current edition of the ″Best of APPC and APPN″ CD-ROM. That book is a little out of date, but is still an excellent introduction to alerts.

- (6.2.3.3) updated DLUS processing of DLUR ACTPU Indicator in XID3
- (13.1.6, D.1.18) added DLC signalling information formats for SDLC switched, V.25 interface, and V.25bis physical interface
- (13.4.2, D.2.9) clarified description of X′8019 000C′ sense code
- (D.1.18.1.1) made SF X′92′ (Port Identifier) required for Frame Relay PVC

The following revisions were made based on review comments to Document Number DLUR-5. These changes are indicated by an ampersand (&) throughout the text in versions up through DLUR-7.

- (11.3.4.4, 11.3.6.1, 12.1.6) added DLUS-served LU registration interaction with central resource registration
- (appendix X) added index

The following revisions were made based on review comments to Document Number DLUR-6 (APPN DLUR Architecture Reference Version 1.0). These changes are indicated by an ampersand (&) throughout the text in DLUR-7.

- (8.1.5, 12.1.4) added DLUR processing of short RSP(BIND) RUs

% The following revisions were made based on review comments to Document
% Number SV40-1010-00/DLUR-7 (APPN DLUR Architecture Reference Version
% 1.1). These changes are indicated by an percent sign (%) throughout the text.

% - (4.6.1.1) corrected sense data encoding in DLUR protocol error alert
% - (4.6.2.1, 4.6.2.2, 4.6.2.3, 4.6.2.4) corrected actions encoding and ordering in
% DLUR configuration error alerts
% - (new 5.3.1, 5.4.1, 5.5.2.1, 5.5.2.2, 6.1.2.1, 6.1.2.2, 6.1.2.3, 6.2, 6.2.2) added
% support for signaling information definition at the DLUR
% - (new 5.3.1, new 5.3.1.1, new 5.3.1.2, 5.4.1, 5.5.2.1, 5.5.2.3, 6.1.2, 6.1.2.3, 6.2,
% 13.1.1, 13.4.3, C.1.24.1, new C.2.3, new C.2.7, new C.2.10) added support for
% PU name definition at the DLUR
% - (6.1, new 6.1.4) added DLUR/S processing for encapsulated FID2 segments on
% CP-SVR pipe to DLUS
% - (8.1.5, 12.1.4) further clarified DLUR processing of short RSP(BIND) RUs and
% added WBRI processing of RSP(BIND) RUs
% - (8.1.6, 12.1.4) added DLUR processing of extended UNBIND RUs
% - (8.2, 12.1.3) clarified DLUR EN tail vector registration processing, including
% permitting an EN DLUR adjacent to its DLUS to register its tail vectors using
% the CP-SVR pipe
% - (6.2.2, 13.1.1.1, C.1.1) added XID1 to the description of CV X′81′ in GDS
% X′1500′

% • (13.1.6, C.1.18.1.1, C.1.18.1.2, C.1.18.1.3, C.1.18.1.4, new C.1.18.1.7, new
% C1.1.8.1.8, new C.1.18.1.9, new C.1.18.1.10) added DLC signalling information
% formats for X.25 PVC and X.25 SVC

% • (C.1.18.1.1, C.1.18.1.2, C.1.18.1.3, C.1.18.1.4, C.1.18.1.5) changed FR DLC
% type to FRPVC

% • (C.1.18.1.1, C.1.18.1.6) added to CV X′4696′ a destination MAC address for
% FRPVC

% • (C.1.18.1.3, C.1.18.1.4) updated V.25/V.25bis destination dial address and calling
% identification number information

% • (C.1.18.1.3, C.1.18.1.4, C.1.18.1.5, C.1.18.1.6) changed encodings of Token-Ring
% signalling information from canonical to noncanonical

% • (C.1.18.1.5) retired CV X′4695′ SDLC station role usage for SDLCNS and
% SDLCSW

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 predefined. 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 change number of sessions (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 5-7.

The CPSVRMGR mode will use the SNASVCMG class of service (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 PU2.0 or T2.1 node providing local support for the dependent LU. This session is created when the SSCP sends an ACTPU to the PU2.0/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 PU2.0 device supporting dependent LUs is Boundary Function (BF)-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 BF-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.

% 5.3.1 DLUR PU Identification

% When the DLUR sends a REQACTPU RU it must identify the PU for which it is requesting SSCP ser-
% vices. The PU will be identified by including with the REQACTPU one or both of the following:

% 1. ***XID*** - This is either an image of an XID received from a PU externally attached to the DLUR or a
% DLUR-generated XID image (for an internal PU or an external PU not supporting XID transmission).

% 2. ***PU TG Descriptor*** - This item consists of one or both of the following:

% a. ***PU Name*** - The PU name can be sent to the DLUS, either to match a predefined PU statement or
% to be used in the creation of a dynamic PU at the DLUS. The PU name can be sent either with or
% without a NETID:

% 1) non-network qualified (without NETID) -in this case the DLUS will match only on PU name.

% 2) network qualified (with NETID) - in this case the DLUS will match on NETID.PUname. It is
% an implementation option for the DLUR to choose how to get the PU name and the NETID
% to put together a network-qualified PU name to send to the DLUS. The DLUR could require
% the user to define a network-qualified PU name, or only define a PU name and use its own
% NETID, or in the case of a downstream PU learn about the NETID via XID (remember: a
% downstream PU could have a NETID different from the DLUR′s NETID; it is the PU′s
% NETID which will be included in the network-qualified PU name).

% b. ***Signaling Information*** - This parameter will identify the DLUR-to-PU connectivity. This information must be coordinated between the DLUR and DLUS nodes:

% 1) it can be defined completely at the DLUR - in this case some or all of the information can optionally be sent to the DLUS to be used for network management, input into the Configuration Services XID exit for connection verification and accounting, and PU correlation

% 2) it can be defined completely at the DLUS - in this case all of the information will be sent to the DLUR on the subsequent ACTPU

% 3) it can be defined partially at the DLUS and partially at the DLUR - in this case the DLUR can optionally send the DLUR-defined signaling information (for the reasons described above) while the DLUS-defined signaling information will be sent on the subsequent ACTPU

% • if upon receipt of the ACTPU, the DLUR determines it has incomplete signaling information, it can optionally proceed with the activation (using defaults) or it can fail the activation

% • if upon receipt of the ACTPU, the DLUR determines it has the same signaling information subfield from both the DLUR and DLUS, but the subfields have different values, it will use the DLUR-defined values (since it was the initiator of the activation)

% When both PU name and CP name are sent to the DLUS, these names must be different.

% 5.3.1.1 DLUS PU Search Order

% When a DLUS receives a REQACTPU, the search order to find the correct PU is as follows:

% 1. PU name - if the DLUR supplies a PU name and if it matches a predefined PU name, the search is over; otherwise,

% 2. IDBLK/IDNUM - if the DLUR supplies an IDBLK/IDNUM and it matches a predefined IDBLK/IDNUM, the search is over (note that the PU names may not match - the DLUR may accept the DLUS-provided name); otherwise,

% 3. CP name - if the DLUR supplies a CP name and it matches a predefined CP name, the search is over (note that the PU names may not match - the DLUR may accept the DLUS-provided name); otherwise,

% 4. user exit - if an exit exists, a dynamic PU will be created (note that the PU names may not match - the DLUR may accept the DLUS-provided name); otherwise, the search will fail

% 5.3.1.2 DLUR PU Search Order

% When a DLUR receives an ACTPU, the search order to find the correct PU is a product option:

% • Option A - PU name does not have to match

% 1. PU name - this name does not have to match a DLUR-defined name for the same PU, and the DLUS-provided PU name will be accepted

% 2. CP name and/or IDBLK/IDNUM - whatever is received must match its corresponding definitions at the DLUR; otherwise, the ACTPU will be rejected (since the CP names and/or IDBLK/IDNUMs do not match)

% • Option B - PU name must match

% 1. PU name - if the DLUS supplies a PU name and if it matches a predefined PU name, the search is over; otherwise,

% 2. IDBLK/IDNUM - if the DLUR supplies an IDBLK/IDNUM and it matches a predefined
% IDBLK/IDNUM, then

% a. if the predefined IDBLK/IDNUM has no PU name defined with it, the search is over and the
% DLUS-supplied PU name will be used; otherwise,

% b. if the predefined IDBLK/IDNUM has a PU name defined with it, the ACTPU will be rejected
% (since the PU names do not match);

% otherwise, the ACTPU will be rejected (since the IDBLK/IDNUMs do not match)

% For more information about SSCP-PU related session initiation information, please refer to 6.2, "SSCP-PU % Session Activation" on page 6-7.

% 5.3.2 DLUS Node Determination

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:

% 5.3.2.1 Predefined DLUS Node Determination

A DLUS node is said to be ***known*** when this information is predefined (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 CP name 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 (e.g., defaults or operator-defined).

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 predefined 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 as session does not already exist, the DLUR uses the Send_Encap_Msg_TP to issue an ALLOCATE verb with the DLUS′s CP name 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 (e.g., DLUS not active or routes not available), 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 (e.g., backup DLUS definition, timer-driven retry, event-driven retry, or operator-driven retry), or wait for a DLUS-driven PU activation as described in 5.4, "DLUS-Initiated CP-SVR Pipe Activation" on page 5-5.

% 5.3.2.2 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 5.3.2.1, "Predefined DLUS Node Determination." 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 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 wil 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 ACTLUs may flow over the SSCP-PU session encapsulated in the CP-SVR pipe to activate the dependent LUs.

When activating a predefined PU, however, the DLUS node must know where to send the ACTPU. 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 CP name 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 CP name of the DLUR node supporting them.

- ***PU TG Descriptor***

% This parameter will identify the DLUR-to-PU connectivity and is made up of signaling information (see % 5.3.1, “DLUR PU Identification” on page 5-2). This parameter must also be coordinated between the % DLUR and DLUS nodes:

% 1. it can be defined completely at the DLUS - in this case all of the information will be sent to the DLUR % on the ACTPU

% 2. it can be defined completely at the DLUR - in this case some or all of the information can optionally be % sent to the DLUS to be used for network management, input into the Configuration Services XID exit % for connection verification and accounting, and PU correlation

% 3. it can be defined partially at the DLUS and partially at the DLUR - in this case the DLUR can % optionally send the DLUR-defined signaling information (for the reasons described above) while the % DLUS-defined signaling information will be sent on the ACTPU

% – if upon receipt of the ACTPU, the DLUR determines it has incomplete signaling information, it can % optionally proceed with the activation (using defaults) or it can fail the activation

% – if upon receipt of the ACTPU, the DLUR determines it has the same signaling information subfield % from both the DLUR and DLUS, but the subfields have different values, it will use the % DLUS-defined values (since the DLUS was the initiator of the activation)

% When the PU is locally supported on the DLUR node, this parameter can identify which PU image is asso- % ciated with the DLUS PU definition. When the PU is located downstream from the DLUR node, this % parameter can identify the TG providing connectivity between the DLUR and PU node. In the case of % leased line connections, this description can 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 can provide the port, MAC and SAP address of the downstream PU. In the case of switched
% callout line connections, the description can specify IDBLK and IDNUM values as well as dial digits.

% A new group of subfields (X′91′ - X′F0′) is 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).

% Whatever signaling information is sent by the DLUR optionally can be echoed back by the DLUS.

% When a DLUR receives an ACTPU, the search order to find the correct PU is a product option:

% – Option A - PU name does not have to match

% 1. PU name - this name does not have to match a DLUR-defined name for the same PU, and the
% DLUS-provided PU name will be accepted

% 2. CP name and/or IDBLK/IDNUM - whatever is received must match its corresponding definitions
% at the DLUR; otherwise, the ACTPU will be rejected (since the CP names and/or
% IDBLK/IDNUMs do not match)

% – Option B - PU name must match

% 1. PU name - if the DLUS supplies a PU name and if it matches a predefined PU name, the search is
% over; otherwise,

% 2. IDBLK/IDNUM - if the DLUR supplies an IDBLK/IDNUM and it matches a predefined
% IDBLK/IDNUM, then

% a. if the predefined IDBLK/IDNUM has no PU name defined with it, the search is over and the
% DLUS-supplied PU name will be used; otherwise,

% b. if the predefined IDBLK/IDNUM has a PU name defined with it, the ACTPU will be rejected
% (since the PU names do not match);

% otherwise, the ACTPU will be rejected (since the IDBLK/IDNUMs do not match)

% For more information about SSCP-PU related session initiation information, please refer to 6.2, “SSCP-PU
% Session Activation” on page 6-7.

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 (e.g., operator-driven retry, timer-based retry, or network management exits).

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." 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 CP name 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. Limited resources are usually indicated as such so that they are used for short-lived connections. CPSVRMGR sessions are long-lived connections which, if established over limited resources, could be very expensive. Even so, CPSVRMGR sessions are permitted over limited resources; implementations may optionally choose to prohibit CPSVRMGR sessions over limited resources. If a CPSVRMGR session is activated over a limited resource, the DLUS and DLUR nodes should ignore the limited resource characteristic of that session and not deactivate the session when the limited resource logic would normally dictate session deactivation due to inactivity. In general, 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 5-19 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 6-7) 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. Next, an incoming Attach for the Receive_Encap_MSG_TP will be received and either a REQACTPU or ACTPU 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 5-28 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, activating 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) 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 5-19 for more information on deactivation of the CP-SVR pipe.)

Figure 5-1 on page 5-9 through Figure 5-3 on page 5-16 illustrate the CP-SVR pipe activation scenarios described thus far.

Notes:

1. 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 6-7.
2. Refer to Table 3-1 on page 3-2 for an explanation of the notations used in the diagrams in this chapter.

5.5.2.1 DLUR-Initiated CP-SVR Pipe Activation

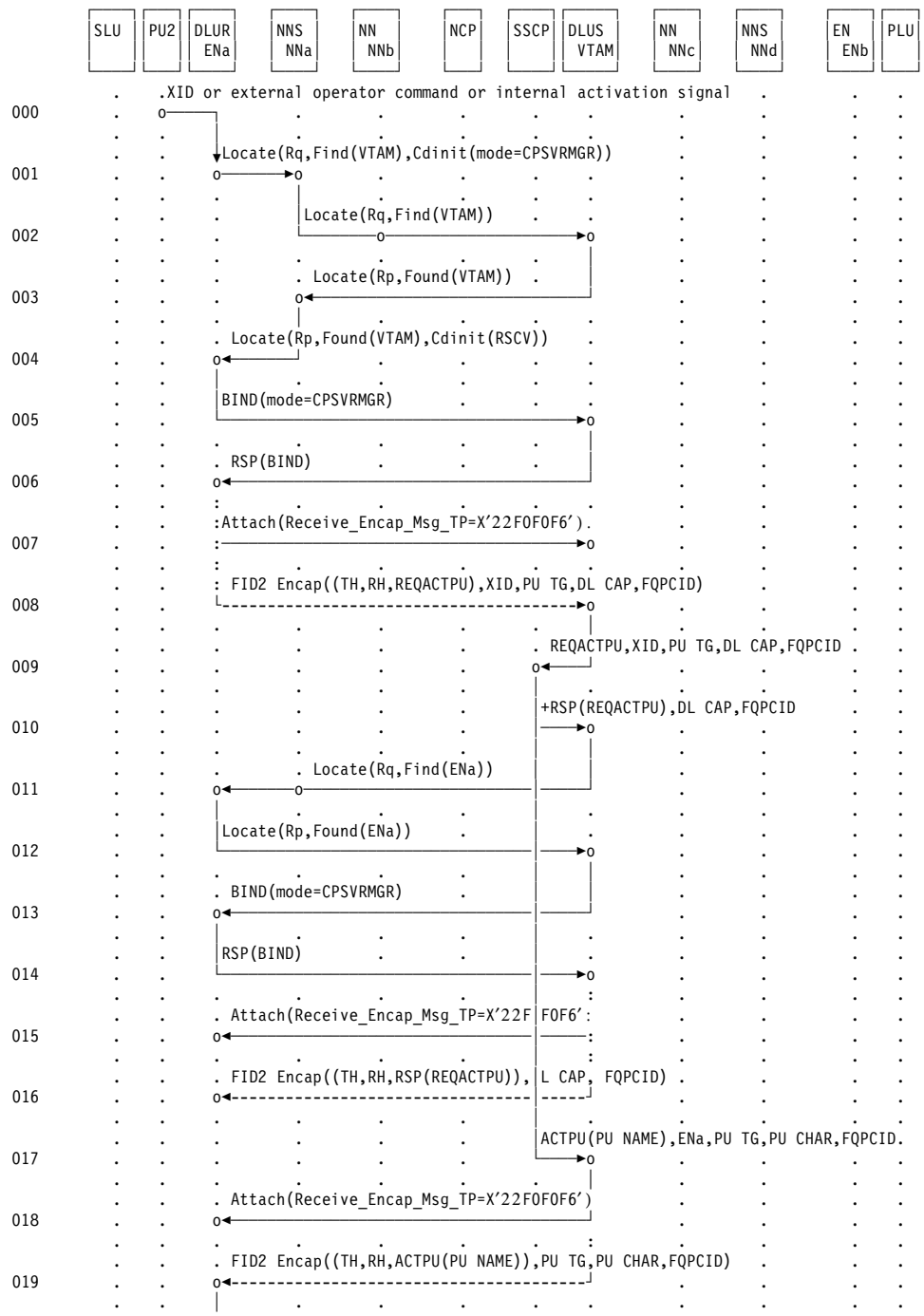
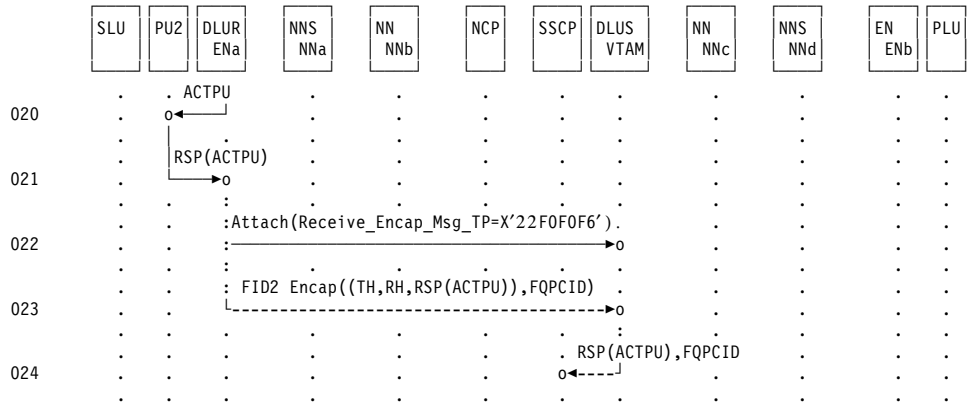


Figure 5-1. DLUR-initiated CP-SVR pipe activation (DLUS known)


SLU
PU2
DLUR ENa
NNS NNa
NN NNb
NCP
SSCP
DLUS VTAM
NN NNc
NNS NNd
EN ENb
PLU
ACTPU
020
RSP(ACTPU)
021
:Attach(Receive_Encap_Msg_TP=X'22F0F0F6').
022
: FID2 Encap((TH,RH,RSP(ACTPU)),FQPCID)
023
RSP(ACTPU),FQPCID
024

1. Upon receipt of an 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 CP name 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 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 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)
   - one or more of the following used to identify the PU:
     - 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 or for an external PU which does not support XID
     - the PU TG control vector(s), identifying the connection between the PU and the DLUR and made up of one or more of the following:
       - — a PU name (CV X'0E') control vector
       - — a signaling information (CV X'46') control vector - this could include some or all of the signaling information defined at 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 on 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 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 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 CP name 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 in response to the REQACTPU received from the
% DLUR. The ACTPU RU includes the name of the PU (which could be predefined at the DLUS, pro-
% vided by the DLUR, or dynamically created by the DLUS).

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 FID2 PIU along with the PU
% TG control vector (which can optionally include the DLUS-defined signaling information and/or an
% echo of the DLUR-defined signaling information), 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.

% • Before storing the PU name, if the DLUR sent a PU name to the DLUS with the REQACTPU, it
% should check to see if the same name came back on the ACTPU. If the DLUR receives a PU name
% on ACTPU different from the name sent with the REQACTPU, it may accept the DLUS-provided
% PU name (see 5.3.1, “DLUR PU Identification” on page 5-2 for details). This can occur, for
% example, when the DLUS does not recognize the PU name sent with the REQACTPU; if other
% information was present, it can find a match based on IDBLK/IDNUM or CP name and return the
% name of the matching predefined PU.

21. When the PU receives the ACTPU, it will respond to the DLUR component with an ACTPU response.

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) to the SSCP component of the DLUS node. The CP-SVR pipe and associated SSCP-PU session have now been activated.

5.5.2.2 DLUS-Initiated CP-SVR Pipe Activation

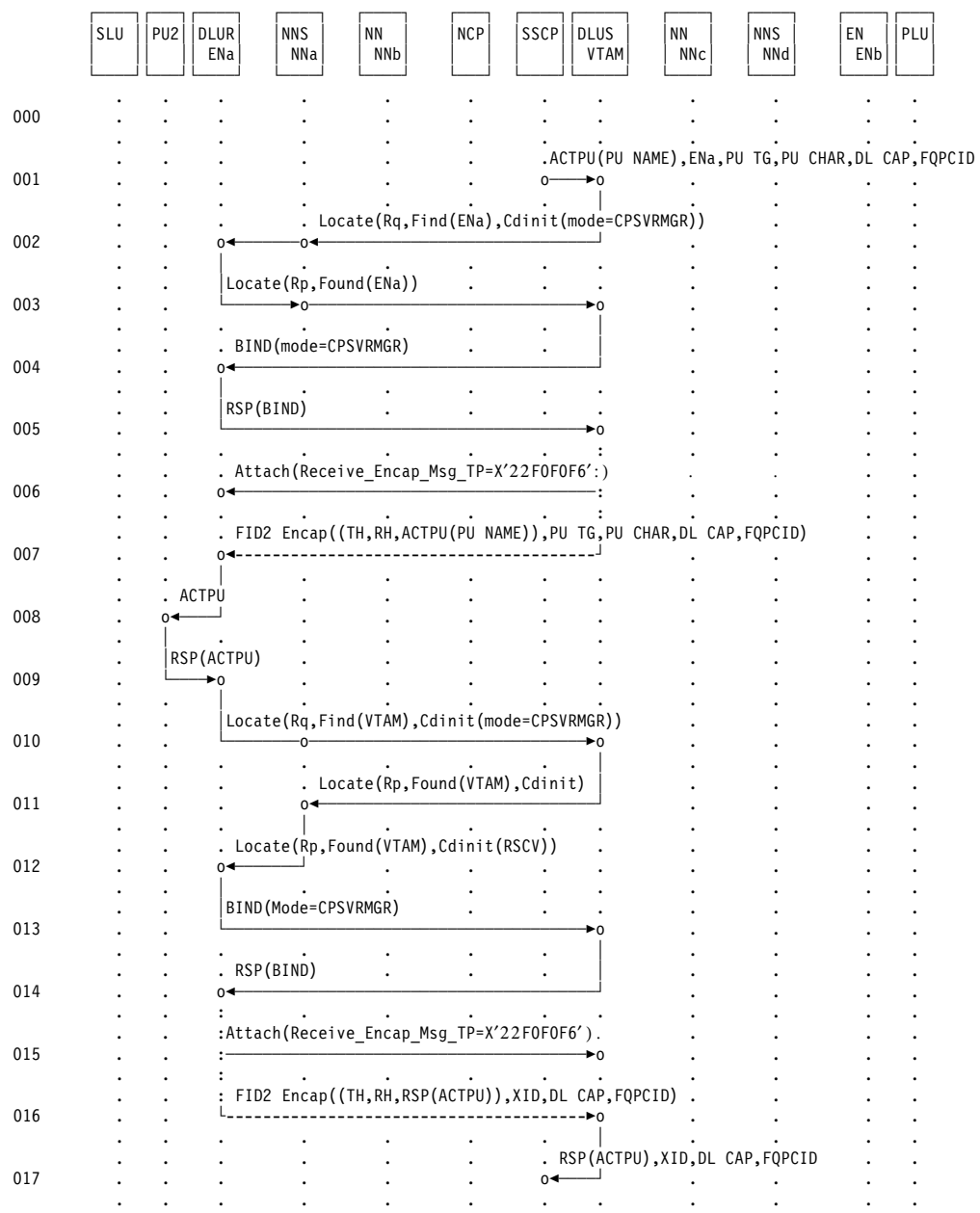


Figure 5-2. DLUS-initiated CP-SVR pipe activation

1. When a DLUS node wishes to activate a PU, the DLUS uses its own internal information (predefined 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 5-5 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
% (optionally including the DLUS-defined signaling information), 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 (optional) 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

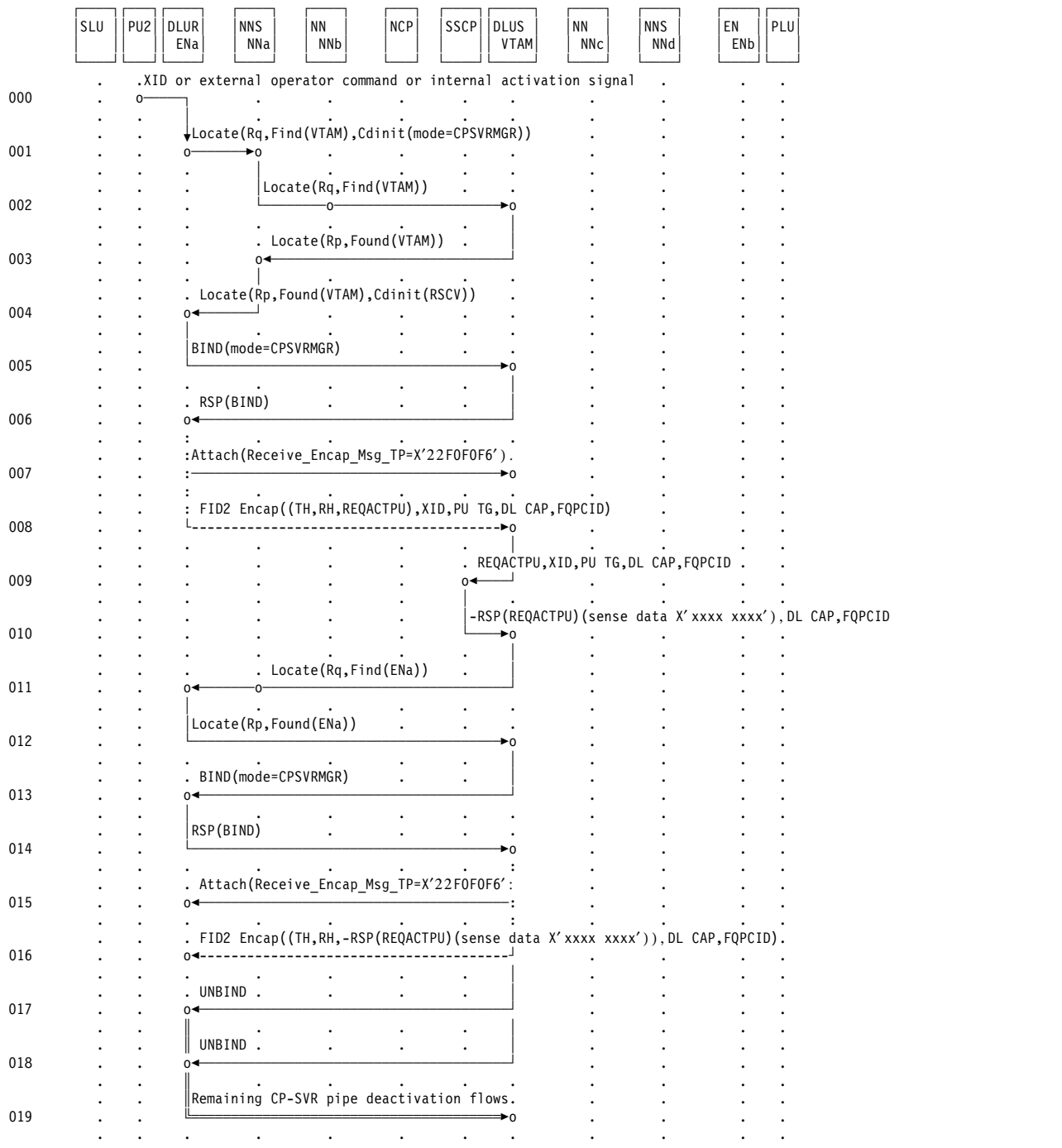


Figure 5-3. DLUR-initiated, DLUS-rejected CP-SVR pipe activation

1. Following a stimulus (flow 0), the DLUR node desires to activate a PU. To do this, the DLUR needs to request an ACTPU from some DLUS node. The DLUR uses predefined information to determine the DLUS node and allocates a conversation to the Receive_Encap_Msg_TP at that DLUS node. Since a conwinner session to that DLUS does not exist, 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 FID2 PIU for the DLUS node.
9. The Receive_Encap_Msg_TP at the DLUS node removes the encapsulation headers and passes the REQACTPU 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 data 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′).
    - % The DLUR-specified network name is known but is a duplicate resource (X′0888 0016′).
    - % The network ID in the Network Name control vector does not match the network ID of the target % resource of the REQACTPU (X′0891 000A′).
    - % The request was rejected due to constraints or policies specific to the receiving DLUS. The request % should not be retried (X′080A 000F′).

    The SSCP generates a negative response to the REQACTPU with appropriate sense data.
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.
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 5-19.

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 (TP) 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 RH will not be present in a FID2 message which is a middle or last segment - see 6.1.4, “Encapsulating Message Segments” on page 6-6 for more information about segmentation of messages on the CP-SVR pipe). 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′ flows 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 provides 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

% (***Note*** - As described in 5.3.1, "DLUR PU Identification" on page 5-2, a DLUR can identify a PU using a
% PU TG descriptor containing a PU name, signaling information, or both. The following section pertains to
% defining signaling information for an internal PU.)

Whether a DLUR node has a single PU or multiple PU images, if any of its PUs are internal (not downstream PUs), each internal PU will be uniquely identified by either the DLUR's CP name 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 CP name instead of IDBLK/IDNUM in the PU TG field.

SF X'92' will carry IDBLK/IDNUM and SF X'93' will carry CP name. The SF combinations that will be permitted when SF X'91' = INTPU are:

Table 6-1. Internal PU Identification

Option	**SF X'92'**	**SF X'93'**
1	internal PU's IDBLK/IDNUM	(not included)
2a	(not included)	CP name
2b	CP's IDBLK/IDNUM	CP name

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 variable X'1500' for:

- REQACTPU and ACTPU when PU activation is DLUR-initiated
- ACTPU when PU activation is DLUS-initiated

% This IDBLK/IDNUM may also be carried in the XID field (X'81') in the GDS variable 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 CP name will be carried in the PU TG field (CV X'4691' = INTPU, CV X'4693' = CP name) in the GDS variable X'1500' for:

- REQACTPU and ACTPU when PU activation is DLUR-initiated
- ACTPU when PU activation is DLUS-initiated

% When a CP name is carried in the PU TG field, the DLUR CP's IDBLK/IDNUM may be carried in the XID field (CV X'81') in the GDS variable 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 CP name will be carried in the PU TG field (CV X′4691′ = INTPU, CV X′4692′ = IDBLK/IDNUM, CV X′4693′ = CP name) in the GDS variable X′1500′ for:

- REQACTPU and ACTPU when PU activation is DLUR-initiated
- ACTPU when PU activation is DLUS-initiated

% When a CP name is carried in the PU TG field, the DLUR CP′s IDBLK/IDNUM may be carried in the XID field (CV X′81′) in the GDS variable 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. See 5.3.1, “DLUR
% PU Identification” on page 5-2 and 5.4.1, “New PU Definition Parameters” on page 5-5 for details about
% the search order used by a DLUR to find a match.

6.1.3 Addressing Per Stage

Much of this is derived from the Address Space Manager (ASM) chapter in *SNA APPN Architecture Reference*.

Local Form Session Identifiers (LFSIDs) are the addresses carried in FID2 THs; these addresses have three parts: Session Identifier High (SIDH) (1 byte), Session Identifier Low (SIDL) (1 byte), Origin Destination Assignment Indicator (ODAI) (1 bit).

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 predefined PU.***

In this case, the SSCP (DLUS node) has a predefined 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 5-5. 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 and send it to the downstream PU.

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

REQACTPU is a new message 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 (see 5.3.1, "DLUR PU Identification" on page 5-2 for more details) to search for a predefined 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.

Data link control (DLC) activation must precede the forwarding of 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., Request Network Address Assignment (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.

6.2.2 Dynamic PU Activation

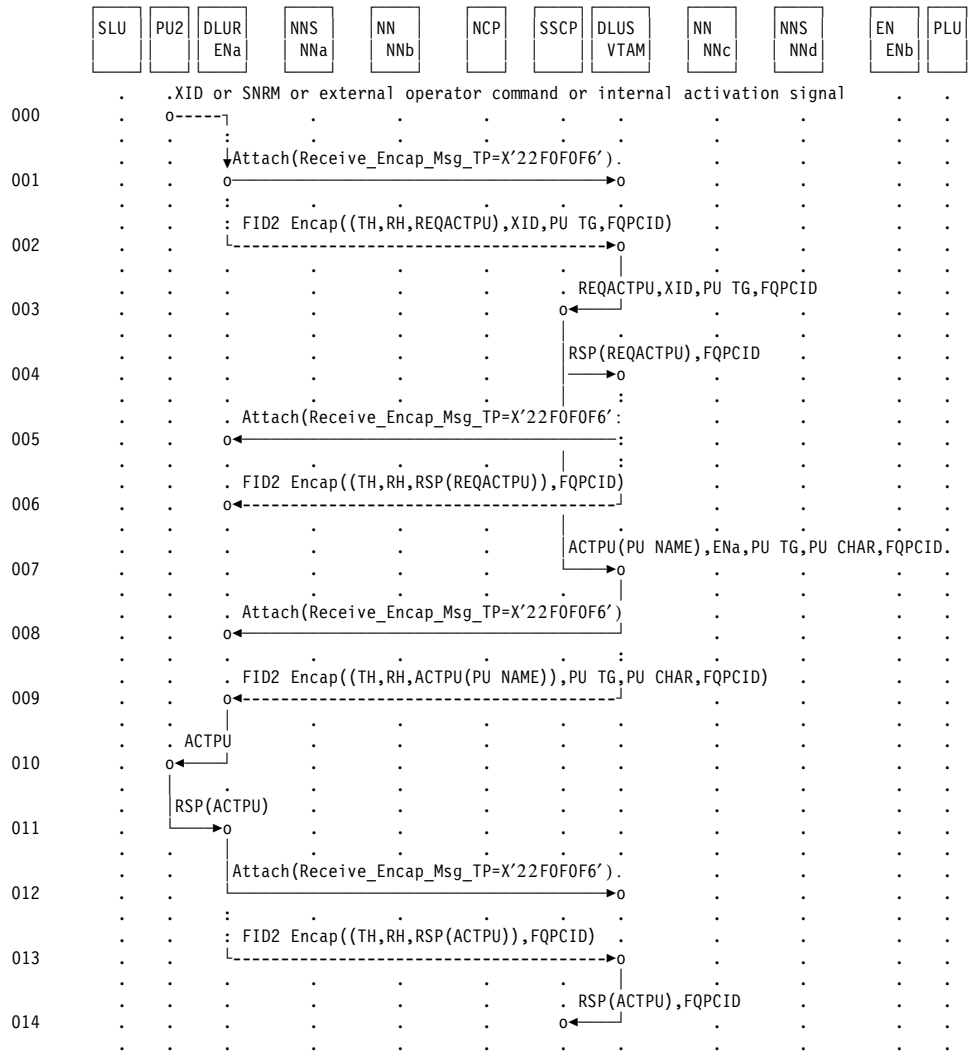


Figure 6-4. Dynamic PU activation

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 5-9). 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 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 may contain an XID0, XID1, or XID3 that will be used by the SSCP to search for a predefined 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 also may carry a PU TG identifier to describe the link attaching the downstream PU to the DLUR.

3. The DLUS component receives the encapsulated REQACTPU 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 predefined 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 3-5. 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 data 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 use the FQPCID to uniquely identify this SSCP-PU session.

Chapter 13. Format Changes

This section describes changes to RUs, control vectors, GDS variables, and sense data for Dependent LU Requester/Server architecture. The actual changes can be found in Appendix C, "Formats For DLUR/S Architecture" on page C-1. These changes will appear in the next edition of *Systems Network Architecture Formats*.

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 includes:

- 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 Network Name (X'0E') control vector (type X'F1' - PU name)
  - 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 is 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′0E′	PU TG	REQACTPU
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 contains either:

% • the I-field image of the XID0, XID1 or XID3 SDLC response of the external PU attached to the DLUR, or

• a DLUR-generated XID0 image for an internal PU or for an external PU which does not support XID.

13.1.2 ACTPU (Activate Physical Unit)

The Network Name (X′0E′) control vector (type X′F1′ - PU name) is 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 predefined or undefined to the SSCP ; it could be internal to the DLUR or downstream of the DLUR.

The REQACTPU has two formats:

- Format X′0′ - external PU
- Format X′1′ - internal PU

A positive response to REQACTPU indicates that the SSCP will send an ACTPU to the PU specified in the REQACTPU′s corresponding FID2 Encapsulation (X′1500′) GDS variable. A negative response to REQACTPU indicates that no ACTPU will be sent, and sense data will be included to specify the reason.

13.4 Sense Data

This section lists new sense data only; new uses of existing sense data 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′ - Upon receiving a Locate request for a DLUS-served dependent LU, the DLUS node has determined that at least one intersubnetwork TG on the path between the DLUS and the PLU was not between two Extended Border Nodes.

13.4.3 REQACTPU

% • X′080A 000F′ - The request was rejected because of constraints or policies specific to the receiving
% DLUS. The request should not be retried.

- X′0852 0002′ - A REQACTPU has been received by an SSCP which has already sent an ACTPU for the same PU.

% • X′0888 0016′ - The DLUR-specified network name is known but is a duplicate resource.

- 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′0891 000A′ - Invalid Network ID: the network ID in the Network Name control vector does not
% match the network ID of the target resource of the REQACTPU.

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

Command Parameters (X′80′) Found Control Vector

Byte	Bit	Content
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.
	3–4	Reserved
	5	Reserved
	6	Owning CP responder indicator: 0 The CP of the DLUS node owning the DLUS-served LU has responded to the Locate/Find. 1 The CP of 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)		Flag Extension Byte
	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.

C.1.24 FID2 Encapsulation (X′1500′) GDS Variable

C.1.24.1 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		GDS Variable Data
4–5		Length (m−3), in binary, of the FID2 PIU, including this Length field
6 – m		FID2 PIU

FID2 Encapsulation (X′1500′) GDS Variable

Byte	Bit	Content
m+1 – n		Control vectors, as described in "Control Vectors" in -- Heading ′CF′ unknown --. *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule LT (see "Substructure Encoding/Parsing Rules" in -- Heading ′CF′ unknown --). X′0E′ Network Name control vector (contains the name of the PU, type X′F1′; see Note) X′25′ Security ID Control control vector (see Note) X′30′ Assign LU Characteristics control vector (see Note) X′43′ Extended SDLC Station control vector (see Note) X′46′ TG Descriptor control vector (see Note) X′51′ DLUR/S Capabilities control vector (see Note) X′60′ Fully Qualified PCID control vector (always present) X′81′ XID Image control vector (see Note) *Note:* See the "Format Changes" chapter in *APPN Dependent LU Requester Architecture Reference* for the conditions of presence for the above control vectors in this GDS variable.

C.1.24.2 FID2 Encapsulation Control Vectors

C.1.24.2.1 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′ (see "Substructure Encoding/Parsing Rules" in -- Heading ′CF′ unknown --)
2 – n		XID I-field image: the bytes received in the information field of the XID response

C.2 Sense Data

C.2.1 0801

0801 Resource Not Available: The LU, PU, link station, 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:

0000 No specific code applies.

C.2.3 080A

080A Permission Rejected: The receiver has denied an implicit or explicit request of the sender.

When sent in response to BIND, it implies either that the secondary LU will not notify the SSCP when a BIND can be accepted, or that the SSCP does not recognize the NOTIFY vector key X′0C′. (See the X′0845′ sense code for a contrasting response.)

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

0000 No specific code applies.

0001 An SSCP has denied permission to establish a session through its resources; the receiving SSCP should not attempt to reroute the request to another SSCP.

0002 An SSCP has denied permission to establish a session through its resources; the receiving SSCP should attempt to reroute the request to another SSCP.

0005 An SSCP has denied permission to continue session setup. A DLU name was presented to an alias selection function (authorization application) that rejected it. The receiving SSCP should not attempt to reroute the request to another SSCP.

0006 An SSCP has denied permission to continue session setup. A DLU name was presented to an alias selection function (authorization application) that rejected it. The receiving SSCP should attempt to reroute the request to another SSCP.

0007 An activation request was received for a resource that has a NETID different from that of the requesting SSCP, and the requesting SSCP indicated previously that it does not support this configuration.

0008 The request specified in the Request Change Control MS major vector was rejected because it did not originate from a valid focal point.

0009 The request specified in the Request Change Control MS major vector was rejected because the ability to support it has been disabled at the receiver.

000A The request was rejected because it would prohibit compliance with the status-reporting requirements specified in the Reporting Level MS Common subvector.

000B The request was rejected because the second-level application, though recognized, operates under the control of a program other than that which has received and is to forward the request to that second-level application program.

000C The request was rejected because the timer/clock at the receiver is protected and cannot be set by the request sender.

000D An SSCP or CP has denied a Locate search request. The receiving SSCP or CP should attempt to reroute the request.

000F The request was rejected because of constraints or policies specific to the receiving DLUS. The request should not be retried.

C.2.4 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:

0000 No specific code applies.

0001 Invalid Request: The specified link station has already received a CONTACT and is therefore under the control of another SSCP. This CONTACT would exceed the share limit (=1).

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).

C.2.5 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:

0000 If the RU is an ACTPU or ACTCDRM, it means that a session has already been activated for the subject destination-origin pair by a session activation request that carried a larger activation request identifier than the current request; the current request is refused.

If the RU is a BIND, it means that the BIND request was received with the same session instance identifier (in the structured subfield X′03′ of the User Data field) as an active session′s; the current request is refused.

0001 A second BIND has been received from a peripheral node PLU while the session was still in the activation process.

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

C.2.6 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.

C.2.7 0888

0888 Name Conflict: A name specified in an RU conflicts with a previous usage, or is unknown, or is known and does not have the required capabilities, or is a duplicate resource for the specified resource type. When a name conflict is detected, further name checking ceases; multiple name conflicts are not reported or detected.

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

0000 No specific code applies.

0001 The specified DLU real network name is known, but identifies a resource that is not LU-LU session capable.

0002 The specified DLU alias network name is known, but identifies a resource that is not LU-LU session capable.

0003 The specified OLU real network name is known, but identifies a resource that is not LU-LU session capable.

0004 The specified OLU alias network name is known, but identifies a resource that is not LU-LU session capable.

0005 Name translation was invalid; that is, a different LU name was returned with the same network ID as the original LU name.

0006 The specified DLU real network name is known, but is a duplicate resource.

0007 The specified DLU alias network name is known, but is a duplicate resource.

0008 The specified OLU real network name is known, but is a duplicate resource.

0009 The specified OLU alias network name is known, but is a duplicate resource.

000B A cross-network DLU name is defined as a shadow resource, but shadow resources are not supported for cross-network sessions.

000C Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

000D When processing a session initiation RU, an SSCP has found two different resource definitions for the OLU, one with the real OLU name and one with the alias OLU name.

000E When processing a session initiation RU, an SSCP has found two different resource definitions for the DLU, one with the real DLU name and one with the alias DLU name.

000F The specified DLU network name is defined as a generic resource. The session should be reinitiated using the name of an LU.

0010 The LU6.2 partner returned a name in the User Data field of its RSP(BIND) that differs from the name it returned in the User Data field of its RSP(BIND) for a previous BIND. Either the partner changed its name or name changes in the network have caused delivery of the latest BIND to a different partner.

0011 The LU6.2 partner receiving a BIND carrying one specific target SLU name returned a name in the User Data field of its RSP(BIND) that is the same as it returned in response to a previous BIND carrying a different target SLU name. Name changes in the network name allowed two names to resolve to the same LU.

0012 The network qualifier of the name returned in the User Data field of a RSP(BIND) is not equal to the network identifier provided by the application that is using network qualified names. Name changes in the network have caused alteration of the network identifier.

0015 A generic name of a resource has been received when only the real name of the resource can be specified.

0016 The DLUR-specified network name is known but is a duplicate resource.

C.2.8 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:

0000 A resource encountered during LU-LU session initiation is not ENA-capable; the session initiation request may be rerouted.

0001 A resource encountered during LU-LU session initiation is not ENA-capable; the session initiation request should not be rerouted.

0002 An SSCP has requested a "pre-ENA compatible" SLU address for an SLU that already has an ENA address.

0003 The gateway node selected by the gateway SSCP from the gateway node list is not ENA-capable when an ENA-capable gateway node is required. Another gateway node may be tried.

0004 During a dynamic path update, the SSCP detected that the update contained a path definition with an ER number greater than 7 and that the target node does not support extended subarea addresses. Therefore, the dynamic path update information for this destination subarea was not forwarded to the target node.

0037 Unknown TG Vectors to Dependent LU Requester: 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.

0038 Too Many Directed Search Subprocedures: A Locate search exceeded the maximum height of the search tree; too many directed search subprocedures were tried; no retry.

0040 Resource Not Found during a Broadcast Search: A Locate GDS variable for a broadcast search was received by a CP that does not provide network services for the search argument resource and neither do any of the CPs searched in its broadcast subtree. This condition is detected by crossing search requests (a CP sends and receives a search request with the same FQPCID and the same search argument resource) or by a local search failure and all CPs in the broadcast subtree returning this sense data.

0048 Neutral Reply Received from an End Node: A Locate reply with no Found and no Extended Sense Data (X′35′) control vector was received from an APPN end node.

0050 Quiesced CP: A CP in the broadcast search tree is in a quiescent state and, therefore, not receiving Locate GDS variables. This condition is detected when a CP in the search subtree is quiesced and no other CP in the subtree found the requested resource.

0060 Storage Not Available: A CP in the broadcast search tree does not have sufficient storage to participate in the search and no other CP in the search subtree found the requested resource.

0070 Session Outage: A CP in the search tree has lost its CP-CP session with a CP that had been sent a Locate GDS variable and no reply had been received.

0080 Duplicate Fully Qualified PCID: A CP in the search tree detected a duplicate fully qualified PCID for a different session request from the session request that first used the fully qualified PCID.

0081 PCID Modifier Too Long: A PCID Modifier List was received that had a length greater than 10 bytes.

0082 PCID Modifier Space Exhausted: A PCID Modifier List was received that contained the maximum of 10 bytes. As the maximum list size has been reached, another list entry cannot be made that was longer than 10 bytes.

C.2.10 0891

0891 Invalid or Missing Network ID.

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

0000 No specific code applies.

0001 PLU Network ID Invalid: The network ID of the PLU is not the same as that of the SSCP(PLU).

0002 Invalid Network ID: The Network ID field in CONNOUT does not match the network ID defined for the target of the CONNOUT.

0003 Invalid network ID: The Network ID field in the RNAA is not the same as the native network ID because of a mismatch between the system definitions of the SSCP and the type 4 node.

0004 The Network Name control vector appended to the received XID3 does not contain a valid network ID. The network ID preceding the CP name must be greater than 0 and less than 8 bytes in length.

Alternatively, a network ID was received as an entry in a Register GDS variable without an accompanying resource name, resulting in an invalid resource name at the receiver; the entry was not registered.

0005 The Network Name control vector appended to the received XID3 does not contain a valid CP name. The CP name, following the network ID, must be greater than 0 and less than 8 bytes in length.

0006 Invalid Network ID: The sender has deactivated CP-CP sessions with the adjacent nonnative CP because neither CP contains border node support (i.e., neither sets byte 9, bit 7, to 1 in the CP Capabilities GDS variable that it sends).

0007 Invalid Network ID: Establishment of a switched link connection failed because the network ID of the destination PU was not equal to that of the requesting SSCP.

000A Invalid Network ID: the network ID in the Network Name control vector does not match the network ID of the target resource of the REQACTPU.

C.2.11 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:

0000 No specific code applies.

0001 The identified LU-LU session had to be deactivated because of a forced deactivation of the associated SSCP-PU or SSCP-LU session, for example, because of a received DACTPU, DACTLU, or DISCONTACT. The LU will send an UNBIND with a reason code of X′0A′ (SSCP Gone).

0002 The LU or session connection manager (SCM) will send UNBIND with a reason code of X′0F′ (Cleanup).