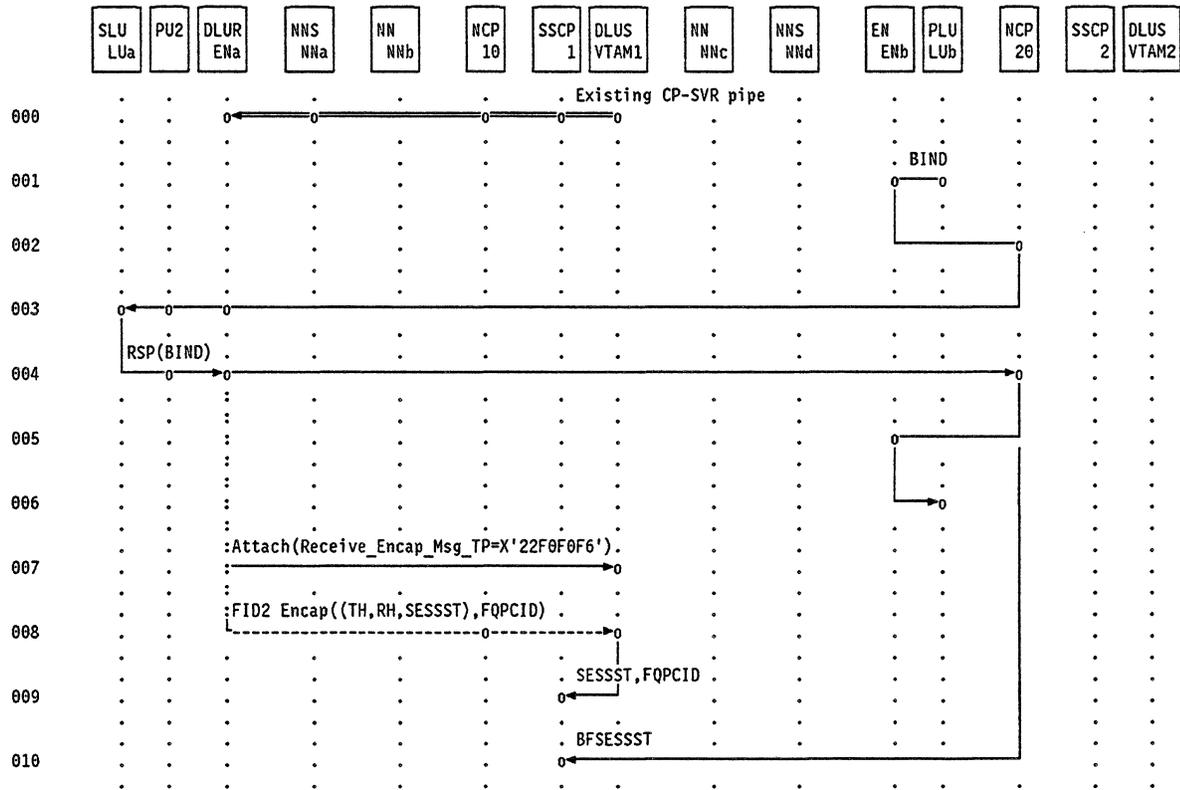


Figure 10-2. LU-LU session activation - SSCP-LU, LU-LU sessions routed through different BF-NCPs (same SSCP)

1. BIND is sent by LUb to LUa, routing through ENb, NNd, NNc, NCP20, NNb, NNa, ENa, and PU2.
- 2.
- 3.
4. RSP(BIND) is sent by LUa to LUb over the same route.
- 5.
- 6.
7. Upon receipt of RSP(BIND), the DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
8. The DLUR in ENa creates a Format 3 SESSST RU and encapsulates it, sending it on the CP-SVR pipe to the DLUS in VTAM1.
9. DLUS de-encapsulates the SESSST and forwards it to SSCP1.
10. Upon receipt of RSP(BIND), NCP20 creates a BFSESSST and sends it to SSCP1. SSCP1 correlates the BFSESSST with the SESSST by the FQPCID contained in each RU.

10.2.1.3 SSCP-LU, LU-LU Sessions Routed Through Different BF-NCPs (Different SSCPs)

In this example, SSCP-PU and SSCP-LU sessions have been established from SSCP1 to PU2 and LUa, respectively, over the CP-SVR pipe between ENa and VTAM1. This pipe routes through the BF in NCP10. The LU-LU session between LUa and LUb will route, however, through NCP20. NCP10 is owned by SSCP1, and NCP20 is owned by SSCP2. Therefore, the dependent LU session awareness RUs concerning LUa will be sent to SSCP1, while LUa's independent LU session awareness RUs will be sent to SSCP2.

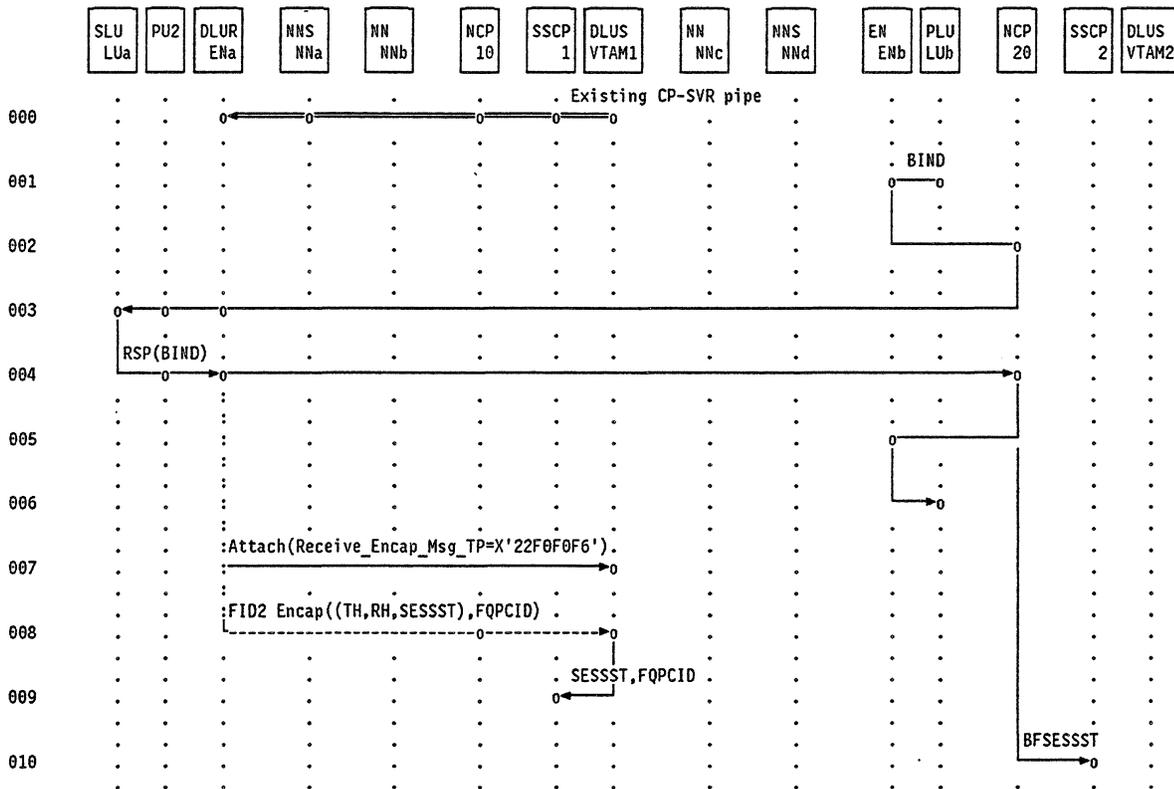


Figure 10-3. LU-LU session activation - SSCP-LU, LU-LU sessions routed through different BF-NCPs (different SSCPs)

1. BIND is sent by LUb to LUa, routing through ENb, NNd, NNc, NCP20, NNb, NNa, ENa, and PU2.
- 2.
- 3.
4. RSP(BIND) is sent by LUa to LUb over the same route.
- 5.
- 6.
7. Upon receipt of RSP(BIND), the DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
8. The DLUR in ENa creates a Format 3 SESSST RU and encapsulates it, sending it on the CP-SVR pipe to the DLUS in VTAM1.
9. DLUS de-encapsulates the SESSST and forwards it to SSCP1.
10. Upon receipt of RSP(BIND), NCP20 creates a BFSESSST and sends it to SSCP2.

10.2.1.4 LU-LU Session Not Routed Through A BF-NCP

In this example, SSCP-PU and SSCP-LU sessions have been established from SSCP1 to PU2 and LUa, respectively, over the CP-SVR pipe between ENa and VTAM1. This pipe routes through the BF in NCP10. The LU-LU session between LUa and LUb will not route, however, through any NCP. Therefore, SSCP1 will only receive dependent LU session awareness RUs concerning LUa.

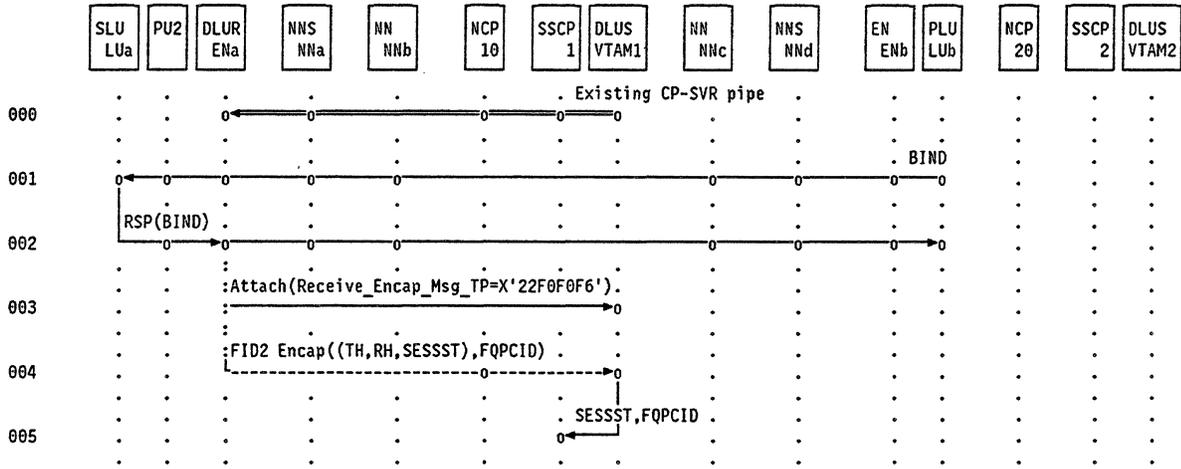


Figure 10-4. LU-LU session activation - LU-LU session not routed through a BF-NCP

1. BIND is sent by LUb to LUa, routing through ENb, NNd, NNc, NNb, NNa, ENa, and PU2.
2. RSP(BIND) is sent by LUa to LUb over the same route.
3. Upon receipt of RSP(BIND), the DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
4. The DLUR in ENa creates a Format 3 SESSST RU and encapsulates it, sending it on the CP-SVR pipe to the DLUS in VTAM1.
5. DLUS de-encapsulates the SESSST and forwards it to SSCP1.

10.2.2 SSCP Takeover & Giveback Flows

It is assumed in this section that the CP-SVR pipe is terminated due to the loss of the DLUS or of the route to the DLUS, and that the SSCP takeover includes the reestablishment of the CP-SVR pipe to the same or another DLUS. So the BF-NCP through which the CP-SVR pipe, which will include the encapsulated SSCP-LU session, is routed must be owned by the takeover DLUS's SSCP.

If the dependent LU's LU-LU session is routed through a different BF-NCP than the SSCP-LU session, and the LU-LU session's BF-NCP loses its SSCP or the route to its SSCP, the takeover SSCP of that BF-NCP does not have to be a DLUS's SSCP.

For SSCP takeover, there will be the following different dependent LU session awareness flow scenarios in the DLUR/S environment:

1. the dependent LU's SSCP-LU and LU-LU sessions route through the same BF-NCP
 - the RSP(ACTLU) and BFSESSINFO are received by the same takeover DLUS's SSCP
2. the dependent LU's SSCP-LU and LU-LU sessions route through different BF-NCPs owned by the same SSCP
 - if both BF-NCPs are acquired by the same takeover SSCP, the RSP(ACTLU) and BFSESSINFO are received by the same takeover SSCP (the takeover DLUS's SSCP)
 - if the BF-NCPs are acquired by different takeover SSCPs, the RSP(ACTLU) and BFSESSINFO are received by different takeover SSCPs
3. the dependent LU's SSCP-LU and LU-LU sessions route through different BF-NCPs owned by different SSCPs
 - if the BF-NCP routing the SSCP-LU session is acquired by a takeover SSCP, the RSP(ACTLU) is received by the takeover SSCP
 - - the takeover SSCP must be a DLUS's SSCP
 - if the BF-NCP routing the LU-LU session is acquired by a takeover SSCP, the BFSESSINFO is received by the takeover SSCP
 - the takeover SSCP may or may not be the SSCP owning the other BF-NCP; the takeover SSCP need not be a DLUS's SSCP
4. the dependent LU's LU-LU session does not route through a BF-NCP
 - no BFSESSINFO is built; only the RSP(ACTLU) is received by the takeover DLUS's SSCP

The DLUR will modify the RSP(ACTLU) differently than the BF-NCP. Several control vectors will not be included, and the Session Information (X'2A') control vector will be different (see 13.3, "Dependent LU Session Awareness Format Changes" on page 13-6 for the new formats).

For SSCP giveback, the session awareness flows are similar to SSCP takeover. The main difference is in the search for a new DLUS, i.e., try activating a CP-SVR pipe with a DLUS other than the one whose SSCP issued the DACTPU(giveback).

All of the flows in this section assume ANS=CONT is defined for the DLUR-attached PU; if instead ANS=STOP is defined, any active LU-LU sessions are deactivated along with the SSCP-PU and SSCP-LU sessions, and no session awareness is sent during takeover or giveback.

10.2.2.1 SSCP-LU, LU-LU Sessions Routed Through Same BF-NCP

In this example, SSCP-PU and SSCP-LU sessions have been established from SSCP1 to PU2 and LUa, respectively, over the CP-SVR pipe between ENa and VTAM1. This pipe routes through the BF in NCP10, as does the LU-LU session between LUa and LUb. NCP10 is owned by SSCP1.

An outage occurs in the SSCP1/VTAM1 host, and after the host is reactivated, a new pipe is established between ENa and VTAM1 (there was no backup host designated to take over for SSCP1). SSCP1 receives all session awareness RUs concerning the active LU-LU session.

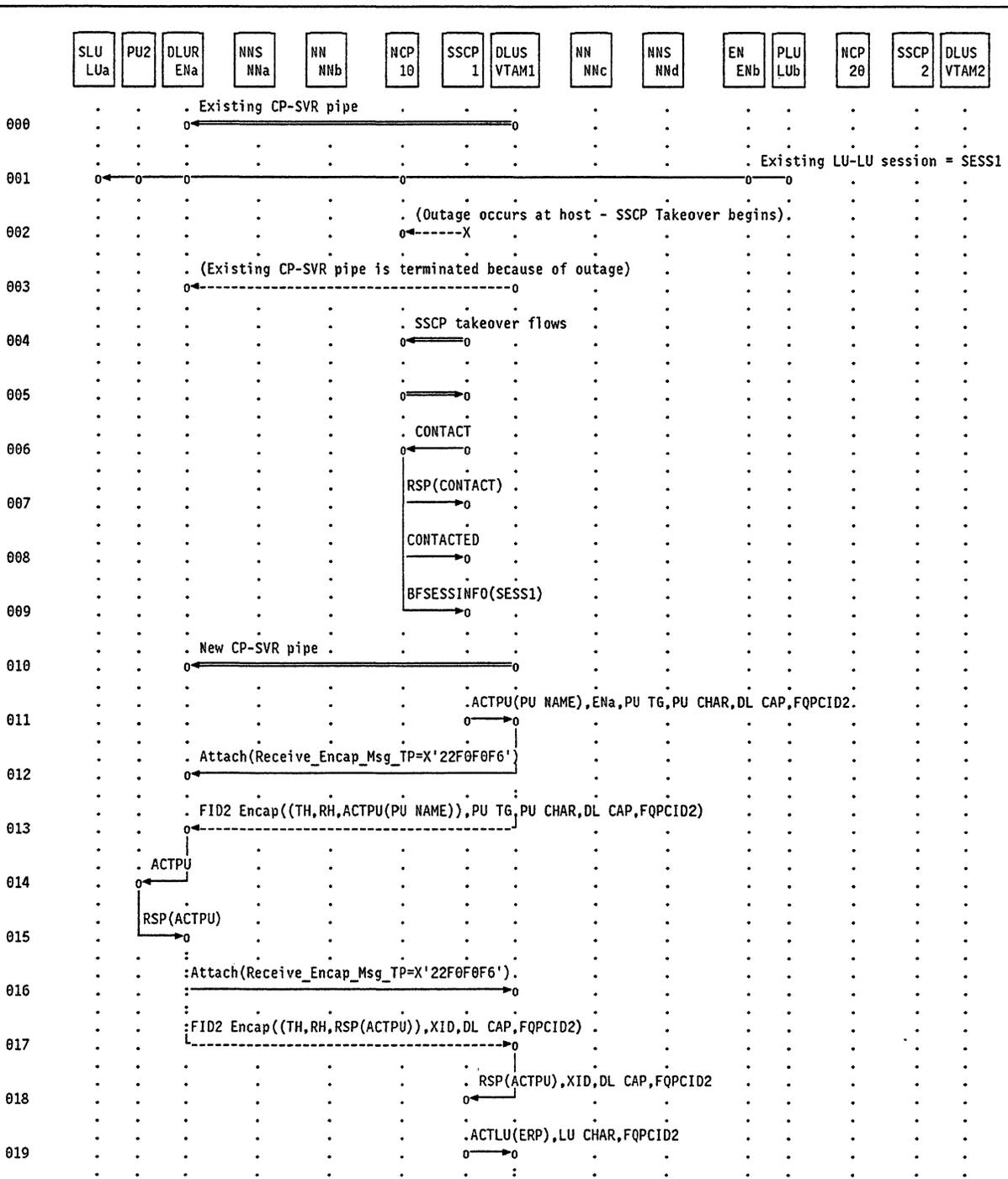
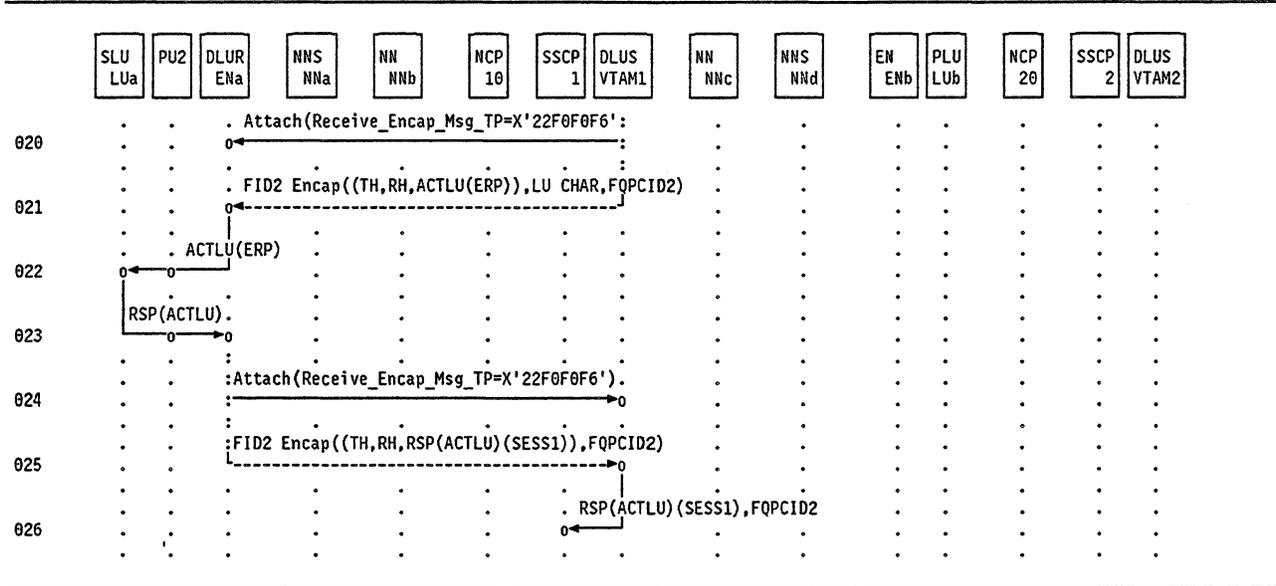


Figure 10-5. SSCP takeover - SSCP-LU, LU-LU sessions routed through same BF-NCP



1. The current LU-LU session between LUa and LUb routes through PU2, ENa, NNa, NNb, NCP10, NNc, NNd, and ENb.
2. The SSCP-NCP session between SSCP1 and NCP10 ends due to a host outage.
3. SSCP1 reacquires ownership of NCP10 and its attached resources.
- 4.
- 5.
- 6.
- 7.
8. When SSCP1 has acquired ownership of the link station through which the LU-LU session is routed, NCP10 builds and sends to SSCP1 a BFSESSINFO RU. This RU indicates that a session exists between LUa (which it believes to be an independent LU) and LUb.
9. The existing CP-SVR pipe is taken down due to the loss of VTAM1.
10. Once VTAM1 is reactivated, a new CP-SVR pipe is established.
11. SSCP1 issues ACTPU to PU2 to restart its SSCP-PU session.
12. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
13. The DLUS VTAM1 encapsulates the ACTPU and forwards it to the DLUR ENa.
14. ENa de-encapsulates the ACTPU and forwards it to PU2.
15. PU2 sends a RSP(ACTPU) to SSCP1 through ENa.
16. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
17. ENa encapsulates the RSP(ACTPU) and forwards it to VTAM1.
18. VTAM1 de-encapsulates the RSP(ACTPU) and passes it on to SSCP1.
19. SSCP1 issues ACTLU to LUa to restart its SSCP-LU session.
20. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
21. VTAM1 encapsulates the ACTLU and forwards it to ENa.
22. ENa de-encapsulates the ACTLU and forwards it to PU2 and on to LUa.
23. LUa sends a RSP(ACTLU) to SSCP1 through PU2, which passes it to ENa.
24. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
25. ENa builds a Session Information (X'2A') control vector, describing the LUa-LUb session, and appends the control vector to the RSP(ACTLU). ENa then encapsulates the RSP(ACTLU) and forwards it to VTAM1.
26. VTAM1 de-encapsulates the RSP(ACTLU) and passes it on to SSCP1. SSCP1 correlates the RSP(ACTLU) with the BFSESSINFO by the FQPCID contained in each RU.

10.2.2.2 SSCP-LU, LU-LU Sessions Routed Through Same BF-NCP

(Takeover & Giveback)

In this example, SSCP-PU and SSCP-LU sessions have been established from SSCP1 to PU2 and LUa, respectively, over the CP-SVR pipe between ENa and VTAM1. This pipe routes through the BF in NCP10, as does the LU-LU session between LUa and LUb. NCP10 is owned by SSCP1.

An outage occurs in the SSCP1/VTAM1 host, and after the host is reactivated, a new pipe is established between ENa and VTAM2 (there was no backup host designated to take over for SSCP1). SSCP2 receives all session awareness RUs concerning the active LU-LU session.

Later SSCP1 gets control back from SSCP2 and receives session awareness RUs about the LU-LU session.

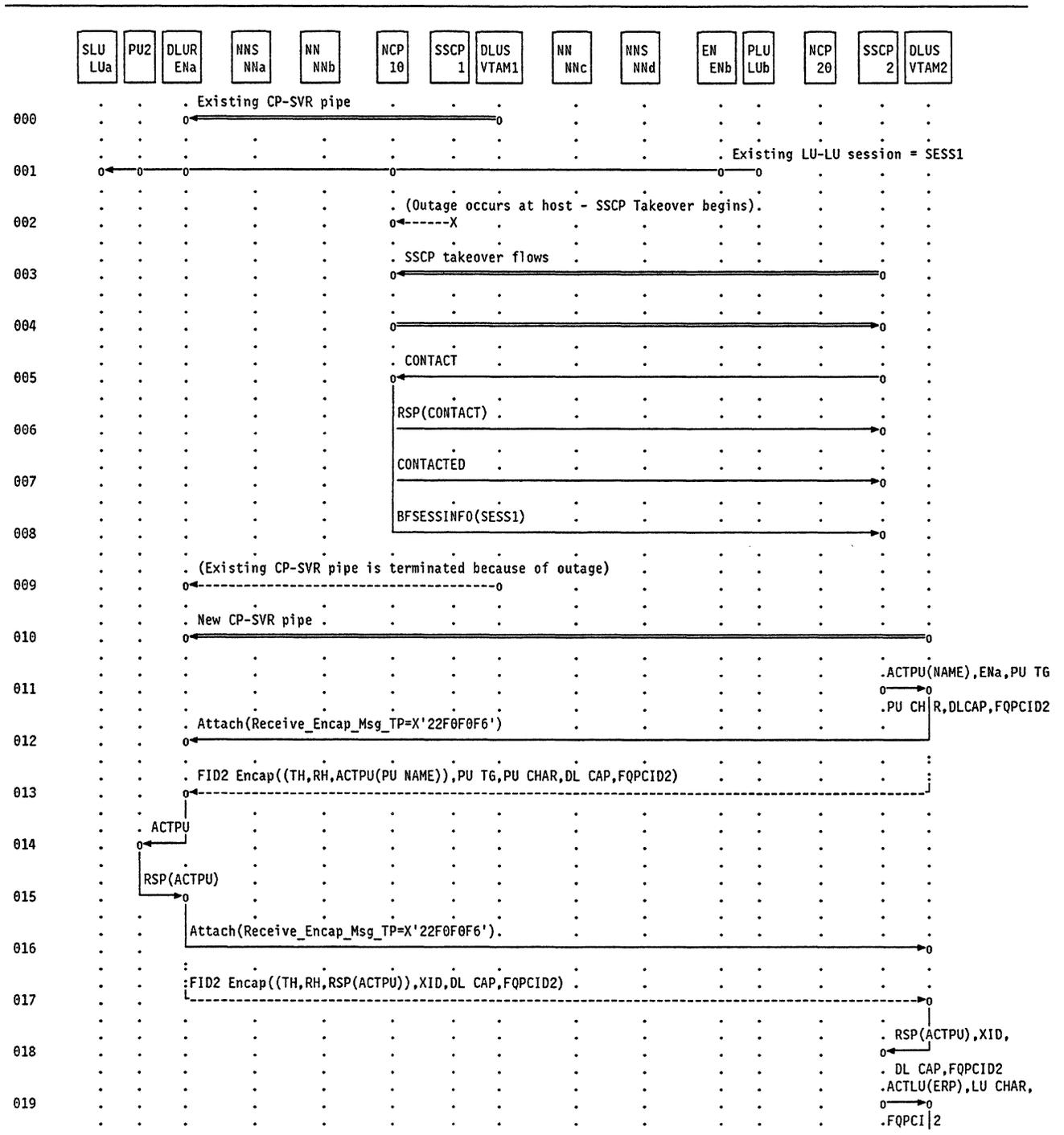
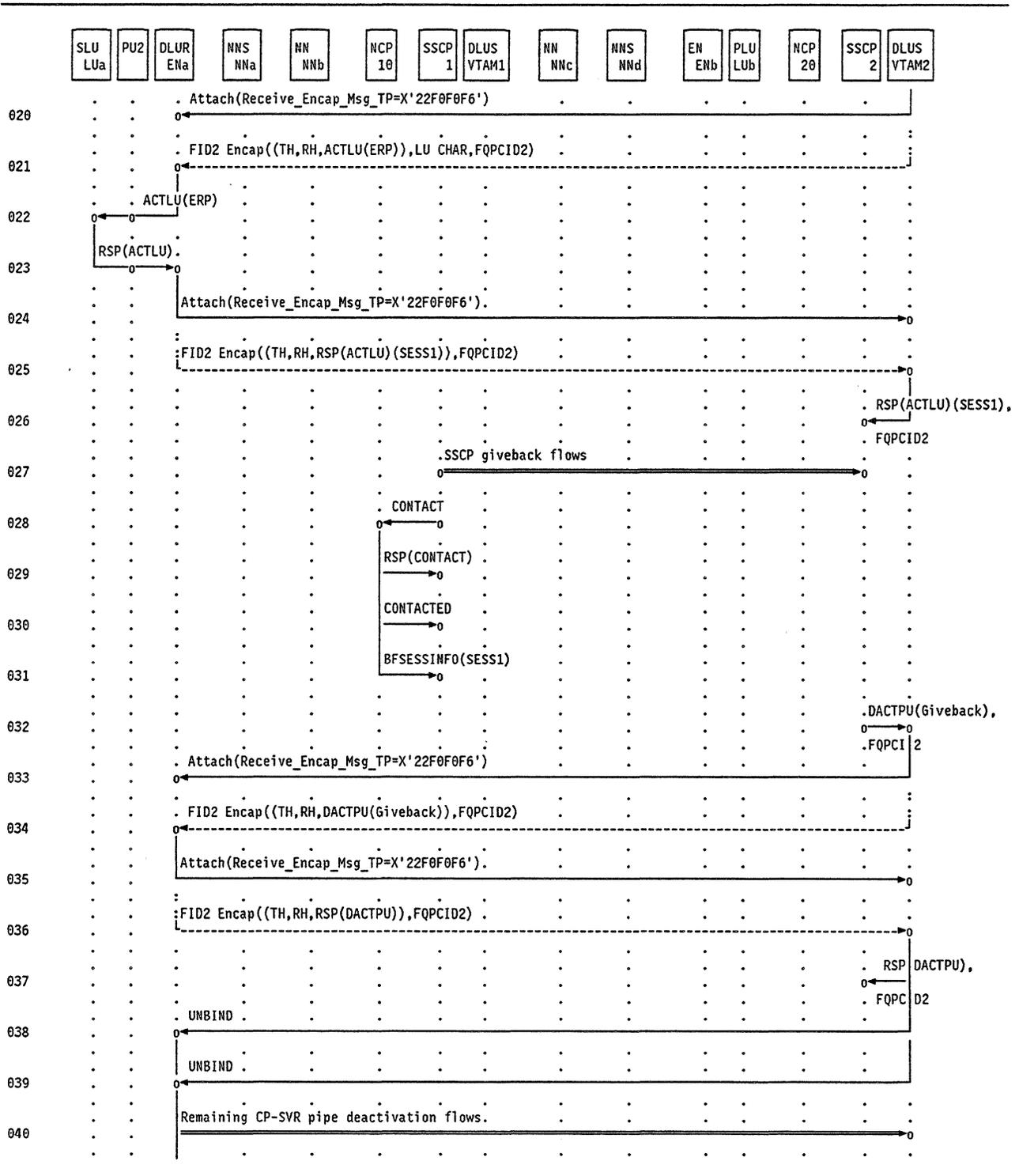
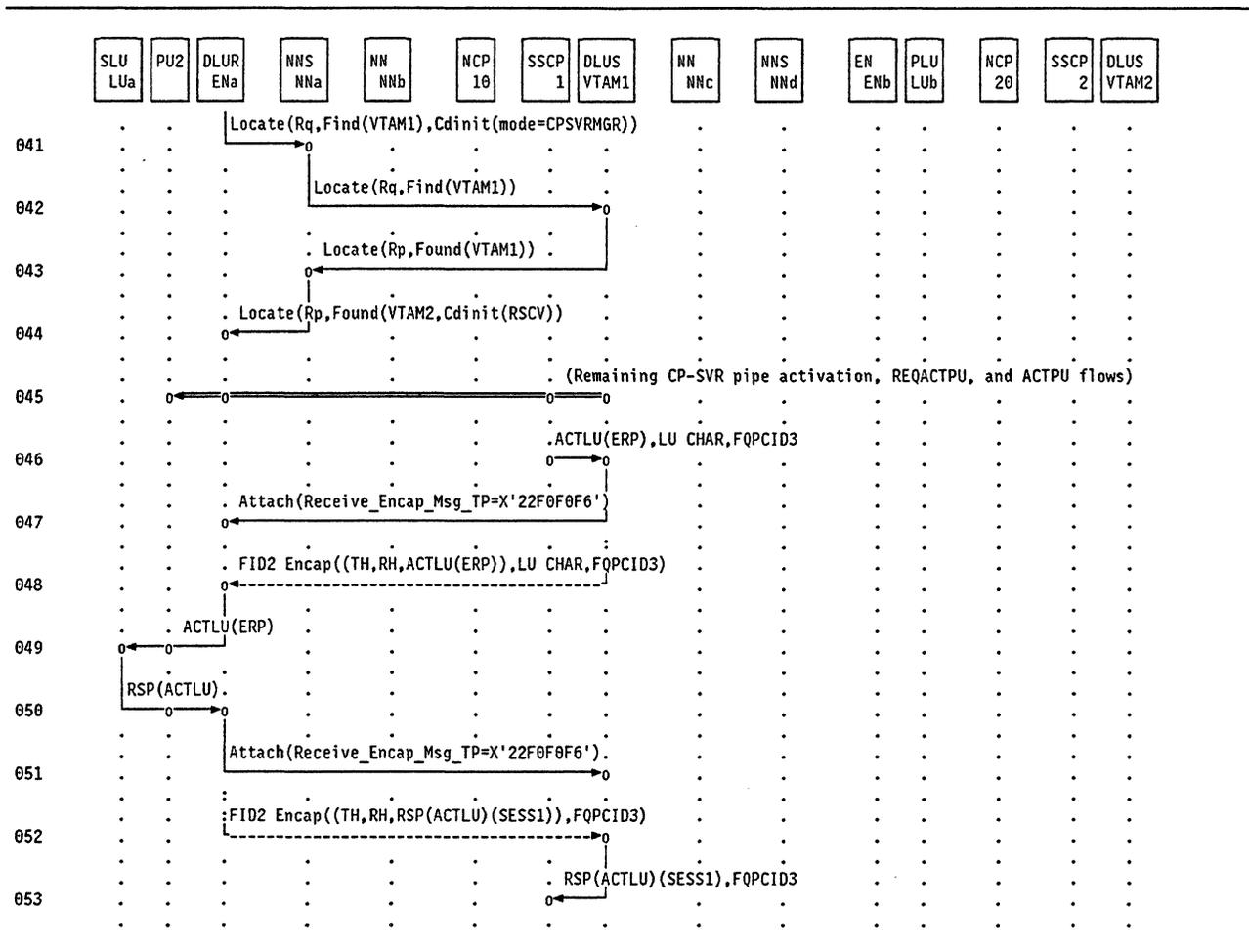


Figure 10-6. SSCP takeover & giveback - SSCP-LU, LU-LU sessions routed through same BF-NCP





1. The current LU-LU session between LUa and LUb routes through PU2, ENa, NNa, NNb, NCP10, NNc, NNd, and ENb.
2. The SSCP-NCP session between SSCP1 and NCP10 ends due to a host outage.
3. SSCP2 acquires ownership of NCP10 and its attached resources.
- 4.
- 5.
- 6.
- 7.
8. When SSCP2 has acquired ownership of the link station through which the LU-LU session is routed, NCP10 builds and sends to SSCP2 a BFSESSINFO RU. This RU indicates that a session exists between LUa (which it believes to be an independent LU) and LUb.
9. The existing CP-SVR pipe is taken down due to the loss of VTAM1.
10. A new CP-SVR pipe is established between ENa and VTAM2.
11. SSCP2 issues ACTPU to PU2 to restart its SSCP-PU session.
12. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
13. The DLUS VTAM2 encapsulates the ACTPU and forwards it to the DLUR ENa.
14. ENa de-encapsulates the ACTPU and forwards it to PU2.
15. PU2 sends a RSP(ACTPU) to SSCP2 through ENa.
16. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
17. ENa encapsulates the RSP(ACTPU) and forwards it to VTAM2.
18. VTAM2 de-encapsulates the RSP(ACTPU) and passes it on to SSCP2.
19. SSCP2 issues ACTLU to LUa to restart its SSCP-LU session.
20. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
21. VTAM2 encapsulates the ACTLU and forwards it to ENa.
22. ENa de-encapsulates the ACTLU and forwards it to PU2 and on to LUa.
23. LUa sends a RSP(ACTLU) to SSCP2 through PU2, which passes it to ENa.
24. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
25. ENa builds a Session Information (X'2A') control vector, describing the LUa-LUb session, and appends the control vector to the RSP(ACTLU). ENa then encapsulates the RSP(ACTLU) and forwards it to VTAM2.
26. VTAM2 de-encapsulates the RSP(ACTLU) and passes it on to SSCP2. SSCP2 correlates the RSP(ACTLU) with the BFSESSINFO by the FQPCID contained in each RU.
27. SSCP1 signals to SSCP2 that it is ready to reacquire ownership of the resources it had owned at the time of the outage. SSCP1 acquires ownership of NCP10 and its attached resources.
- 28.
- 29.
- 30.

31. When SSCP1 has acquired ownership of the link station through which the LU-LU session is routed, NCP10 builds and sends to SSCP1 a BFSESSINFO RU. This RU indicates that a session exists between LUa (which it believes to be an independent LU) and LUb.
32. SSCP2 issues DACTPU(giveback) to PU2 to end its SSCP-PU session.
33. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
34. The DLUS VTAM2 encapsulates the DACTPU and forwards it to the DLUR ENa.
35. ENa de-encapsulates the DACTPU and converts it to a RSP(DACTPU). The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
36. The DLUR re-encapsulates the RSP(DACTPU) and forwards it to VTAM2.
37. VTAM2 de-encapsulates the RSP(DACTPU) and passes it on to SSCP2.
38. VTAM2 determines that there are now no active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it sends an UNBIND to ENa on VTAM2's conwinner session to begin deactivation of the pipe.
39. VTAM2 also sends an UNBIND on its conloser session to ENa.
40. The DLUR node sends a RSP(UNBIND) to the DLUS for the conwinner session and another RSP(UNBIND) for the conloser session. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.
41. ENa determines that an attached LU has an active LU-LU session, and it restarts activation of the CP-SVR pipe to VTAM1 (it will not attempt activation of the CP-SVR pipe to VTAM2, since it received a DACTPU(giveback) from VTAM2's SSCP). ENa sends a locate for VTAM1.
42. The NNS(ENa) issues either a broadcast or directed search for the CP_LU of the DLUS = VTAM1.
43. The DLUS node responds to the search request with a Locate Found reply.
44. The NNS(ENa) uses the information in the Locate reply to calculate a route to the DLUS node and passes this back to the DLUR in the form of an RSCV.
45. The CP-SVR pipe activation completes successfully; ENa sends a REQACTPU RU to SSCP1, and SSCP1 sends an ACTPU to restart its SSCP-PU session with PU2.
46. SSCP1 issues ACTLU to LUa to restart its SSCP-LU session.
47. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
48. VTAM1 encapsulates the ACTLU and forwards it to ENa.
49. ENa de-encapsulates the ACTLU and forwards it to PU2 and on to LUa.
50. LUa sends a RSP(ACTLU) to SSCP1 through PU2, which passes it to ENa.
51. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
52. ENa builds a Session Information (X'2A') control vector, describing the LUa-LUb session, and appends the control vector to the RSP(ACTLU). ENa then encapsulates the RSP(ACTLU) and forwards it to VTAM1.
53. VTAM1 de-encapsulates the RSP(ACTLU) and passes it on to SSCP1. SSCP1 correlates the RSP(ACTLU) with the BFSESSINFO by the FQPCID contained in each RU.

10.2.2.3 SSCP-LU, LU-LU Sessions Routed Through Different BF-NCPs

(Same Takeover SSCP)

In this example, SSCP-PU and SSCP-LU sessions have been established from SSCP1 to PU2 and LUa, respectively, over the CP-SVR pipe between ENa and VTAM1. This pipe routes through the BF in NCP10, which is owned by SSCP1. The LU-LU session between LUa and LUb routes through the BF in NCP20, which is also owned by SSCP1.

An outage occurs in the SSCP1/VTAM1 host, and SSCP2 takes over ownership of the resources previously owned by SSCP1. This includes taking over ownership of both NCP10 and NCP20 as well as establishing a new pipe between ENa and VTAM2. SSCP2 receives all session awareness RUs concerning the active LU-LU session.

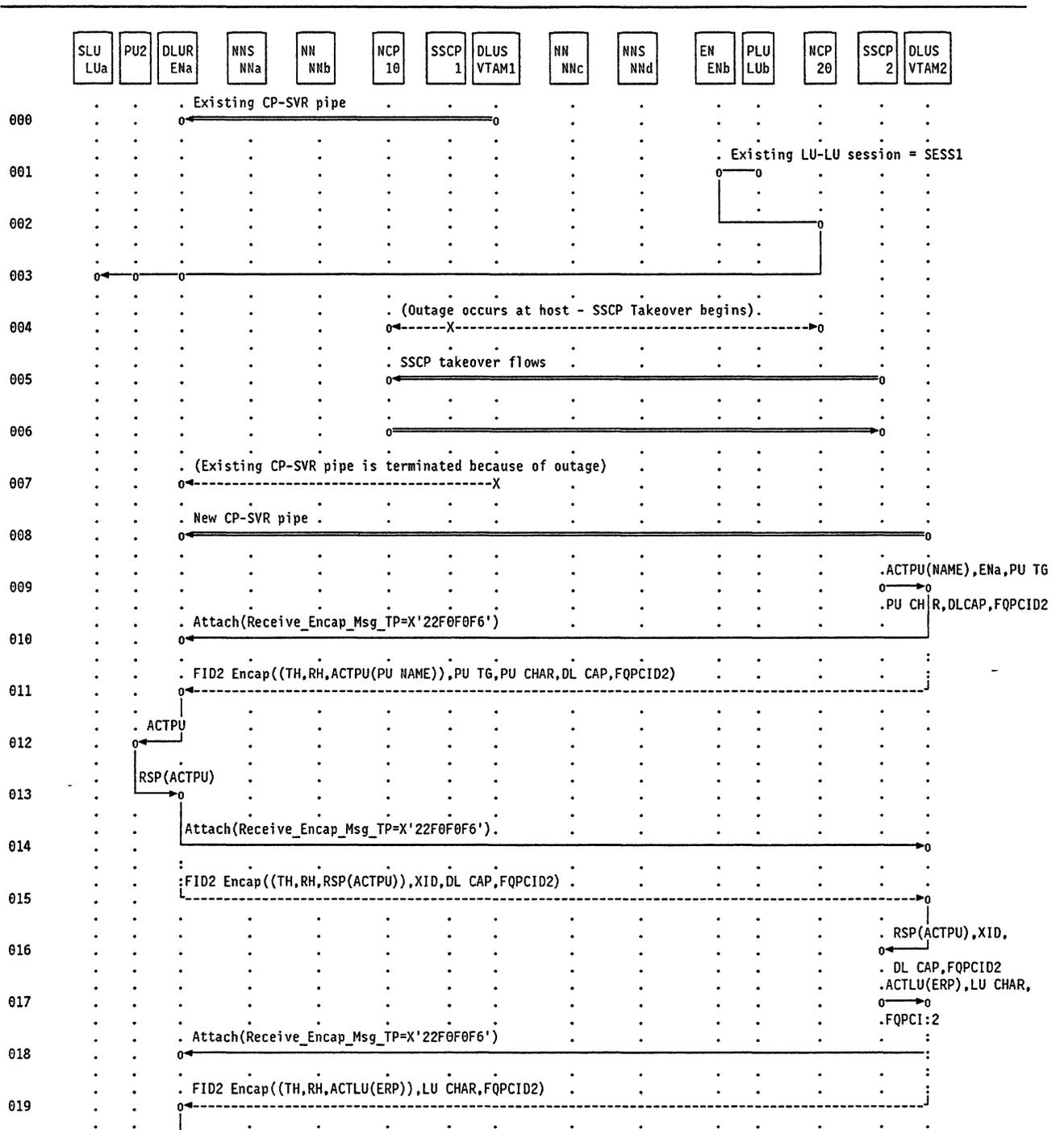
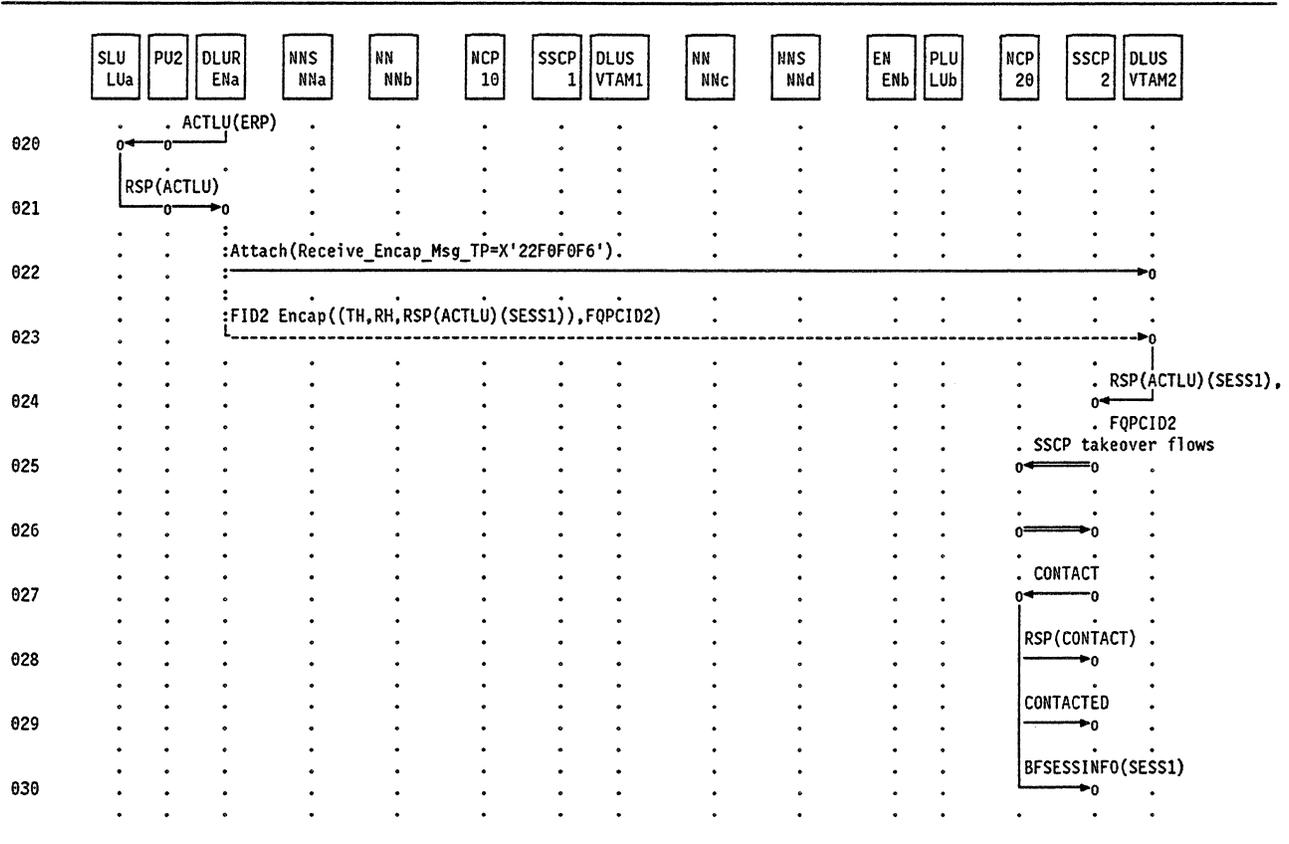


Figure 10-7. SSCP takeover - SSCP-LU, LU-LU sessions routed through different BF-NCPs / same takeover SSCP



1. The current LU-LU session between LUa and LUb routes through PU2, ENa, NNa, NNb, NCP20, NNc, NNd, and ENb.
- 2.
- 3.
4. The SSCP-NCP sessions between SSCP1 and NCP10 and between SSCP1 and NCP20 end due to a host outage.
5. SSCP2 acquires ownership of NCP10 and its attached resources previously owned by SSCP1.
- 6.
7. The existing CP-SVR pipe is taken down due to the loss of VTAM1.
8. A new CP-SVR pipe is established between ENa and VTAM2.
9. SSCP2 issues ACTPU to PU2 to start its SSCP-PU session.
10. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
11. The DLUS VTAM2 encapsulates the ACTPU and forwards it to the DLUR ENa.
12. ENa de-encapsulates the ACTPU and forwards it to PU2.
13. PU2 sends a RSP(ACTPU) to SSCP2 through ENa.
14. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
15. ENa encapsulates the RSP(ACTPU) and forwards it to VTAM2.
16. VTAM2 de-encapsulates the RSP(ACTPU) and passes it on to SSCP2.
17. SSCP2 issues ACTLU to LUa to start its SSCP-LU session.
18. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
19. VTAM2 encapsulates the ACTLU and forwards it to ENa.
20. ENa de-encapsulates the ACTLU and forwards it to PU2 and on to LUa.
21. LUa sends a RSP(ACTLU) to SSCP2 through PU2, which passes it to ENa.
22. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
23. ENa builds a Session Information (X'2A') control vector, describing the LUa-LUb session, and appends the control vector to the RSP(ACTLU). ENa then encapsulates the RSP(ACTLU) and forwards it to VTAM2.
24. VTAM2 de-encapsulates the RSP(ACTLU) and passes it on to SSCP2.
25. SSCP2 acquires ownership of NCP20 and its attached resources.
- 26.
- 27.
- 28.
- 29.
30. When SSCP2 has acquired ownership of the link station through which the LU-LU session is routed, NCP20 builds and sends to SSCP2 a BFSESSINFO RU. This RU indicates that a session exists between LUa (which it believes to be an independent LU) and LUb. SSCP2 correlates the BFSESSINFO with the RSP(ACTLU) by the FQPCID contained in each RU.

10.2.2.4 SSCP-LU, LU-LU Sessions Routed Through Different BF-NCPs (Different Takeover SSCPs)

In this example, SSCP-PU and SSCP-LU sessions have been established from SSCP1 to PU2 and LUa, respectively, over the CP-SVR pipe between ENa and VTAM1. This pipe routes through the BF in NCP10, which is owned by SSCP1. The LU-LU session between LUa and LUb routes through the BF in NCP20, which is also owned by SSCP1.

An outage occurs in the SSCP1/VTAM1 host, and SSCP2 and SSCP3 take over ownership of the resources previously owned by SSCP1. This includes SSCP2 taking over ownership of NCP10, SSCP3 taking over ownership of NCP20, and ENa and VTAM2 establishing a new CP-SVR pipe. The dependent LU session awareness RUs concerning the active LU-LU session will be sent to SSCP2, while the session's independent LU session awareness RUs will be sent to SSCP3.

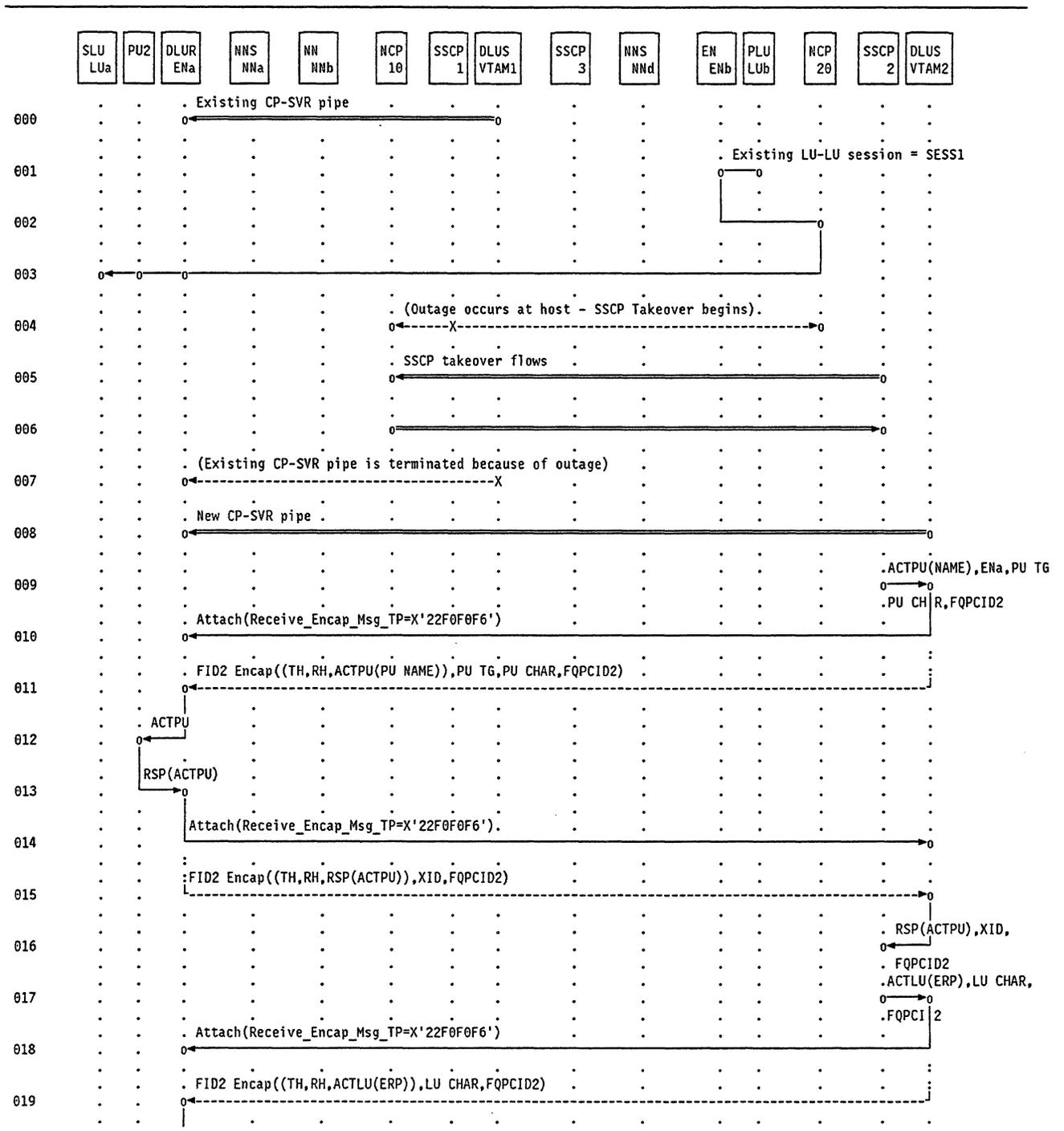
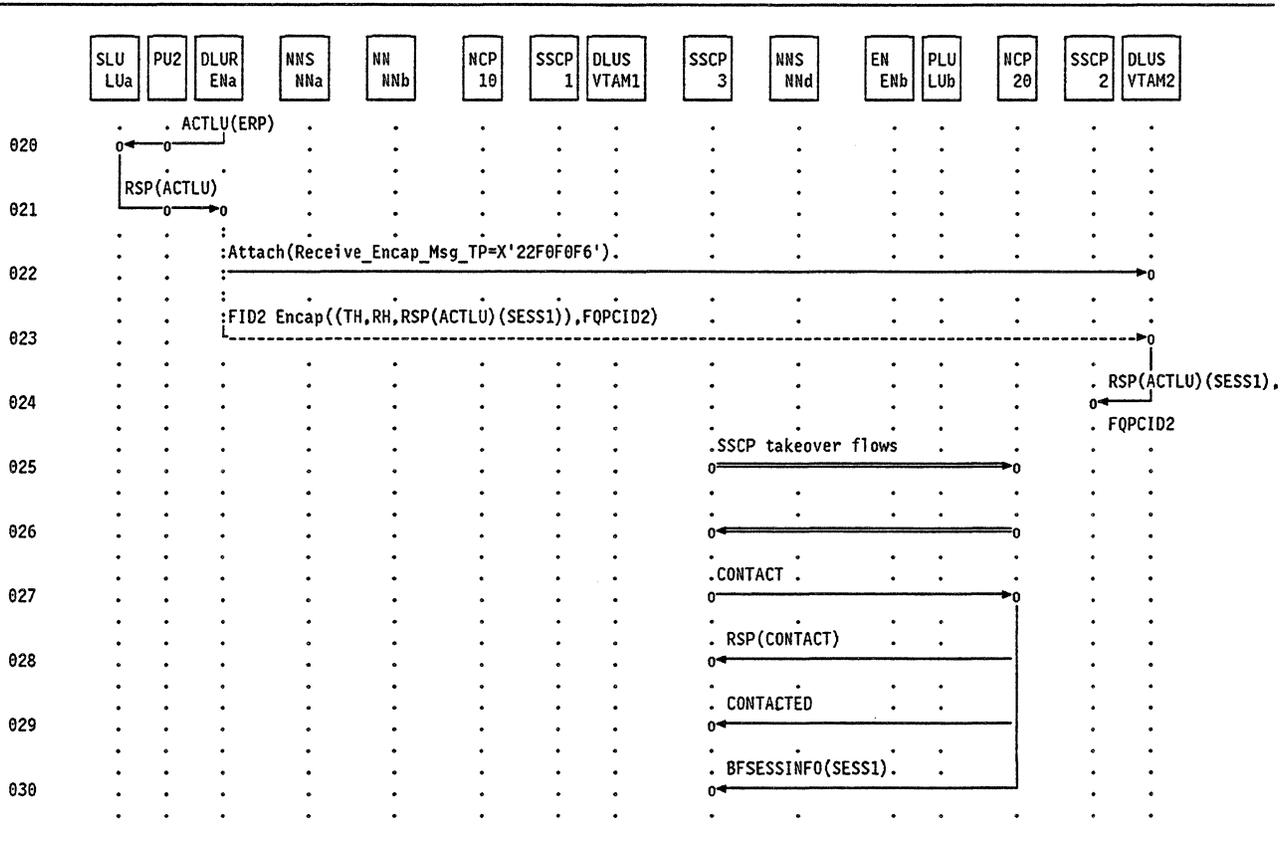


Figure 10-8. SSCP takeover - SSCP-LU, LU-LU sessions routed through different BF-NCPs / different takeover SSCPs



1. The current LU-LU session between LUa and LUb routes through PU2, ENa, NNa, NNb, NCP20, NNc, NNd, and ENb.
- 2.
- 3.
4. The SSCP-NCP sessions between SSCP1 and NCP10 and between SSCP1 and NCP20 end due to a host outage.
5. SSCP2 acquires ownership of NCP10 and its attached resources previously owned by SSCP1.
- 6.
7. The existing CP-SVR pipe is taken down due to the loss of VTAM1.
8. A new CP-SVR pipe is established between ENa and VTAM2.
9. SSCP2 issues ACTPU to PU2 to start its SSCP-PU session.
10. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
11. The DLUS VTAM2 encapsulates the ACTPU and forwards it to the DLUR ENa.
12. ENa de-encapsulates the ACTPU and forwards it to PU2.
13. PU2 sends a RSP(ACTPU) to SSCP2 through ENa.
14. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
15. ENa encapsulates the RSP(ACTPU) and forwards it to VTAM2.
16. VTAM2 de-encapsulates the RSP(ACTPU) and passes it on to SSCP2.
17. SSCP2 issues ACTLU to LUa to start its SSCP-LU session.
18. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
19. VTAM2 encapsulates the ACTLU and forwards it to ENa.
20. ENa de-encapsulates the ACTLU and forwards it to PU2 and on to LUa.
21. LUa sends a RSP(ACTLU) to SSCP2 through PU2, which passes it to ENa.
22. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
23. ENa builds a Session Information (X'2A') control vector, describing the LUa-LUb session, and appends the control vector to the RSP(ACTLU). ENa then encapsulates the RSP(ACTLU) and forwards it to VTAM2.
24. VTAM2 de-encapsulates the RSP(ACTLU) and passes it on to SSCP2.
25. SSCP3 acquires ownership of NCP20 and its attached resources.
- 26.
- 27.
- 28.
- 29.
30. When SSCP3 has acquired ownership of the link station through which the LU-LU session is routed, NCP20 builds and sends to SSCP3 a BFSESSINFO RU. This RU indicates that a session exists between LUa (which it believes to be an independent LU) and LUb.

10.2.3 Normal LU-LU Session Deactivation Flows

For LU-LU session deactivation, there will be the following different dependent LU session awareness flow scenarios in the DLUS environment:

1. the dependent LU's SSCP-LU and LU-LU sessions route through the same BF-NCP
 - the SESSEND and BFSESEND are received by the DLUS's SSCP
2. the dependent LU's SSCP-LU and LU-LU sessions route through different BF-NCPs owned by the same SSCP (the DLUS's SSCP)
 - the SESSEND and BFSESEND are received by the DLUS's SSCP
3. the dependent LU's SSCP-LU and LU-LU sessions route through different BF-NCPs owned by different SSCPs
 - only the SESSEND is received by the DLUS's SSCP; the BFSESEND is received by a different SSCP
4. the dependent LU's LU-LU session does not route through a BF-NCP
 - no BFSESEND is built; only the SESSEND is received by the DLUS's SSCP

The DLUR will build a new format of SESSEND, Format 3; it differs from the current Format 2 in that it will not carry all of the same control vectors (see 13.3, "Dependent LU Session Awareness Format Changes" on page 13-6 for the new SESSEND format).

10.2.3.1 SSCP-LU, LU-LU Sessions Routed Through Same BF-NCP

In this example, SSCP-PU and SSCP-LU sessions have been established from SSCP1 to PU2 and LUa, respectively, over the CP-SVR pipe between ENa and VTAM1. This pipe routes through the BF in NCP10, as does the LU-LU session between LUa and LUb. NCP10 is owned by SSCP1, which will receive all of the session awareness RUs concerning LUa.

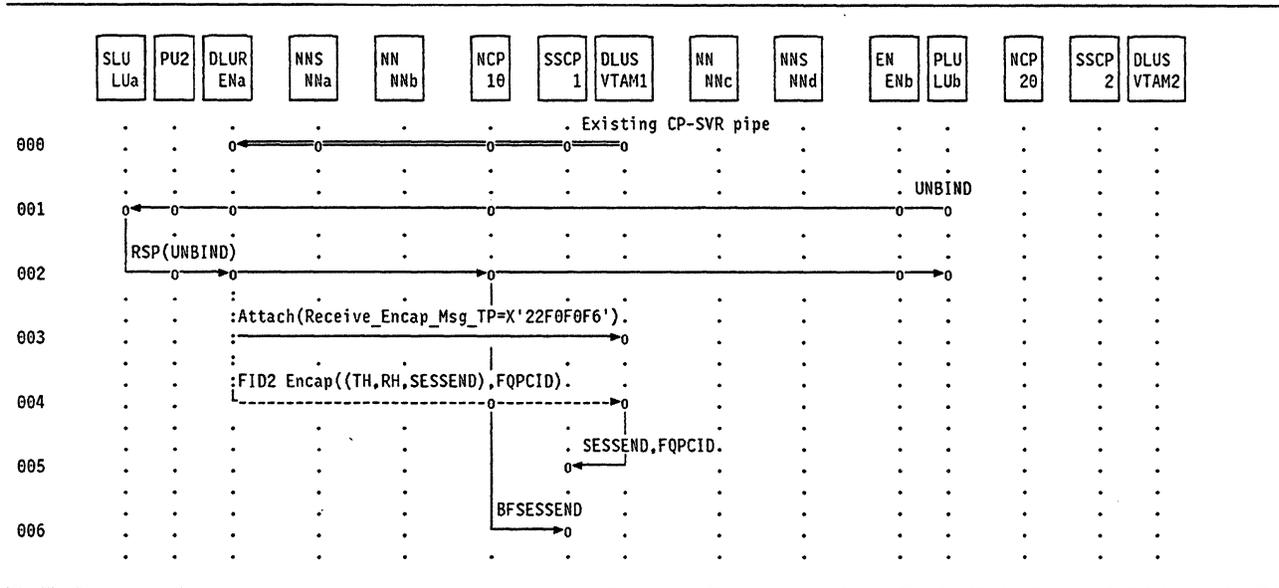


Figure 10-9. LU-LU session deactivation - SSCP-LU, LU-LU sessions routed through same BF-NCP

1. UNBIND is sent by LUb to LUa, routing through ENb, NNd, NNc, NCP10, NNb, NNa, ENa, and PU2.
2. RSP(UNBIND) is sent by LUa to LUb over the same route.
3. Upon receipt of RSP(UNBIND), the DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
4. The DLUR in ENa creates a Format 3 SESSEND RU and encapsulates it, sending it on the CP-SVR pipe to the DLUS in VTAM1.
5. DLUS de-encapsulates the SESSEND and forwards it to SSCP1.
6. Upon receipt of RSP(UNBIND), NCP10 creates a BFSESSEND and sends it to SSCP1. SSCP1 correlates the BFSESSEND with the SESSEND by the FQPCID contained in each RU.

10.2.3.2 SSCP-LU, LU-LU Sessions Routed Through Different BF-NCPs (Same SSCP)

In this example, SSCP-PU and SSCP-LU sessions have been established from SSCP1 to PU2 and LUa, respectively, over the CP-SVR pipe between ENa and VTAM1. This pipe routes through the BF in NCP10. The LU-LU session between LUa and LUb routes, however, through NCP20. Both NCPs are owned by SSCP1, so all session awareness RUs concerning LUa will be sent to SSCP1.

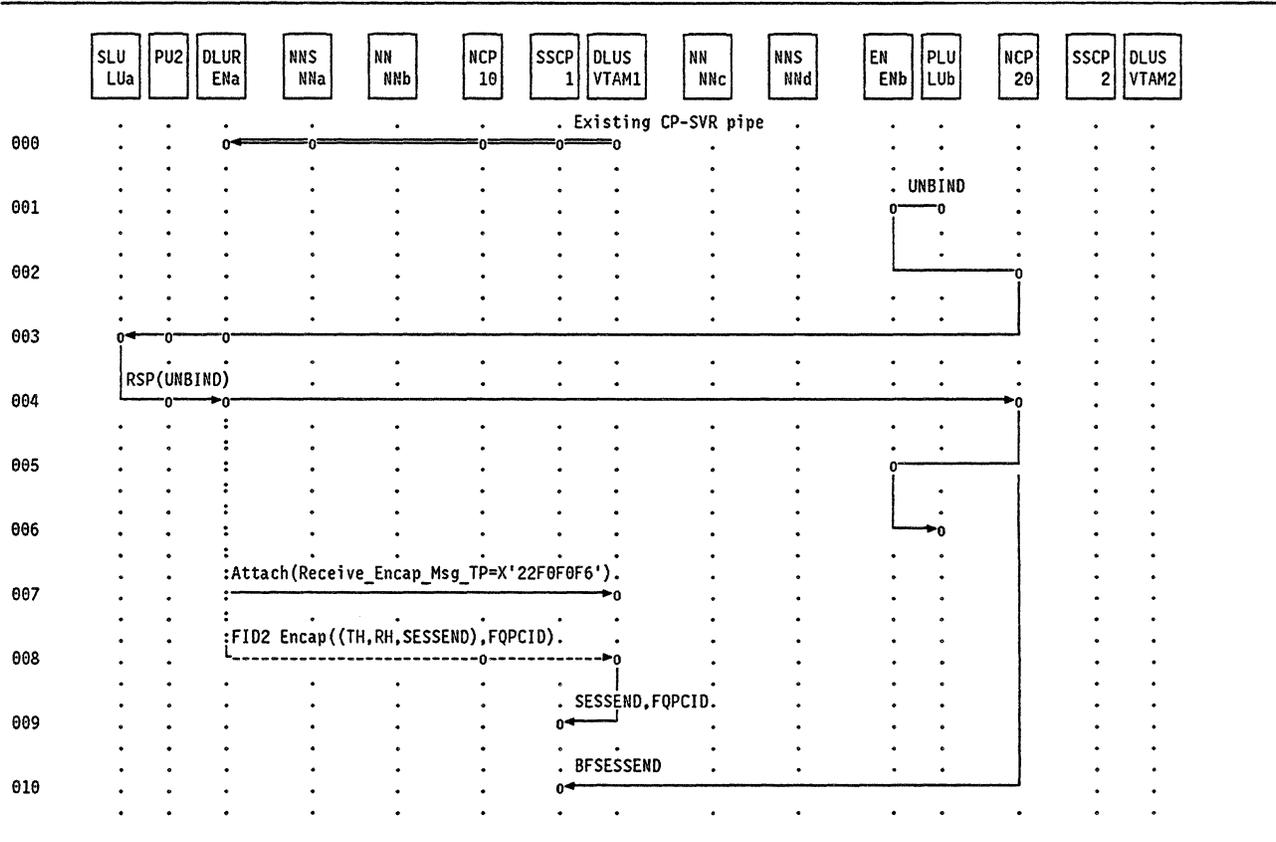


Figure 10-10. LU-LU session deactivation - SSCP-LU, LU-LU sessions routed through different BF-NCPs (same SSCP)

1. UNBIND is sent by LUb to LUa, routing through ENb, NNd, NNc, NCP20, NNb, NNa, ENa, and PU2.
- 2.
- 3.
4. RSP(UNBIND) is sent by LUa to LUb over the same route.
- 5.
- 6.
7. Upon receipt of RSP(UNBIND), the DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
8. The DLUR in ENa creates a Format 3 SESSEND RU and encapsulates it, sending it on the CP-SVR pipe to the DLUS in VTAM1.
9. DLUS de-encapsulates the SESSEND and forwards it to SSCP1.
10. Upon receipt of RSP(UNBIND), NCP20 creates a BFSESEND and sends it to SSCP1. SSCP1 correlates the BFSESEND with the SESSEND by the FQPCID contained in each RU.

10.2.3.3 SSCP-LU, LU-LU Sessions Routed Through Different BF-NCPs (Different SSCPs)

In this example, SSCP-PU and SSCP-LU sessions have been established from SSCP1 to PU2 and LUa, respectively, over the CP-SVR pipe between ENa and VTAM1. This pipe routes through the BF in NCP10. The LU-LU session between LUa and LUb routes, however, through NCP20. NCP10 is owned by SSCP1, and NCP20 is owned by SSCP2. Therefore, the dependent LU session awareness RUs concerning LUa will be sent to SSCP1, while LUa's independent LU session awareness RUs will be sent to SSCP2.

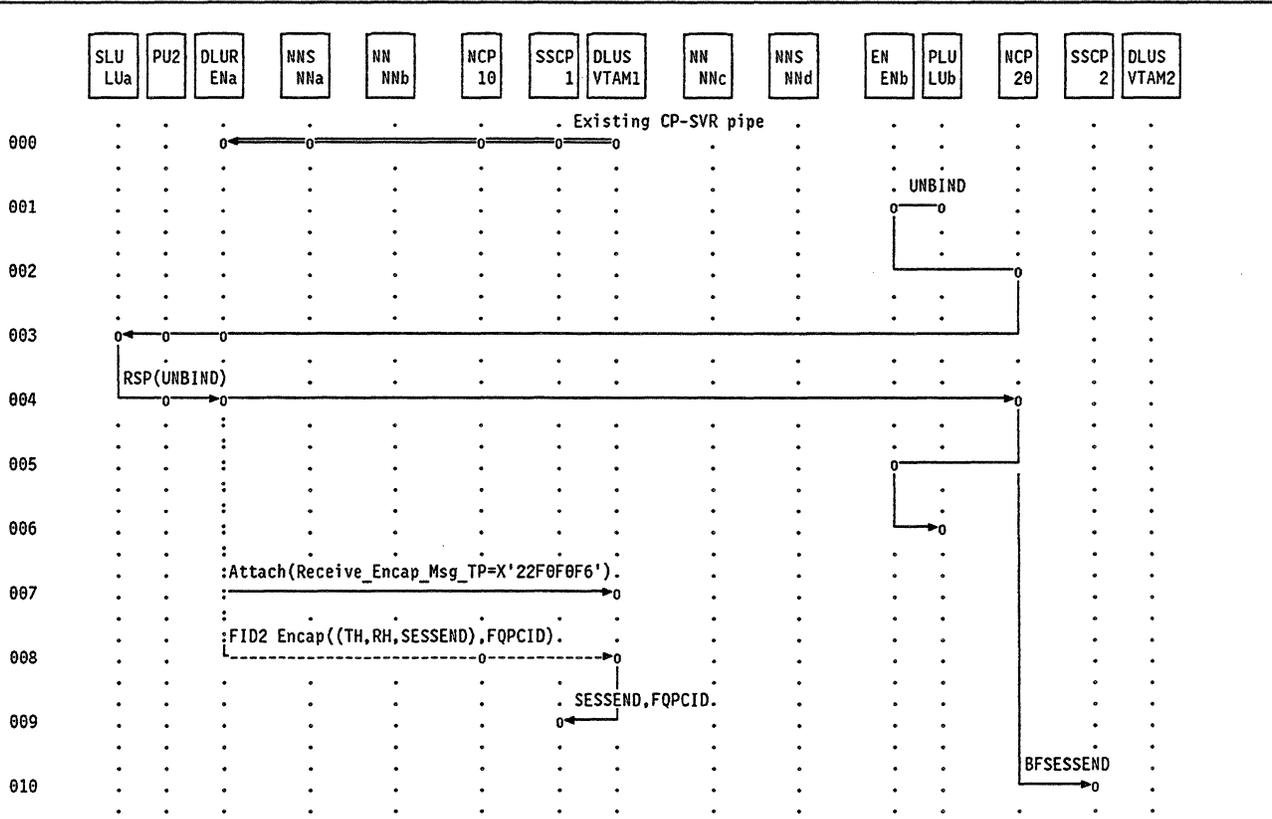


Figure 10-11. LU-LU session deactivation - SSCP-LU, LU-LU sessions routed through different BF-NCPs (different SSCPs)

1. BIND is sent by LUb to LUa, routing through ENb, NNd, NNc, NCP20, NNb, NNa, ENa, and PU2.
- 2.
- 3.
4. RSP(UNBIND) is sent by LUa to LUb over the same route.
- 5.
- 6.
7. Upon receipt of RSP(UNBIND), the DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
8. The DLUR in ENa creates a Format 3 SESSEND RU and encapsulates it, sending it on the CP-SVR to the DLUS in VTAM1.
9. DLUS de-encapsulates the SESSEND and forwards it to SSCP1.
10. Upon receipt of RSP(UNBIND), NCP20 creates a BFSESEND and sends it to SSCP2.

10.2.3.4 LU-LU Session Not Routed Through A BF-NCP

In this example, SSCP-PU and SSCP-LU sessions have been established from SSCP1 to PU2 and LUa, respectively, over the CP-SVR pipe between ENa and VTAM1. This pipe routes through the BF in NCP10. The LU-LU session between LUa and LUb does not route, however, through any NCP. Therefore, SSCP1 will only receive dependent LU session awareness RUs concerning LUa.

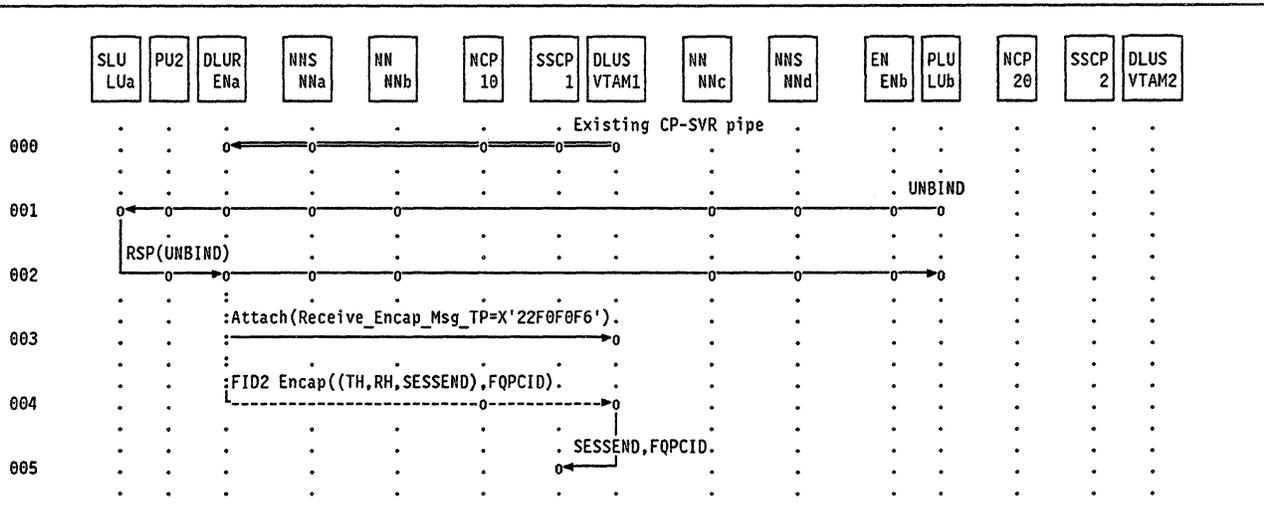


Figure 10-12. LU-LU session deactivation - LU-LU session not routed through a BF-NCP

1. UNBIND is sent by LUb to LUa, routing through ENb, NNd, NNc, NNb, NNa, ENa, and PU2.
2. RSP(UNBIND) is sent by LUa to LUb over the same route.
3. Upon receipt of RSP(UNBIND), the DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
4. The DLUR in ENa creates a Format 3 SESEND RU and encapsulates it, sending it on the CP-SVR pipe to the DLUS in VTAM1.
5. DLUS de-encapsulates the SESEND and forwards it to SSCP1.

10.2.4 Abnormal LU-LU Session Deactivation Flows

The same session deactivation flow scenarios apply to abnormal deactivation situations. The new Format 3 SESSEND will also be used in these cases.

10.2.4.1 Abnormal LU-LU Session Deactivation

In this example, SSCP-PU and SSCP-LU sessions have been established from SSCP1 to PU2 and LUa, respectively, over the CP-SVR pipe between ENa and VTAM1. This pipe routes through the BF in NCP10, as does the LU-LU session between LUa and LUb. NCP10 is owned by SSCP1, which will receive all of the session awareness RUs concerning LUa.

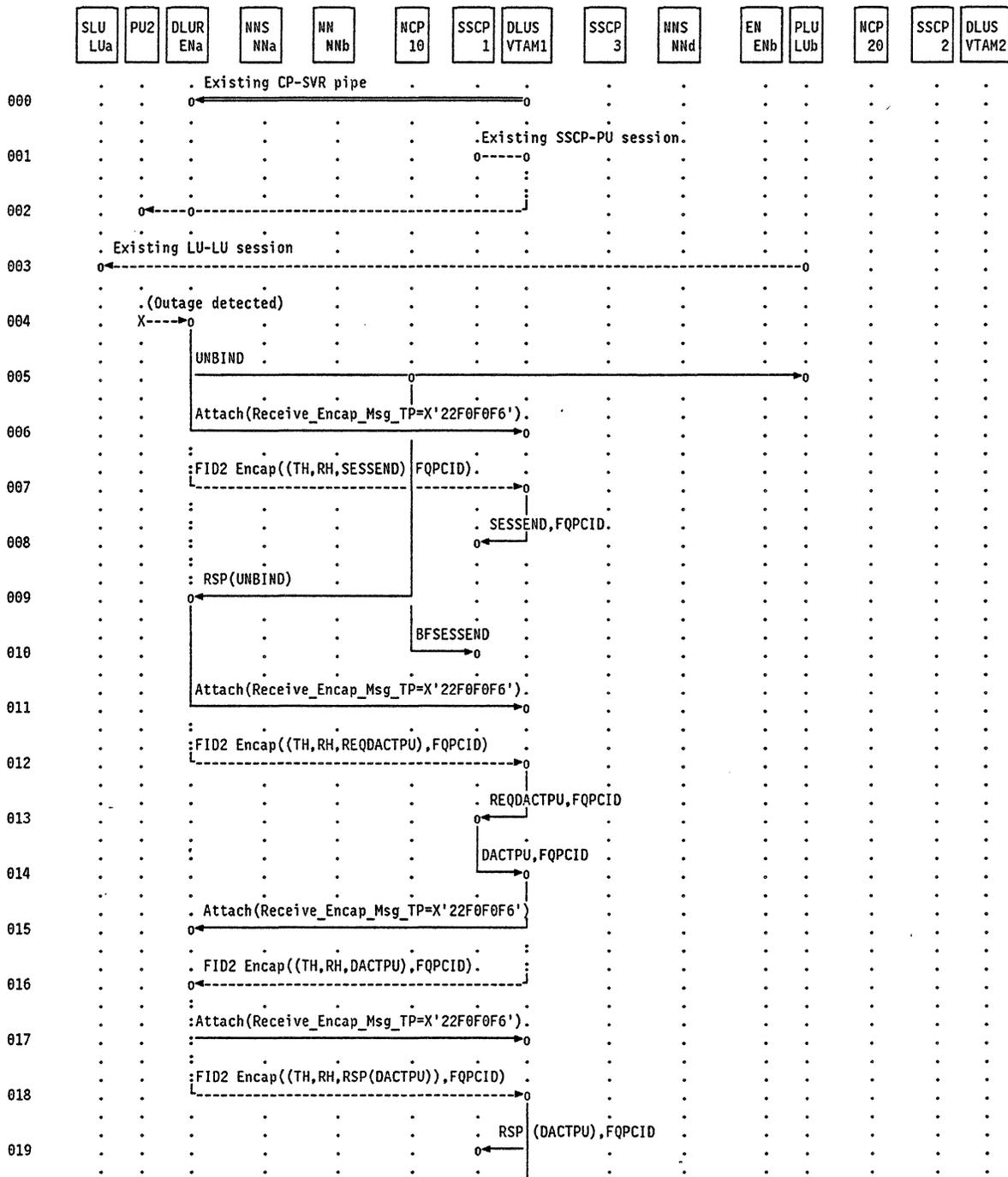
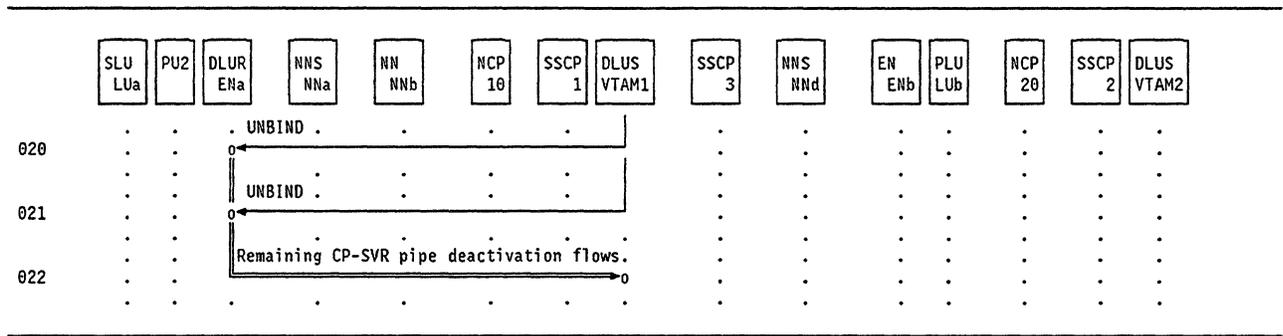


Figure 10-13. LU-LU session deactivation (abnormal)



1. There is an active session between SSCP1 and PU2. It is the only active SSCP-PU session using the CP-SVR pipe between VTAM1 and ENa.
- 2.
3. There is an active session between LUa and LUb.
4. ENa detects an outage on its connection to PU2.
5. UNBIND is sent by the DLUR in ENa to LUb, routing through NNa, NNb, NCP10, NNd, and ENb. The UNBIND type is X'08' (route extension inoperative).
6. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
7. The DLUR in ENa also creates a Format 3 SESSEND RU and encapsulates it, sending it on the CP-SVR pipe to the DLUS in VTAM1.
8. DLUS de-encapsulates the SESSEND and forwards it to SSCP1.
9. Upon receipt of UNBIND, NCP10 creates a RSP(UNBIND) and sends it to ENa while forwarding the UNBIND on to LUb.
10. NCP10 also creates a BFSESEND and sends it to SSCP1. SSCP1 correlates the BFSESEND with the SESSEND by the FQPCID contained in each RU.
11. The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
12. To signal to SSCP1 that its SSCP-PU session with PU2 is broken, ENa builds a REQDACTPU, encapsulates it, and forwards it to VTAM1 on its CPSVRMGR pipe.
13. VTAM1 de-encapsulates the REQDACTPU and forwards it to SSCP1.
14. SSCP1 responds to the REQDACTPU by building a DACTPU and passing it on to VTAM1.
15. The DLUS node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUR node.
16. VTAM1 encapsulates the DACTPU and forwards it to ENa.
17. ENa de-encapsulates the DACTPU, and since it no longer has a connection to PU2, converts the DACTPU into a positive RSP(DACTPU). The DLUR node sends an FMH5 Attach, to initiate the Receive_Encap_Msg_TP at the DLUS node.
18. ENa then encapsulates the RSP(DACTPU) and sends it to VTAM1.
19. VTAM1 de-encapsulates the RSP(DACTPU) and forwards it to SSCP1.
20. VTAM1 also determines that there are now no active or pending SSCP-PU sessions on its CP-SVR pipe to ENa, so it sends an UNBIND to ENa on VTAM1's conwinner session to begin deactivation of the pipe.
21. VTAM1 also sends an UNBIND on its conloser session to ENa.
22. The DLUR node sends a RSP(UNBIND) to the DLUS for the conwinner session and another RSP(UNBIND) for the conloser session. For more details on CP-SVR pipe deactivation see 5.6, "CP-SVR Pipe Deactivation" on page 5-16.

Chapter 11. Multi-Subnet DLUR/S

11.1 Overview

So far this document has described DLUR/S protocols when the DLUS, DLUR, and PLU have all resided in the same APPN subnet. This chapter will describe the additional functions necessary to allow one or more of these nodes (DLUS, DLUR, PLU) to reside in different APPN subnets. These functions will impact:

- DLUS
- DLUR
- Extended Border Node (EBN)
- Central Directory Server (CDS)

! For DLURs unable to support full multi-subnet function, there is a limited DLUR multi-subnet support capability. A DLUR with limited multi-subnet support must reside in the same subnet as its DLUS, but a limited multi-subnet DLUR and the PLU can reside in different subnets. A DLUR with limited multi-subnet support has all searches for its DLUS-supported dependent LUs handled by its DLUS.

11.2 Multiple Subnet DLUR/S Configurations

There are five possible configurations to consider (there may or may not be intermediate subnets also, but the same considerations apply):

Table 11-1. Multiple Subnet DLUR/S Configurations						
Configura-tion	DLUS	DLUR	PLU	Case 1	Case 2	Case 3
1	Subnet A	Subnet A	Subnet A			
2	Subnet A	Subnet B	Subnet B	X	X	
3	Subnet A	Subnet B	Subnet A	X		X
4	Subnet A	Subnet A	Subnet B		X	X
5	Subnet A	Subnet B	Subnet C	X	X	X

These configurations can fall into three cases, each with their respective requirements.

11.2.1 Case 1: DLUS And DLUR In Different Subnets

- Requirements/Solutions
 1. CP-SVR pipe activation rules require that the pipe not cross through a subarea network. In multiple subnet paths, this cannot be currently enforced.
 - A new indicator will be needed to suppress subarea searching in every subnet when a multi-subnet CP-SVR pipe is established.

2. Tail vector registration when the DLUR is in a different subnet cannot be used by the DLUS to provide route information to the PLU.
 - The DLUS will have to indicate that the tail vectors are not usable and that another Locate must be sent to the DLUR to acquire correct tail vectors. Also that Locate will need a special indicator so that the DLUR will return a positive reply (normally the DLUR always responds negatively to Locates for a DLUR-attached dependent LU).

11.2.2 Case 2: DLUS And PLU In Different Subnets

- Requirements/Solutions

1. Border Nodes need to send Locates for the SLU to the DLUS. However, they should not send BINDs for the SLU to the DLUS, since this would set up a suboptimal LU-LU session path.
 - Responsibility for acquiring tail vectors will depend on whether or not the DLUS and DLUR reside in the same subnet:
 - If the DLUS and DLUR are in the same subnet, the tail vectors returned by the DLUS, along with specifying ENCP=DLUR and NNS=DLUS, will provide enough information to calculate an LU-LU session route which does not include the DLUS (see 8.2, “DLUR EN Tail Vector Registration” on page 8-4 for more information - note tail vector registration only occurs when the DLUR is an EN).
 - If the DLUS and DLUR are in different subnets, the Extended Border Node nearest the PLU will need to send a Locate to the DLUR to acquire the correct tail vectors. The Border Node also will modify the response to include NNS=BN. This will provide enough information to calculate an LU-LU session route which does not include the DLUS or even the DLUS’s subnet.
2. DLUS needs to send Locates for the PLU through a Border Node. However, the BIND from the PLU should not necessarily be sent back through the same Border Node (especially when PLU and DLUR are in the same subnet).
 - Responsibility for acquiring tail vectors and modifying the Locate to appear as if it came from the DLUR will depend on whether or not the DLUS and DLUR reside in the same subnet:
 - If the DLUS and DLUR are in the same subnet, the tail vectors will be returned by the DLUS. The DLUS will modify the Locate, specifying ENCP=DLUR and NNS=DLUS. The LU-LU session route will go through the same Border Nodes as the Locate from the DLUS to the PLU.
 - If the DLUS and DLUR are in different subnets, the Extended Border Node nearest the PLU will need to send a Locate to the DLUR to acquire the correct tail vectors. The Border Node also will use this information to modify the the Locate for the PLU to appear as if it came from the DLUR. This will provide enough information to calculate an LU-LU session route which does not include the DLUS or even the DLUS’s subnet.

11.2.3 Case 3: PLU And DLUR In Different Subnets

- Requirements/Solutions

1. Border Nodes need to send Locates for the SLU to the DLUS. However, they should not send BINDs for the SLU to the DLUS, since this would set up a suboptimal LU-LU session path.
 - Responsibility for acquiring tail vectors will depend on whether or not the DLUS and PLU reside in the same subnet:

- If the DLUS and PLU are in the same subnet, it is the DLUS's responsibility to provide the correct tail vectors to the PLU. The DLUS will need to send a Locate to the DLUR to acquire the correct tail vectors. This will provide enough information to calculate an LU-LU session route which does not include the DLUS.
 - If the DLUS and PLU are in different subnets, see 11.2.2, "Case 2: DLUS And PLU In Different Subnets" on page 11-2.
2. DLUS needs to send Locates for the PLU through a Border Node. However, the BIND from the PLU should not necessarily be sent back through the same Border Node.
 - Responsibility for acquiring tail vectors and modifying the Locate to appear as if it came from the DLUR will depend on whether or not the DLUS and PLU reside in the same subnet:
 - If the DLUS and PLU are in the same subnet, it is the DLUS's responsibility to provide the correct tail vectors and to modify the Locate to appear as if it came from the DLUR. The DLUS will need to send a Locate to the DLUR to acquire the correct tail vectors. This will provide enough information to calculate an LU-LU session route which does not include the DLUS.
 - If the DLUS and PLU are in different subnets, see 11.2.2, "Case 2: DLUS And PLU In Different Subnets" on page 11-2.
 3. Blind BINDs (BINDs sent without a prior Locate) for DLUR-attached dependent LUs need to be intercepted by a Border Node to allow the DLUR to be involved in LU-LU session activation.
 - Normally a Extended Border Node would generate a Locate for the SLU and the DLUR would respond not found. The Border Node will need to include an indicator in that Locate so that the DLUR will return a positive reply.

11.2.4 Configuration Restrictions

Complications arise in the Locate/Search procedures for a dependent LU served by DLUR/S nodes across multiple subnets. Rather than complicate the DLUR by requiring that it or its NNS handle Session Services Extensions Locate functions, it was decided to have the DLUS handle those functions. To do this, the DLUS must initiate and respond to Locate Searches on behalf of the SLU even though BINDs to the SLU must go through the DLUR. This means that a Locate search initiated by a PLU for a SLU must (crossing intercluster links if necessary) reach the DLUS of the SLU which handles Session Services Extensions complications to the search. The DLUR must also be located (either through a search or by information passed back by the DLUS) since it will be the recipient of the BIND. The need to have two apparent locations for the SLU is the root cause of the complications for multiple subnet DLUR/S. These separate locations must be cached by BNs, DLUS, and central directory servers.

The following restrictions must be adhered to in configuring multiple subnets to provide DLUR/S support:

1. The CP-SVR pipe from the DLUS to the DLUR must not pass through a subarea network.
2. A path through EBNs must exist between the PLU and DLUS in different subnets to support Session Services Extensions type flows. This includes a requirement for a EBN in the PLU's subnet as well as a requirement that if the PLU is in a subarea network, the connection to the subnet containing the ICN (to that subarea) must be a EBN.
3. Within its subnet, the DLUR may have either NNs or R1BNs (Release 1 Border Nodes), which must connect to a EBN across an Intercluster Link (ICL) along the path to a PLU in another subnet. The EBN must handle blind BINDs to the SLU. Notice that there are no Session Services Extensions type flows between the DLUR and the PLU.

- ! 4. Within its subnet, the DLUR may have either NNs or RIBNs, which must connect to a EBN across an
! Intercluster Link (ICL) along the path to the DLUS in another subnet.

11.3 Functional Description

In the previous section various solutions to requirements for multi-subnet DLUR/S support were outlined. What follows are the details of this support.

Added function must be put in Extended Border Nodes, DLUS nodes, DLUR nodes, and Central Directory Servers (CDS). The function required differs depending on the distribution of the PLU, DLUR, and DLUS among subnets. If the DLUR and DLUS are both in the same subnet, no redirected Locate Searches to Locate the DLUR are required. In all other cases, either the DLUS or a EBN initiates a second search. Which node conducts the extra search and when it does so depends on the configuration of subnets and where EBNs are located between subnets. CDS must cache two locations for each SLU, one being the Location of the DLUS and the second being the location of the NNS(DLUR).

11.3.1 Border Node Terminology

- Origin Border Node (OBN)
 - The OBN is the nearest EBN to the OLU along a Locate path. This BN need not be in the same subnet as the OLU. If a BN receives a search and the Intercluster Links Crossed count is zero, then the BN can assume that it must serve as the Origin Border Node. The OBN may be required to perform searches to Locate the DLUR of a SLU in the case of a PLU-initiated search.
- Destination Border Node (DBN)
 - The DBN is the nearest EBN to the DLU along a Locate path. This BN need not be in the same subnet as the DLU, however, in general, the BN will not be able to determine that it is a DBN unless it is in the same subnet or across an ICL from the subnet containing the DLU. The DBN may be required to initiate a search for the DLUR of the SLU in the case of a SLU-initiated search.

11.3.2 Multi-Subnet DLUR/S Indicators

- DSR - DLUR Search Required
 - one bit field carried in Locate
 - indicates that the tail vectors in the CDINIT are not usable in the BIND and a follow up search to locate the resource on the DLUR is required. This bit always applies to the SLU. This bit is turned on by the DLUS when it generates a Found if the DLUS and DLUR are in different subnets (this information is available when the CP-SVR pipe is set up). The DSL bit (see below) is always set when the DSR is set.
- DSL - DLUS-Served LU
 - one bit field carried in the Find and Found GDS variables
 - indicates that the SLU is a DLUS-served LU. The DSL bit is set in a SLUinit in the Locate Find by the DLUS node initiating the search. The DSL bit is also set by the DLUS or DLUR in the Found in the Locate reply if the found resource is a DLUS-served LU.
- OCR - Owning CP Respond
 - one bit field carried in the Find and Found GDS variables

- If the OCR is set in the Find, the DLUR owning the resource will respond with a Found. If the OCR is not set, a DLUR will respond not found to the resource if it is a DLUS-served LU owned by the DLUR.
- PSP - Prevent Subarea Path
 - one bit field carried in the Locate GDS variable
 - When this bit is set, the BN does not reset the Suppress Subarea Search bit. This bit is set when the DLUS or DLUR is sending a Locate as part of the CP-SVR pipe activation since the pipe may not cross a subarea network.
- ! • Limited DLUR Multi-Subnet Support
 - ! – one bit field carried in the DLUR/S Capabilities control vector
 - ! – When this bit is set, the DLUR will not respond to searches with the OCR bit set; the DLUS will respond to these searches.

11.3.3 Multi-Subnet CP-SVR Pipe Activation

When the DLUS wants to set up the CP-SVR pipe, the DLUS must initiate a Locate for the DLUR. When the DLUR wants to set up the CP-SVR pipe, the DLUR must initiate a Locate for the DLUS. In either case the Locate must not cross a subarea. To insure this, the node initiating the Locate must set both the Suppress Subarea Search (SSS) bit in the Locate and the Prevent Subarea Path (PSP) bit in the Locate. Once the partner (DLUS or DLUR) is located without crossing a subarea, the CP-SVR pipe may be established just like in the single subnet case.

- ! During CP-SVR pipe activation, if a DLUS detects:
 - ! • the CP-SVR pipe crosses a subnet boundary, and
 - ! • the DLUR only supports limited multi-subnet function in the DLUR/S Capabilities control vector
- ! then the DLUS will deactivate the CP-SVR pipe, using sense code X'088E 000F' in the CV X'35' of the UNBIND(s) (see 5.6.2.2, "DLUR/S Capabilities Mismatches" on page 5-26 for more information on DLUR/S capabilities mismatch processing).

! 11.3.4 Multi-Subnet EN DLUR Capabilities

- ! A DLUR-capable End Node must set the Search Status Indicator in the ENCP Search Control CV (X'33') during the CP Capabilities exchange with its network node server.
- ! An End Node may choose not to set this indicator if it registers all LUs with its NNS (this significantly cuts down domain broadcast traffic). An End Node cannot register its DLUS-owned LUs with the NNS, since the NNS will delete the directory entry as soon as the EN rejects a Locate (which it must do if the OCS bit is not set). Even if a DLUR-capable EN registers all its non-DLUS-owned LUs, it must still set the Search Status Indicator.

11.3.5 Multi-Subnet DLUR/S Locate Searches

The DLUS must respond to PLU-initiated Locates for DLUS-served LUs. The DLUS must initiate searches on behalf of the SLUs it serves. In both the SLU-initiated and PLU-initiated searches, the ensuing BIND must go from the PLU to the DLUR (not the DLUS) before reaching the SLU. The mismatch between the endpoints of the BIND and the Locate must be accounted for. The DLUS and EBN redirected searches required to handle this are described below.

11.3.5.1 PLU-initiated searches

For PLU-initiated searches, either the DLUS or OBN (if one exists) making the Locate response to the PLU must indicate that the BIND should be sent to the DLUR. If the DLUS does not already know the location of the DLUR (relative to the PLU), either the DLUS or the OBN must find it for the PLU.

If the DLUS and DLUR are in the same subnet, the DLUS will know the TGs for the DLUR and can return them in the Locate Found. In this case no additional search for the DLUR is required. The only modification the DLUS will need make to the Locate reply is return the DLUR TGs with the DLUR listed as ENCP in the reply. Both the DLUS and DLUR for the SLU must be located before the session is established. The DLUS(SLU) must be located first. In the case where the DLUR and DLUS are in the same subnet, only the Locate to the DLUS is necessary since the DLUS knows the TGs to the DLUR which can be used to determine the best route to the DLUR.

When the DLUS is in a different subnet than the DLUR, a separate search to the DLUR must be conducted to find a multisubnet route between the PLU and DLUR. This search will be initiated by the OBN when one is available. If there is no OBN, then the PLU is in the same subnet as the DLUS. In this case the DLUS must initiate the search (which verifies a COS-acceptable route across subnets). When the DLUR is located by the OBN or DLUS in this manner, the Locate Found returned to the PLU is modified so that it will be possible for the PLU to establish a session with the SLU which goes through the DLUR but need not go through the DLUS or even the subnet containing the DLUS.

What follows is the specific role of the DLUR node in a multi-subnet DLUR/S PLU-initiated search:

11.3.5.1.1 DLUR / PLU-initiated search: The function of the DLUR is to serve as the intermediary between the dependent LU and the SSCP on the DLUS. The session between a PLU and a DLUS-served SLU must pass through the DLUR to reach the SLU.

- ! A DLUR will return a positive reply to a search for one of its dependent resources only if the Find has the
- ! OCR bit set and the DLUR supports full multi-subnet function. Only a EBN or DLUS should initiate a search with the OCR bit set. Except when the EBN generates a search in response to a blind BIND, no searches for a dependent LU with the OCR bit set should be initiated unless the DLUS has first received a search without the bit set. Searches initiated by downlevel nodes will never have this bit set and will always locate the DLUS rather than the DLUR.

The DLUR Found response will have the DSL bit set, indicating that the SLU is a dependent LU. On a DLUR search reply, the DLUR should set the NNS(DLUR) as the NNS(SLU), the DLUR as ENCP and the DLUS is carried in the CV X'40'.

11.3.5.2 SLU-initiated searches

When a SLU wishes to initiate a search, the DLUS must begin the search on behalf of the SLU to find the PLU. The DSL bit is set to indicate that the SLU is a DLUS-served LU. The Locate is initiated at the DLUS. However, when the PLU receives the Locate, it must send a BIND to the SLU by way of the DLUR (and not back to the DLUS). The complication of the SLU-initiated search comes about because the multi-subnet route that the BIND takes from the PLU to the SLU by way of the DLUR may pass through a different set of BNs than the Locate did in its path from the DLUS to the PLU. The SLU-initiated Locate sent to the PLU must contain the appropriate node or TGs to get the BIND to the correct destination. If the PLU is local to the DLUR subnet, the TGs to the DLUR (if an EN) or an RSCV containing the DLUR as last entry (if a NN) are carried by the Locate to the PLU. When the PLU is not local to the DLUR subnet, the TGs will be those appropriate to leave the PLU subnet along a path to the DLUR. The DLUS will be carried as a CV'40' when a EBN has been crossed in the path between the DLUS and the PLU.

11.3.5.2.1 DLUR / SLU-initiated search: The function of the DLUR is to serve as the intermediary between the dependent LU and the SSCP on the DLUS. The session between a PLU and a DLUS-served SLU must pass through the DLUR to reach the SLU. A DLUR will return a positive reply to a search for one of its dependent resources only if the Find has the OCR bit set and the DLUR supports full multi-subnet function. Only a EBN or DLUS should initiate a search with the OCR bit set. Searches initiated by downlevel nodes will never have this bit set and will always locate the DLUS rather than the DLUR. The DLUR Found response will have the DSL bit set, indicating that the SLU is a dependent LU. On a DLUR search reply, the DLUR should set the NNS(DLUR) as the NNS(SLU), the DLUR as ENCP and the DLUS is carried in the CV X'40'.

11.3.6 Multi-Subnet Functional Impact Summary

The following DLUR functions are required for DLUR/S support in a multiple subnet environment:

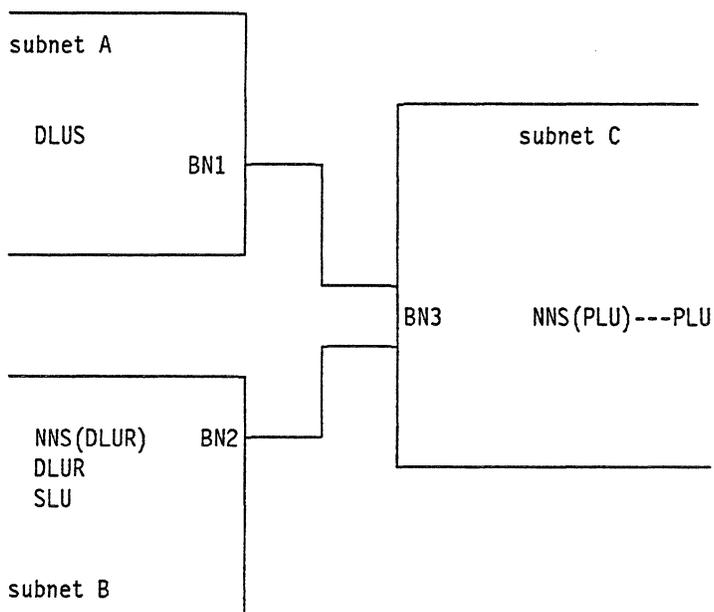
11.3.6.1 DLUR / Multi-Subnet Impacts

- ! • The DLUR returns not found for Locates for its DLUS-served resources unless the OCR bit is set and the DLUR supports full multi-subnet function.
- ! • The DLUR returns Found (with the DSL bit set) if the OCR bit is set in the Find and the DLUR owns the resource (identifies the SLU as served by a DLUS) and the DLUR supports full multi-subnet function.
- ! • The DLUR returns Found with CV'40' indicating the DLUS name for use by management services.
- ! • The DLUR sets the PSP bit when initiating a Locate for the DLUS for the purpose of setting up the CP-SVR pipe. This prevents setting up a pipe across a subarea.
- ! • Wildcards do not apply to DLUS-served LUs. All DLUS-served LUs must be explicitly defined on the DLUR and on the DLUS. No wildcard responses are made with the DSL bit set.
- ! • A DLUR-capable End Node must set the Search Status Indicator in the ENCP Search Control CV (X'33') during the CP Capabilities exchange with its network node server.

11.4 Multi-Subnet DLUR/S Flows

11.4.1 Multi-Subnet PLU-Initiated Searches

11.4.1.1 DLUS, DLUR, PLU In Different Subnets



SUBNET lists

BN1		BN2		BN3	
Search for net	at BN	search for net	at BN	search for net	at BN
NETC(a11)	BN3	NETC(a11)	BN3	NETC(a11)	BN3
NETB(OCR off)	BN1,BN3	NETB(OCR off)	BN2,BN3	NETB(OCR off)	BN2,BN1
NETB(OCR on)	BN3	NETB(OCR on)	BN2	NETB(OCR on)	BN2
NETA(a11)	BN1	NETA(a11)	BN3	NETA(a11)	BN1

Figure 11-1. DLUS, DLUR, PLU in different subnets / one BN per subnet

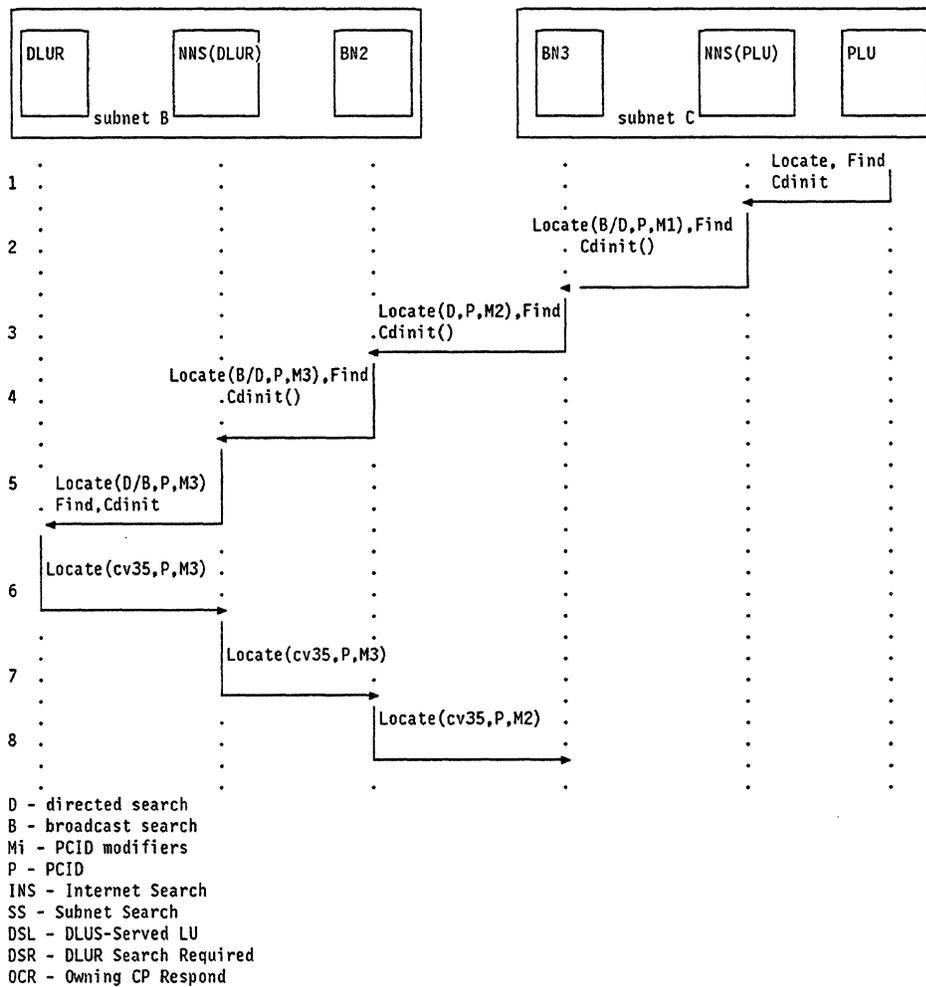


Figure 11-2. PLU-initiated search for dependent LU when DLUS, DLUR and PLU are all in different subnets.

Notes:

1. The search for NETB.SLU is originated by the PLU in subnet C which forwards the search to its Network Node Server.
2. The NNS(PLU) does not have the location of SLU cached and sends a broadcast which reaches a Border Node (BN3). The "DLUS-Served LU" (DSL) bit is off.
3. BN3 checks its subnet list and finds two entries for NETB with the DSL bit off. The search is forwarded from BN3 to BN2 since BN2 is the first entry in the list. BN3 caches the PCID and remembers that the DSL bit was off.
4. BN2 checks its subnet list for NETA and finds entries for its local net and for BN3. BN2 first searches its local net by sending out a broadcast with the internet search bit on (to prevent exiting the subnet). The DSL bit is off. BN2 caches the PCID for the search along with the information that the DSL bit was off.
5. NNS(DLUR) forwards the search to DLUR.

6. DLUR returns a not Found since it does not respond to a search for the dependent LU (SLU) unless the search comes in with the OCR bit on which indicates that the DLUS has properly set up the pipe between the DLUS and the DLUR.

7. NNS(DLUR) returns a not Found to BN2.

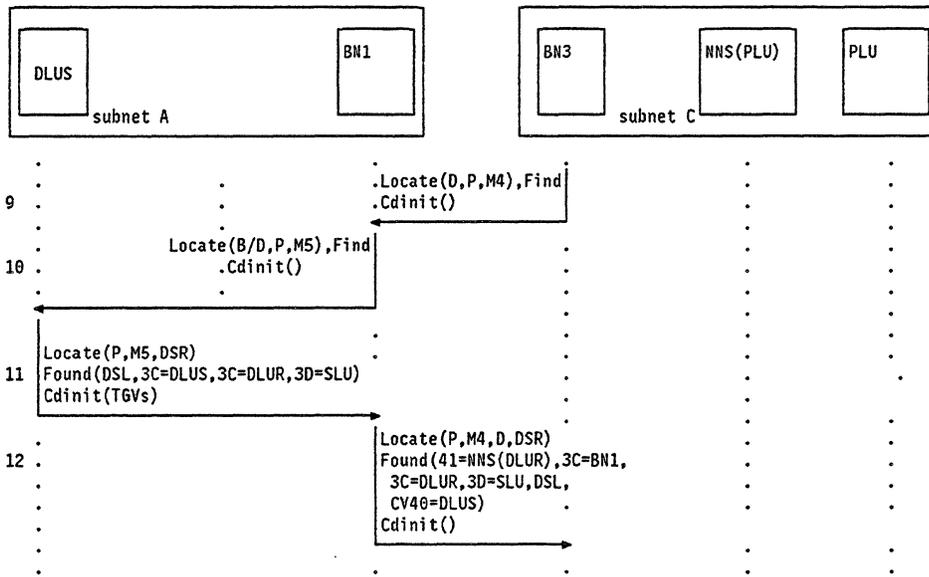


Figure 11-3. PLU-initiated search for dependent LU support when DLUS, DLUR and PLU are all in different subnets (cont.).

8. BN2 has discovered that the SLU is not in subnet B so BN2 checks its subnet list for the next place to search for SLU. Based on the subnet list, BN2 sends a not Found to BN3.
9. BN3 checks its subnet list and finds that a resource with NETID NETB may be in BN1 and forwards a search to BN1.
10. BN1 initiates a search of its local subnet (NETA) by sending out a broadcast.
11. DLUS receives the search for SLU, and realizes that it supports that particular resource through DLUR. DLUS assumes the role of network node server of SLU with DLUR as the ENCP (even if DLUR had been a NN, it is treated as an ENCP in the hierarchy). The Found returned to BN1 contains these control vectors indicating the hierarchy: X'3C'(DLUS,X'F6'), X'3C'(DLUR,X'F4'), and X'3D'(SLU,X'F3'). The DSL bit is set since it is known that SLU is a DLUS-served LU. The DSR bit is set since the TGs returned by DLUS are not appropriate to reach the DLUR. The Found is returned to BN1.
12. BN1 replaces the name in the first control vector (X'3C') with its own name and returns it to BN3. The DLUS name which was initially in this CV X'3C' is put in a CV X'40'. BN1 caches the location of DLUS(SLU) for future searches.

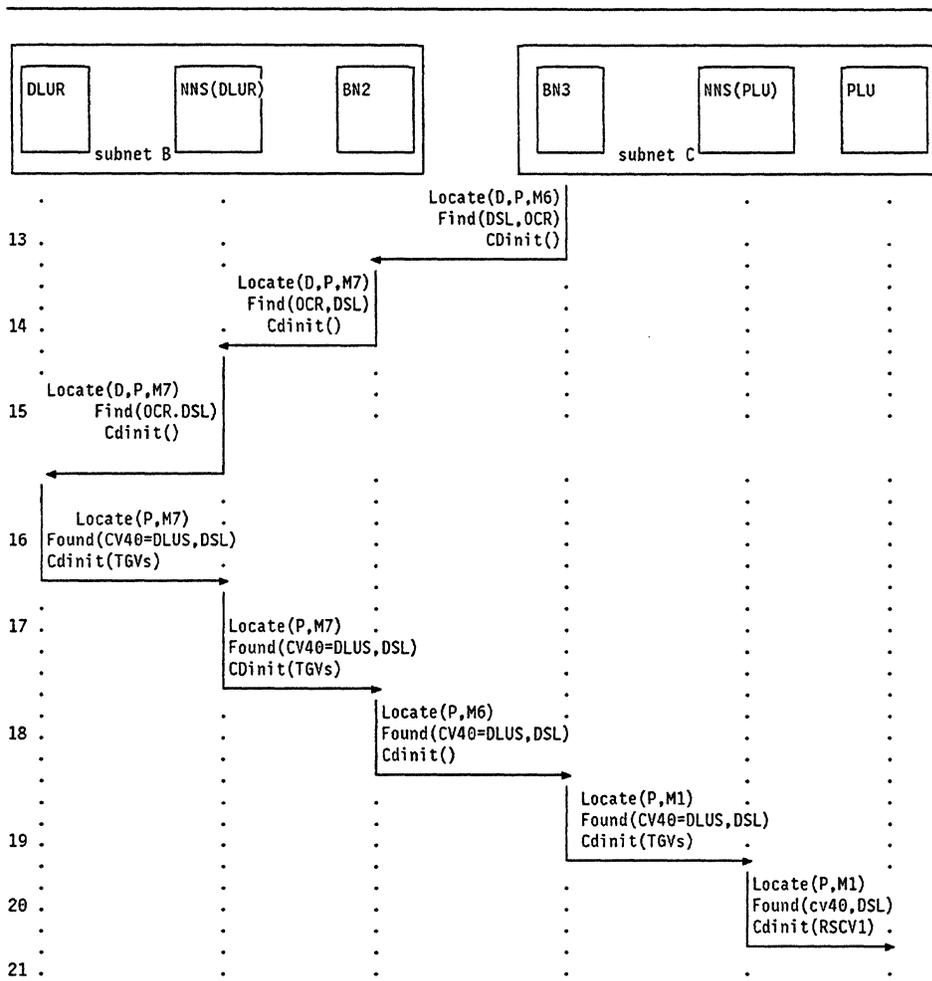


Figure 11-4. PLU-initiated search for dependent LU support when DLUS, DLUR and PLU are all in different subnets (cont.).

13. BN3 replaces the name in the first control vector (X'3C') with its name. It caches the location of SLU by way of the DLUS as BN1. Since BN3 is the originating BN for this search it has extra responsibilities. The DLUS has been located. Now BN3 must send a Locate Find (with the OCR bit set and an incremented PRN) to locate the SLU by way of the DLUR. BN3 sees from its subnet list (and the OCR bit set) that the appropriate BN to send the search to is BN2.
14. BN2 had previously cached the PCID for this search but with a different PRN. Since this search has a new PRN, BN2 will accept this search (and cache the fact that this PCID with its PRN was received). BN2 sees from its subnet list that NETB.SLU might be in its local subnet. BN2 checks its cache, and if the SLU by way of the DLUR is in its cache, BN2 sends a directed search to NNS(DLUR) for SLU. Otherwise BN2 sends a broadcast search for the SLU with the INS and SS bits set to prevent exiting the subnet.
15. When NNS(DLUR) receives the search it is sent to DLUR.
16. DLUR sees that the OCR bit is on, therefore it responds with a Found. The DSL bit is set and CV X'40' = DLUS is carried on the Found.
17. NNS(DLUR) returns the Found to BN2 with CV X'3C' = NNS(DLUR), CV X'3C' = DLUR, and CV X'3D' = SLU.

18. BN2 replaces the first CV X'3C' with its own name but does not change the CV X'40'. BN2 caches the location of the SLU by way of the DLUR as being at NNS(DLUR). The Found is sent to BN3.
19. BN3 caches the location of SLU by way of the DLUR as BN2. BN3 inserts its own name as the CV X'3C' indicating NNS and returns the Found to NNS(PLU).
20. NNS(PLU) caches BN3 as the NNS of SLU and sends the Found back to PLU.
21. PLU sends a BIND to BN3. BN3 checks its cache (associates the Locate with the BIND PCID and the highest PRN) and sends the BIND on to BN2. BN2 checks the cache and forwards the BIND on to NNS(DLUR) and to the DLUR from there. The BIND response is sent back through BN2 and BN3 to PLU.

11.4.2 Multi-Subnet SLU-Initiated Searches

11.4.2.1 DLUS, DLUR, PLU In Different Subnets

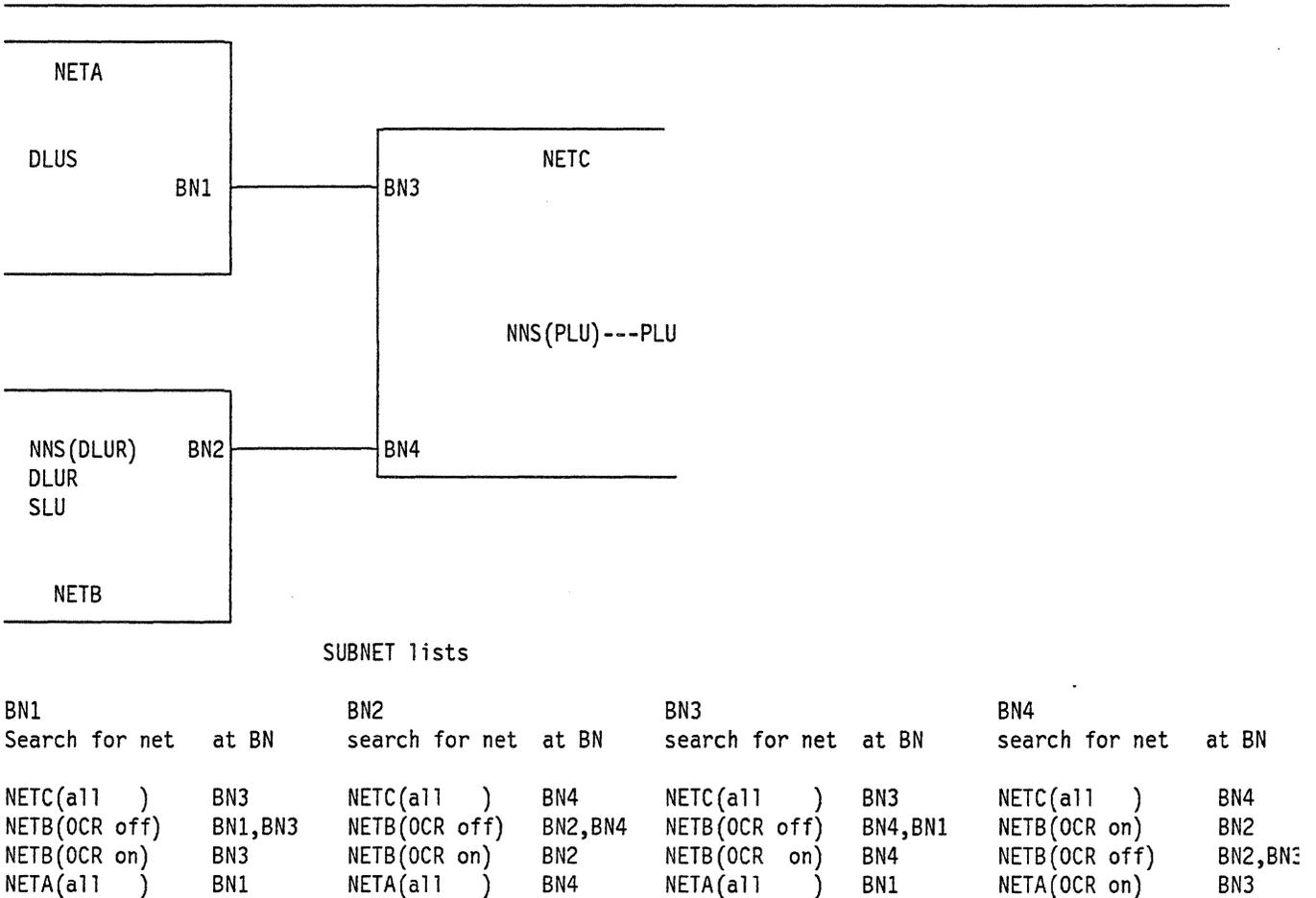


Figure 11-5. DLUS, DLUR, PLU in different subnets / more than one BN per subnet

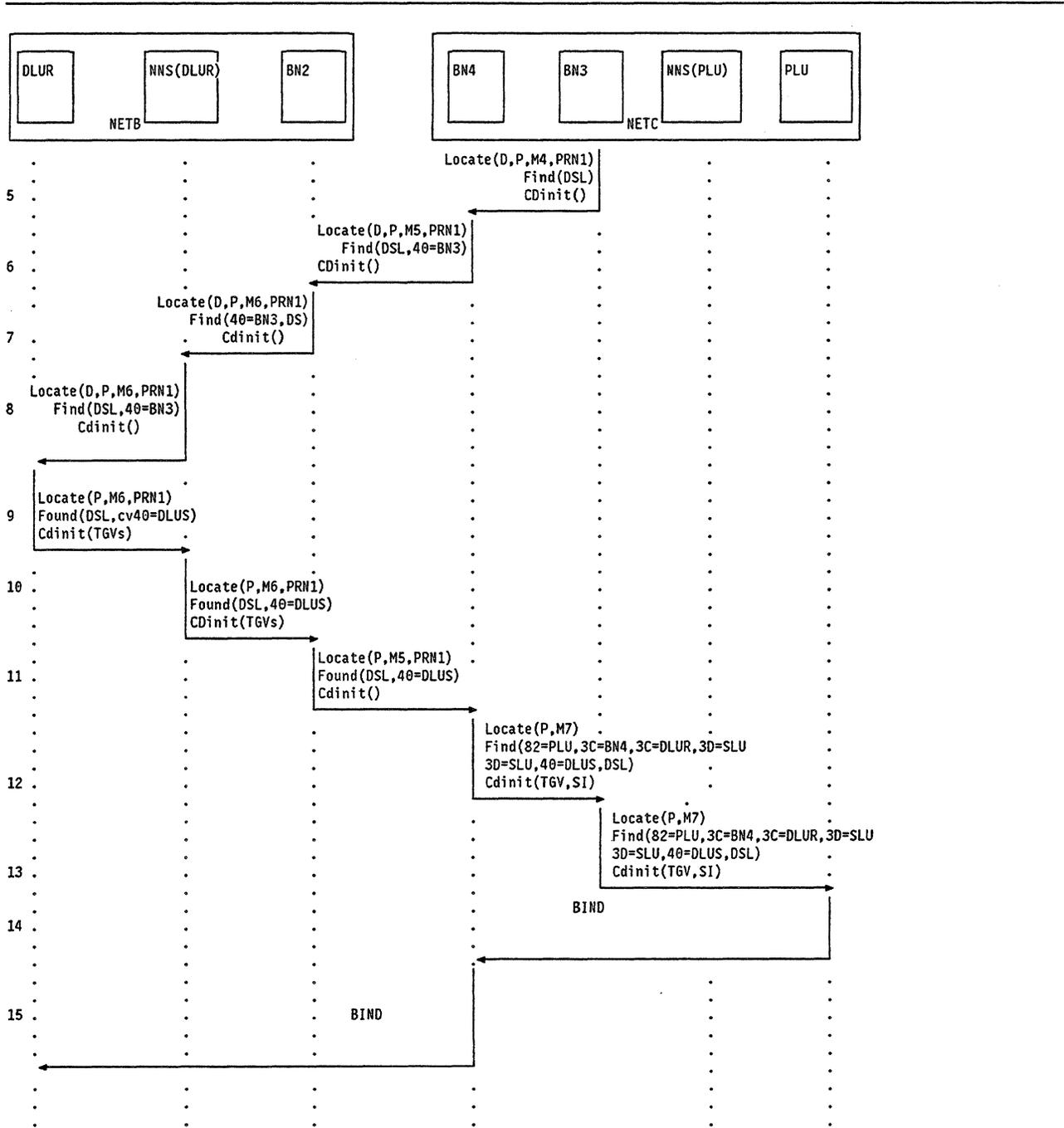


Figure 11-7. SLU-initiated search when DLUS, DLUR and PLU are all in different subnets (continued).

- BN3 replaces the name in the first (NNS) control vector X'3C' with its name. It caches the location of SLU by way of the DLUS as BN1. Since BN3 is the originating BN for this search and the Find has the DSR bit set, the BN has extra responsibilities. Since the DLUS has been located, its name is in the CV X'40'. This information is saved and will later be sent back to the PLU. Now BN3 must send a Locate Find (with the OCR bit set and an incremented PRN) to locate SLU. BN3 sees from its subnet list that the appropriate BN to send the search to is BN4.

6. BN4 receives the search from BN3 and indicates itself to be the NNS and puts BN3 in the CV X'40'. BN4 sees from its subnet list that it must send the search to BN2.
7. BN2 will accept this search (and cache the fact that this PCID/PRN was received). BN2 sees from its subnet list that NETB.SLU might be in its local subnet. BN2 will check its cache for the location of SLU by way of the DLUR. If SLU is found, BN2 sends a directed search to NNS(DLUR) for SLU; otherwise a broadcast search restricted to BN2's local subnet (NETB) will be initiated.
8. When NNS(DLUR) receives the search it sends it on to DLUR.
9. DLUR sees that the OCR bit is on, therefore it responds with a Found (DSL bit on). A CV X'40' is appended with the name of DLUS.
10. NNS(DLUR) returns Found to BN2 with CV X'3C'=NNS(DLUR), CV X'3C'=DLUR, CV X'3D'=SLU and CV X'40'=DLUS.
11. BN2 replaces the first CV X'3C' with its own name. BN2 caches the location of the SLU by way of the DLUR as being at NNS(DLUR). The Found is sent to BN4.
12. BN4 accepts the Found, indicates itself as being the NNS(SLU), caches the location of SLU by way of the DLUR as being at BN2, and sends the search back to BN3 with a TG from BN4 to BN2.
13. BN3 caches the location of SLU by way of the DLUR as BN4. BN3 now has the information it needs to modify the original SLU-initiated search and send it on to NNS(PLU). The search must indicate that BN4 is the NNS since it is the BN on the route to the DLUR. The PCID, modifier, and PRN are set indicate that the original SLU-initiated search is continuing. The TG from BN4 to BN2 is put in the CDINIT. BN3 keeps the name of BN4 as the NNS(SLU). The CV X'40'=DLUS is still included in the Find (for management services at the PLU). The modified Find is sent on to NNS(PLU).
14. NNS(PLU) caches BN4 as the NNS of SLU. It uses the TG in the CDINIT, computes a RSCV to BN4, and sends PLU instructions to initiate a BIND.
15. PLU sends BIND to BN4. BN4 checks its cache and sends the BIND on to BN2. BN2 checks the cache and forwards the BIND on to NNS(DLUR) and to the DLUR from there. The BIND response is sent back through BN2 and BN4 to the PLU.

Chapter 12. DLUR Option Set Content

12.1 DLUR Tower Requirements

Any product that implements the DLUR tower must implement the following functions as described in this document:

12.1.1 CP-SVR Pipe

The DLUR node will have to be able to establish the CP-SVR pipes in conjunction with the activation or reactivation of a given SSCP-PU session as described in Chapter 5, "CP-SVR Pipe" on page 5-1. This will specifically include the ability to:

1. Support the new CPSVRMGR mode and COS.
2. Know the DLUS node to be contacted when a PU requires ACTPU.
3. De-activate CP-SVR pipes when idle.
4. Bring down a conwinner session when the conloser fails.

12.1.2 SSCP-PU & SSCP-LU Session Activation/Encapsulation

The DLUR node will also be required to activate SSCP-PU and SSCP-LU sessions over the CP-SVR pipes. This will include both logic to establish the sessions as well as logic to encapsulate and de-encapsulate the associated flows. Specific requirements include the ability to:

1. Generate an FQPCID for a PU requiring activation.
2. Correlate SSCP-PU and SSCP-LU sessions to PU images on the basis of the FQPCID.
3. Generate the REQACTPU RU and understand related responses and sense codes.
4. Interrogate ACTLU commands to build a table that relates SLU names to the PU image that supports them.
5. Maintain an active session count for each dependent LU.
6. Examine +RSP(ACTLU) for extended BIND and network-qualified name support information.
7. Examine INIT-SELF to acquire PLU name and URC information.
8. Optionally provide SDDL NMT support to allow for dynamic dependent LU definition in the DLUS (required only if dynamic definition is supported).
9. Detect ACTPU race conditions and resolve them appropriately.
10. Generate the REQDACTPU RU and understand related responses and sense codes.
11. Encapsulate and de-encapsulate SSCP-PU and SSCP-LU FID2 PIUs.

12.1.3 EN TGV Reporting

DLUR ENs will be required to report their TGVs up to the DLUS node only if the DLUR and the DLUS are in the same subnet. These TDUs will be sent as events that effect them occur. In the case where the DLUR and the DLUS are non-adjacent (or no CPSVCMG session exists between them), these TDUs will flow over the CP-SVR pipe. In the case where the DLUR and DLUS are also connected via a CPSVCMG session, these TDUs will flow over that pipe.

12.1.4 LU-LU BIND Processing

1. Unextend extended BINDs destined for nodes that do not support receipt of such BINDs; reextend unextended + RSP(BIND)s when the corresponding BIND request was unextended.
2. Remove NETIDs from LU names in BINDs destined for nodes that do not support network-qualified names.
3. Replace PLU names in BINDs destined for nodes which specified a PLU name in the INIT-SELF different from the PLU name received in the BIND.
4. Reject BINDs received for already active dependent LUs.
5. Properly set the adaptive session pacing indicator in the BIND and + RSP(BIND) for the REX and APPN stages.
6. Route BINDs to the destination PU/LU based upon the SLU name specified in the NS field of the BIND and information maintained in the SLU name table.
7. Perform intermediate routing functions when the LU is located on a downstream PU. This includes:
 - Intermediate routing function
 - Session outage notification
 - Maintaining pacing counts/indicators in both directions

12.1.5 Session Awareness Reporting

In order to assist the DLUS node in providing session limit management, the DLUR node must provide the DLUS node with session awareness data. The DLUR node will be required to:

1. Upon receipt of a positive RSP(ACTLU) from the attached dependent LU, if the dependent LU has an active LU-LU session, the DLUR must build a Session Information (X'2A') control vector, append it to the RSP(ACTLU), and send the RU to the SSCP by encapsulating the RU and sending it on the CP-SVR pipe to the DLUS.
2. Upon receipt of a positive RSP(BIND) from the attached dependent LU, the DLUR must build a SESSST RU and send it to the SSCP by encapsulating the RU and sending it on the CP-SVR pipe to the DLUS.
3. Upon receipt of a RSP(UNBIND) from the attached dependent LU, the DLUR must build a SESSEND RU and send it to the SSCP by encapsulating the RU and sending it on the CP-SVR pipe to the DLUS.

12.1.6 Multi-Subnet Support

- ! 1. The DLUR returns not found for Locates for its DLUS-served resources unless the OCR bit is set and the DLUR supports full multi-subnet function.
- ! 2. The DLUR returns Found (with the DSL bit set) if the OCR bit is set in the Find and the DLUR owns the resource (identifies the SLU as served by a DLUS) and the DLUR supports full multi-subnet function.
- ! 3. The DLUR returns Found with CV'40' indicating the DLUS name for use by management services.
- ! 4. The DLUR sets the PSP bit when initiating a Locate for the DLUS for the purpose of setting up the CP-SVR pipe. This prevents setting up a pipe across a subarea.
- ! 5. Wildcards do not apply to DLUS-served LUs. All DLUS-served LUs must be explicitly defined on the DLUR and on the DLUS. No wildcard responses are made with the DSL bit set.
- ! 6. A DLUR-capable End Node must set the Search Status Indicator in the ENCP Search Control CV (X'33') during the CP Capabilities exchange with its network node server.

Chapter 13. Format Changes

This section describes changes to RUs, control vectors, GDS variables, and sense codes for Dependent LU Requester/Server architecture. The actual proposed changes can be found in Appendix D, "Formats For DLUR/S Architecture" on page D-1.

13.1 CP-SVR Pipe Format Changes

13.1.1 FID2 Encapsulation (X'1500') GDS Variable

This is a new GDS variable used to carry SSCP-PU and SSCP-LU sessions on the CP-SVR pipe. This variable will include:

- a field indicating the length of the GDS variable
- a GDS ID of X'1500'
- a field indicating the length of the FID2 PIU
- the entire FID2 PIU (TH, RH, RU (including optional control vectors))
- one or more control vectors
 - a Security ID Control (X'25') control vector
 - an Assign LU Characteristics (X'30') control vector
 - an Extended SDLC Station (X'43') control vector
 - a TG Descriptor (X'46') control vector. See 13.1.6, "TG Descriptor (X'46') Control Vector" on page 13-3 for details about the format changes for this control vector when it specifies the connection between the DLUR and the PU.
 - a new DLUR/S Capabilities (X'51') control vector
 - a Fully-Qualified Procedure Correlation Identifier (FQPCID) (X'60') control vector

An FQPCID includes a PCID, a network-qualified CP name, and an optional PCID modifier. The network-qualified CP name will be defined as follows:

 - DLUS-generated - the network-qualified SSCP name
 - DLUR-generated - the network-qualified CP(DLUR) name
 - a new XID Image (X'81') control vector

Table 13-1. GDS X'1500' Control Vectors		
	term used in flows	RUs CV appears with
X'25'		RSP(ACTPU), REQACTPU
X'30'	LU CHAR	ACTLU
X'43'	PU CHAR	ACTPU
X'46'	PU TG	ACTPU, REQACTPU
X'51'	DL CAP	ACTPU, RSP(ACTPU), REQACTPU, RSP(REQACTPU)
X'60'	FQPCID	all RUs
X'81'	XID	REQACTPU, RSP(ACTPU)

13.1.1.1 XID Image (X'81') FID2 Encapsulation Control Vector

- * This is a new control vector, locally defined to this GDS variable, which will contain either:
 - * • the I-field image of either the XID3 or XID0 SDLC response of the external PU attached to the DLUR,
 - * or
 - * • a DLUR-generated XID0 image for an internal PU.

13.1.2 ACTPU (Activate Physical Unit)

The Network Name (X'0E') control vector (type X'F1' - PU name) will be included with the ACTPU when sent by a DLUS to a DLUR-serviced PU.

13.1.3 REQACTPU (Request ACTPU)

This is a new RU built by the DLUR and sent to the SSCP in the DLUS requesting that an ACTPU be sent to establish an SSCP-PU session with a DLUR-attached PU. This PU could be pre-defined or undefined to the SSCP; it could be internal to the DLUR or downstream of the DLUR.

The REQACTPU will have two formats:

- * • Format X'0' - external PU
- * • Format X'1' - internal PU

A positive response to REQACTPU will indicate that the SSCP will send an ACTPU to the PU specified in the REQACTPU RU. A negative response to REQACTPU will indicate that no ACTPU will be sent, and a sense code will be included to specify the reason.

13.1.4 REQDACTPU (Request DACTPU)

This is a new RU built by the DLUR and sent to the SSCP in the DLUS requesting that an DACTPU be sent to take down an active SSCP-PU session with a DLUR-attached PU.

% The REQDACTPU will specify which PU's SSCP-PU is to be deactivated as well as the cause for the request, e.g.:

- % • X'00': loss of connectivity from the DLUR to the downstream PU
- % • X'01': REQDISCONT(normal) received from the downstream PU
- % • X'02': REQDISCONT(immediate) received from the downstream PU

% A positive response to REQDACTPU will indicate that the SSCP will send an DACTPU to the PU specified in the REQDACTPU RU, except for cause X'01', where the DACTPU may be preceded by LU-LU and SSCP-LU session deactivation commands. A negative response to REQDACTPU will indicate that no DACTPU will be sent, and a sense code will be included to specify the reason.

13.1.5 Network Name (X'0E') Control Vector

The Network Name (X'0E') control vector (type X'F1' - PU name) will be optionally network-qualified. The DLUS will include a network-qualified PU name when the PU's NETID is different from the SSCP's NETID.

13.1.6 TG Descriptor (X'46') Control Vector

! When this control vector is included in a FID2 Encapsulation (X'1500') GDS variable, the TG Identifier (X'80') subfield will not be included since the link being described by the CV in this case is not a TG but a boundary link.

A new subfield (X'91') will be created to identify the DLC type of a PU internal to or externally attached to the DLUR. Information carried in this subfield will include:

- Subfield X'91' - DLC type, e.g.,
 - Data Link Switching
 - Ethernet
 - FDDI
 - Frame relay
 - Internal PU
 - SDLC leased
 - SDLC switched
 - Token-Ring
 - X.25

Additional new subfields (X'92'-X'F0') will be defined for each DLC type to carry its signaling information, e.g., addresses and dial digits.

Some of the proposed changes can be found in Appendix D, "Formats For DLUR/S Architecture" on page D-1.

13.1.7 DLUR/S Capabilities (X'51') Control Vector

This is a new control vector which will describe the capabilities of the DLUS or DLUR to its partner.

For this, the initial level of DLUR/S support, there will be:

- a one byte release field indicating Release 1 DLUR/S support (X'01'). Future releases will have their own values to aid in uplevel/downlevel support determination.
- a one byte indicator specifying whether the node is a NN (X'00') or an EN (X'01').
- a one byte indicator specifying whether the node is a DLUR (X'00') or a DLUS (X'01').
- a four byte Flow Reduction Sequence Number (FRSN)
- a one byte support indicator field, including:
 - an indicator for support of the RECEIVE_TDU_TP (X'22F0F0F4') service transaction program - when set, the sending node supports receipt of TDUs on this session
 - ! – an indicator for limited DLUR support of ANS - when set, the DLUR only supports ANS = STOP; when not set, the DLUR supports both ANS = CONT and ANS = STOP;
 - ! – an indicator for limited DLUR multi-subnet support - when set, the DLUR will not respond to dependent LU searches with the OCR bit set; when not set, the DLUR will respond to dependent LU searches with the OCR bit set;
- four reserved bytes (for future use)

13.2 Multi-Subnet Session Format Changes

13.2.1 Locate (X'12C4') GDS Variable

13.2.1.1 Intersubnetwork Search (X'82') Locate Control Vector

13.2.1.1.1 DLUR Search Required (DSR) Indicator: This is a new indicator to specify whether or not a search to obtain the DLUR TGVs is required.

13.2.1.1.2 Prevent Subarea Path (PSP) Indicator: This is a new indicator to specify whether or not to prevent the R2BN from resetting the Suppress Subarea Search indicator.

13.2.2 Find Resource (X'12CA') GDS Variable

13.2.2.1 Command Parameters (X'80') Find Control Vector

13.2.2.1.1 DLUS-Served LU (DSL) Indicator: This is a new indicator to specify whether or not the search origin LU is a dependent LU served by a Dependent LU Server.

13.2.2.1.2 Owing CP Respond (OCR) Indicator: This is a new indicator to specify whether the DLUS or the DLUR should respond to the Locate-Find.

13.2.3 Found Resource (X'12CB') GDS Variable

13.2.3.1 Command Parameters (X'80') Found Control Vector

13.2.3.1.1 DLUS-Served LU (DSL) Indicator: This is a new indicator to specify whether or not the search destination LU is a dependent LU served by a Dependent LU Server.

13.2.3.1.2 Owing CP Respond (OCR) Indicator: This is a new indicator to specify whether the DLUS or the DLUR has responded to the Locate-Find.

13.3 Dependent LU Session Awareness Format Changes

13.3.1 SESSST (Session Started)

There is a new format, X'03', defined in byte 3 of the SESSST RU. Comparing format X'03' (which is built by the DLUR) to format X'01' (which is built by the BF-NCP),

- The DLUR will not be aware of the network address of the dependent LU's LU-LU session partner, so no session key will be included in format X'03'.
- The DLUR will not be aware of ER or VR information, so the VR-ER Mapping Data (X'1E') control vector will not be included in format X'03'.
- The DLUR will not be aware of XRF information, so the Related Session Identifier (X'28') and the XRF/Cryptography (X'68') control vectors will not be included in format X'03'.
- For network management, specifically problem determination, the DLUR will include the BIND Image (X'31') control vector.
- To return name information, the Session Information (X'2A') control vector will be included in format X'03', replacing the Local-Form Session Identifier (X'23') and the Fully-Qualified PCID (X'60') control vectors, which are already included in the X'2A' control vector (see 13.3.4, "Session Information (X'2A') Control Vector" on page 13-7).

Therefore, following the format byte, format X'03' will include only the Session Information (X'2A') and the BIND Image (X'31') control vectors.

13.3.2 SESSEND (Session Ended)

There is a new format, X'3', defined in byte 3, bits 0-3 of the SESSEND RU. Comparing format X'3' (which is built by the DLUR) to format X'2' (which is built by the BF-NCP),

- The DLUR will not be aware of the network address of the dependent LU's LU-LU session partner, so no session key will be included in format X'3'.

Therefore, following the format byte, format X'3' will include the cause byte, a retired byte, and, conditionally, the Extended Sense Data (X'35') and the Fully-Qualified PCID (X'60') control vectors.

13.3.3 RSP(ACTLU)

The RSP(ACTLU) RU, when built by a DLUR, will have several changes to it:

- The DLUR will not be aware of XRF information, so the Related Session Identifier (X'28') and the XRF/Cryptography (X'68') control vectors will not be included in the RSP(ACTLU) processed by a DLUR.
- The Session Information (X'2A') control vector built and included in the RSP(ACTLU) by the DLUR will be different (see 13.3.4, "Session Information (X'2A') Control Vector" on page 13-7 for details).

Therefore, following the FM profile, RSP(ACTLU), when processed by a DLUR, will include only the SSCP-LU Session Capabilities (X'00'), the LU-LU Session Services Capabilities (X'0C'), and the modified Session Information (X'2A') control vectors.

13.3.4 Session Information (X'2A') Control Vector

The Session Information (X'2A') control vector, when built by a DLUR, will have several changes to it:

- The DLUR will not be aware of the network address of the dependent LU's LU-LU session partner, so the Fully-Qualified Network Address Pair (X'15') control vector will not be included.
- The DLUR will not be aware of ER or VR information, so the VR-ER Mapping Data (X'1E') control vector will not be included.

Therefore, when built by a DLUR, the Session Information (X'2A') control vector will include only the Network Name (X'0E'), the Local-Form Session Identifier (X'23') and the Fully-Qualified PCID (X'60') control vectors.

13.4 Sense Codes

This section lists new sense codes only; new uses of existing sense codes also appear throughout the document.

13.4.1 ACTPU

- X'0801 0026' - The PU is not available because the Dependent LU Server-Dependent LU Requester connection could not be established.
- X'082C 0002' - Invalid Request: The specified PU has already received an ACTPU and is therefore under the control of another SSCP. This ACTPU would exceed the share limit (= 1).
- X'088E 0009' - The session could not be established because the Dependent LU Requester detected an incompatibility between its capabilities and those of its Dependent LU Server.

13.4.2 LOCATE

- X'0890 0037' - Search failure: TG vectors to the DLUR are unknown. A resubmitted Locate search for a dependent LU at its DLUR was unsuccessful. This condition arises only after the dependent LU server has verified the existence of the dependent LU; retry.
- X'8019 000C' - Routing exception: A Border Node is not in PLU's subnet when searching for DLUS-supported LU. This occurs when a DLUS node determines that the PLU node's subnet did not use a Border Node for subnet connectivity when sending out a Locate request for a DLUS-served dependent LU.

13.4.3 REQACTPU

- X'0852 0002' - A REQACTPU has been received by an SSCP which has already sent an ACTPU for the same PU.
- X'088E 0008' - The session could not be established because the Dependent LU Server detected an incompatibility between its capabilities and those of its Dependent LU Requester.

13.4.4 REQDACTPU

- X'0801 0028' - REQDACTPU received for a PU which is known but already inactive
- X'0806 0034' - REQDACTPU received for an unknown PU

13.4.5 UNBIND

- ! • X'0877 0056' - A CPSVRMGR session cannot be established over a LEN connection that is not of type TCP.
- ! • X'088E 0008' - The session could not be established because the Dependent LU Server detected an incompatibility between its capabilities and those of its Dependent LU Requester.
- ! • X'088E 0009' - The session could not be established because the Dependent LU Requester detected an incompatibility between its capabilities and those of its Dependent LU Server.
- ! • X'088E 000F' - An attempt was made to establish a CP-SVR pipe across a subnet boundary between a dependent LU server and a dependent LU requester with limited multi-subnet support.

- X'08A0 0007' - DLUS-DLUR session deactivation (disruptive): LU-LU sessions for DLUR-supported dependent LUs should be reset
- X'08A0 0008' - DLUS-DLUR session deactivation (non-disruptive): LU-LU sessions for DLUR-supported dependent LUs should not be reset
- X'08A0 0009' - DLUS-DLUR session deactivation (non-disruptive): protocol violation detected (LU-LU sessions for DLUR-supported dependent LUs should not be reset)
- X'08A0 000A' - DLUS-DLUR session deactivation (non-disruptive): DLUR should wait for DLUS reactivation of DLUS-DLUR session (LU-LU sessions for DLUR-supported dependent LUs should not be reset)

Appendix A. Definition of Terms

DLUS node. Dependent LU Server node. This is a node that is part SSCP and part APPN Network Node. It provides SSCP services to remote Secondary LUs in the form of SSCP-PU and SSCP-LU sessions which are encapsulated on an LU6.2 session pipe with some DLUR node.

DLUR node. Dependent LU Requester node. This is an APPN T2.1 node (EN or NN) that supports dependent Secondary LUs (SLUs) either

locally or directly attached by establishing an LU6.2 pipe to a DLUS node for SSCP services.

CP-SVR pipe. This term is used to describe the two LU6.2 sessions between a DLUS and DLUR node. These sessions are similar to CP-CP sessions in that each node (DLUS and DLUR) has a contention-winner and contention-loser session to the other. The CP-SVR pipe is established using a new mode=CPSVRMGR which uses the SNASVCMG COS. More information can be found in Chapter 5, "CP-SVR Pipe" on page 5-1

Appendix B. Alternate Solutions

B.1 LU-LU Session Route Calculation

As an alternative to the modifications of the associated resource hierarchy and TGVs indicated in Chapter 8, "LU-LU Session Routing" on page 8-1, the DLUS node could indicate the DLUR node to be an EN served by the DLUS node with *dummy TGVs* when the DLUR is, in fact, an NN. This would be done to avoid any incompatibilities with existing APPN PLU nodes that cannot accept a resource hierarchy of NNCP, ENCP with no TGVs. In this case, (when the DLUR node is an NN), SS at the CP(PLU) will send to TRS a Request_Route message indicating the origin node (CP(PLU)), the destination node (DLUR), and the a dummy tail vector (received on the associated resource hierarchy from the DLUS node) pointing back to the DLUR node. This dummy tail vector will specify a TG number of 1 and the partner CPname will be the DLUR node. TRS will then believe DLUR to be an EN connected to only one NN whose name happens to be the CPname of the DLUR node. This being the case, TRS will compute a route to the DLUR node (thinking it is an adjacent NN) and then use the TG number from the dummy tail vector and change the CPname to the destination node (which is also the DLUR node). So the resultant RSCV will specify a route from the origin to the DLUR node with an extra hop using TG number 1 to the DLUR node again. When the DLUR node receives such an RSCV, it will detect that the last hop represents a dummy TGV by a bit specified in the CV'4680'. It will consequently ignore this hop and route the BIND to the appropriate LU on this node.

Appendix C. Sample Network Configurations

C.1 Sample Network Configuration Number 1

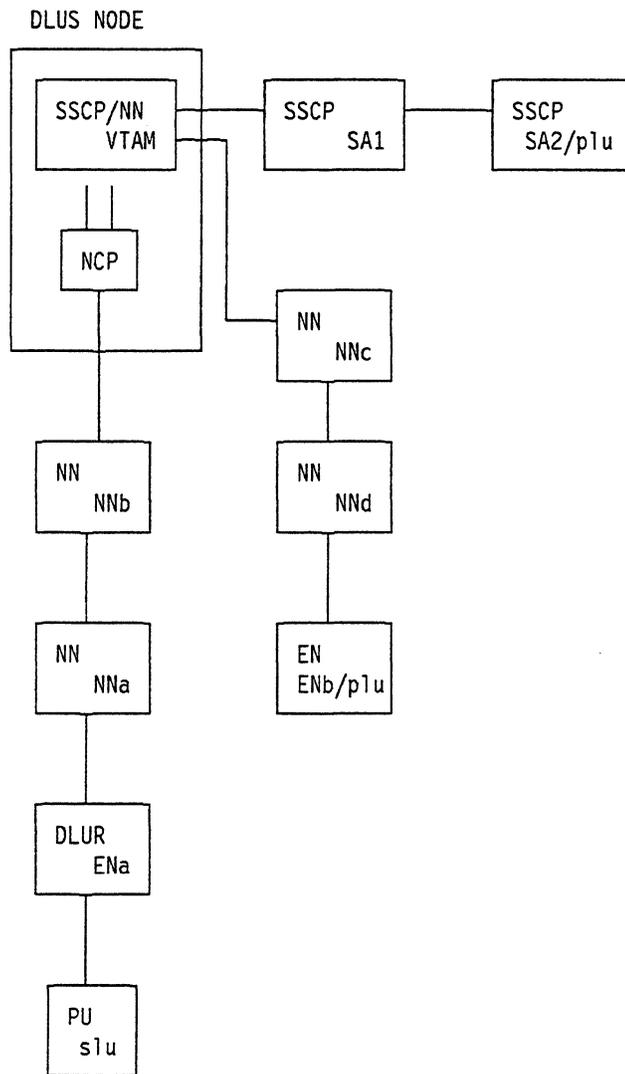


Figure C-1. Sample network configuration number 1

C.1.1 Sample Network Interpretation

In the illustration above, the node functions are listed in the upper left hand corner of the box. The node's CPname is listed in the lower right hand corner. The VTAM and NCP boxes listed in the upper left hand corner of the diagram represent a composite node image. This composite node presents a single T2.1 NN image to the APPN portion of the network and a Subarea image to the Subarea portion of the network. This node is an interchange node and is also the Dependent LU Server (DLUS) of the network. Its CPname is VTAM.

The LUs in this network have been named SLU and PLU. The SLU is located on a PU which is physically downstream from the Dependent LU Requester (DLUR) End Node whose CPname is ENa. The PLU has been recorded in two places. The PLU is located in the local domain of SSCP known as SA2. PLU is also located on End Node ENb. Although it is not correct to have two LUs with the same name in a single network, this does not present a problem in the flows described throughout the document because both SA2 and ENb are never both reflected in the network at the same time.

C.2 Sample Network Configuration Number 2

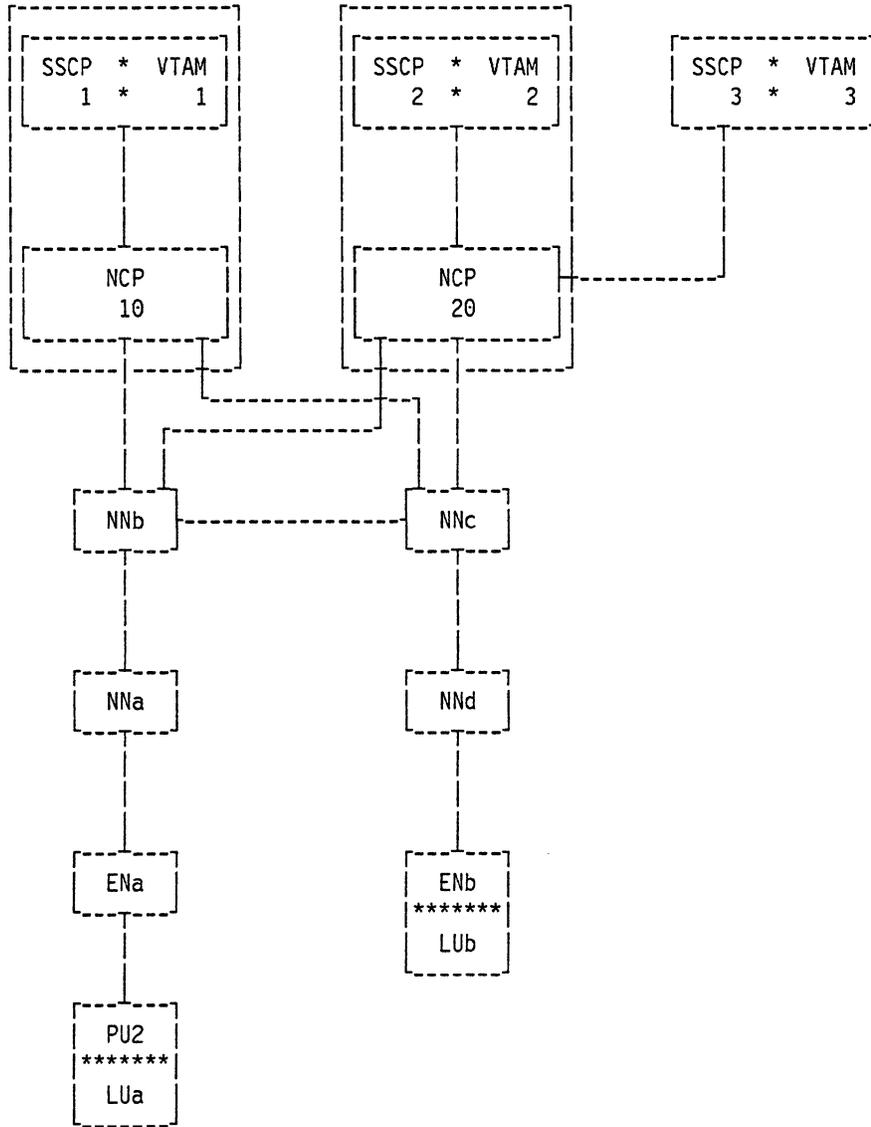


Figure C-2. Sample Network Configuration Number 2

Appendix D. Formats For DLUR/S Architecture

D.1 Formats

! Change bars indicate architecture formats changes. Some formats appear in this section without change bars; ! these formats are pertinent to DLUR/S but did not previously appear in the open SNA formats document ! (they appeared in the licensed version).

D.1.1 ACTLU

D.1.1.1 ACTLU (ACTIVATE LOGICAL UNIT)

SSCP → LU, Exp; SC

ACTLU is sent from an SSCP to an LU to activate a session between the SSCP and the LU and to establish common session parameters.

ACTLU (ACTIVATE LOGICAL UNIT)

Byte	Bit	Content
0		X'0D' request code
1		Indicators:
	0	Enhanced address management indicator: 0 sender does not support enhanced address management 1 sender supports enhanced address management
	1	Static/dynamic address indicator (reserved if byte 1, bit 0 = 0): 0 sender considers the LU address to be static 1 sender considers the LU address to be dynamic
	2-5	Reserved
	6-7	Type activation requested: 01 cold (retired) 10 ERP
2	0-3	FM profile: X'0' FM profile 0
	4-7	TS profile: X'1' TS profile 1 (only value defined)
3-n		Control vector as described in the section &CV. in -- Heading 'CF' unknown -- <i>Note:</i> The following control vector may be included; it is parsed according to sub-field parsing rule KL. X'0E' Network Name control vector (contains the LU name, type X'F3'; included if byte 1, bit 0 = 1)

D.1.2 ACTPU

D.1.2.1 ACTPU (ACTIVATE PHYSICAL UNIT)

SSCP → PU, Exp; SC

ACTPU is sent by the SSCP to activate a session with the PU, and to obtain certain information about the PU.

ACTPU (ACTIVATE PHYSICAL UNIT)

Byte	Bit	Content
0		X'11' request code
1	0-3	Format: X'0' Format 0 X'1' Format 1—same as Format 0, except that it always includes one or more control vectors in bytes 9-n (sent only to type 2.1 nodes that use XID3 with byte 10, bit 3 set to 1)
	4-7	Type activation requested: X'1' cold (retired) X'2' ERP
2	0-3	FM profile: X'0' FM profile 0
	4-7	TS profile: X'1' TS profile 1
3-8		A 6-byte field that specifies the ID of the SSCP issuing ACTPU; the first four bits specify the format for the remaining bits: 0-3 Format: 0000 (only value defined) 4-7 Type of the node containing the SSCP 8-47 Implementation and installation-dependent binary identification
9-n		Control vectors as described following this RU or in the section &CV. in -- Heading 'CF' unknown -- <i>Note:</i> The following control vectors may be included; they are parsed according to subfield parsing rule KL (see &ruspars. in -- Heading 'CF' unknown --). X'0E' Network Name control vector (contains the name of the PU, type X'F1'; optionally present when sent to a PU T2 T2.1) X'80' PU Capabilities control vector (present for Format 1 only)

D.1.2.1.1 PU Capabilities (X'80') ACTPU Control Vector**PU Capabilities (X'80') ACTPU Control Vector**

Byte	Bit	Content
0-1		Vector header; Key = X'80'

PU Capabilities (X'80') ACTPU Control Vector

Byte	Bit	Content
2	0	<u>Vector Data</u> Unsolicited NMVT support: 0 Sending node does not support unsolicited NMVTs for PSID. 1 Sending node supports unsolicited NMVTs for PSID.
	1-7	Reserved

D.1.3 REQACTPU

D.1.3.1 REQACTPU (REQUEST ACTIVATE PHYSICAL UNIT)

PU → SSCP, Norm; FMD NS(c)

REQACTPU is sent from the PU to an SSCP to request that ACTPU be sent to the PU named in the RU or corresponding GDS variable.

REQACTPU (REQUEST ACTIVATE PHYSICAL UNIT)

Byte	Bit	Content
0-2		X'41023E' NS header
3-4		Reserved
5	0-3	Format: X'0' Format 0: external PU X'1' Format 1: internal PU
	4-7	Reserved

D.1.4 REQDACTPU

D.1.4.1 REQDACTPU (REQUEST DEACTIVATE PHYSICAL UNIT)

PU → SSCP, Norm; FMD NS(c)

REQDACTPU is sent from the PU to an SSCP to request that DACTPU be sent to the PU.

REQDACTPU (REQUEST DEACTIVATE PHYSICAL UNIT)

Byte	Bit	Content
0-2		X'41023F' NS header

REQDACTPU (REQUEST DEACTIVATE PHYSICAL UNIT)

Byte	Bit	Content
3-4		Reserved
5	0-3	Format: X'0' Format 0 (only value defined)
	4-7	Reserved
		<u>Format 0</u>
6		Cause: X'00' Loss of connectivity between the Dependent LU Requester and its serviced PU X'01' Dependent LU Requester received REQDISCONT(normal) from its serviced PU X'02' Dependent LU Requester received REQDISCONT(immediate) from its serviced PU

D.1.5 REQDISCONT**D.1.5.1 REQDISCONT (REQUEST DISCONTACT)**

PU T1|2 → SSCP, Norm; FMD NS(c)

With REQDISCONT, the PU T2 requests the SSCP to start a procedure that will ultimately discontact the secondary station in the T2 node.

REQDISCONT (REQUEST DISCONTACT)

Byte	Bit	Content
0-2		X'01021B' NS header
3	0-3	Type: X'0' normal X'8' immediate
	4-7	CONTACT information: X'0' do not send CONTACT immediately X'1' send CONTACT immediately <i>Note:</i> Bits 4-7 are reserved for switched connections.

D.1.6 SESSEND

D.1.6.1 SESSEND (SESSION ENDED)

LU → SSCP, Norm; FMD NS(s)

SESEND is sent, with no-response requested, by the LU (or boundary function on behalf of the LU in a peripheral node) to notify the SSCP that the session between the specified LUs has been successfully deactivated.

SESEND (SESSION ENDED)

Byte	Bit	Content
0-2		X'810688' NS header
3	0-3	Format: X'3' format 3
	4-7	Reserved
		<u>Format 3</u>
4		Cause: indicates the reason for the deactivation of the LU-LU session (see UNBIND for values)
5		Retired
6-n		Control vectors as described in the section &CV. in -- Heading 'CF' unknown -- <i>Note:</i> The following control vectors may be included; they are parsed according to subfield parsing rule KL. X'35' Extended Sense Data control vector (conditionally present) X'60' Fully-qualified PCID control vector (conditionally present)

D.1.7 SESSST**D.1.7.1 SESSST (SESSION STARTED)**

LU → SSCP, Norm; FMD NS(s)

SESSST is sent, with no response requested, to notify the SSCP that the session between the specified LUs has been successfully activated.

SESSST (SESSION STARTED)

Byte	Bit	Content
0-2		X'810686' NS header
3		Format: X'03' Format 3: control vectors present in bytes 4-n
		<u>Format 3</u>

SESSST (SESSION STARTED)

Byte	Bit	Content
4-n		Control vectors as described in the section &CV. in -- Heading 'CF' unknown -- <i>Note:</i> The following control vectors may be included; they are parsed according to subfield parsing rule KL. X'2A' Session Information control vector X'31' BIND Image control vector

D.1.8 RSP(ACTLU)**D.1.8.1 RSP(ACTLU)**

LU → SSCP, Exp; SC

RSP(ACTLU)

Byte	Bit	Content
0		X'0D' request code
1		Type of activation selected: X'01' cold (retired) X'02' ERP
2	0-3 4-7	FM profile: Same as the corresponding request TS profile: same as the corresponding request

Note: Two versions of this RU are defined.*A full response can be sent in which bytes 0-m are present.*

3-m		Control vectors <i>Note:</i> The following control vectors may be included; they are parsed according to subfield parsing rule KL. X'00' SSCP-LU Session Capabilities control vector (present to override the defaults of a 2-byte response, in which case always first) X'0C' LU-LU Session Services Capabilities control vector (present to override the defaults of a 2-byte response, in which case always second) <i>Note:</i> The following control vector is appended by the Dependent LU Requester X'2A' Session Information control vector (present when the LU already has an active session)
-----	--	--

*A two-byte response may be sent; it means maximum RU size = 256 bytes, LU-LU session limit = 1, the LU can act as a secondary LU, and all other fields in control vectors X'00' and X'0C' are defaulted to 0's.***D.1.9 LU-LU Session Services Capabilities (X'0C') Control Vector**

D.1.9.1 LU-LU Session Services Capabilities (X'0C') Control Vector

Note: Do not confuse control vector X'0C' with NOTIFY vector X'0C', which carries similar information.

LU-LU Session Services Capabilities (X'0C') Control Vector

Byte	Bit	Content
0-1		Vector header; Key = X'0C' (see -- Heading 'RUSPARS' unknown -- and -- Table 'CV' unknown --)
2-m		<u>Vector Data</u>
2	0-3	Primary LU capability (used between a subarea LU and its SSCP; also used between a peripheral LU and its SSCP if the BF(LU) supports the receipt of CINIT; otherwise, reserved): 0000 PLU capability is inhibited: Sessions can be neither queued nor started. 0001 PLU capability is disabled: Sessions can be queued but not started. 0010 reserved 0011 PLU capability is enabled: Sessions can be queued or started.
	4-7	Secondary LU capability: 0000 SLU capability is inhibited: Sessions can be neither queued nor started. 0001 SLU capability is disabled: Sessions can be queued but not started. 0010 reserved 0011 SLU capability is enabled: Sessions can be queued or started.
3-4		LU-LU session limit: 0000 no session limit specified (only value allowed for subarea LUs) 0001 session limit of 1 (only value allowed for peripheral LUs)
5-6		LU-LU session count: the number of LU-LU sessions that are not reset for this LU and for which SESSEND will be sent to the SSCP <i>Note:</i> For XRF, this field applies to only the XRF-active session. XRF-backup sessions are not included in this count.

LU-LU Session Services Capabilities (X'0C') Control Vector

Byte	Bit	Content
7	0	Parallel session support:
		0 parallel sessions not supported 1 parallel sessions supported
	1	NOTIFY support:
		0 NOTIFY not supported 1 NOTIFY supported
	2	SESSST capability (used between a subarea or dependent LU or BF and its SSCP; otherwise, reserved):
		0 SESSST RU is suppressed if SLU 1 SESSST RU is sent if SLU
	3	XRF Session Activation (X'27') control vector support (used between a BF and its SSCP; otherwise, reserved):
		0 XRF Session Activation (X'27') control vector not supported on BIND 1 XRF Session Activation (X'27') control vector supported on BIND
	4	Peripheral node extended BIND support indicator (used between a dependent LU and its BF; otherwise, reserved):
		0 Dependent LU does not support receipt of extended BINDs. 1 Dependent LU does support receipt of extended BINDs.
	5	Network-qualified names support indicator (used between an LU and its SSCP; passed through the BF to the SSCP if sent by a peripheral LU):
		0 A BIND received by this LU cannot contain network-qualified LU names in bytes k + 2-m and p + 2-r.
		1 A BIND received by this LU can contain network-qualified LU names in bytes k + 2-m and p + 2-r.
	6	Subarea node extended BIND support indicator (used between a subarea LU or BF and its SSCP; otherwise, reserved):
0 Subarea LU or BF(LU) does not support sending and receiving extended BIND.		
1 Subarea LU or BF(LU) does support sending and receiving extended BIND.		
7	Retired (set to 0)	
8-15		Retired (set to X'4040404040404040')
16(=m)	0	Unrecognized-control-vectors-on-CINIT support indicator:
		0 CINIT containing unrecognized control vectors will be rejected. 1 CINIT containing unrecognized control vectors will be accepted.
	1-7	Reserved

D.1.10 Network Name (X'0E') Control Vector**D.1.10.1 Network Name (X'0E') Control Vector**

Network Name (X'0E') Control Vector

Byte	Bit	Content
0-1		Vector header; Key = X'0E' <i>Note:</i> A null X'0E' control vector consists of a vector header with no vector data. The length field is set appropriately.
2-n		<u>Vector Data</u>
2		Network name type: X'F1' PU name X'F3' LU name X'F4' CP name (see Note) X'F5' SSCP name X'F6' NNCP name X'F7' link station name (not network-qualified) X'F8' CP name of CP(PLU) X'F9' CP name of CP(SLU) <i>Note:</i> When this control vector is carried in some message units, such as XID3 or BIND, X'F4' means simply "CP name," without specifying the CP type (e.g., EN or NN), and X'F6' is not used; see each individual message-unit structure in which this control vector appears for details on such usage.
3-n		Network-qualified name: a 1 to 17-byte name consisting of an optional qualifier concatenated to a 1 to 8-byte type-1134 symbol-string name; when present, the qualifier contains a 1 to 8-byte type 1134-symbol-string network identifier concatenated with a period (when the qualifier is not present, the period is omitted). The network-qualified name appears, for example, as follows: NETID.NAME, with optional (but not significant) trailing, but no imbedded, space (X'40') characters. Implementation usage constrains the leading character of the name to be alphabetic. <i>Note:</i> When this control vector is carried in an RSCV, the net ID may be "trimmed" from this field in contiguous control vectors <i>after</i> the first in a series pertaining to the same net ID subnetwork (to conserve RSCV total length).

D.1.11 Local-Form Session Identifier (X'23') Control Vector

D.1.11.1 Local-Form Session Identifier (X'23') Control Vector

Local-Form Session Identifier (X'23') Control Vector

Byte	Bit	Content
0-1		Vector header; Key = X'23' (see -- Heading 'RUSPARS' unknown -- and -- Table 'CV' unknown --)
2-n		<u>Vector Data</u>
2		Format: X'02' Format 2: FID2 session identifier

Local-Form Session Identifier (X'23') Control Vector

Byte	Bit	Content
<i>For format 2—FID2</i>		
3–n		Session identifier (SID) for Format 2—FID2
3		High-order byte of session identifier (SIDH): OAF' in the TH of the BIND request
4		Low-order byte of session identifier (SIDL): DAF' in the TH of the BIND request
5(=n)		Flags:
	0–5	Reserved
	6	ODAI field from TH of the BIND request
	7	Reserved

D.1.12 Security ID Control (X'25') Control Vector**D.1.12.1 Security ID Control (X'25') Control Vector**

The Security ID control vector is available for sending identifier information for a switched call to the SSCP. The information may be provided by the switched telephone or X.25 network.

Security ID Control (X'25') Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X'25' (see -- Heading 'RUSPARS' unknown -- and -- Table 'CV' unknown --)
2–q		<u>Vector Data</u>
2		Reserved
3–q		<u>Security ID</u>
3		Length (q – 4, in binary, of security ID)
4–q		Security ID: a string of EBCDIC characters <i>Note:</i> In the X.25 environment, the security ID is the calling DTE address provided by the network, once converted from packed digits to EBCDIC characters.

D.1.13 Session Information (X'2A') Control Vector

D.1.13.1 Session Information (X'2A') Control Vector

Session Information is included, once for each session, on RSP(ACTLU) and SESSST to report the existence of a dependent LU's session to an SSCP taking over control of the LU.

Session Information (X'2A') Control Vector

Byte	Bit	Content
0-1		Vector header; Key=X'2A' (see -- Heading 'RUSPARS' unknown -- and -- Table 'CV' unknown --)
2-n		<u>Vector Data</u>
2	0	LU role: 0 The subject LU is the SLU in this session. 1 The subject LU is the PLU in this session.
	1-7	Reserved
3-n		Control vectors, as described in the &CV. discussion in -- Heading 'CF' unknown --
		<i>Note:</i> The following control vectors may be included; they are parsed according to parsing rule KL.
		X'0E' Network Name control vector: the network-qualified name of the partner LU for the reported session (always present)
		X'15' Fully Qualified Network Address Pair control vector: PLU and SLU, respectively (present except for an LU attached to a Dependent LU Requester)
		X'1E' VR-ER Mapping Data control vector (present except for an LU attached to a Dependent LU Requester)
		X'23' Local-Form Session Identifier control vector
		X'60' Fully Qualified PCID control vector (present, when present in the corresponding BIND)

D.1.14 Assign LU Characteristics (X'30') Control Vector

D.1.14.1 Assign LU Characteristics (X'30') Control Vector

Assign LU Characteristics (X'30') Control Vector

Byte	Bit	Content
0-1		Vector header; Key=X'30' (see -- Heading 'RUSPARS' unknown -- and -- Table 'CV' unknown --)
2-n		<u>Vector Data</u>

Assign LU Characteristics (X'30') Control Vector

Byte	Bit	Content
2-3		Reserved session resources: the number, in binary, of session resources the boundary function is to reserve for the use of the LU identified in the RU
4	0	REX stage pacing parameters: 0 adaptive session pacing allowed on the REX stage 1 fixed session pacing is to be used on REX stage
	1	Reserved
	2-7	Pacing window size. This value is used as the pacing window size for fixed pacing if bit 0 of this byte is ON, or it is used as the batch mode pacing window size if bit 0 of this byte is OFF.
5(=n)	0	Subarea stage pacing parameters: 0 adaptive session pacing allowed on the subarea stage 1 fixed session pacing is to be used on subarea stage
	1	Reserved
	2-7	Pacing window size. This value is used as the pacing window size for fixed pacing if bit 0 of this byte is ON, or it is used as the batch mode pacing window size if bit 0 of this byte is OFF.
6-7		Maximum number of simultaneous LU-LU sessions in which this LU can participate. If this value is zero, then the node receiving this control vector X'30' should use the locally defined default value.

D.1.15 BIND Image (X'31') Control Vector**D.1.15.1 BIND Image (X'31') Control Vector****BIND Image (X'31') Control Vector**

Byte	Bit	Content
0-1		Vector header; Key = X'31' (see -- Heading 'RUSPARS' unknown -- and -- Table 'CV' unknown --)
2-n		<u>Vector Data</u>
2-n		Bytes 1-r (i.e., through the SLU Name field) of the BIND image for the session being activated
		<i>Note:</i> This control vector normally contains bytes 1-r of BIND, but the number of bytes present may be qualified in the specific descriptions of the RUs or GDS variables that carry it.

D.1.16 Extended SDLC Secondary Station (X'43') Control Vector

D.1.16.1 Extended SDLC Station (X'43') Control Vector
--

Extended SDLC Station (X'43') Control Vector

Byte	Bit	Content
0-1		Vector header; Key = X'43' (see -- Heading 'RUSPARS' unknown -- and -- Table 'CV' unknown --)
2-n		<u>Vector Data</u>
2	0-4	Reserved
	5-7	Node type identifier: 100 T1 node 010 T2 node 001 T4 5 node
3		Type modifier:
	0	TS profile usage (reserved except when byte 2 indicates T1 node): 0 - TS profile 2 1 TS profile 2
	1	0 Discontinue link-level contact with adjacent T1 2 node if the subarea node initiates an auto network shutdown procedure for the SSCP controlling that T1 2 node or, in the case of a switched subarea link, discontinue link-level contact with adjacent T4 5 node if auto network shutdown procedure is initiated. 1 Continue link-level contact with adjacent T1 2 node if the subarea node initiates an auto network shutdown procedure for the SSCP controlling that T1 2 node or, in the case of a switched subarea link, continue link-level contact with adjacent T4 5 node if an auto network shutdown procedure is initiated.
	2	0 Use SNRM polling for the secondary station. 1 Use null XID polling for the secondary station.
	3	0 Modem test support for the secondary station is as specified during system definition for the link. 1 Modem test is not supported for the secondary station.
	4	Data mode: 0 half-duplex 1 full-duplex
	5	LPDA-2 Information field (bytes 17-18) validity indicator: 0 Bytes 17-18 are valid. 1 Bytes 17-18 are to be ignored.
	6	Retry Control Information field (bytes 7-8) validity indicator: 0 Bytes 7-8 are to be ignored. 1 Bytes 7-8 are valid.
	7	Reserved

Extended SDLC Station (X'43') Control Vector

Byte	Bit	Content
4		Frame send control value: <ul style="list-style-type: none"> • For SDLC, the maximum number of frames (or BTUs) that the node can send before it must receive an acknowledgment from its partner link station. This value has an implied modulus for the SDLC send and receive counts: less than 8 implies a modulus of 8, while 8 or greater implies a modulus of 128. • For frame-relay switching equipment (FRSE), the maximum send queue size that, when reached, causes the node to discard frames, without forwarding, until room once again exists in the queue. Reaching half this maximum queue size causes the node to set the appropriate congestion indicators in available frames to inform its partner FRSEs of the congestion condition.
5		Maximum consecutive BTUs sent from the primary station to the specified secondary station without another secondary station on the link being polled or being sent BTUs
6		Error retry indicator: X'00' no immediate retry X'10' immediate retry
7–8		<u>Retry Control Information</u>
7		Length of pause between retry sequences
8		Maximum number of retry sequences
9–10		Byte count of maximum BTU size permitted to be sent to the adjacent link station represented by the specified secondary station
11–14		<u>Threshold Control Information</u>
11–12		Total number of transmissions
13–14		Total number of error retries
15–16		Average number of bytes expected when the station is polled
17–18		<u>LPDA-2 Information</u>
17		Number (1 or 2), in binary, of link connection segments
18		<u>Local Modem Addresses</u>
	0–3	Local modem address of link segment 1: X'0'–X'F' are valid addresses
	4–7	Local modem address of link segment 2: X'0'–X'F' are valid addresses; a value of 0 is used if only one link segment exists
19		3174 group poll address (present when byte 5, bit 4 in control vector X'0B' was set by the T4 node and the SSCP has APPN capability; otherwise not present): nonzero value to identify a group poll address; 0 if no group poll address exists
<i>Note:</i>		When this control vector flows on an RNAA, it may be abbreviated (indicated by the length field); in this case, bytes 7–8 are reserved, and byte 19 or bytes 11–19 are omitted. When this control vector is sent on a SETCV, byte 3, bits 2–7, and bytes 5–8 are reserved.

D.1.17 TG Descriptor (X'46') Control Vector

D.1.17.1 TG Descriptor (X'46') TDU Control Vector

The TG Descriptor control vector identifies a transmission group (TG). All implementations, as a base requirement, always accept and forward the TG Descriptor (X'46') control vector in its entirety.

TG Descriptor (X'46') TDU Control Vector

Byte	Bit	Content
0-1		Vector header; Key = X'46' (see -- Heading 'RUSPARS' unknown -- and -- Table 'CV' unknown --)
2-n		<u>Vector Data</u> <i>Note:</i> The following subfields (described following this control vector) may be included. They are parsed according to subfield parsing rule LT.
	X'80'	TG Identifier subfield (always present except when the control vector is included in a FID2 Encapsulation GDS variable)
	X'82'	DLC Signaling Information subfield (present only when the X'80' subfield indicates the TG is to a link connection network, except not present on XID3)
	X'83'	Real Partner CP name subfield (present in a CD-Initiate in a Locate reply [or request] when a border node modifies the associated resource hierarchy such that the CP(DLU) [or CP(OLU)] is not adjacent to the NNS(DLU) [or NNS(OLU)]; or in an RSCV when an NNS(OLU) calculates a route that includes a TG vector carrying it): when present, used in preference to the TG-partner node's CP name in the TG Identifier (X'80') subfield
	X'91'	DLC Signaling Type Information subfield (present when ...)

D.1.17.1.1 DLC Signaling Type (X'91') TG Descriptor Subfield

DLC Signaling Type (X'91') TG Descriptor Subfield

Byte	Bit	Content
0-1		Subfield Header; Key = X'91' Length, in binary, of DLC Signaling Type subfield
2-n		<u>Subfield Data</u>

DLC Signaling Type (X'91') TG Descriptor Subfield

Byte	Bit	Content
2-n		DLC type (EBCDIC characters):
		DLSW Data Link Switching
		ETHERNET Ethernet
		FDDI FDDI
		FRELAY Frame Relay
		INTPU Internal PU
		SDLCNS SDLC non-switched
		SDLCSW SDLC switched
		TR Token Ring
		X25 X.25

D.1.17.1.2 DLC Signaling Information (X'92') TG Descriptor Subfield

DLC Signaling Information (X'92') TG Descriptor Subfield

Byte	Bit	Content
0-1		Subfield Header; Key = X'92' Length, in binary, of DLC Signaling Information subfield (see -- Heading 'RUSPARS' unknown --)
2-n		<u>Subfield Data</u>
<i>For DLC Signaling Type = INTPU:</i>		
2-5(=n)		Internal PU Identifier:
	0-11	Block number
	12-31	ID number

D.1.17.1.3 DLC Signaling Information (X'93') TG Descriptor Subfield

DLC Signaling Information (X'93') TG Descriptor Subfield

Byte	Bit	Content
0-1		Subfield Header; Key = X'93' Length, in binary, of DLC Signaling Information subfield (see -- Heading 'RUSPARS' unknown --)
2-n		<u>Subfield Data</u>
<i>For DLC Signaling Type = INTPU:</i>		
2-n		DLUR Network Qualified CP name (up to 17 characters)

D.1.18 DLUR/S Capabilities (X'51') Control Vector

D.1.18.1 DLUR/S Capabilities (X'51') Control Vector

DLUR/S Capabilities control vector is exchanged by the Dependent LU Requester and the Dependent LU Server to describe their DLUR/S capabilities once a DLUS-DLUR session is activated.

DLUR/S Capabilities (X'51') Control Vector

Byte	Bit	Content
0-1		Vector header; Key = X'51'
2-n		<u>Vector Data</u>
2		Dependent LU support level: X'01' Release 1
3		Node type: X'00' Network Node X'01' End Node
4		DLUR/S node type: X'00' DLUR X'01' DLUS
5-8		Flow reduction sequence number: a value that identifies the latest Topology Database Update GDS variable received by the sender of the DLUR/S Capabilities control vector from the receiver of the DLUR/S Capabilities control vector
9	0	Support indicators (bit is set to 1 if the sender supports the function): RECEIVE_TDU_TP (X'22F0F0F4') service transaction program supported: The sending CP supports receipt of TDUs on this session
	1	Limited DLUR ANS support: The DLUR only supports ANS = STOP
	2	Limited DLUR multi-subnet support: The DLUR will not respond to dependent LU searches with the OCR bit set
	3-7	Reserved
10-13		Reserved

D.1.19 Locate (X'12C4') GDS Variable

D.1.19.1 Locate (X'12C4') GDS Variable

The Locate (X'12C4') GDS variable is used in conjunction with other GDS variables by the SEND_ and RECEIVE_NETWORK_SEARCH service transaction programs.

Locate (X'12C4') GDS Variable

Byte	Bit	Content
0-1		Length (n + 1), in binary, of the GDS variable, including the Length field
2-3		Key: X'12C4'
4		<u>GDS Variable Data</u>
	0	Locate chain indicator: 0 discard 1 keep
	1-3	Request-reply status (r = reserved): 00r request 01r incomplete reply (sent or received only by NNCPs) a complete reply (bits 1-3 = 10r 11r) will follow 10r complete reply 11r complete reply, but eligible resources may exist that could not be located because of an outage on the search route
	4	Locate chain keep support (reserved except on a directed Locate request, i.e., except when bits 1-3 = 000, control vector X'80' is present, and control vector X'2B' is not present): set to 1 by the NNS(OLU) if it can keep Locate chains after processing their replies; any node (i.e., a back-level implementation) along the directed search path that cannot support keeping Locate chains in this way sets this field to 0. Once set to 0, this field remains 0 for the remainder of the request path. 0 not supported (i.e., byte 4, bit 0 must be set to 0 on the reply) 1 supported (byte 4, bit 0 may be set to 0 or 1 according to need)
	5-7	Reserved
5-6		Retired
7-8		Search number (sent or received only by NNCPs; otherwise, reserved): a binary value used as a secondary key, in conjunction with the Fully Qualified PCID (X'60') and PCID Modifier (X'81') control vectors, to uniquely identify a search subprocedure (control block) of a Locate procedure; echoed in the search reply

Locate (X'12C4') GDS Variable

Byte	Bit	Content
9-n		Control vectors <i>Note:</i> The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT. X'0E' Network Name control vector: name of the destination control point (present in a request when the destination network name is known) X'2B' Route Selection control vector (present on a directed Locate search request exchanged between NNCPs to specify the CPs along a directed Locate procedure path) X'35' Extended Sense Data control vector (present on a reply to indicate a Locate error) X'60' Fully Qualified PCID control vector (always present) X'80' Search Scope control vector (present between NNCPs to define the scope of a broadcast search request) X'81' PCID Modifier control vector (always present on searches initiated at the current level NNS) X'82' Intersubnetwork Search control vector (present when the Locate is being sent over an intersubnetwork TG)

D.1.19.2 Locate Control Vectors

D.1.19.2.1 Search Scope (X'80') Locate Control Vector

Search Scope (X'80') Locate Control Vector

Byte	Bit	Content
0-1		Vector header; Key = X'80'
2-n		<u>Vector Data</u>
2		Hop count: a binary value specifying the number of hops that may be traversed for a broadcast search (set by the broadcast origin CP and decremented, on the search request, by intermediate CPs participating in the broadcast search)

D.1.19.2.2 PCID Modifier (X'81') Locate Control Vector

PCID Modifier (X'81') Locate Control Vector

Byte	Bit	Content
0-1		Vector header; Key = X'81'

PCID Modifier (X'81') Locate Control Vector

Byte	Bit	Content
2-n		<u>Vector Data</u>
2-3		Procedure Resubmit Number
4		Last significant half-byte in the PCID Modifier List field (zero-origin index to the last half-byte in the list that has been claimed by a node on the procedure path)
5-n		PCID modifier list <i>Note:</i> The PCID Modifier List provides up to 20 half-bytes of list entries, with each list entry containing a count of the number of new search subprocedures created by a particular node. The Last Significant Half-byte field is used to determine the next available half-byte in the PCID Modifier List. The PCID modifier list has a minimum length of 1 byte, and must always contain an integral number of bytes. Thus, if byte 4 specifies an odd number of half-bytes (i.e., byte 4 is even), then the last byte in the list is padded with X'0' to form a full byte. A node must be able to receive at least twenty entries in the list.

D.1.19.2.3 Intersubnetwork Search (X'82') Locate Control Vector

Intersubnetwork Search (X'82') Locate Control Vector

Byte	Bit	Content
0-1		Vector header; Key = X'82'
2-n		<u>Vector Data</u>
2		Subnetwork controls:
	0	Search scope indicator: 0 Search may span subnetwork boundaries. 1 Search may not span subnetwork boundaries.
	1	Adjacent subnetwork search indicator 0 Do not limit search to adjacent subnetwork. 1 Limit search to adjacent subnetwork.
	2-3	Reserved
	4	Prevent Subarea Path indicator 0 Do not prevent the BN from resetting the Suppress Subarea Search bit 1 Prevent the BN from resetting the Suppress Subarea Search bit
	5	DLUR Search Required indicator 0 A search to obtain the DLUR TGVs is not required 1 A search to obtain the DLUR TGVs is required
	6-7	Reserved

D.1.20 Find Resource (X'12CA') GDS Variable

D.1.20.1 Find Resource (X'12CA') GDS Variable

The Find Resource (X'12CA') GDS variable is used to request a node to search its directory for the search arguments provided.

Find Resource (X'12CA') GDS Variable

Byte	Bit	Content
0-1		Length (n + 1), in binary, of the GDS variable, including the Length field
2-3		Key: X'12CA'
4-n		<u>GDS Variable Data</u>
4-n		Control vectors, as described in D.1.20.2, "Find Control Vectors" on page D-22. <i>Note:</i> The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT.
	X'80'	Command Parameters (X'80') Find control vector (always present, always first)
	X'3C'	Associated Resource Entry (X'3C') control vector: used to identify the search origin end node CP or network node server CP information to be saved at the search destination (e.g., in the server's directory as a cache entry); hierarchical associations are indicated by the order of X'3C' and X'3D' control vectors, those appearing first being hierarchically above those that follow (an EN(OLU) generates one for its own CP; an EN(DLU) receives one for the OLU's network node server CP and, if the OLU resides in an end node, one for the OLU's ENCP)
	X'3D'	Directory Entry (X'3D') control vector: provides information about the search origin (always present)
	X'40'	Real Associated Resource Entry (X'40') control vector: used to identify the name of the real associated resource of the resource identified in the Directory Entry (X'3D') control vector (present only when an Associated Resource Entry (X'3C') control vector in the hierarchy does not represent the real hierarchical superior of the target resource but rather is a surrogate)
	X'82'	Search Argument Directory Entry (X'82') Find control vector: used to specify the search argument directory entry (always present)

D.1.20.2 Find Control Vectors

D.1.20.2.1 Command Parameters (X'80') Find Control Vector

Command Parameters (X'80') Find Control Vector

Byte	Bit	Content
0-1		Vector header; Key = X'80'
2-m		<u>Vector Data</u>
2	0	Origin information present indicator: 1 present (only value defined)
	1-7	Reserved
3	0-1	Reserved
	2	DLUS-served LU indicator: 0 the LU is not served by a Dependent LU Server. 1 the LU is served by a Dependent LU Server.
	3-5	Reserved
	6	Owning CP Respond Indicator: 0 The DLUS node owning the DLUS served LU should reply to the Locate-Find. 1 The DLUR node owning the DLUS served LU should reply to the Locate-Find.
4	7	Surrogate resource owner indicator (reserved for end nodes): 0 This Find is from the real resource owner. 1 This Find may be from a surrogate resource owner.
	0-1	Reserved
	2	Net ID authenticity indicator: 0 Target resource Net ID is authentic. 1 Target resource Net ID is not authentic.
	3-7	Reserved

D.1.20.2.2 Search Argument Directory Entry (X'82') Find Control Vector

Search Argument Directory Entry (X'82') Find Control Vector

Byte	Bit	Content
0-1		Vector header; Key = X'82'
2-m		<u>Vector Data</u>

Search Argument Directory Entry (X'82') Find Control Vector

Byte	Bit	Content
2-3		Resource type: X'00F3' logical unit X'00F4' ENCP X'00F6' NNCP
4-m		Resource name: a 1- to 17-byte name consisting of an optional qualifier concatenated to a 1- to 8-byte type-1134 symbol-string name; when present, the qualifier contains a 1- to 8-byte type-1134 symbol-string network ID concatenated with a period (which is omitted if the network ID is omitted) <i>Note:</i> The network ID is always present when different from the network ID of the receiver.

D.1.21 Found Resource (X'12CB') GDS Variable**D.1.21.1 Found Resource (X'12CB') GDS Variable**

The Found Resource (X'12CB') GDS variable is a positive reply to a Find Resource (X'12CA') GDS variable; it provides the requested data.

Found Resource (X'12CB') GDS Variable

Byte	Bit	Content
0-1		Length (n + 1), in binary, of the GDS variable, including the Length field
2-3		Key: X'12CB'
4-n		<u>GDS Variable Data</u>
4-n		Control vectors, as described in D.1.21.2, "Found Control Vectors" on page D-24. <i>Note:</i> The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT. X'80' Command Parameters (X'80') Found control vector (always present, always first) X'3C' Associated Resource Entry (X'3C') control vector: used to identify the search destination end node CP or network node server CP to be saved at the search origin (e.g., in the server's directory as a cache entry); hierarchical associations are indicated by the order of X'3C' and X'3D' control vectors, those appearing first being hierarchically above those that follow (an EN(DLU) generates one for its own CP; an EN(OLU) receives one for the DLU's network node server CP and, if the DLU resides in an end node, one for the DLU's ENCP) X'3D' Directory Entry (X'3D') control vector: identifies the requested destination directory entry (always present)

Found Resource (X'12CB') GDS Variable

Byte	Bit	Content
		X'40' Real Associated Resource Entry (X'40') control vector: used to identify the name of the real associated resource of the resource identified in the Directory Entry (X'3D') control vector (present only when an Associated Resource Entry (X'3C') control vector in the hierarchy does not represent the real hierarchical superior of the target resource but rather is a surrogate); multiple Real Associated Resource (X'40') control vectors are arranged in hierarchical order, with an earlier-appearing one being hierarchically above one that follows.

D.1.21.2 Found Control Vectors**D.1.21.2.1 Command Parameters (X'80') Found Control Vector****Command Parameters (X'80') Found Control Vector**

Byte	Bit	Content
0-1		Vector header; Key = X'80'
2-m		<u>Vector Data</u>
2	0	Target information present indicator: 1 present (only value defined)
	1	Reserved
	2	DLUS served LU indicator: 0 the LU is not served by a Dependent LU Server. 1 the LU is served by a Dependent LU Server.
	3-5	Reserved
	6	Owning CP Respond Indicator: 0 The DLUS node owning the DLUS served LU has responded to the Locate-Find. 1 The DLUR node owning the DLUS served LU has responded to the Locate-Find.
	7	Wild-card directory entry: 0 The directory entry for this located resource is an explicit or partially specified name. 1 The directory entry for this located resource is a wild-card entry.
3(=m)	0-6	Reserved
	7	Surrogate resource owner indicator: 0 This Found is from the real resource owner. 1 This Found may be from a surrogate resource owner.

D.1.22 FID2 Encapsulation (X'1500') GDS Variable

D.1.22.1 FID2 Encapsulation (X'1500') GDS Variable

The FID2 Encapsulation GDS variable transports control session information and data between an SSCP and a dependent LU or its associated PU.

FID2 Encapsulation (X'1500') GDS Variable

Byte	Bit	Content
0-1		Length (n + 1), in binary, of the GDS variable, including the Length field
2-3		Key: X'1500'
4-n		<u>GDS Variable Data</u>
4-5		Length (m-3), in binary, of the FID2 PIU, including this Length field
6-m		FID2 PIU
m+1-n		Control vectors <i>Note:</i> The following control vectors may be included; they are parsed according to subfield parsing rule LT. X'25' Security ID Control control vector (conditionally present) X'30' Assign LU Characteristics control vector (conditionally present) X'43' Extended SDLC Station control vector (conditionally present) X'46' TG Descriptor control vector (conditionally present) X'51' DLUR/S Capabilities control vector (conditionally present) X'60' Fully Qualified PCID control vector (always present) X'81' XID Image control vector (conditionally present)

D.1.22.2 FID2 Encapsulation Control Vectors

D.1.22.2.1 XID Image (X'81') FID2 Encapsulation Control Vector

XID Image (X'81') FID2 Encapsulation Control Vector

Byte	Bit	Content
0-1		Vector Header; Key = X'81'
2-n		XID I-field image: the bytes received in the information field of the SDLC XID response

D.2 Sense Codes

D.2.1 0801

0801 Resource Not Available: The LU, PU, link station, channel link, or link specified in an RU is not available.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

| 0026 The PU is not available because the Dependent LU Server-Dependent LU
| Requester connection could not be established.

| 0028 REQDACTPU received for a PU which is known but already inactive.

D.2.2 0806

0806 Resource Unknown: For example, the request contained a name or address not identifying a PU, LU, SSCP, link, or link station known to the receiver or the sender.

Note: In an interconnected network environment, this sense code may be set by an SSCP in whose subnetwork and domain the LU was expected to reside; it is not set by an SSCP that is only an intermediary on the session-setup path. A gateway SSCP examines the Resource Identifier control vector in a session setup request (for example, CDINIT), to determine whether the LU is in the SSCP's subnetwork and domain.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

| 0034 REQDACTPU received for an unknown PU.

D.2.3 082C

082C Resource-Sharing Limit Reached: The request received from an SSCP was to activate a half-session, a link, or a procedure, when that resource was at its share limit.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

| 0002 Invalid Request: The specified PU has already received an ACTPU and is therefore under
| the control of another SSCP. This ACTPU would exceed the share limit (= 1).

D.2.4 0852

0852 Duplicate Session Activation Request: Two session activation requests have been received with related identifiers. The relationship of the identifiers and the resultant action varies by request.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

| 0002 A REQACTPU has been received by an SSCP which has already sent an ACTPU for the
| same PU.

D.2.5 0877

0877 Resource Mismatch: The receiver of a request has detected a mismatch between two of the following: (1) its definition of an affected resource, (2) the actual configuration, and (3) the definition of the resource as implied in the request.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

- 0000 No specific code applies.
- 0001 Link Defined as Switched Is Nonswitched: A link defined to an ACTLINK receiver as being switched was found to be nonswitched during the activation attempt.
- 0002 Link Defined as SDLC Is Non-SDLC: A link defined to an ACTLINK receiver as being SDLC was found to be non-SDLC during the activation attempt.
- 0003 Link Defined as Having Automatic Connect-Out Capability Does Not: A link defined to an ACTLINK receiver as having automatic connect-out capability was found to lack it during the activation attempt.
- 0004 ACTLINK Received for a Resource Other Than a Link: An ACTLINK was received that resolved to a local device address representing a device other than a link.
- 0005 Link defined as X.21 is not X.21.
- 0006 Link defined as LPDA-capable is configured in NRZI mode.
- 0007 A request that is allowed only for a primary link station was received for a link station that is defined to the receiver as secondary.
- 0008 A request for link problem determination for modems was received for a link that is defined to the receiver as not supporting link problem determination for modems.
- 0009 A request for link problem determination for modems was received for a link that is defined to the receiver as supporting link problem determination for modems, but no link station or resource supporting link problem determination for modems was found on the link.
- 000A A request that is allowed only for a nonswitched link was received for a link that is defined to the receiver as switched.
- 000B A request that is allowed only for a link with a modem not using the multiplexed links feature was received for a link that is defined to the receiver as having a modem using the multiplexed links feature.
- 000C Resource Definition Mismatch for Modems: A request that is allowed only for a link with a nontailed modem was received for a link that is defined to the receiver as having a tailed modem.
- 000D The sending SSCP and the receiving T4 node have conflicting system definitions. A BIND has been received for an LU address that is currently being used by an active LU-LU session. The LU address is primary on this active session. The LU address cannot be used for a secondary role on a new session.
- 000E The sending SSCP and the receiving T4 node have conflicting system definitions. A BIND has been received for an independent LU, but the LU specified is not in a T2.1 node.

- 000F The sending SSCP and the receiving T4 node have conflicting system definitions. The SSCP owner is the same as the SSCP sending the nonactivation CONTACT PIU, but the node to be contacted is not a T2.1. The CONTACT is for a T2.1 node, but the node to be contacted is not defined as a T2.1 node to the receiver.
- 0010 The BFCLEANUP is for an independent LU, but the LU specified is not an independent LU.
- 0011 The subarea address portion of an addressed LU is not equal to the subarea address of the T4 node. The LU is not in the same subarea as the T4 node.
- 0012 A BFCLEANUP is for a resource that is not a BF LU, and hence the request is rejected. This is a situation where the function is not supported by the target resource. It can be caused by a system definition mismatch between the T4 node and the SSCP.
- 0013 The network ID in the BIND SLU name is not equal to the network ID of the boundary function, or the SLU name is not equal to the LU name in the boundary function control block for the LU, or the network ID in the BIND SLU name is not equal to the LU network ID in the boundary function control block for the LU.
- 0014 The LU specified in the FNA is not associated with the PU specified in the FNA; that is, an LU address (bytes 7–n) is not associated with the PU target address specified.
- 0015 BFCINIT Name Mismatch: The BIND cannot be built from the BFCINIT because the network-qualified PLU name does not match. The session activation is rejected by the boundary function with a BFTERM.
- 0016 Invalid Target Address: Either of the following conditions holds:
- The PU with which the specified LUs are to be associated is not type 1 or type 2; i.e., the SSCP attempts to add an LU to a PU, but the boundary function has defined that PU as a type 4.
 - The SSCP sent an RNAA assignment type X'0' or X'5' with a PU or LU specified instead of a link. This is caused by a definition mismatch.
- 0017 An entire network address including subarea and element is required for pre-ENA address assignment: If an entire network address is not specified and an RNAA requesting a pre-ENA address is received, the RNAA is rejected.
- 0018 An RNAA type 4 was received requesting an auxiliary address on a dependent LU.
- 001A The target LU specified in BFCLEANUP or BFCINIT is not associated with the same link station that is associated with the session indicated in the URC control vector.
- 001B The target link station specified in a BFCLEANUP is not the same link station as the session indicated in the URC control vector.
- 001C Resource Definition Mismatch for BFCINIT: The sending SSCP and the receiving T4 node have conflicting system definition. A BFCINIT has been received for an LU address that is currently being used by an active LU-LU session. The LU address is primary on this already active session. The LU address cannot be used for a secondary role on a new session.
- 001D The LU address in the BFCINIT is a secondary address; the BFCINIT is rejected.
- 001E The subject LU specified in the BFSESSINFO RU is not defined to the SSCP as an independent LU; this is a mismatch between the SSCP and the BF.

- 001F A dependent LU is attached to a PU that indicates ACTPU is to be suppressed; the SSCP cannot activate the LU because ACTLU is not supported.
- 0020 A peripheral node supporting independent LUs has received an ACTLU request for an LU. This request is rejected, as an independent LU does not support ACTLU.
- 0021 An RNAA(Add) was received by the boundary function for a resource defined at system definition time, which is not allowed.
- 0022 The link for which ACTLINK was issued is a S/370 channel that has been defined for connections only to a T2.1 node. However, the SSCP that sent ACTLINK had previously indicated it does not support T2.1 connections.
- 0023 Modem test support cannot be changed. The RNAA or SETCV containing the SDLC Secondary Station (X'03') or the Extended SDLC Secondary Station (X'43') control vector is rejected.
- 0024 The data mode cannot be changed. The RNAA or SETCV containing the SDLC Secondary Station (X'03') or the Extended SDLC Secondary Station (X'43') control vector is rejected.
- 0025 The receiving node is unable to process a BIND for the LU type specified for the given LU name.
- 0027 A link connection request for a nonempty active link connection configuration was received by the management services element; the active link connection configuration of the DLC element is empty; i.e., it has no link connection components present.
- 0028 An RNAA(Move) was received for a link station, and the link station's primary-secondary role is incompatible with the target link.
- 0029 The RU refers to a resource, and the sender and receiver disagree about its status. One considers it a static resource, the other a dynamic resource.
- 002A A session cannot be activated because the node does not support segment generation and the maximum link BTU size is too small to satisfy a requirement on the minimum send RU size as defined for the session mode.
- 002B A session cannot be activated because the node does not support segment reassembly and the maximum link BTU size is too small to satisfy a requirement on the minimum receive RU size as defined for the session mode.
- 002C A BFINIT session request was received from a PLU that is not in the same network as this SSCP, or a BFSESSINFO was received reporting a subject LU in another network.
- 002D BFSESSINFO was received for an independent subject LU, but the reported LU is considered by the receiver as a dependent LU.
- 002E BFSESSINFO was received reporting a dynamic subject LU that the receiver considers to be located under a different ALS from that reported in the BFSESSINFO. The SSCP will attempt to correct this configuration mismatch.
- 002F BFSESSINFO was received reporting a subject LU that the receiver considers to be located under a different ALS from that reported in the BFSESSINFO. The SSCP cannot correct this configuration mismatch.
- 0030 BFSESSINFO was received for a subject LU, but the receiver has the address associated with a different LU, which it considers to be static.

- 0031 BFSESSINFO was received for a subject LU, but the receiver has the address associated with anything other than a static LU or an other-domain resource.
- 0032 BFSESSINFO was received for a subject LU that is verified, but, for a given session, either the partner LU is reported as the primary and the receiver does not consider that LU to be primary-capable, or the partner LU is reported as the secondary and the receiver does not consider that LU to be secondary-capable.
- 0033 Upon receipt of BFSESSINFO, the receiver considers the control block associated with a partner LU to be an other-domain resource that is not active or an application program that is not active.
- 0034 Upon receipt of BFSESSINFO, an SSCP is unable to associate the information received about a partner LU to be associated with an LU, an other-domain resource, or an application program.
- 0035 A network address was returned in RSP(RNAA) that the receiver believes is already associated with a different resource.
- 0036 BFSESSINFO received containing an invalid ALS address. For example, the ALS does not represent a T2.1 node.
- 0037 BFSESSINFO received for a subject LU, where the secondary address specified in the BFSESSINFO does not match the secondary address the SSCP believes is associated with the LU.
- 0038 The subject LU specified in the BFSESSINFO RU is not defined to the SSCP as an LU or an other-domain resource.
- 0039 A request that is valid only for a switched subarea link was received for a link that is not subarea-capable.
- 003A A request that is valid only for a nonswitched subarea link was received for a subarea dial link.
- 003B An RNAA(Add) was received for an LU; however, an LU with the same name but a different local address already exists under the specified ALS.
- 0041 Takeover processing completed, but the SSCP did not receive a BFSESSINFO for a resource that the SSCP believed to be a static, independent LU.
- 0042 A BFINIT session request was received from a PLU that is not controlled by this SSCP.
- 0043 A request was received for a nonswitched resource that is valid only for a switched resource.
- 0044 A CONNOUT requested X.21 dial and auto-call capability was not present; resource mismatch.
- 0050 The element address of an intra-FRSE PVC segment subport specified in a SETCV resides on the same frame-relay port as another subport within a subport set.
- 0051 The maximum frame size in the system-definition differs for any two partners in an intra-FRSE PVC segment subport set specified in a SETCV.
- 0052 Adjacent frame-relay equipment management protocols are not supported on either of the frame-relay ports for the primary or its backup subport specified in the SETCV for the intra-FRSE PVC segment subport set.

0056 A CPSVRMGR session cannot be established over a LEN connection that is not of type TCP.

D.2.6 088E

088E Capability Mismatch: A network component detected a capability mismatch between different resources involved in the same network function. For example, an SSCP detects that an LU has been assigned a subarea address too large for one of the other resources involved in the session initiation to support.

Bytes 2 and 3 following the sense code contains sense code specific information. Settings allowed are:

0008 The session could not be established because the Dependent LU Server detected an incompatibility between its capabilities and those of its Dependent LU Requester.

0009 The session could not be established because the Dependent LU Requester detected an incompatibility between its capabilities and those of its Dependent LU Server.

000F An attempt was made to establish a CP-SVR pipe across a subnet boundary between a dependent LU server and a dependent LU requester with limited multi-subnet support.

D.2.7 0890

0890 Search Failure.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0037 TG vectors to the Dependent LU Requester are unknown: A resubmitted Locate search for a dependent LU at its Dependent LU Requester was unsuccessful. This condition arises only after the Dependent LU Server has verified the existence of the dependent LU; retry.

D.2.8 08A0

08A0 Session Reset: An LU or PU is resetting an LU-LU session.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0007 DLUS-DLUR session deactivation (disruptive): LU-LU sessions for DLUR-supported dependent LUs should be reset

0008 DLUS-DLUR session deactivation (non-disruptive): LU-LU sessions for DLUR-supported dependent LUs should not be reset

0009 DLUS-DLUR session deactivation (non-disruptive): protocol violation detected (LU-LU sessions for DLUR-supported dependent LUs should not be reset)

000A DLUS-DLUR session deactivation (non-disruptive): DLUR should wait for DLUS reactivation of DLUS-DLUR session (LU-LU sessions for DLUR-supported dependent LUs should not be reset)

D.2.9 8019

8019 Routing Exception: A node is unable to perform the required routing function for a request.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

| 000C A border node is not in PLU's subnet when searching for DLUS-supported LU: This
| occurs when a DLUS node determines that the PLU node's subnet did not use a Border
| Node for subnet connectivity when sending out a Locate request for a DLUS-served
| dependent LU.

			D-22, D-22, D-22, D-22, D-24, D-24, D-24, D-24, D-24, D-24, D-24, D-24, D-24
RUBIN2	RU	D-1	D-1, D-1
RUBIN3	RU	D-1	D-18, D-18, D-18, D-18
RUBIN4	RU	D-1	D-7, D-7, D-7, D-7, D-7, D-7, D-7, D-7, D-7, D-7
RUBIN5	RU	D-1	
RUBIN6	RU	D-1	
RUBIN7	RU	D-1	
RUBIN8	RU	D-1	
MSCOPTB	RU	D-1	
MSCOPTH	RU	D-1	

Figures

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
DLSFIG1	DLRFUNCT	3-2	3-1
DLSDEFC	DLRDEFC	5-7	5-1
DLSCPSE	DLRCPSE	5-11	5-2 6-11
DLSCPRE	DLRCPRE	5-13	5-3 6-7
DLSPD1	DLRPD1	5-17	5-4
DLSPD3	DLRPD3	5-19	5-5
DLSPD4	DLRPD4	5-22	5-6
DLSCPR5	DLRCPR5	5-27	5-7
DLSCPR3	DLRCPR3	5-29	5-8
DLSCPR2	DLRCPR2	5-32	5-9
DLSCPR4	DLRCPR4	5-36	5-10
DLSCPR7	DLRCPR7	5-38	5-11
DLSCPR6	DLRCPR6	5-40	5-12
DLSPD2	DLRPD2	5-43	5-13
DLSPD5	DLRPD5	5-46	5-14
DLSPD6	DLRPD6	5-49	5-15
DLSRC1	DLRRC1	5-53	5-16 6-28
DLSRC3	DLRRC3	5-56	5-17
DLSRC4	DLRRC4	5-59	5-18
DLSRC5	DLRRC5	5-61	5-19
DLSRC6	DLRRC6	5-65	5-20
DLSADD1	DLRENCAP	6-4	6-1
DLSADD2	DLRENCAP	6-5	6-2
DLSPAPU	DLRPAPU	6-8	6-3
DLSDAPU	DLRDAPU	6-10	6-4

DLSPR1	DLRPR1		
DLSPR2	DLRPR2	6-14	6-5
DLSPR3	DLRPR3	6-17	6-6
DLSPALU	DLRPALU	6-20	6-7
DLSDALU	DLRDALU	6-23	6-8
DLSPLD1	DLRPLD1	6-26	6-9
DLSPLD2	DLRPLD2	6-30	6-10
		6-33	6-11
DLSPLD3	DLRPLD3		6-28
		6-35	6-12
DLSPLD4	DLRPLD4	6-37	6-13
DLSUSSM	DLRUSSM	7-2	7-1
DLSUSSL	DLRUSSL	9-2	9-1
			9-4
DLSUSSP	DLRUSSP	9-5	9-2
DLSPLUA	DLRPLUA	9-7	9-3
DLSUSSS	DLRUSSS	9-10	9-4
DLSUSST	DLRUSST	9-14	9-5
DLSSA1	DLRSA1	10-5	10-1
DLSSA2	DLRSA2	10-6	10-2
DLSSA3	DLRSA3	10-8	10-3
DLSSA4	DLRSA4	10-10	10-4
DLSS2	DLRST2	10-13	10-5
DLSS5	DLRST5	10-17	10-6
DLSS3	DLRST3	10-23	10-7
DLSS4	DLRST4	10-27	10-8
DLSSD1	DLRSD1	10-31	10-9
DLSSD2	DLRSD2	10-32	10-10
DLSSD3	DLRSD3	10-34	10-11
DLSSD4	DLRSD4	10-36	10-12
DLSSD5	DLRSD5	10-38	10-13
DLS7371	DLRXSN1	11-8	11-1
DLS737D	DLRXP1	11-9	11-2
DLS737A	DLRXP1	11-10	11-3
DLS737B	DLRXP1	11-11	11-4
DLS7372	DLRXSN2	11-12	11-5
DLS737C	DLRXS1	11-13	11-6
DLS737E	DLRXS1	11-14	11-7
NTWK1	DLRFIG	C-1	C-1
NTWK2	DLRFIG	C-3	C-2

Headings			
<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
NONGEND	DLRFUNCT	3-4	3.2.2, Configuration Services XID Exit Support 6-11
SDDLU	DLRFUNCT	3-4	3.2.3, Self-Defining Dependent LU Support 3-4, 6-22
DLSNM	DLUR SCRIPT	4-1	Chapter 4, Network Management Concerns
DLSNAM	DLRNM	4-1	4.1, Naming Of DLUS-Supported Resources
CPSVR	DLUR SCRIPT	5-1	Chapter 5, CP-SVR Pipe 12-1, A-1
COS	DLRCPSVR	5-1	5.2, CPSVRMGR Mode 5-5
DLRSI	DLRCPSVR	5-2	5.3, DLUR-Initiated CP-SVR Pipe Activation 5-5, 5-9, 5-34, 5-41
DLKWN	DLRCPSVR	5-2	5.3.1, Pre-Defined DLUS Node Determination 5-3, 5-51
DLSUNK	DLRCPSVR	5-3	5.3.2, Dynamic DLUS Node Determination 5-51
DLSSI	DLRCPSVR	5-4	5.4, DLUS-Initiated CP-SVR Pipe Activation 5-3, 5-51
NEWPARM	DLRCPSVR	5-4	5.4.1, New PU Definition Parameters 5-12, 5-30, 5-37, 5-39, 6-6, 6-7
DLKWN2	DLRCPSVR	5-4	5.4.2, DLUS-Driven PU Activation 5-5
DLSUNK2	DLRCPSVR	5-5	5.4.3, Dynamic DLUR Node Determination
SESSEST	DLRCPSVR	5-5	5.5, CP-SVR Pipe Activation 5-1
SPD	DLRCPSVR	5-16	5.6, CP-SVR Pipe Deactivation 5-5, 5-6, 5-15, 5-28, 5-31, 5-35, 5-37, 5-39, 5-41, 5-44, 5-48, 5-50, 5-54, 5-58, 5-60, 5-67, 10-21, 10-40
NRMSPD	DLRCPSVR	5-16	5.6.1, Normal CP-SVR Pipe Deactivation
ABNSPD	DLRCPSVR	5-25	5.6.2, Abnormal CP-SVR Pipe Deactivation 5-6
ABNSPDU	DLRCPSVR	5-25	5.6.2.1, UNBIND-Initiated CP-SVR Pipe Deactivation 5-51
ABNSPDM	DLRCPSVR	5-26	5.6.2.2, DLUR/S Capabilities Mismatches 11-5
SPSSPD	DLRCPSVR	5-41	5.6.3, Abnormal SSCP-PU/SSCP-LU Session Deactivation (DLUR-Initiated) 6-28
SPSSPD1	DLRCPSVR	5-42	5.6.3.1, Downstream PU Outage
SPSSPD2	DLRCPSVR	5-44	5.6.3.2, Receipt Of REQDISCONT From Downstream PU
SESSLGC	DLRCPSVR	5-51	5.7, Pipe And Session Failure Logic 5-16, 5-25
SESSRCV	DLRCPSVR	5-51	5.7.1, CPSVRMGR Session Recovery
PULURCV	DLRCPSVR	5-51	5.7.2, SSCP-PU And SSCP-LU Session Recovery
RACERCV	DLRCPSVR	5-51	5.7.3, Pipe And Session Activation/Recovery Race Conditions
RAC2RCV	DLRCPSVR	5-52	5.7.3.1, Multi-DLUS Race Conditions
RAC1RCV	DLRCPSVR	5-55	5.7.3.2, Single DLUS Race Conditions
SSCPPU	DLRENCAP		

		6-6	6.2, SSCP-PU Session Activation 5-6, 5-6
SSCPLU	DLRENCAP		
		6-22	6.3, SSCP-LU Session Activation 8-1, 9-6, 9-8, 9-13
PLDACT	DLRENCAP		
		6-28	6.4, SSCP-PU/SSCP-LU Session Deactivation (DLUS-Initiated) 5-41, 5-44, 6-29
PLANS	DLRENCAP		
PLFLOWS	DLRENCAP	6-28	6.4.1, DLUR ANS Support
DLSUSS	DLUR SCRIPT	6-29	6.4.2, SSCP-PU/SSCP-LU Session Deactivation Flows
ROUTE	DLUR SCRIPT	7-1	Chapter 7, Unformatted Session Services Management
		8-1	Chapter 8, LU-LU Session Routing 9-6, 9-6, 9-8, 9-13, B-1
RSPACLU	DLRRROUTE		
TAIL	DLRRROUTE	8-2	8.1.2, +RSP(ACTLU)
		8-4	8.2, DLUR EN Tail Vector Registration 11-2
SAW	DLUR SCRIPT		
		10-1	Chapter 10, Dependent LU Session Awareness Management 7-3
SAWOV	DLRSAW		
PRESAW	DLRSAW	10-2	10.1, Functional Overview
DLSSAW	DLRSAW	10-2	10.1.1, Pre-DLUR/S Dependent LU Session Awareness
		10-2	10.1.2, DLUR/S Dependent LU Session Awareness 6-38, 8-2, 8-2, 8-3
SAWFLO	DLRSAW		
SAWACT	DLRSAW	10-4	10.2, Dependent LU Session Awareness Flows
SAWACF1	DLRSAW	10-4	10.2.1, LU-LU Session Activation Flows
		10-5	10.2.1.1, SSCP-LU, LU-LU Sessions Routed Through Same BF-NCP
SAWACF2	DLRSAW		
		10-6	10.2.1.2, SSCP-LU, LU-LU Sessions Routed Through Different BF-NCPs (Same SSCP)
SAWACF3	DLRSAW		
		10-8	10.2.1.3, SSCP-LU, LU-LU Sessions Routed Through Different BF-NCPs
SAWACF4	DLRSAW		
SAWSTA	DLRSAW	10-10	10.2.1.4, LU-LU Session Not Routed Through A BF-NCP
		10-11	10.2.2, SSCP Takeover & Giveback Flows 5-51
SAWSTF2	DLRSAW		
		10-12	10.2.2.1, SSCP-LU, LU-LU Sessions Routed Through Same BF-NCP
SAWSTF5	DLRSAW		
		10-16	10.2.2.2, SSCP-LU, LU-LU Sessions Routed Through Same BF-NCP
SAWSTF3	DLRSAW		
		10-22	10.2.2.3, SSCP-LU, LU-LU Sessions Routed Through Different BF-NCPs
SAWSTF4	DLRSAW		
		10-26	10.2.2.4, SSCP-LU, LU-LU Sessions Routed Through Different BF-NCPs
SAWDAC	DLRSAW		
SAWDAF1	DLRSAW	10-30	10.2.3, Normal LU-LU Session Deactivation Flows
		10-31	10.2.3.1, SSCP-LU, LU-LU Sessions Routed Through Same BF-NCP
SAWDAF2	DLRSAW		
		10-32	10.2.3.2, SSCP-LU, LU-LU Sessions Routed Through Different BF-NCPs (Same SSCP)
SAWDAF3	DLRSAW		
		10-34	10.2.3.3, SSCP-LU, LU-LU Sessions Routed Through Different BF-NCPs
SAWDAF4	DLRSAW		
		10-36	10.2.3.4, LU-LU Session Not Routed Through A BF-NCP
SAWDAB	DLRSAW		
		10-37	10.2.4, Abnormal LU-LU Session Deactivation Flows
SAWDAF5	DLRSAW		
		10-37	10.2.4.1, Abnormal LU-LU Session Deactivation
DLXNET	DLUR SCRIPT		
		11-1	Chapter 11, Multi-Subnet DLUR/S

XNETOV	DLRXNET	11-1	11.1, Overview
XNETCNF	DLRXNET	11-1	11.2, Multiple Subnet DLUR/S Configurations
XNETCR1	DLRXNET	11-1	11.2.1, Case 1: DLUS And DLUR In Different Subnets
XNETCR2	DLRXNET	11-2	11.2.2, Case 2: DLUS And PLU In Different Subnets 11-3, 11-3
XNETCR3	DLRXNET	11-2	11.2.3, Case 3: PLU And DLUR In Different Subnets
XNETCR4	DLRXNET	11-3	11.2.4, Configuration Restrictions
XNETFD	DLRXNET	11-4	11.3, Functional Description
XNETDEF	DLRXNET	11-4	11.3.1, Border Node Terminology
XNETIND	DLRXNET	11-4	11.3.2, Multi-Subnet DLUR/S Indicators
XNETPIP	DLRXNET	11-5	11.3.3, Multi-Subnet CP-SVR Pipe Activation
XNETEN1	DLRXNET	11-5	11.3.4, Multi-Subnet EN DLUR Capabilities
XNETLOC	DLRXNET	11-6	11.3.5, Multi-Subnet DLUR/S Locate Searches
XNETPLU	DLRXNET	11-6	11.3.5.1, PLU-initiated searches
XNETSLU	DLRXNET	11-7	11.3.5.2, SLU-initiated searches
XNETFIS	DLRXNET	11-7	11.3.6, Multi-Subnet Functional Impact Summary
XNETFI2	DLRXNET	11-7	11.3.6.1, DLUR / Multi-Subnet Impacts
XNETFLO	DLRXNET	11-7	11.4, Multi-Subnet DLUR/S Flows
XNETFLP	DLRXNET	11-7	11.4.1, Multi-Subnet PLU-Initiated Searches
XNETFP1	DLRXNET	11-8	11.4.1.1, DLUS, DLUR, PLU In Different Subnets
XNETFLS	DLRXNET	11-12	11.4.2, Multi-Subnet SLU-Initiated Searches
XNETFS1	DLRXNET	11-12	11.4.2.1, DLUS, DLUR, PLU In Different Subnets
DLSTOWE	DLUR SCRIPT	12-1	Chapter 12, DLUR Option Set Content
DLR	DLRTOWER	12-1	12.1, DLUR Tower Requirements
DLSFORM	DLUR SCRIPT	13-1	Chapter 13, Format Changes
CPSFMT	DLRFORM	13-1	13.1, CP-SVR Pipe Format Changes
ENCFMT	DLRFORM	13-1	13.1.1, FID2 Encapsulation (X'1500') GDS Variable
EN81FMT	DLRFORM	13-2	13.1.1.1, XID Image (X'81') FID2 Encapsulation Control Vector
RQACPU	DLRFORM	13-2	13.1.2, ACTPU (Activate Physical Unit)
RQAFMT	DLRFORM	13-2	13.1.3, REQACTPU (Request ACTPU)
RDQFMT	DLRFORM	13-3	13.1.4, REQDACTPU (Request DACTPU)
NNMFMT	DLRFORM	13-3	13.1.5, Network Name (X'0E') Control Vector
TGDFMT	DLRFORM	13-3	13.1.6, TG Descriptor (X'46') Control Vector 13-1
EN80FMT	DLRFORM	13-4	13.1.7, DLUR/S Capabilities (X'51') Control Vector
SESFMT	DLRFORM	13-5	13.2, Multi-Subnet Session Format Changes
LOCFMT	DLRFORM	13-5	13.2.1, Locate (X'12C4') GDS Variable
LO80FMT	DLRFORM	13-5	13.2.1.1, Intersubnetwork Search (X'82') Locate Control Vector
FINFMT	DLRFORM	13-5	13.2.2, Find Resource (X'12CA') GDS Variable
FI80FMT	DLRFORM	13-5	13.2.2.1, Command Parameters (X'80') Find Control Vector
FOUFMT	DLRFORM	13-5	13.2.3, Found Resource (X'12CB') GDS Variable

FO80FMT	DLRFORM			
SAWFMT	DLRFORM	13-5		13.2.3.1, Command Parameters (X'80') Found Control Vector
		13-6		13.3, Dependent LU Session Awareness Format Changes 10-3, 10-4, 10-11, 10-30
SSTFMT	DLRFORM			
SNDFMT	DLRFORM	13-6		13.3.1, SESSST (Session Started)
RALFMT	DLRFORM	13-6		13.3.2, SESSEND (Session Ended)
SINFMT	DLRFORM	13-6		13.3.3, RSP(ACTLU)
		13-7		13.3.4, Session Information (X'2A') Control Vector 13-6, 13-6
DLSSNS	DLRFORM			
ACPSNS	DLRFORM	13-8		13.4, Sense Codes
LOCSNS	DLRFORM	13-8		13.4.1, ACTPU
RQASNS	DLRFORM	13-8		13.4.2, LOCATE
RQDSNS	DLRFORM	13-8		13.4.3, REQACTPU
UNBSNS	DLRFORM	13-8		13.4.4, REQDACTPU
ALT	DLUR SCRIPT			
ALTRTE	DLRALT	B-1		Appendix B, Alternate Solutions
		B-1		B.1, LU-LU Session Route Calculation 8-1
DLSNWKS	DLUR SCRIPT			
NTWK1	DLRFIG	C-1		Appendix C, Sample Network Configurations
NTWK2	DLRFIG	C-1		C.1, Sample Network Configuration Number 1
DLSFMTS	DLUR SCRIPT	C-3		C.2, Sample Network Configuration Number 2
		D-1		Appendix D, Formats For DLUR/S Architecture 13-1, 13-4
CF	?	?	?	
RUSPARS	?	?	?	D-1, D-2, D-2, D-5, D-6, D-11
				D-7, D-9, D-10, D-11, D-11, D-12, D-13, D-15, D-16, D-17
RUCV0E	RUCV0E DLRFMT	D-8		D.1.10.1, Network Name (X'0E') Control Vector
RUCV25	RUCV25 DLRFMT	D-10		D.1.12.1, Security ID Control (X'25') Control Vector
RUCV30	RUCV30 DLRFMT	D-11		D.1.14.1, Assign LU Characteristics (X'30') Control Vector
RUCV31	RUCV31 DLRFMT	D-12		D.1.15.1, BIND Image (X'31') Control Vector
RUCVLOC	RUGV12C4 DLRFMT	D-19		D.1.19.2, Locate Control Vectors
RUGVFIN	RUGV12CA DLRFMT	D-21		D.1.20.1, Find Resource (X'12CA') GDS Variable
RUCVFIN	RUGV12CA DLRFMT	D-22		D.1.20.2, Find Control Vectors D-21
RUCVFI0	RUGV12CA DLRFMT	D-22		D.1.20.2.1, Command Parameters (X'80') Find Control Vector
RUCVFI2	RUGV12CA DLRFMT	D-22		D.1.20.2.2, Search Argument Directory Entry (X'82') Find Control Vector
RUGVFOV	RUGV12CB DLRFMT	D-23		D.1.21.1, Found Resource (X'12CB') GDS Variable
RUCVFOU	RUGV12CB DLRFMT	D-24		D.1.21.2, Found Control Vectors D-23
RUCVFO0	RUGV12CB DLRFMT	D-24		D.1.21.2.1, Command Parameters (X'80') Found Control Vector

Footnotes

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
MODEL	DLRFUNCT	3-6	1 3-6
DLRTGV	DLRRROUTE	8-1	1 8-1

Revisions

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
V1	DLUR SCRIPT	i	iii, iv, 1-2, 1-2, 3-4, 3-4, 5-1, 5-1, 5-1, 5-1, 5-2, 5-2, 5-4, 5-4, 5-4, 5-5, 5-5, 5-7, 5-8, 5-9, 5-9, 5-42, 5-42, 5-44, 5-48, 5-51, 5-51, 5-67, 5-68, 6-3, 6-5, 6-6, 6-6, 6-6, 6-6, 6-10, 6-10, 6-11, 6-11, 6-11, 6-11, 6-11, 6-11, 6-11, 6-11, 6-11, 6-14, 6-14, 6-17, 6-18, 6-39, 6-39, 8-3, 8-3, 9-12, 9-12, 13-2, 13-2, 13-2, 13-2
V2	DLUR SCRIPT	i	iv, iv, 5-42, 5-42, 5-44, 5-44, 5-44, 5-47, 5-48, 5-48, 5-48, 5-48, 5-48, 5-48, 5-48, 5-48, 5-50, 8-3, 8-3, 13-3, 13-3
V3	DLUR SCRIPT	i	iii, iii, iv, vi, 5-6, 5-6, 5-6, 5-6, 5-13, 5-13, 5-14, 5-14, 5-15, 5-15, 5-25, 5-25, 5-25, 5-26, 5-26, 5-41, 5-44, 5-44, 5-49, 5-49, 5-50, 5-50, 5-55, 5-55, 5-65, 5-67, 5-68, 5-68, 6-1, 6-3, 6-28, 6-28, 6-28, 6-29, 6-33, 6-33, 6-34, 6-34, 6-34, 6-34, 6-34, 6-35, 6-35, 6-36, 6-36, 6-38, 6-39, 8-2, 8-2, 8-2, 8-2, 8-2, 8-2, 8-2, 8-3, 8-3, 8-3, 8-4, 8-4, 11-1, 11-1, 11-3, 11-4, 11-5, 11-5, 11-5, 11-5, 11-5, 11-5, 11-5, 11-6, 11-6, 11-7, 11-7, 11-7, 11-7, 11-7, 11-7, 11-7, 12-3, 12-3, 12-3, 12-3, 13-3, 13-3, 13-4, 13-4, 13-8, 13-8, D-1, D-1
007516	DLUR SCRIPT	i	D-2, D-2, D-3, D-3, D-3, D-4, D-5, D-5, D-5, D-5, D-5, D-5, D-5, D-6, D-6, D-6, D-8, D-8, D-9, D-9, D-11, D-11, D-11, D-11, D-11, D-15, D-15, D-17, D-18, D-20, D-20, D-22, D-22, D-22, D-22, D-24, D-24, D-24, D-24, D-24, D-25, D-26, D-30, D-31, D-31, D-31, D-31, D-31, D-31, D-32, D-32
007759	DLUR SCRIPT	i	D-15, D-15, D-15, D-16, D-16, D-16, D-16, D-17
007779	?	?	?
007898	?	?	D-11, D-11 D-30, D-30

Tables

<u>id</u>	<u>File</u>	<u>Page</u>	<u>References</u>
ENCAPID	DLRENCAP	6-2	6-1
ENCAPPU	DLRENCAP	6-39	6-2
SAWTAB	DLRSAW	10-2	10-1
XNETTAB	DLRXNET	11-1	11-1
GDSTAB	DLRFORM	13-2	13-1
CV	?	?	?

Processing Options

Runtime values:

Document fileid	DLUR SCRIPT
Document type	USERDOC
Document style	DEFAULT
Profile	EDFPRF40
Service Level	0022
SCRIPT/VS Release	4.0.0
Date	94.02.03
Time	09:35:56
Device	38PPN
Number of Passes	3
Index	YES
SYSVAR G	INLINE
SYSVAR X	YES
SYSVAR 0	ERTDVCF7

Formatting values used:

Annotation	NO
Cross reference listing	YES
Cross reference head prefix only	NO
Dialog	LABEL
Duplex	YES
DVCF conditions file	ERTDVCF7
DVCF value 1	(none)
DVCF value 2	(none)
DVCF value 3	(none)
DVCF value 4	(none)
DVCF value 5	(none)
DVCF value 6	(none)
DVCF value 7	(none)
DVCF value 8	(none)
DVCF value 9	(none)
Explode	NO
Figure list on new page	YES
Figure/table number separation	YES
Folio-by-chapter	YES
Head 0 body text	(none)
Head 1 body text	Chapter
Head 1 appendix text	Appendix
Hyphenation	YES
Justification	YES
Language	ENGL
Keyboard	395
Layout	OFF
Leader dots	YES
Master index	(none)
Partial TOC (maximum level)	4
Partial TOC (new page after)	INLINE
Print example id's	NO
Print cross reference page numbers	YES
Process value	(none)
Punctuation move characters	,,
Read cross-reference file	(none)
Running heading/footing rule	NONE
Show index entries	NO
Table of Contents (maximum level)	3
Table list on new page	YES
Title page (draft) alignment	RIGHT
Write cross-reference file	(none)

Imbed Trace

Page 0	DSMDVCF
Page iii	DLRPREF
Page 1-1	DLRINTR
Page 2-1	DLRREQS
Page 3-1	DLRFUNCT
Page 4-1	DLRNM
Page 5-1	DLRCPSVR
Page 5-7	DLRDEFC
Page 5-11	DLRCPSE
Page 5-13	DLRCPRE
Page 5-17	DLRPD1
Page 5-19	DLRPD3
Page 5-22	DLRPD4
Page 5-27	DLRCPR5
Page 5-29	DLRCPR3
Page 5-32	DLRCPR2
Page 5-36	DLRCPR4
Page 5-38	DLRCPR7
Page 5-40	DLRCPR6
Page 5-43	DLRPD2
Page 5-46	DLRPD5
Page 5-49	DLRPD6
Page 5-53	DLRRC1
Page 5-56	DLRRC3
Page 5-59	DLRRC4
Page 5-61	DLRRC5
Page 5-65	DLRRC6
Page 6-1	DLRENCAP
Page 6-8	DLRPAPU
Page 6-10	DLRDAPU
Page 6-14	DLRPR1
Page 6-17	DLRPR2
Page 6-20	DLRPR3
Page 6-23	DLRPALU
Page 6-26	DLRDALU
Page 6-30	DLRPLD1
Page 6-33	DLRPLD2
Page 6-35	DLRPLD3
Page 6-37	DLRPLD4
Page 7-1	DLRUSS
Page 7-1	DLRUSSM
Page 8-1	DLROUTE
Page 9-1	DLRSESS
Page 9-2	DLRUSSL
Page 9-5	DLRUSSP
Page 9-7	DLRPLUA
Page 9-10	DLRUSSS
Page 9-14	DLRUSST
Page 10-1	DLRSAW
Page 10-5	DLRSA1
Page 10-6	DLRSA2
Page 10-8	DLRSA3
Page 10-10	DLRSA4
Page 10-13	DLRST2
Page 10-17	DLRST5
Page 10-23	DLRST3
Page 10-27	DLRST4
Page 10-31	DLRSD1
Page 10-32	DLRSD2
Page 10-34	DLRSD3
Page 10-36	DLRSD4
Page 10-38	DLRSD5
Page 11-1	DLRXNET
Page 11-7	DLRXSN1
Page 11-8	DLRXP1
Page 11-12	DLRXSN2
Page 11-12	DLRXS1
Page 12-1	DLRTOWER
Page 13-1	DLRFORM
Page A-1	DLRDEF
Page B-1	DLRALT
Page C-1	DLRFIG
Page D-1	DLRFMTS
Page D-1	SDLRSETU
Page D-1	DFMTREV
Page D-1	RU
Page D-1	RUACLU DLRFMT

Page D-1	RUACPU DLRFMT
Page D-3	RURQACPU DLRFMT
Page D-3	RURQDCPU DLRFMT
Page D-4	RURQDISC DLRFMT
Page D-4	RUSE DLRFMT
Page D-5	RUSS DLRFMT
Page D-6	RUACLU# DLRFMT
Page D-6	RUCV0C DLRFMT
Page D-8	RUCV0E DLRFMT
Page D-9	RUCV23 DLRFMT
Page D-10	RUCV25 DLRFMT
Page D-10	RUCV2A DLRFMT
Page D-11	RUCV30 DLRFMT
Page D-12	RUCV31 DLRFMT
Page D-12	RUCV43 DLRFMT
Page D-15	RUCV46 DLRFMT
Page D-15	RUCV4691 DLRFMT
Page D-16	RUCV4692 DLRFMT
Page D-16	RUCV4693 DLRFMT
Page D-17	RUCV51 DLRFMT
Page D-18	RUGV12C4 DLRFMT
Page D-20	RUGV12CA DLRFMT
Page D-23	RUGV12CB DLRFMT
Page D-24	RUGV1500 DLRFMT
Page D-26	SDT0801 DLRFMT
Page D-26	SDT0806 DLRFMT
Page D-26	SDT082C DLRFMT
Page D-26	SDT0852 DLRFMT
Page D-26	SDT0877 DLRFMT
Page D-31	SDT088E DLRFMT
Page D-31	SDT0890 DLRFMT
Page D-31	SDT08A0 DLRFMT
Page D-31	SDT8019 DLRFMT