

33 OSI Primitives on the Protocol Spreadsheet

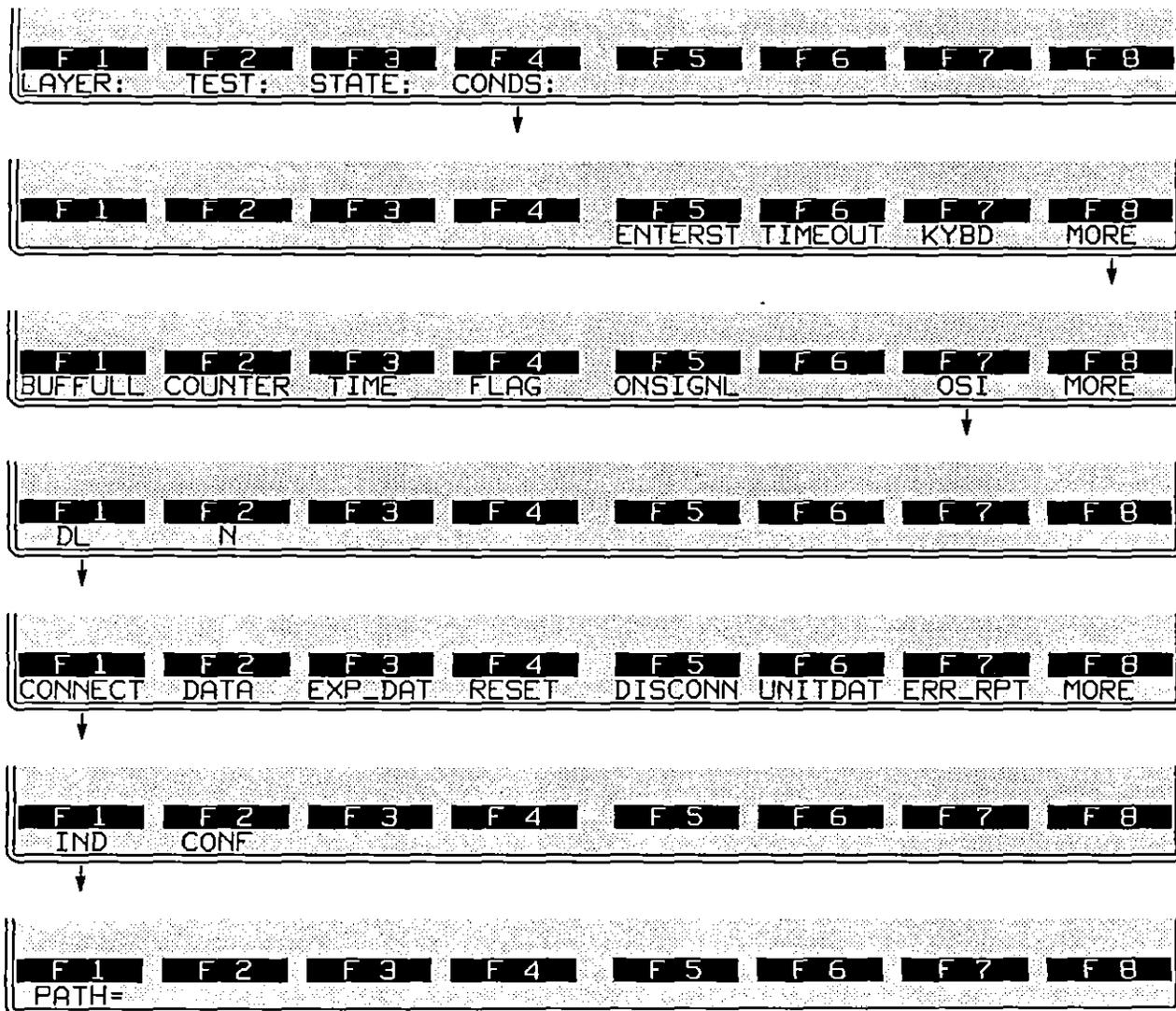


Figure 33-1 The softkey path for a *DL_CONNECT_IND* condition primitive at Layer 3.

33 OSI Primitives on the Protocol Spreadsheet

Primitives are defined in the Open Systems Interconnect (OSI) Reference Model as protocol-independent interactions between adjacent layers of the model.

For example, data that comes into the INTERVIEW at Layer 1 or starts down the OSI "ladder" at a layer above Layer 1 is stored in a structure called an *IL* (interlayer) buffer. This buffer is passed between layers along with a primitive data unit (PDU), or primitive.

Since primitives are layer-specific, they are not available on the Trigger menus which offer conditions and actions at Layer 1. You must use the Protocol Spreadsheet to send, receive, or monitor primitives.

The Protocol Spreadsheet is divided into seven layers in accordance with the OSI model. By giving the operator control of the boundaries between these layers, primitives make layered programming possible.

Primitives for a given OSI layer may be entered in the Protocol Spreadsheet whether or not a protocol personality package is loaded in for that layer. Table 33-2 lists the primitives that may be entered on the current Protocol Spreadsheet. Due to the uncomplicated, "always-connected" nature of the RS-232/V.24, V.35, and RS-449 interfaces, Layer 1 primitives are automatic and do not appear on the Protocol Spreadsheet for that layer. OSI service for Layers 2 through 7 currently is available.

NOTE: Unless a Layer 1 package (such as DDCMP) is loaded in, primitives are not available when Format: SYNC is selected on the Line Setup screen.

On the Protocol Spreadsheet, primitives take the form of conditions and actions. A condition primitive monitors the layer boundaries for action primitives that are sent down from above or up from below. An action primitive at any layer is sent either up or down to the next layer. Each primitive is shared by two layers. DL_CONNECT IND, for example, is an action primitive at Layer 2. At Layer 3, the same primitive is a condition. The prefix (DL) is an abbreviation for the name of the lowermost layer (Data Link) which shares the primitive. Table 33-1 lists all primitive prefixes and the layers which share them.

Table 33-1
Primitive Prefixes and Associated Layers

Prefix	Lowest Layer of Operation	Shares with Layer
PH	Physical (Layer 1)	2
DL	Data Link (Layer 2)	3
N	Network (Layer 3)	4
T	Transport (Layer 4)	5
S	Session (Layer 5)	6
P	Presentation (Layer 6)	7

33.1 Softkey Selections

The condition and action primitives specific to a given layer will be arrayed on softkeys that appear when you press the softkey for OSI. OSI is **F7** on the second rack of condition softkeys. Figure 33-1 shows the softkey path to an OSI condition primitive at Layer 3. OSI is **F1** on the third rack of action softkeys. Layer-specific softkey racks corresponding to the following general categories appear successively as selections are made:

(A) Direction

Indicate the direction from which the condition primitive will come. At Layer 3, for example, the first choice (DL) will detect primitives handed up from the layer below; the second selection (N) will detect primitives handed down from the layer above. As an action, you select the direction which you wish the primitive to go: the first choice (DL) sends the primitive to the layer below; the second selection (N) sends the primitive to the layer above.

(B) Type

Choose among the primitive types offered at each layer. Each layer has its own set of primitives, but they all can be grouped into four major phases: establishment, data transfer, release, and debug and error reporting.

In the establishment phase, a layer establishes a connection with the layer above and/or below. The activate and connect primitives provide this function. Data transfer is accomplished via the data, expedited data, and reset primitives. Deactivate and disconnect primitives break the connection between layers in the release phase. Debug and error reporting primitives include debug, error report, and unitdata.

(C) Request/Response

For some primitives, you must indicate whether you are searching for—or sending—a request (REQ or IND) or a response (RESP or CONF). INDications and CONFirms come from the layer below or go to the layer above; REQuests and RESPonses come from the layer above or go to the layer below.

(D) Path

Provide a path, if necessary. Interlayer primitives must handle channel or “path” information in order to insure that data moving down from Layer 4 is given the correct logical channel at Layer 3, or that data moving from Layer 3 to Layer 2 bears the correct frame address when it goes out on the data link.

A softkey sequence that leads to the PATH= selection for a primitive on the Protocol Spreadsheet is shown in Figure 33-2.

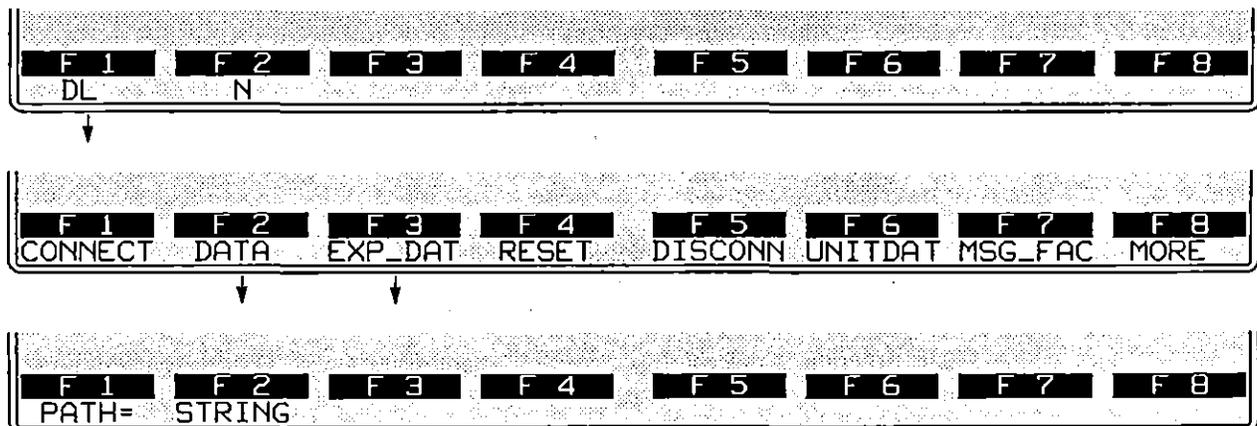


Figure 33-2 DATA and EXPEDITE_DATA action-primitives may carry path information as well as a data string.

Refer to Section 37.1(E) for a discussion of how paths are tied to call parameters (and directly or indirectly to logical channel numbers) via user entries on the Packet Level Setup screen (Figure 37-2) at X.25 Layer 3.

Primitive paths are only an important consideration when more than one layer is multiaddress or multichannel. In that situation, the vertical path numbers should match. Layer 3 might provide several logical channels, for example, while Layer 2 services more than one link address. When a set of call parameters is assigned by the user to path #1 at Layer 3, path #1 on the setup screen at Layer 2 should reference the appropriate link address for that call.

Remember that data primitives along with their path parameters usually are handled automatically (see Section 34). Automatic primitives will carry the same path information as the SEND or GIVE_DATA actions that generated them.

(E) String

Optional *strings* may be added to DATA or EXPEDITED_DATA action-primitives at any layer. A string is external data that is referenced in the list node of an interlayer buffer. (See Figure 66-1.) This buffer is passed with the selected primitive. One special use of the string field is to identify an IL buffer that has just been handed down from above. The macro N_DATA (or T_DATA or PH_DATA) enclosed in double parens in a data-primitive string field will identify the buffer that was just received from above. When the current action primitive is processed, the IL buffer will be passed to the layer below. One softkey sequence leading to a string selection is given in Figure 33-2. Always enclose a string in double quotation marks.

Here is an example of a data primitive at Layer 4 passing a string down to the next layer below:

```
LAYER: 4
  STATE: transport
  CONDITIONS: KEYBOARD " "
  ACTIONS: N_DATA REQ "«(FOX)»"
LAYER:3
  STATE: network
  CONDITIONS: N_DATA REQ
  ACTIONS: SEND DATA PATH= 0 "«(N_DATA)»"
LAYER:2
  STATE: datalink
  CONDITIONS: DL_CONNECT REQ
  ACTIONS: DL_CONNECT CONF
  CONDITIONS: DL_DATA REQ
  ACTIONS: SEND INFO "«(DL_DATA)»"
```

This program is designed as a "quick" demonstration of OSI service primitives and will transmit a fox message out on the interface (and display it on the INTERVIEW screen) whether or not an actual link and call have been established. (Layer packages must of course be loaded in at Layers 2 and 3.) Note the following:

- The action at Layer 4 forces the fox message down to Layer 3.
- The SEND DATA action at Layer 3 adds an appropriate Layer 3 header to whatever data is referenced in the action.
- The string that contains the macro «(N_DATA)» indicates that the Layer 3 header should be copied into the IL buffer that was passed with the N_DATA primitive from Layer 4.
- The same SEND action at Layer 3 triggers an automatic DL_CONNECT REQUEST primitive, since Layer 3 does not send *packets* to Layer 2 unless the link has been established.
- The Layer 2 program bypasses link-establishment by forcing a DL_CONNECT CONFIRM primitive up to 3.

- Now the data packet can be passed down to Layer 2, where a SEND INFO action inserts a frame header in the buffer received (in a DL_DATA REQ primitive) from Layer 3.
- Layer 1 primitives are automatic.

The fox message will also be transmitted if DATA REQUEST primitives are used instead of SEND actions at Layers 2 and 3 (or if no protocol packages are loaded); but the data in that case will not receive protocol headers.

33.2 Sample Primitives: CONNECT INDs and CONNECT REQs

Figure 33-3 and Figure 33-4 illustrate the flow of "connect" primitives between Layer 2 and Layer 3. The primitives in the figures are the labeled arrows positioned between the layers.

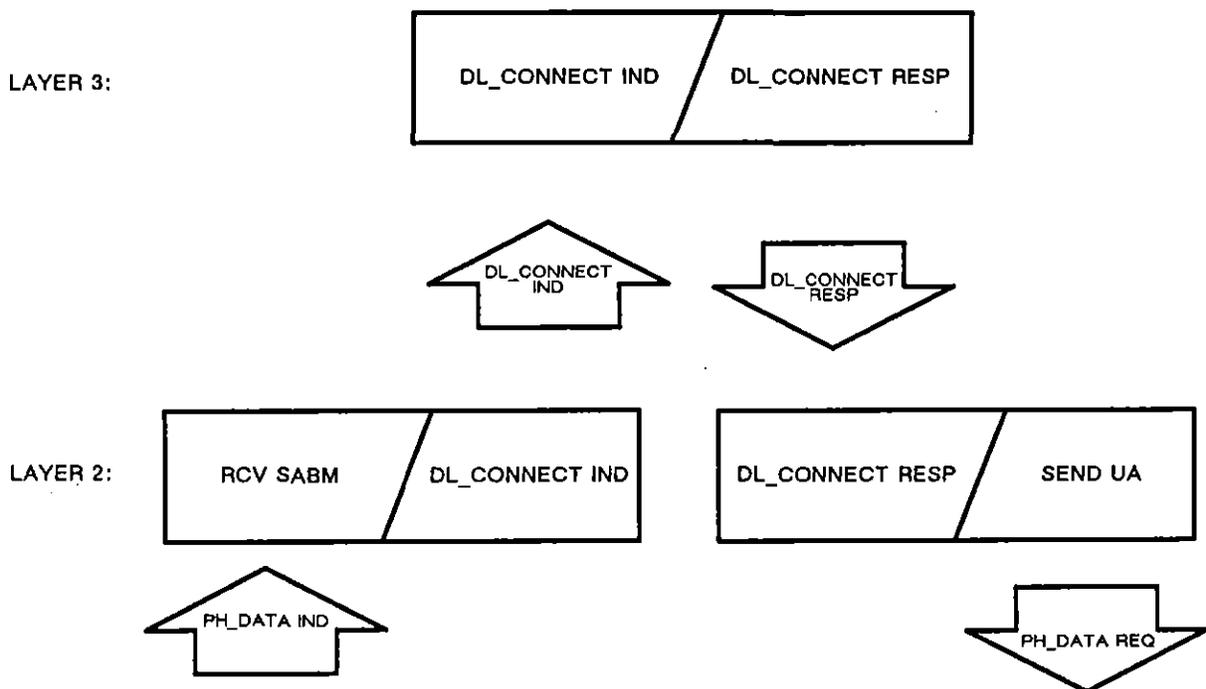


Figure 33-3 The (arrow-shaped) primitives moving between Layers 2 and 3 are intended to satisfy Layer 2 that a Layer 3 entity really is "up there." (The three rectangles contain spreadsheet conditions and actions.)

In Figure 33-3, Layer 2 receives a Set Mode command (SABM) from the data link. Before it responds positively (UA) to this command, Layer 2 passes up a DL_CONNECT IND primitive in order to verify that a Layer 3 entity really is "up there." When the active status of a Layer 3 entity is confirmed, Layer 2 sends the positive response (UA) down to Layer 1 and out onto the link to invite its Layer 2 peer to begin transferring data (Info frames).

The spreadsheet block that accomplished this exchange of primitives would be the following:

```

LAYER: 2
STATE: establish_link
CONDITIONS: RCV SABM
ACTIONS: DL_CONNECT IND
CONDITIONS: DL_CONNECT RESP
ACTIONS: SEND UA
LAYER: 3
STATE: dl_connect
CONDITIONS: DL_CONNECT IND
ACTIONS: DL_CONNECT RESP
    
```

In Figure 33-4, the request for confirmation of an adjacent layer is downward. Layer 3 wishes to send a Restart packet to the Layer 3 entity on the other side of the link; but it doesn't want to pass the packet down to Layer 2 if there is no mechanism at that layer to handle it. So Layer 3 precedes the Restart packet with a DL_CONNECT REQ primitive.

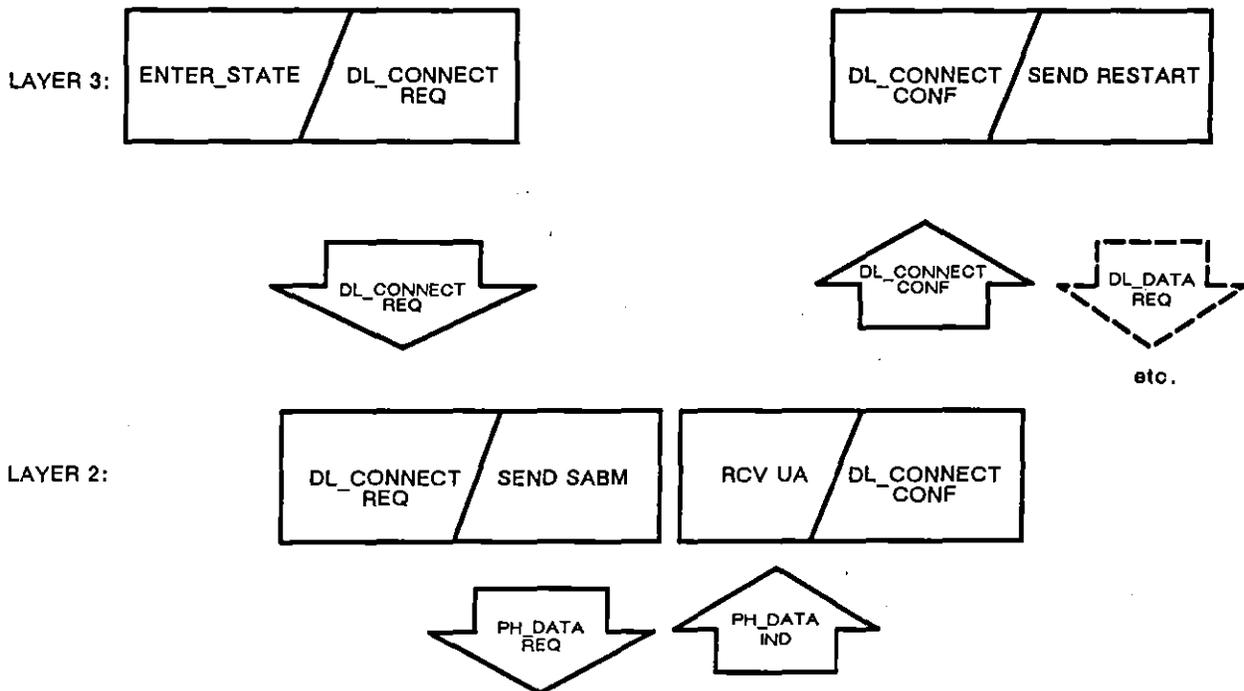


Figure 33-4 Layer 3 uses connect primitives to be sure that the Layer 2 entity below has established a link.

In the scenario illustrated in the figure, Layer 2 has not yet established the link. It responds to the connect-request primitive by sending a SABM, the X.25 command that initiates "connection" between link-level peers. When the SABM is acknowledged in a UA response, Layer 2 gives the DL_CONNECT CONF primitive up to Layer 3.

33.3 Sample Primitives: DATA INDs and DATA REQs

Figure 33-5 illustrates the primitives that are generated and monitored by the Protocol Spreadsheet when data is passed in both directions through an intermediate protocol layer (Layer 2). In this example, X.25 is the protocol package loaded in for both Layer 2 and Layer 3. Here a call-request packet is passed up through Layer 2 and received at Layer 3, and a call-confirm packet is sent down by Layer 3. The primitives in the figure are the labeled arrows positioned between the layers.

Data is in the form of physical-layer (PH) data when it moves in either direction between Layer 1 and Layer 2. PH_DATA primitives control the movement of this data. In between Layers 2 and 3, the data takes the form of data-link-layer (DL) data, with DL_DATA primitives responsible for data-delivery.

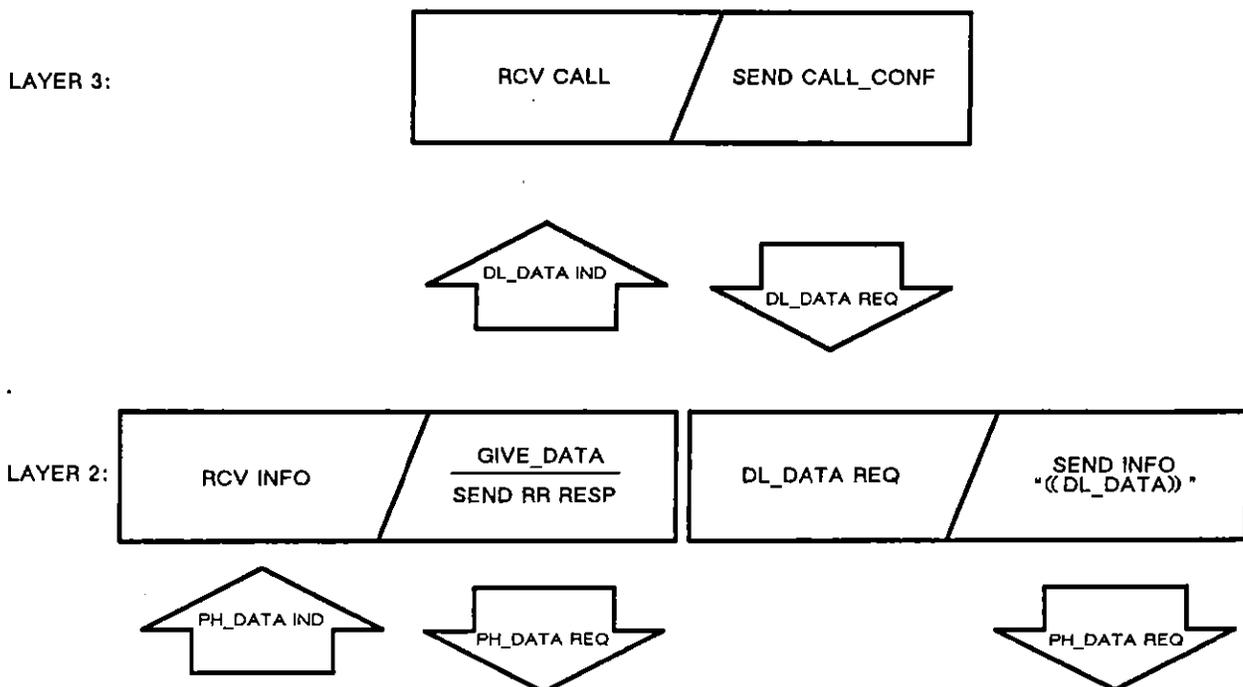


Figure 33-5 These (arrow-shaped) primitives are generated and monitored at Layer 2 when Layer 3 receives and sends data. (The three rectangles contain X.25 conditions and actions.)

The PH and DL versions of the data are not exactly the same. DL data has one less layer of protocol attached to it. In the example in Figure 33-5, the Layer 2 protocol was stripped off by the GIVE_DATA action when the call-request packet was being passed upward. On the call-confirm packet's trip down through the layers, Layer 2 protocol was added to the DL data by the SEND action—thus yielding PH data.

Note that "DL data" refers to data that moves above the DL layer (Layer 2), not below it. "DL data" can be taken literally to mean that as far as the DL layer is concerned, this is pure data, with no protocol that is recognizable at Layer 2.

When data is being passed upward, the primitives that signal the data are called indications (DATA INDS). When data is sent downward, the primitives at each layer are termed requests (DATA REQS).

Table 33-2
OSI Service Primitives

Layer	Conditions		Actions	
	From Layer Below	From Layer Above	To Layer Below	To Layer Above
7	P_CONNECT IND P_CONNECT CONF P_DATA IND P_EXPEDITED_DATA IND P_RELEASE IND P_RELEASE CONF P_UNITDATA IND P_ERROR_REPORT IND P_MGT_FACILITY IND P_DEBUG IND P_TD_DATA IND P_RD_DATA IND P_TD_EXPEDITED_DATA IND P_RD_EXPEDITED_DATA IND P_TD_UNITDATA IND P_RD_UNITDATA IND		P_CONNECT REQ P_CONNECT RESP P_DATA REQ P_EXPEDITED_DATA REQ P_RELEASE REQ P_RELEASE RESP P_UNITDATA REQ P_MGT_FACILITY REQ P_DEBUG REQ	
6	S_CONNECT IND S_CONNECT CONF S_DATA IND S_EXPEDITED_DATA IND S_RELEASE IND S_RELEASE CONF S_UNITDATA IND S_ERROR_REPORT IND S_MGT_FACILITY IND S_DEBUG IND S_TD_DATA IND S_RD_DATA IND S_TD_EXPEDITED_DATA IND S_RD_EXPEDITED_DATA IND S_TD_UNITDATA IND S_RD_UNITDATA IND	P_CONNECT REQ P_CONNECT RESP P_DATA REQ P_EXPEDITED_DATA REQ P_RELEASE REQ P_RELEASE RESP P_UNITDATA REQ P_MGT_FACILITY REQ P_DEBUG REQ	S_CONNECT REQ S_CONNECT RESP S_DATA REQ S_EXPEDITED_DATA REQ S_RELEASE REQ S_RELEASE RESP S_UNITDATA REQ S_MGT_FACILITY REQ S_DEBUG REQ	P_CONNECT IND P_CONNECT CONF P_DATA IND P_EXPEDITED_DATA IND P_RELEASE IND P_RELEASE CONF P_UNITDATA IND P_ERROR_REPORT IND P_MGT_FACILITY IND P_DEBUG IND P_TD_DATA IND P_RD_DATA IND P_TD_EXPEDITED_DATA IND P_RD_EXPEDITED_DATA IND P_TD_UNITDATA IND P_RD_UNITDATA IND

Table 33-2 (Continued)

Layer	Conditions		Actions	
	From Layer Below	From Layer Above	To Layer Below	To Layer Above
5	T_CONNECT IND T_CONNECT CONF T_DATA IND T_EXPEDITED_DATA IND T_DISCONNECT IND T_UNITDATA IND T_ERROR_REPORT IND T_MGT_FACILITY IND T_DEBUG IND T_TD_DATA IND T_RD_DATA IND T_TD_EXPEDITED_DATA IND T_RD_EXPEDITED_DATA IND T_TD_UNITDATA IND T_RD_UNITDATA IND	S_CONNECT REQ S_CONNECT RESP S_DATA REQ S_EXPEDITED_DATA REQ S_RELEASE REQ S_RELEASE RESP S_UNITDATA REQ S_MGT_FACILITY REQ S_DEBUG REQ	T_CONNECT REQ T_CONNECT RESP T_DATA REQ T_EXPEDITED_DATA REQ T_DISCONNECT REQ T_UNITDATA REQ T_MGT_FACILITY REQ T_DEBUG REQ	S_CONNECT IND S_CONNECT CONF S_DATA IND S_EXPEDITED_DATA IND S_RELEASE IND S_RELEASE CONF S_UNITDATA IND S_ERROR_REPORT IND S_MGT_FACILITY IND S_DEBUG IND S_TD_DATA IND S_RD_DATA IND S_TD_EXPEDITED_DATA IND S_RD_EXPEDITED_DATA IND S_TD_UNITDATA IND S_RD_UNITDATA IND
4	N_CONNECT IND N_CONNECT CONF N_DATA IND N_DATA_ACK IND N_EXPEDITED_DATA IND N_RESET IND N_RESET CONF N_DISCONNECT IND N_UNITDATA IND N_ERROR_REPORT IND N_MGT_FACILITY IND N_DEBUG IND N_TD_DATA IND N_RD_DATA IND N_TD_EXPEDITED_DATA IND N_RD_EXPEDITED_DATA IND N_TD_UNITDATA IND N_RD_UNITDATA IND	T_CONNECT REQ T_CONNECT RESP T_DATA REQ T_EXPEDITED_DATA REQ T_DISCONNECT REQ T_UNITDATA REQ T_MGT_FACILITY REQ T_DEBUG REQ	N_CONNECT REQ N_CONNECT RESP N_DATA REQ N_DATA_ACK REQ N_EXPEDITED_DATA REQ N_RESET REQ N_RESET RESP N_DISCONNECT REQ N_UNITDATA REQ N_MGT_FACILITY REQ N_DEBUG REQ	T_CONNECT IND T_CONNECT CONF T_DATA IND T_EXPEDITED_DATA IND T_DISCONNECT IND T_UNITDATA IND T_ERROR_REPORT IND T_MGT_FACILITY IND T_DEBUG IND T_TD_DATA IND T_RD_DATA IND T_TD_EXPEDITED_DATA IND T_RD_EXPEDITED_DATA IND T_TD_UNITDATA IND T_RD_UNITDATA IND

Table 33-2 (Continued)

Layer	Conditions		Actions	
	From Layer Below	From Layer Above	To Layer Below	To Layer Above
3	DL_CONNECT IND DL_CONNECT CONF DL_DATA IND DL_EXPEDITED_DATA IND DL_RESET IND DL_RESET CONF DL_DISCONNECT IND DL_UNITDATA IND DL_ERROR_REPORT IND DL_MGT_FACILITY IND DL_DEBUG IND DL_TD_DATA IND DL_RD_DATA IND DL_TD_EXPEDITED_DATA IND DL_RD_EXPEDITED_DATA IND DL_TD_UNITDATA IND DL_RD_UNITDATA IND	N_CONNECT REQ N_CONNECT RESP N_DATA REQ N_DATA_ACK_REQ N_EXPEDITED_DATA REQ N_RESET REQ N_RESET RESP N_DISCONNECT REQ N_UNITDATA REQ N_MGT_FACILITY REQ N_DEBUG REQ	DL_CONNECT REQ DL_CONNECT RESP DL_DATA REQ DL_EXPEDITED_DATA REQ DL_RESET REQ DL_RESET RESP DL_DISCONNECT REQ DL_UNITDATA REQ DL_MGT_FACILITY REQ DL_DEBUG REQ	N_CONNECT IND N_CONNECT CONF N_DATA IND N_DATA_ACK_IND N_EXPEDITED_DATA IND N_RESET IND N_RESET CONF N_DISCONNECT IND N_UNITDATA IND N_ERROR_REPORT IND N_MGT_FACILITY IND N_DEBUG IND N_TD_DATA IND N_RD_DATA IND N_TD_EXPEDITED_DATA IND N_RD_EXPEDITED_DATA IND N_TD_UNITDATA IND N_RD_UNITDATA IND
2	PH_ACTIVATE IND PH_ACTIVATE CONF PH_DATA IND PH_RESET IND PH_RESET CONF PH_DEACTIVATE IND PH_ERROR_REPORT IND PH_MGT_FACILITY IND PH_DEBUG IND PH_TD_DATA IND PH_RD_DATA IND	DL_CONNECT REQ DL_CONNECT RESP DL_DATA REQ DL_EXPEDITED_DATA REQ DL_RESET REQ DL_RESET RESP DL_DISCONNECT REQ DL_UNITDATA REQ DL_MGT_FACILITY REQ DL_DEBUG REQ	PH_ACTIVATE REQ PH_ACTIVATE RESP PH_DATA REQ PH_RESET REQ PH_RESET RESP PH_DEACTIVATE REQ PH_MGT_FACILITY REQ PH_DEBUG REQ	DL_CONNECT IND DL_CONNECT CONF DL_DATA IND DL_EXPEDITED_DATA IND DL_RESET IND DL_RESET CONF DL_DISCONNECT IND DL_UNITDATA IND DL_ERROR_REPORT IND DL_MGT_FACILITY IND DL_DEBUG IND DL_TD_DATA IND DL_RD_DATA IND DL_TD_EXPEDITED_DATA IND DL_RD_EXPEDITED_DATA IND DL_TD_UNITDATA IND DL_RD_UNITDATA IND

34 Automatic OSI Primitives

Often the Protocol Spreadsheet primitives that operate below a given layer are handled automatically by the protocol package at that layer.

Data primitives are automatic any time a SEND or GIVE_DATA softkey action is entered on the spreadsheet.

Connect Requests are automatic when the first spreadsheet data primitive is passed downward. The DL_CONNECT REQ in Figure 33-4 does not have to be entered in the user program. The connect request (but not the confirm) is handled automatically by the layer-package software, which assumes that the user never wishes to pass data *downward* to an empty layer.

The Connect Ind and Connect Resp primitives in Figure 33-3, on the other hand, are not automatic. They are completely at the discretion of the programmer of the Protocol Spreadsheet. If the programmer wishes Layer 2 to complete the link setup and begin transferring Info frames without the active participation of a higher layer, that is a viable alternative.

In the sequence in Figure 33-5 all of the primitives designated by arrows—with one exception—are generated and monitored automatically by the RCV, GIVE_DATA, and SEND spreadsheet entries. (The lone exception is the DL_DATA REQ primitive that is used as a condition in Layer 2.) This automatic handling of primitives frees the user at the top layer from programming considerations outside of his own immediate protocol.

When protocol packages are loaded, monitor primitives (such as DL_TD_DATA IND) are passed up the layers automatically. These primitives allow the Layer 2 and Layer 3 protocol-trace screens to display frame and packet information even when emulate primitives have not been passed up.

Automatic handling of primitives will vary with different protocols. Refer to the sections on the individual protocols for information on which primitives are tied to which protocol conditions and actions.

35 X.21 Layer 1

```

** Layer Setup **

DRIVE: HRD   Layer 1 Package: X.21   Selections   Packages Loaded
DRIVE: HRD   Layer 2 Package: X.25   X.25        NO PACKAGE
DRIVE: HRD   Layer 3 Package: X.25   X.25        HRD
DRIVE: HRD   Layer 4 Package: NO PACKAGE NO PACKAGE  HRD
DRIVE: HRD   Layer 5 Package: NO PACKAGE NO PACKAGE
DRIVE: HRD   Layer 6 Package: NO PACKAGE NO PACKAGE
DRIVE: HRD   Layer 7 Package: NO PACKAGE NO PACKAGE

Depress XEQ Key To Load The Selected Packages

Select Protocol Package
F1  F2  F3  F4  F5  F6  F7  F8
NO PCKG SS7_CMP X.21 DDCMP ISDN_D
    
```

Figure 35-1 In addition to being a Test Interface Module, X.21 is a "layer-personality package" of softkey functions at Layer 1.

```

F1  F2  F3  F4  F5  F6  F7  F8
LAYER: TEST: STATE: CONDS:

↓

F1  F2  F3  F4  F5  F6  F7  F8
DTE  DCE  RECEIVE  LEADS  ENTERST  TIMEOUT  KYBD  MORE

↓

F1  F2  F3  F4  F5  F6  F7  F8
T  C  R  I  UA
    
```

Figure 35-2 A special set of leads are available for monitoring once the X.21 package has been loaded in.

35 X.21 Layer 1

In addition to being a Test Interface Module (see Section 49), X.21 is a "layer personality package" of functions loaded into memory from disk via the Layer Setup screen. Figure 35-1 shows the Layer Setup screen configured to load in the X.21 package from the hard-disk drive. Refer to Section 8 for information on operating the Layer Setup screen.

The X.21 package consists of a group of conditions and actions at Layer 1 on the Protocol Spreadsheet that facilitate X.21 programming. Figure 35-2 shows the softkey path to a rack of condition softkeys that represent X.21 leads. These softkey conditions allow you to detect lead changes and lead status. Of the conditions on the first rack of softkeys below CONDS:, only LEADS is specific to X.21 and will be documented fully in this section.

Figure 35-3 shows the highest rack of softkeys containing actions that are specific to X.21. The SEND softkey includes a CALL_SETUP_SEND function that sends text messages always in ASCII code (consistent with X.21 call-setup protocol). The LEADS softkey gives the user control of X.21 control and data leads in emulation mode. The PROTOCL softkey includes functions that switch the line setup back and forth from ASCII 7-bit odd parity for call setups to whatever line setup the user has configured for data transfer on the Line Setup menu.

Other softkey actions in Figure 35-3 are not specific to X.21 and are discussed elsewhere in the manual.

A group of Figures at the end of this section, Figure 35-10 through Figure 35-13, shows the INTERVIEW emulating either the user or the switch in calling, called, clearing, and cleared scenarios. The "conditions" and "actions" in these drawings are softkey conditions and actions in the X.21 Layer 1 personality package. The "new states" in the drawings are standard X.21 state names which may be borrowed as state names on the Protocol Spreadsheet.

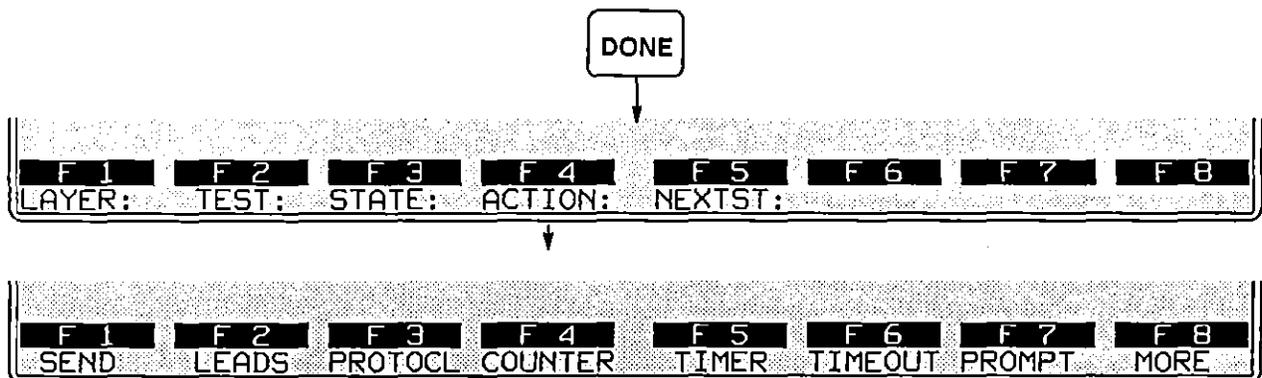


Figure 35-3 The *SEND*, *LEADS*, and *PROTOCL* softkeys branch to actions that are specific to X.21.

35.1 X.21 bis

The X.21 Layer 1 package also will work with the standard RS-232/V.24 TIM in an X.21 *bis* configuration. With the standard RS-232 TIM installed, the LEADS softkey shown in Figure 35-2 will be replaced by the EIA softkey that branches to standard EIA control-lead names: RTS, CTS, etc. The X.21 *bis* recommendation maps X.21 data and control leads to EIA leads according to the following conversions:

T	=	TD
C	=	DTR
R	=	RD
I	=	DSR

The LEADS softkey in Figure 35-3 changes to EIA in X.21 *bis*. When the RS-232 TIM is installed, data leads can be set to one of the standard X.21 idle conditions (+, ♯, ♪, ♫, ♬, and so forth) only via the IDLE_LN softkey action.

35.2 Transmitter/Receiver Phases

X.21 requires that data such as selection signals (the destination phone number) be transmitted during call setup. The data is transmitted in the following synchronous format: ASCII code, ♪♪ sync pattern, 7 data bits, odd parity, no BCC.

Once the call is established, a different format and code may be used at the link level and above. In order to monitor and transmit X.21 data and higher-level data correctly without exiting Run mode and reconfiguring the line setup, the X.21 layer package provides two different "phases" of the transmitter and the receivers. These phases are called CALL_SETUP and DATA_TRANSFER, and they are entered into the program via softkey. See Section 35.5(C), below.

When the program is in call-setup phase, data is monitored and sent according to the synchronous format and ASCII code defined above. In data-transfer phase, the format and code are as defined by the user on the Line Setup menu.

35.3 Sending From Layer 2

When Layer 1 is in data-transfer phase, a SEND action at Layer 2 will cause a transmission to go out onto the line automatically. No SEND action at Layer 1 is required.

When Layer 1 is in call-setup phase, a SEND action at Layer 2 will be ignored. If Layer 1 wishes to communicate to Layer 2 its readiness to send data, it must do so by SIGNAL action (see Section 30.4), since primitives are not currently operative at Layer 1.

35.4 X.21 Conditions

To bring up the bank of softkey conditions for Layer 1, first press the CONDITIONS softkey. This key becomes available when the cursor enters a programming block at the state level.

The first three condition softkeys—DTE, DCE, and RECEIVE—are common to all Layer 1 configurations. The fourth condition softkey, LEADS, is specific to the X.21 test-interface module. To the right of the LEADS softkey are general (layer-independent) conditions discussed in Section 30.

LEADS is a transitional/status condition and may be combined with other conditions (including other LEADS conditions). Refer to Section 30.2 for a discussion of how conditions may be combined.

(A) Data

The first three X.21 conditions can monitor one of the two data leads for a specific data event. This event can be any of several characters, a string of characters, a good BCC following the character or string, an error revealed by a block or parity check, and so on. When DTE is selected, data on the T lead will be monitored. A DCE condition monitors the R lead. RECEIVE conditions are intended for use in the emulate modes. When RECEIVE is selected, the INTERVIEW will always monitor the lead opposite its own transmit lead.

The fourth X.21 condition, LEADS, also can monitor both data leads. Figure 35-4 shows that T and R leads can be monitored for ZERO or ONE status. A data lead will satisfy one of these conditions when it is valid zero or valid one—that is, when it has retained its zero or one status for sixteen consecutive bit times.

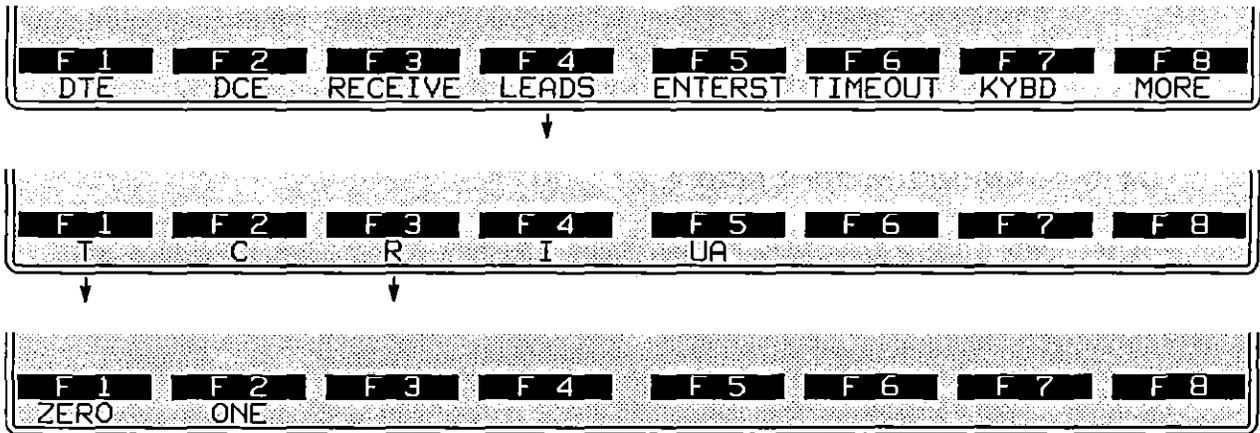


Figure 35-4 T and R leads can be monitored for *zero* or *one* status.

A mere transition from zero to one (or from one to zero) has no significance in X.21 protocol and cannot be detected by a LEADS T or LEADS R condition. (Control leads C and I, on the other hand, may be monitored either for a mere transition or for a valid status—see below.)

(B) Control Leads

X.21 conditions can monitor the status of C and I control leads. The C and I softkeys are in the conditions rack below LEADS. See Figure 35-5.

NOTE: Before you may monitor the status of C and I leads, the buffering of control leads must be enabled on the Front-End Buffer Setup menu. See Section 9.1(B).

C and I may be tested for true status or for valid status. In the X.21 protocol, the state of the lead is valid if it has been true for sixteen bit times. LEADS C ON, for example, checks the true state of the C lead. If the condition is alone in a CONDITIONS block, any momentary transition of the C lead from *off* to *on* will satisfy the condition. If the condition is used in a context where it is static rather than transitional—see Section 30.2 for a definition of this context—the true *on* state at the moment the lead is checked will satisfy the condition.

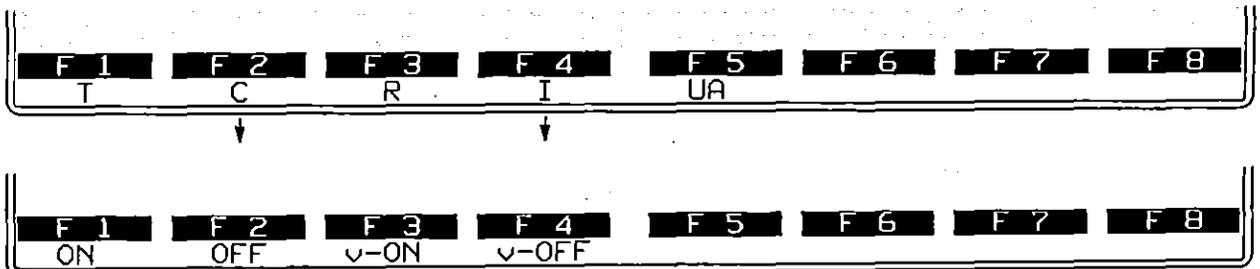


Figure 35-5 C and I leads may be tested for true status or for valid status.

The LEADS C ON_VALID condition requires not only that the state be true but also that it be valid. Valid conditions may be transitional or static, depending on how they are combined with other conditions in the same CONDITIONS block. When LEADS C ON_VALID is used alone in a CONDITIONS block, it is transitional. A transitional ON_VALID condition will be valid sixteen bit times after the transition from off to on—assuming that it retains its true status for the entire sixteen clock pulses.

(C) User Assigned

A LEADS UA condition detects an *on* or *off* state only if that state is valid. If a data lead is patched to the UA input, ON equals zero and OFF equals one.

35.5 X.21 Actions

When a block of conditions has been entered, press **Done** to access the ACTIONS softkey. The actions that pertain to the X.21 Layer 1 personality package are SEND, LEADS, and PROTOCL, shown in Figure 35-3. SEND and LEADS actions are operative in emulate modes only.

(A) Data Leads

Data leads may be programmed to send character strings via the SEND softkey. They also may be programmed to idle constant mark, constant space, bell characters, plus characters, sync characters, and an alternating pattern of 0's and 1's via the LEADS softkey.

1. *Data-transfer send.* Figure 35-6 shows the three send options that branch under the SEND softkey. If you press SND_DTA, the keyword SEND is written to the spreadsheet screen and this prompt appears below the screen: "Enter Transmit String." This is a normal Layer 1 send action and it is appropriate whenever you are in Data Transfer state according to the X.21 protocol.

Press the SND_DTA softkey or type SEND followed by space to begin the entry. Enter the string inside of quotation marks. After quotation marks are typed to close the transmit string, a set of softkeys appears for the error-checking value that will be appended to the transmit string—GOODBCC, BAD_BCC, NO_BCC, or ABORT.

To execute a data-transfer send, the program must be in data-transfer phase—see below. In this phase, the transmitter and receivers are obeying the code and format options selected on the Line Setup menu.

2. *Call-setup send.* The SEND softkey includes a CALL_SETUP_SEND function that sends text messages always in a code and format that is consistent with X.21 call-setup protocol. Press the SND_CLL softkey (Figure 35-6) or type CALL_SETUP_SEND followed by space to begin the entry. Enter the string inside of quotation marks.

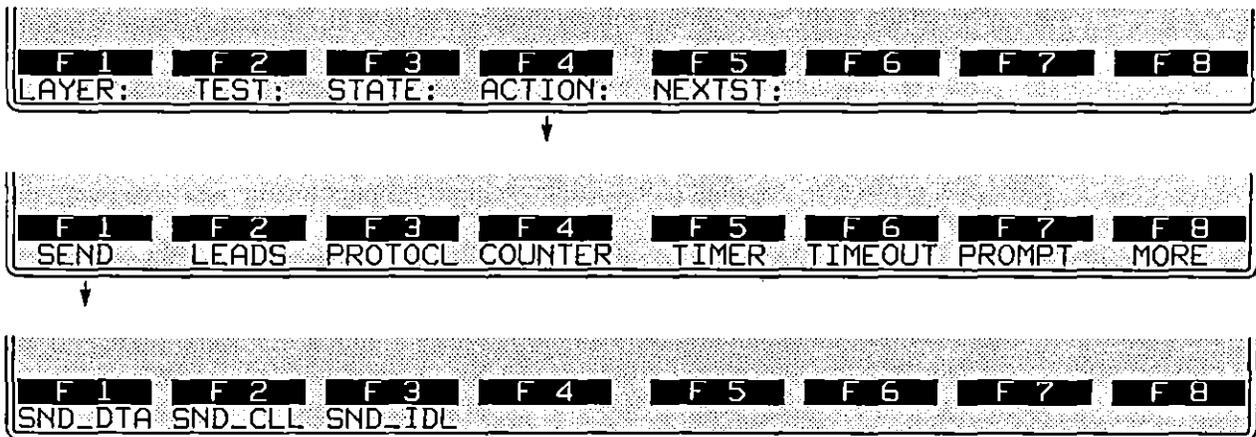


Figure 35-6 Two separate *SEND* actions are used to transmit either in data-transfer or call-setup format.

The code and format of a call-setup send action always is the same: ASCII code, 7 data bits, odd parity, no block check transmitted. Synchronization characters are $\$ \$$ (hex $\backslash \backslash$). The synchronization pattern must be provided in the transmit string—it is not automatic.

To execute a call-setup send, the program must be in call-setup phase—see 35.5(C), below. In this phase, the transmitter and receivers are disregarding the code and format options selected on the Line Setup menu.

3. *Call-setup send idle.* The SEND softkey also includes a CALL_SETUP_SEND_IDLE function. This action combines the CALL_SETUP_SEND action and the LEADS action that specifies an idle character. When entered as these two separate actions, the change in idle may occur slightly before or after the transmission.

Especially during high-speed operation, use the CALL_SETUP_SEND_IDLE action (and the NEW_IDL selection below it) to guarantee that the specified change in the idle character occurs *during* the string transmission. Press NEW_IDL and enter the (ASCII) idle character inside double quotation marks. Use the $\boxed{\text{hex}}$ key to enter hexadecimal characters.

4. *Idle.* Data leads also may be programmed to idle constant mark, constant space, bell characters, plus characters, sync characters, and alternating 0 and 1. Figure 35-7 shows the softkey path going through LEADS and T or R to the various idle states.
5. *One or zero.* Select ONE to idle constant mark, ZERO to idle constant space. Assuming that the program is in call-setup phase, a data lead idling mark will appear on the data display as r . Space idle will be displayed as b .
6. *Plus, bell, or sync.* Plus (+) characters, bell (r) characters, and sync (v) characters also may be transmitted as contiguous idle characters. A transmit

stream of any of these characters will be preceded by two ASCII sync characters (hex 16). In other respects, the action LEADS R BELLS has the same effect as the action IDLE_LINE "a".

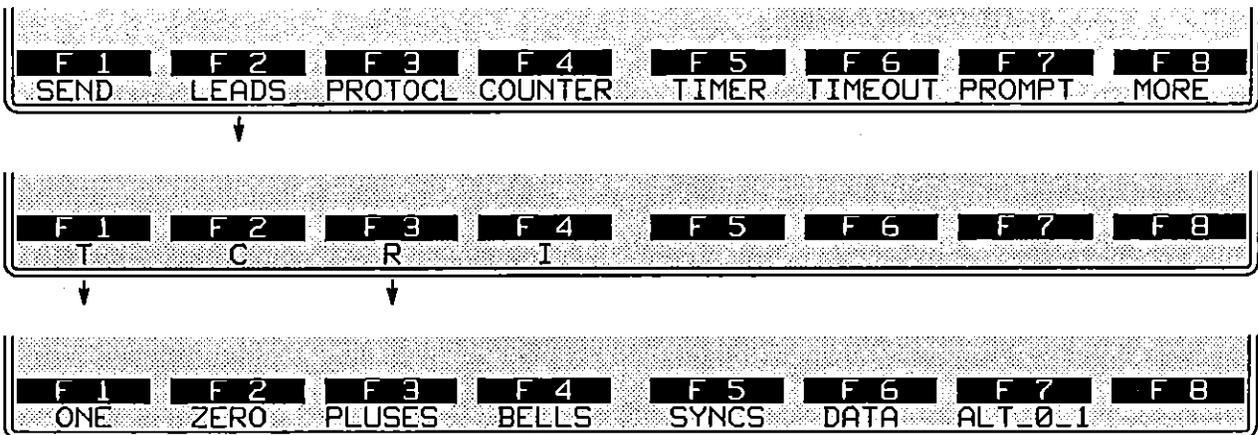


Figure 35-7 X.21 data leads T and R sometimes perform a "control" function by idling various characters.

Plus-, bell-, and sync-character idle actions do not take effect unless the program is in call-setup phase at the time the action is executed. The LEADS R ZERO action or the LEADS T ONE action will take effect even in data-transfer phase. The monitors may not be set up properly to detect or display the idle state, however, and we recommend that the programmer switch to call-setup phase as soon as one of the control leads first signals a clear request or indication.

7. *Data.* A ONE or ZERO leads action will clamp the line to the requisite voltage level. Once a lead is clamped, it must be unclamped before it can be used again for data. The DATA softkey shown in Figure 35-7 represents the "unclamp" action.

To change from idling space to transmitting selection signals, for example, you would insert the unclamp action (LEADS T DATA) shown here:

```
STATE: call_request
CONDITIONS: ENTER STATE
ACTIONS: LEADS T ZERO
        PROMPT "Press S to send selection signals"
CONDITIONS: KEYBOARD "Ss"
ACTIONS: LEADS T DATA
        CALL_SETUP_SEND "S S 123123+"
```

The PLUSES, BELLS, SYNCs, or ALT_0_1 action also will unclamp the line automatically.

8. *Alternating 0/1.* Press the softkey for ALT_0_1 to send an alternating series of zeroes and ones. The sequence is not preceded by sync characters. This idle action does not take effect unless the program is in call-setup phase at the time the action is executed.

(B) Control Leads

Control leads C and I may be controlled by spreadsheet action. Press the LEADS action softkey to bring up the rack of X.21 leads shown in Figure 35-8. Press C or I to access the softkeys that allow you to set a control lead to ON or OFF voltage.

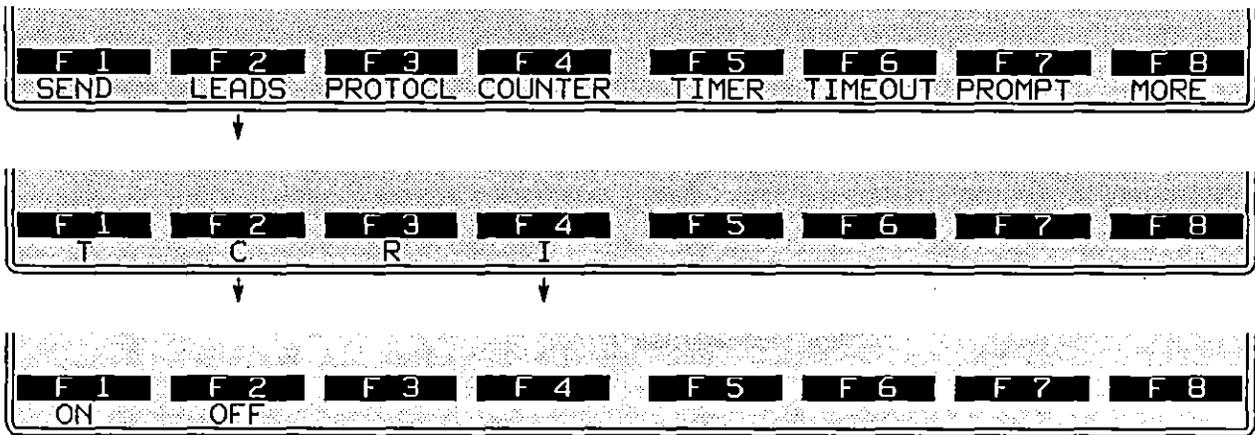


Figure 35-8 Control leads C and I may be turned on or off via softkey.

(C) Two Phases

The X.21 layer package provides two different "phases" of the transmitter and the receivers. These phases are called CALL_SETUP and DATA_TRANSFER, and they are entered into the program via softkey in the ACTIONS softkey rack that branches below PROTOCL. See Figure 35-9.

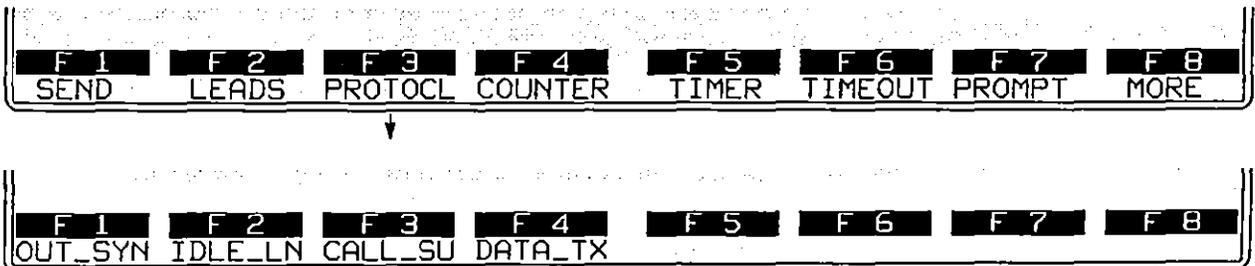


Figure 35-9 The X.21 layer package provides two different "phases" of the transmitter and the receivers, *Call Setup* and *Data Transfer*.

The initial configuration phase that the program adopts upon entering Run mode is selectable on the X.21 Interface Control setup menu. See Section 49.8. The default program-initiating phase is data transfer.

1. *Call setup.* When the program is in call-setup phase, data is monitored and sent according to the synchronous format and ASCII code defined above in Section 35.2. Idle display is automatically on. This means that when receivers encounter a condition that normally would send them out of sync (such as one or more `^F`-idle characters), the receivers begin looking for sync as they normally would but the raw data continues to be displayed, in reverse video.

With idle display automatically on, the transition will appear on the screen as a series of `^N` (NULL) characters when the data lead goes to zero to signal a call request or a clear request. For this reason, we recommend that the program adopt `CALL_SETUP` phase as soon as possible after a clear request or clear indication is signalled. In this way, the screen display of `^N` characters will record the clear request. In data-transfer phase, the steady zero signal will not be preceded by a special sync pattern and, depending on the line setup, may not be displayed.

Figure 35-10 shows the INTERVIEW on the user side of the X.21 interface. Here the INTERVIEW adopts call-setup phase prior to clamping its leads to signal `dte_clear_request`.

Figure 35-11 shows the INTERVIEW on the network-switch side of the X.21 interface. When the user side clears a call, the INTERVIEW programs a change to call-setup phase prior to clamping its leads to signal `doe_clear_confirmation`.

The appropriate `SEND` action in call-setup phase is `CALL_SETUP_SEND`. See Section 35.5(A)2. The simple `SEND` action, appropriate for data-transfer phase, will not be executed in call-setup phase.

When Layer 1 is in call-setup phase, a `SEND` action at Layer 2 will be ignored. If Layer 1 attains data-transfer phase and wishes to notify Layer 2 that it now is ready to send data, it must do so by `SIGNAL` action (see Section 30.4), since primitives are not currently operative at Layer 1.

2. *Data transfer.* When you press the softkey for `DATA_TX` (Figure 35-9) or type `DATA_TRANSFER` in an Actions block on the spreadsheet, you are sending the unit into data-transfer phase. In this phase, the unit monitors and sends according to the parameters selected by the user on the Line Setup menu. See Section 5.

The appropriate `SEND` action in data-transfer phase is entered on the Protocol Spreadsheet via the `SEND` softkey labeled `SND_DTA`. This action is written to screen simply as `SEND`. `CALL_SETUP_SEND` cannot be executed in data-transfer phase.

When Layer 1 is in data-transfer phase, a `SEND` action at Layer 2 will cause a transmission to go out onto the line automatically. No `SEND` action at Layer 1 is required.

EMULATE USER, USER CALLING AND CLEARING

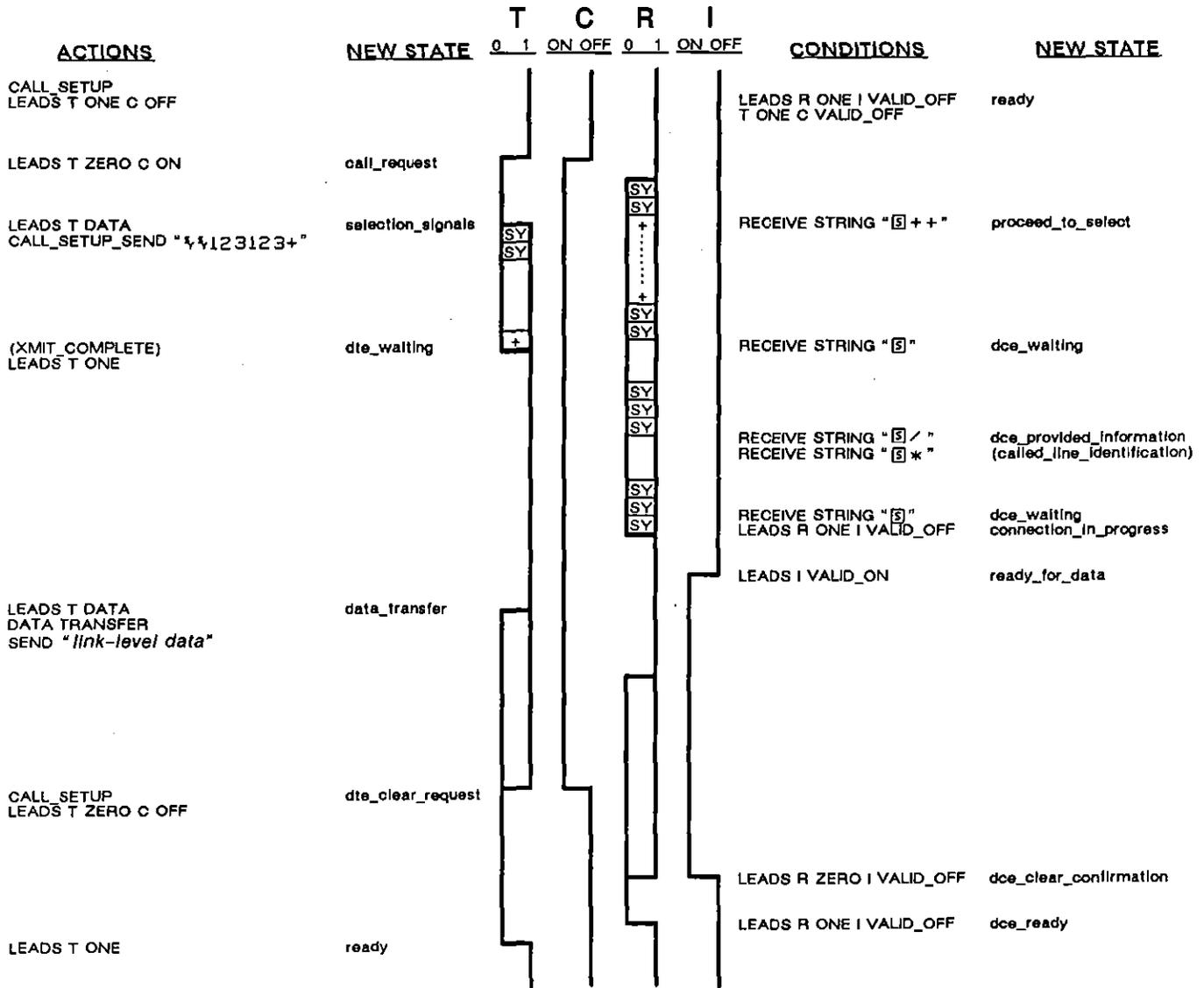


Figure 35-10 In this DTE-calling-and-clearing scenario, the INTERVIEW is on the user (DTE) side of the X.21 interface.

EMULATE SWITCH, USER CALLING AND CLEARING

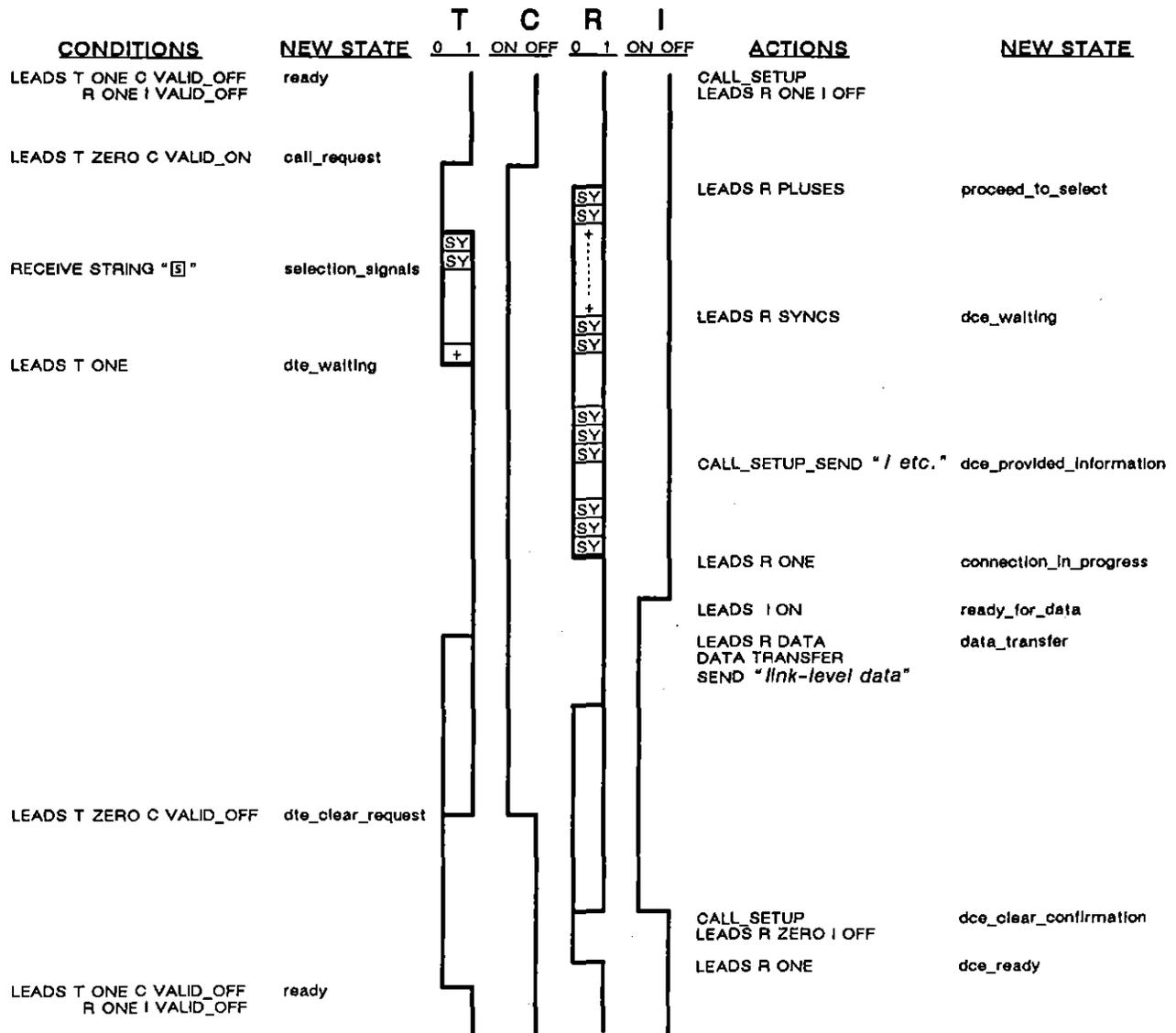


Figure 35-11 In this DTE-calling-and-clearing scenario, the INTERVIEW is on the network/switch side of the interface.

EMULATE USER, USER CALLED AND CLEARED

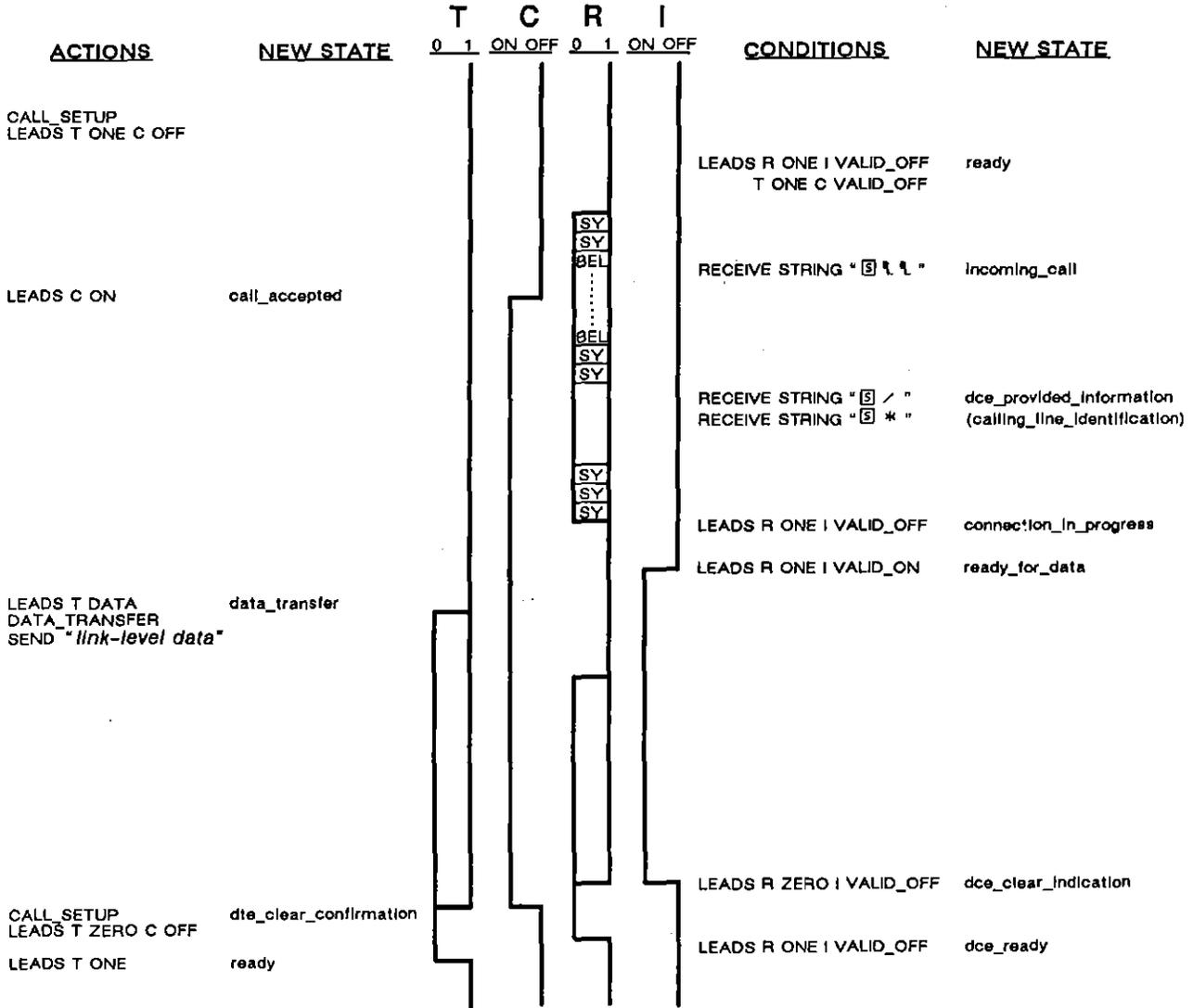


Figure 35-12 In this DTE-called-and-cleared scenario, the INTERVIEW is on the user (DTE) side of the X.21 interface.

EMULATE SWITCH, USER CALLED AND CLEARED

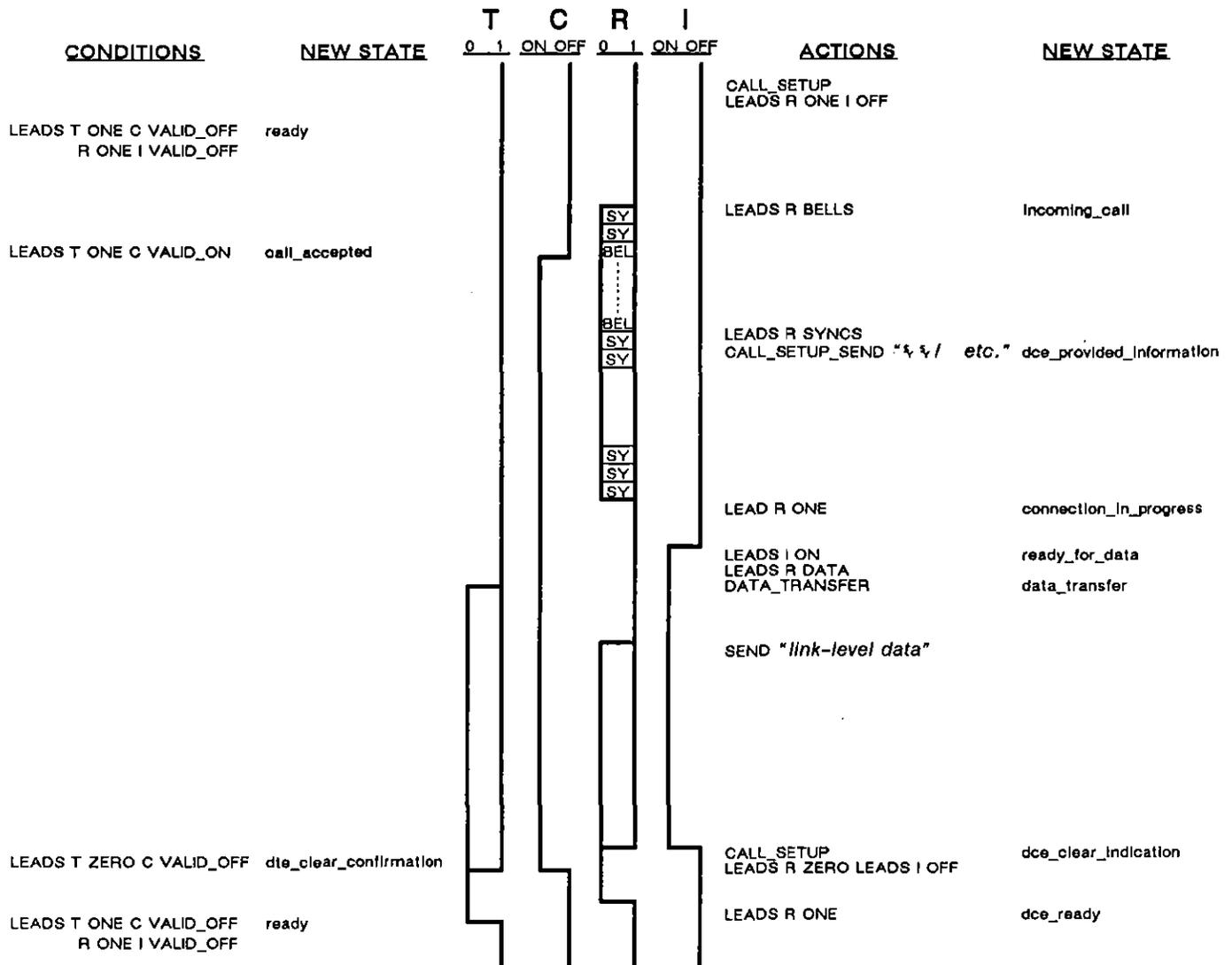


Figure 35-13 In this DTE-called-and-cleared scenario, the INTERVIEW is on the network/switch side of the interface.

36 X.25 Layer 2

```

** Layer Setup **

DRIVE: HRD   Layer 1 Package: NO PACKAGE   NO PACKAGE
DRIVE: FD2   Layer 2 Package: X.25         SDLC      FD1
DRIVE: FD2   Layer 3 Package: X.25         NO PACKAGE
DRIVE: HRD   Layer 4 Package: NO PACKAGE   NO PACKAGE
DRIVE: HRD   Layer 5 Package: NO PACKAGE   NO PACKAGE
DRIVE: HRD   Layer 6 Package: NO PACKAGE   NO PACKAGE
DRIVE: HRD   Layer 7 Package: NO PACKAGE   NO PACKAGE

Depress XEQ Key To Load The Selected Packages

Select Layer
F1  F2  F3  F4  F5  F6  F7  F8
LAYER-1 LAYER-2 LAYER-3 LAYER-4 LAYER-5 LAYER-6 LAYER-7 PROTSSEL
    
```

Figure 36-1 The X.25 personality package for Layer 2 is loaded from the Layer Setup screen.

```

** X.25 Frame Level Setup **

T1 (for INFO frame): 1.0 sec
Emulate: LOGICAL DTE
Mode of operation: MOD 8
Window size: 7

Enter Window Size (1 to 7) For Outstanding Frame: 7
F1  F2  F3  F4  F5  F6  F7  F8
    
```

Figure 36-2 Protocol Configuration screen for X.25 Layer 2.

36 X.25 Layer 2

Layer 2 X.25 is a "layer personality package" of functions that are loaded into memory from disk via the Layer Setup screen. Figure 36-1 shows the Layer Setup screen configured to load in the Layer 2 X.25 package from floppy-disk drive #2. Refer to Section 8 for details on operating the Layer Setup screen.

The Layer 2 X.25 package consists of the following:

- A special X.25 Frame Level Setup screen that controls certain parameters when the unit is tracing or emulating X.25.
- A protocol trace (illustrated in Figure 36-3) that distills from X.25 data the Level 2 events that have protocol significance. This trace is accessible by softkey in Run mode at all times.
- A group of conditions and actions at Layer 2 on the Protocol Spreadsheet that facilitate X.25 programming. Figure 36-8 shows the softkey path to the first rack of condition softkeys when the X.25 package is loaded in at Layer 2.

36.1 Frame-Level Setup

The parameters on the X.25 Frame Level Setup screen must be configured correctly for an accurate trace display and for proper emulation.

To bring up this screen, first go to the Layer Setup screen (press **PROG**, **F5**). Execute the X.25 selection at Layer 2; X.25 should appear in the **Packages Loaded** column. Press **F8** (labeled **PROTSEL**) to bring up a prompt to **Select Protocol Configuration Screen**. Then press **F2** (**LAYER-2**) to call up the X.25 Frame Level Setup screen.

The four parameter fields on this screen are shown in Figure 36-2. **T1**, **Emulate**, and **Window Size** apply to interactive (emulate) tests only. **Mode of Operation** must be configured correctly for the protocol trace as well as for proper emulation.

(A) T1

Enter a four-digit (including decimal point) T1 timeout value in this field. The largest valid entry is 65.5 seconds. The smallest entry is .001 second, or 1 millisecond.

T1 is the name given to the retransmission timer for INFO frames. When a value is entered in the T1 field on this menu, the layer 2 package will handle T1 timings correctly, as follows:

- Whenever the INTERVIEW sends an I-frame at Layer 2 and *there are no previous frames sent by the INTERVIEW currently outstanding (unacknowledged)*, the timer starts timing down from the value entered on the Frame Level Setup screen.
- An acknowledgment by the device under test of the most recent frame transmitted by the INTERVIEW *stops* the timer (so that it does not expire).
- An acknowledgment by the device under test of a frame that is not the most recent frame transmitted by the INTERVIEW—an “incomplete” acknowledgment—restarts the T1 timer to the value selected on the configuration screen.

Expiration of this Frame Level Setup timeout can only be detected by a T1_EXPIRED condition on the Protocol Spreadsheet at Layer 2. This particular timeout cannot be detected by a generic condition of TIMEOUT T1.

According to the protocol, a T1_EXPIRED condition should result in a RESEND action.

(B) Emulate Logical DTE/DCE

There are two selections in the Emulate field on the X.25 Frame Level Setup screen, LOGICAL DTE and LOGICAL DCE. The entry in this field determines the Layer 2 address bytes used by the INTERVIEW during interactive testing.

Configured as a logical DTE, the INTERVIEW uses address α_1 for commands and α_2 for responses. Usually a logical DTE is the PAD at the user site.

Configured as a logical DCE, the INTERVIEW sends commands to address α_2 and responds using address α_1 . Usually a logical DCE is a network switch.

Use the Mode selection (EMULATE DTE or EMULATE DCE) on the Line Setup menu to regulate the *physical* interface—whether to use pin 2 or pin 3 to transmit, and so on.

(C) Mode of Operation

The **Mode of Operation** field refers to the mode of numbering INFO and supervisory frames. There are two options, **MOD 8** and **MOD 128**.

MOD 8 uses sequence numbers 0-7. MOD 128 adds an extra byte to the control field in INFO, RR, RNR, REJ, and SREJ frames. See Figure 36-5. This extra byte allows sequence numbers in a range of 0-127.

The correct "modulus" must be selected in this field in order to conduct interactive communications and also to generate an accurate X.25 Layer 2 trace.

(D) Window Size

Any window size may be entered up to the current modulus minus one: 7 or 127. The window size is the maximum number of unacknowledged I-frames that Layer 2 will buffer for retransmission. When the limit is reached, any further INFO frames that are named in SEND actions triggered at Layer 2 will be passed to Layer 1 for transmission but not buffered for retransmission.

The window is a queue that buffers frames for retransmission in case one or more transmissions are lost or in error. A RESEND action will resend the first (earliest) frame in the window. Successive RESENDS will send successive frames until there are no more frames to resend; or until the window is reset by an acknowledgment or by a RESEND FIRST action.

36.2 Protocol Trace

The Layer 2 X.25 package includes an automatic frame-trace display that summarizes link-level activity. This trace mode is enabled whenever the unit is in Run mode, both real-time and frozen.

While the unit is in Run mode, press the softkey for L2TRACE to bring the protocol trace for X.25 Layer 2 to the screen. (If the X.25 package for Layer 3 is also loaded in, the L2TRACE softkey will appear after you have pressed PROTOCL, **F2** on the primary rack of display-mode softkeys.)

Figure 36-3 is an example of the Layer 2 trace display. Each horizontal row in the trace represents a frame.

(A) The Protocol Trace in Freeze Mode

Press **FREEZE** to prevent the addition of new data to *all* the display buffers, including the trace buffers. The frozen trace display may be scrolled through or paged through. The top line always is the cursor line (though there is no actual cursor on the trace display). Pressing **DOWN** or **↓** moves the viewing "window" down relative to the data to add one line of fresher data to the bottom of the screen. Pressing **UP** or **↑** moves the viewing window up to add a line of older data to the top of the screen.

(B) Trace Columns

The columns in the protocol trace for Layer 2 X.25 are explained below.

1. *Source.* The SRC column identifies the lead on which the frame was monitored, TD (DTE) or RD (DCE). This column identifies the *physical* source of the frame, not the *logical* source. The physical DTE uses the TD lead to transmit. The physical DCE uses RD to transmit.

Just as on the data display, RD data is underlined.

2. *Address.* The address octet (see Figure 36-5) is given in the ADDR column, with its two hexadecimal quartets presented as full-size alphanumerics. The address may be $\%_1$ or $\%_3$ in single-link operation.

This column identifies the logical DTE and DCE. The logical DTE uses address $\%_1$ for *INFO frames and other command frames*, and address $\%_3$ for responses. The logical DCE uses address $\%_3$ for INFO frames and other commands, and address $\%_1$ for responses.

3. *Type.* The mnemonic (abbreviated) names for eleven frame types as they appear in the TYPE column of the protocol trace are shown in Figure 36-5 under "CONTROL." The control field, therefore, indicates the frame type. If a control octet does not fit any of the patterns in the figure, the frame is listed in the TYPE column as UNKWN followed by the hexadecimal value of the control byte: UNKWN=F3.

If the number of bytes in the frame is below the required minimum, the frame is posted as INVALID.

4. *N(R) and N(S).* One column on the frame-level trace is devoted to N(R) values, and one column to N(S). The frame types that include N(R) or N(S) fields in their control fields are indicated in Figure 36-5. N(R) and N(S) occupy three bits each in modulo 8, seven bits each in modulo 128.

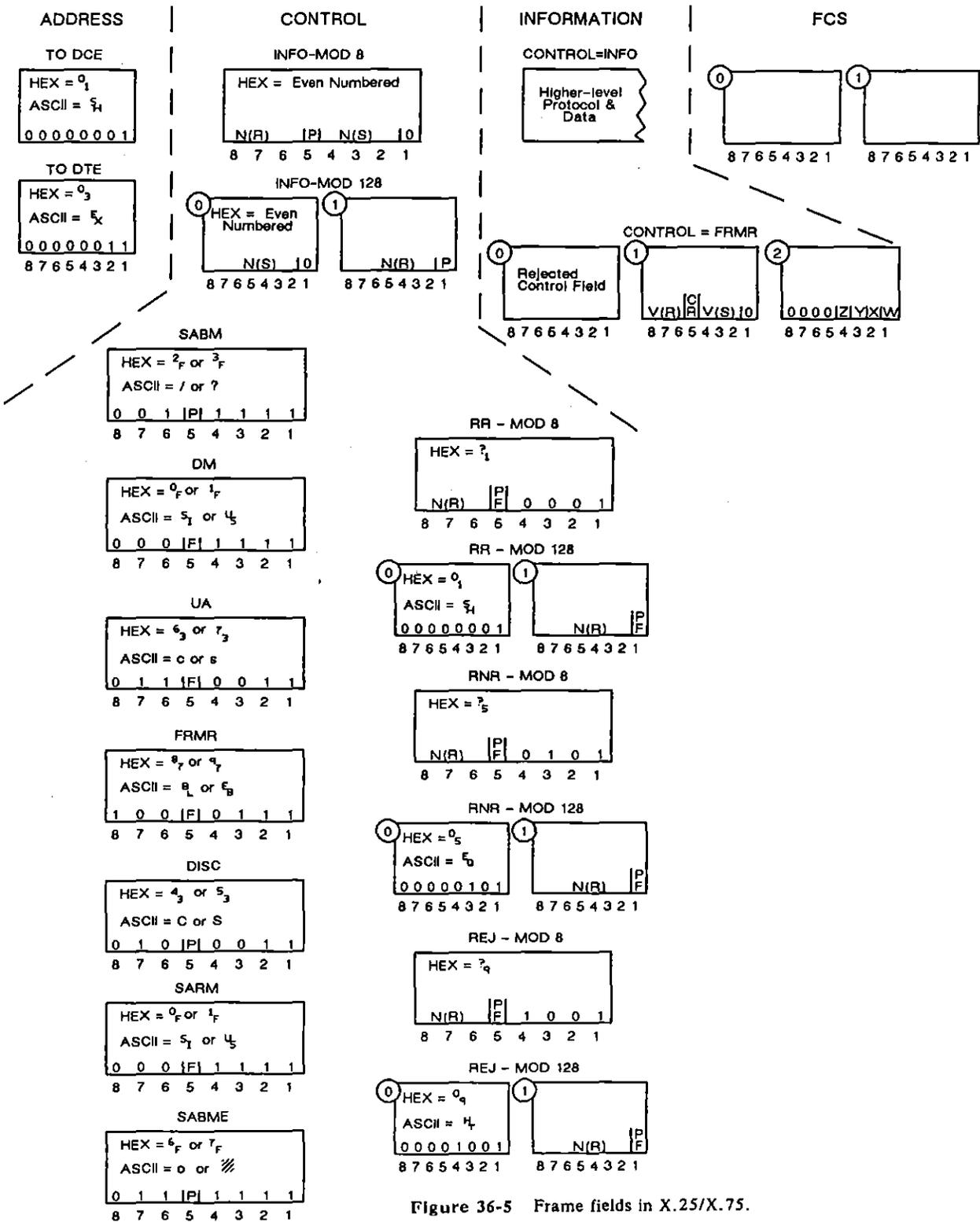


Figure 36-5 Frame fields in X.25/X.75.

MON/DISK/FD2		BLK=00913		
ASCII/8/NONE/BOP				
SRC	ADDR	TYPE	Nr	Ns
DTE	01	INFO	12	7A
DTE	01	INFO	12	7B
DTE	01	INFO	12	7C
DCE	01	RR	7C	
DTE	01	INFO	12	7C
DTE	01	INFO	12	7D
DTE	01	INFO	12	7E
DTE	01	INFO	12	7F
DTE	01	INFO	12	00
DTE	01	INFO	12	01
DTE	01	INFO	12	02
DTE	01	INFO	12	03
DTE	01	INFO	12	04
DTE	01	INFO	12	05
DTE	01	INFO	12	06
F 1 F 2 F 3 F 4				
L2TRACE L3TRACE				

Figure 36-6 MOD 128 sequence numbers are displayed in two-digit hexadecimal characters.

N(R) and N(S) values are presented in decimal format in modulo-8 traces. Each column displays a single digit that represents a 3-bit binary value. For modulo 128, the values 0 to 7 are given in "character" format, where the columns contain a two-digit hexadecimal character (see Figure 36-6).

SRC	ADDR	TYPE	Nr	Ns
DCE	03	DISC		
DTE	03	UA		
DTE	01	SABM		
DCE	01	UA		
DTE	01	INFO	0	0
DCE	03	INFO	0	0
DTE	03	RR	1	1
DCE	01	RR	1	1
DTE	01	INFO	1	1
DCE	01	RR	2	2
DCE	03	INFO	2	2
DTE	03	RR	2	2
DCE	03	INFO	2	2
DTE	03	RR	3	3
DTE	01	INFO	3	2

Figure 36-7 Nr and Ns columns are staggered, with the outside columns representing the sequence of DCE numbered I-frames.

Note that the Nr and Ns columns on the trace are staggered to suggest four columns. The two outside columns, comprised of the DTE's N(R) value and

the DCE's N(S), form a numbering sequence for DCE I-frames. The arrows in Figure 36-7 indicate the sequence: the DTE expects to receive frame 0, the DCE sends frame 0. The DTE expects frame 1; it asks for frame 1 again; finally the DCE sends frame 1. And so on.

The two inside columns reveal a similar pattern for DTE I-frames.

5. *P and F.* The status of the poll or the final bit is given in the P/F column. Whether this bit is the P or F bit is indicated for most frame types in Figure 36-5 (under "CONTROL").

The setting of the P bit in an INFO frame often denotes the retransmission of an unacknowledged frame following a T1 timeout.

6. *Size.* The number of bytes in each frame is given in this column in four decimal digits. The count begins with the address byte and excludes the two-byte FCS. Frames without I-fields show a count of two.
7. *Time.* The time of the arrival of the *end of the frame* at the DTE or DCE monitor is provided by a 24-hour clock and posted to the trace display. The clock is accurate to the millisecond.

When **Time Ticks: ON** is selected on the Front-End Buffer Setup screen (see Section 9), time values are incorporated into the data itself. As a result, times posted to the trace display will not be affected when recorded data is played back, even at varying speeds.

If **Time Ticks: OFF** was selected instead during live recording, times on the trace during playback will be influenced by "local conditions" such as playback speed, idle suppression, etc.

8. *Frame checking.* An X.25 frame ends as soon as a flag or seven 1-bits in a row are detected. If a flag ends the frame, a frame check is performed and the result is posted both to the data display and to the BCC column of the trace display. The symbol denotes a good frame check, while symbolizes a bad frame.

for abort is posted to the display when a frame is ended by seven 1-bits.

36.3 Monitor Conditions

When the Layer 2 X.25 personality package is loaded in (via the Layer Setup screen), a set of conditions checks DTE and DCE leads both in monitor and emulate modes. This set of conditions is accessed by the DTE and DCE selections on the first rack of condition softkeys at Layer 2. See Figure 36-8.

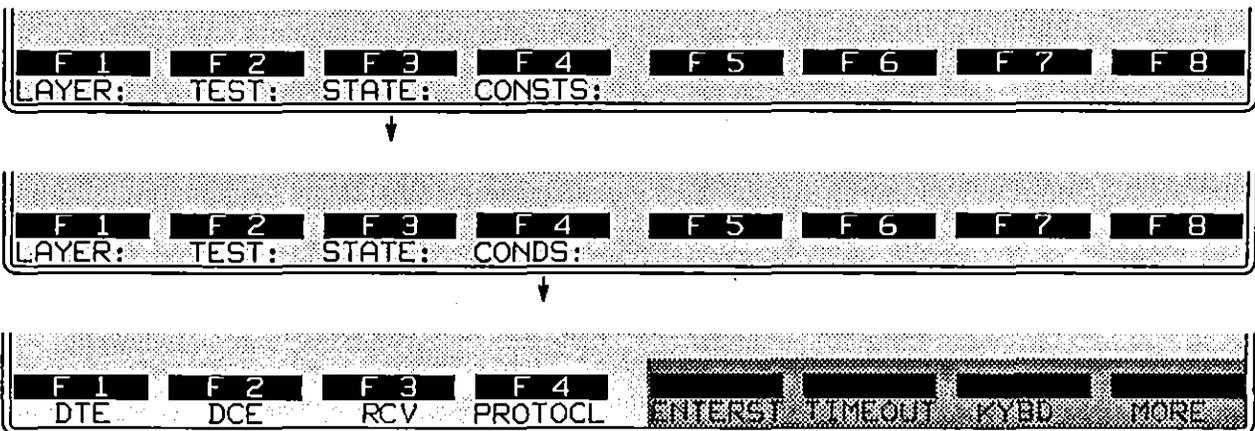


Figure 36-8 Unlike RCV conditions, the softkeys for DTE and DCE are valid when the INTERVIEW is monitoring the line passively.

After the keyword DTE (or DCE) is written to the spreadsheet, a rack of softkeys appears that represent *types* of frames: INFO, SABM, UA, and so forth.

(A) Frame Types

The softkeys for INFO, supervisory, unnumbered, and "other" frames are illustrated in Figure 36-9. Press a softkey to write one of these frame types to the Layer 2 spreadsheet. DTE or DCE followed by a frame-type mnemonic—DTE INFO, for example, or DCE SABM—is a complete condition and will come true if a matching frame is monitored. Address, poll/final, and BCC conditions may be added to the simple frame mnemonic, but they are optional.

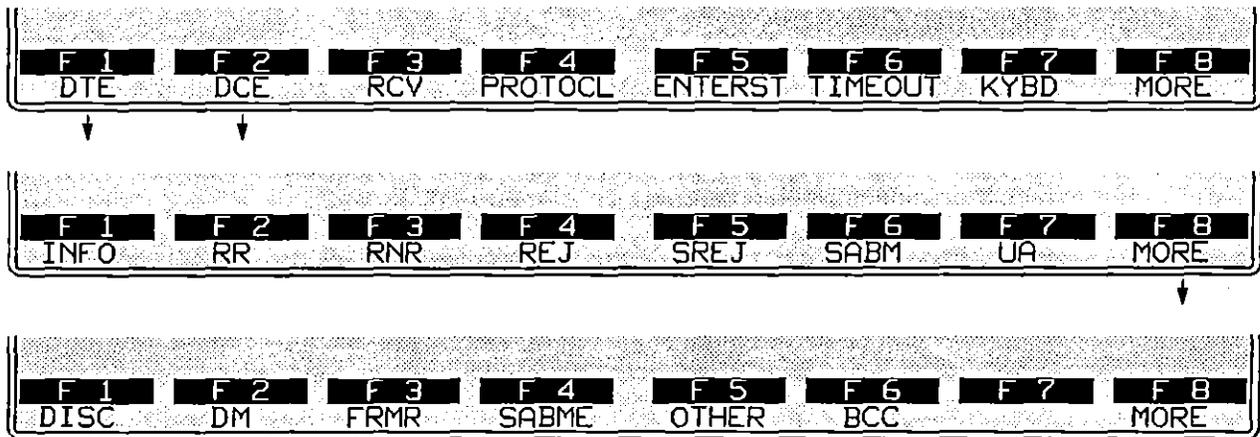


Figure 36-9 Frame types.

1. *Info frames.* INFO frames differ for MOD 8 and MOD 128 numbering schemes. (See Figure 36-5.) For spreadsheet conditions to match I-frames accurately, the correct numbering system ("Mode of Operation") should be selected on the Frame Level Setup screen.
2. *Supervisory frames.* The four supervisory-frame types that can be searched for on the data leads are RR (Receive Ready), RNR (Receive Not Ready), REJect, and SREJ (Selective Reject). These frames always contain N(R) fields (see Figure 36-5) and serve mainly to acknowledge or reject INFO frames.

Like INFO frames, supervisory frames are constructed differently according to the numbering scheme, MOD 8 or MOD 128.

3. *Unnumbered frames.* Unnumbered frames generally assist in link-setup and takedown. Different set-mode commands are used in different protocols: SABM for LAPB MOD 8 and SABME for LAPB MOD 128.
4. *Other frames.* Any frame type may be entered as a hexadecimal value instead of by name. Press the softkey for OTHER. See Figure 36-10. Then enter the hex byte in the form of two alphanumerics. Here, for example, is a SABM (with the P bit set) entered as a hexadecimal:

CONDITIONS: DCE OTHER 3F

Address, poll/final, and BCC conditions may be appended to OTHER conditions. In MOD 8, the P/F bit is already specified in the hex entry, and a P/F condition will be ignored.

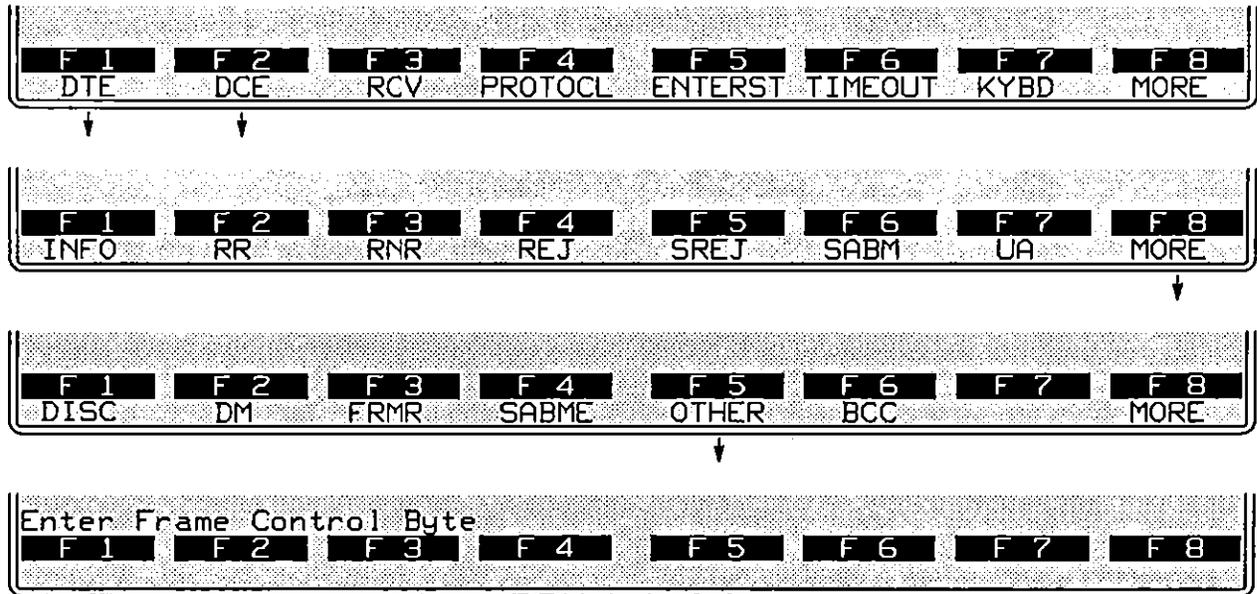


Figure 36-10 The hex value of any frame may be specified under *OTHER*.

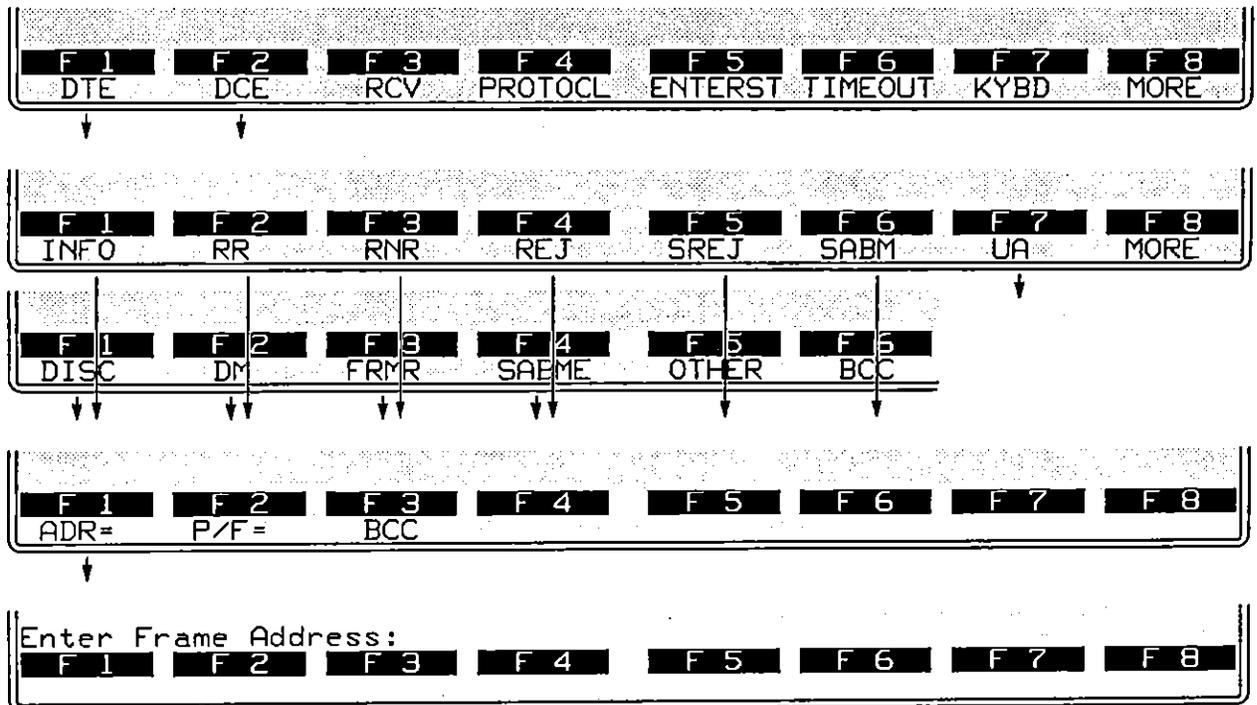


Figure 36-11 The hex value of the address byte is entered as two alphanumerics for *all* frame types.

5. *Address*. An address condition may be added to INFO, supervisory, unnumbered, and OTHER conditions. Press the softkey for ADR=, shown in Figure 36-11. Then enter the hexadecimal address octet as two alphanumerics. The address octet 01, for example, appears as follows:

CONDITIONS: DTE INFO ADR= 01

To bypass the ADR= selection (as well as the other options on the same rack of softkeys in Figure 36-11) press **Done**.

6. *Poll/final bit*. P/F conditions are optional for all frame types. P/F values of 0 or 1 are entered by the softkey sequence in Figure 36-12.

Press **Done** to bypass the P/F= condition and the other conditions on the same softkey level in Figure 36-12.

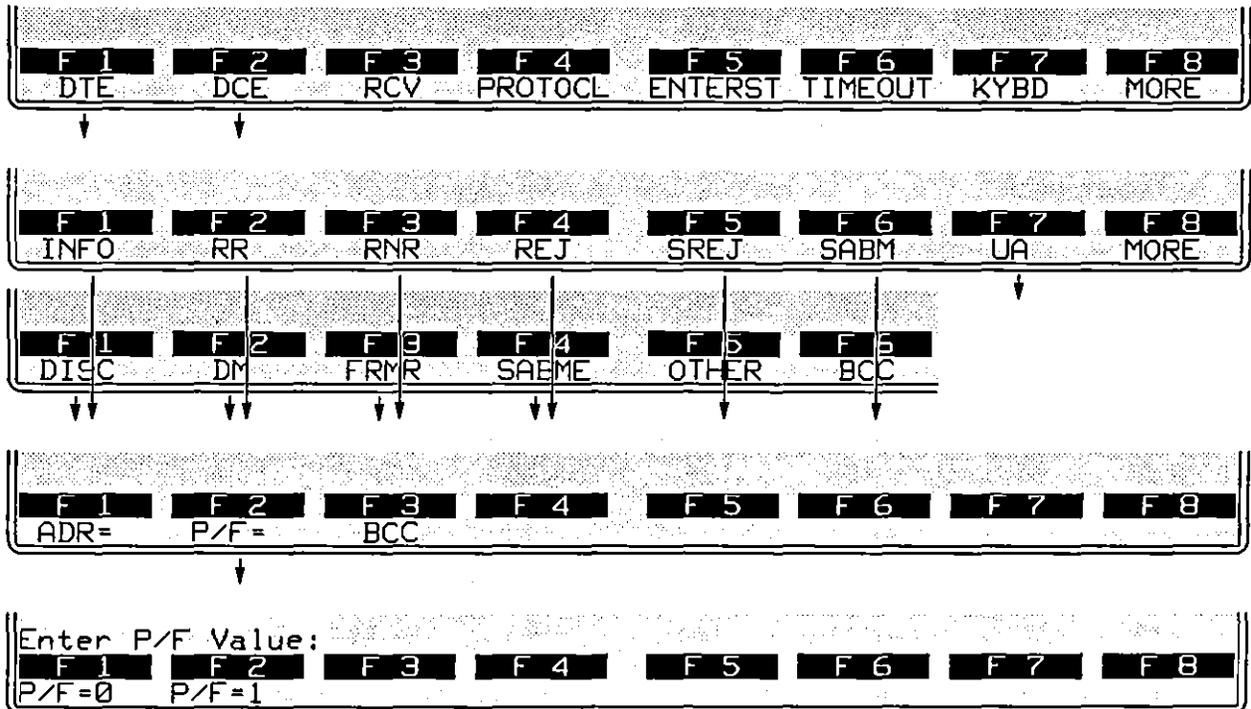


Figure 36-12 The value of the P/F bit may be chosen as a condition.

(B) BCC Conditions

DTE and DCE frames may be monitored for good and bad frame checks and for aborts. All DTE or DCE frames may be monitored with respect to frame checking, as in this example:

CONDITIONS: DTE BDBCC

The softkey sequence for this spreadsheet entry is given in Figure 36-13.

Or a particular *type* of frame may have a BCC or abort condition appended to it:

CONDITIONS: DCE INFO ABORT

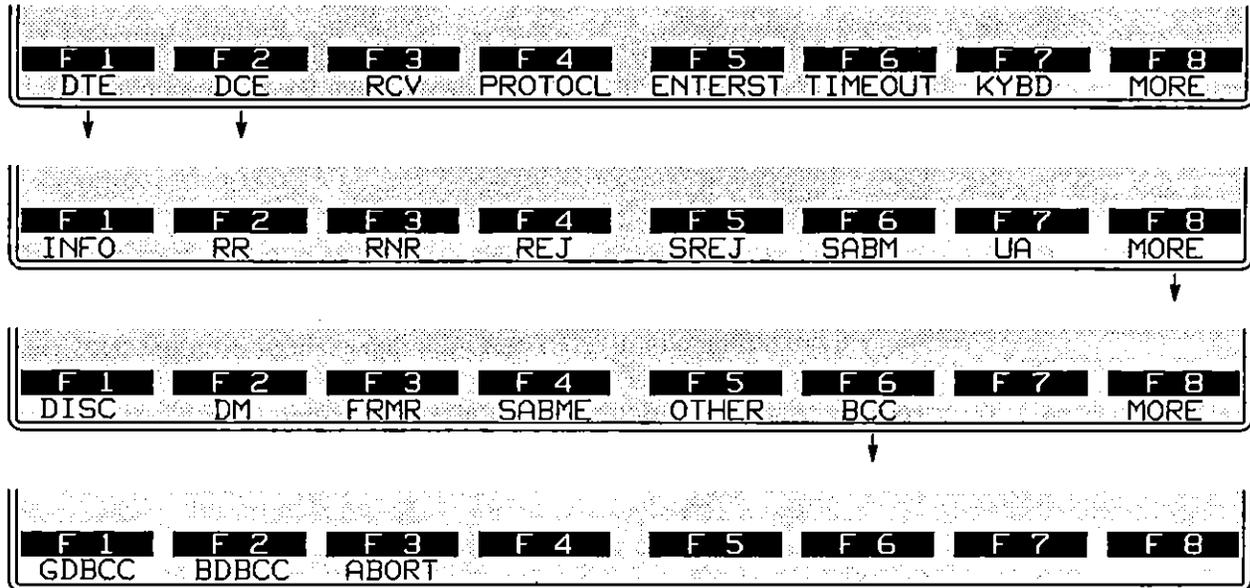


Figure 36-13 A condition may search for *all* good, bad, or aborted frames.

36.4 Emulate-Mode Conditions

The remaining conditions are functional only when the Line Setup menu is configured for Mode: `EMULATE DTE` or `EMULATE DCE`.

(A) Receive Conditions

Like DTE and DCE conditions, RCV conditions monitor a data lead for X.25 frame types. RCV conditions operate only in emulate modes, and they check only the data lead that the INTERVIEW is not using to transmit. While a RCV condition may look like a DTE or DCE condition—RCV INFO P/F=1 looks the same as DCE INFO P/F=1—there are important differences that are noted below.

1. *Valid frame sequencing.* To satisfy RCV conditions, numbered frames must have correct N(R) and N(S) sequencing.
2. *Good BCC.* RCV conditions cannot match frames with bad frame checks, nor can they match aborted frames. (Emulate-mode conditions are designed

for ease of programming, and the assumption is that as an X.25 emulator, you are not required to acknowledge—or negative-acknowledge—bad or aborted frames.)

If you wish to count bad BCCs or aborts, use DTE or DCE conditions instead of RCV conditions.

3. *Address in supervisory frames.* An address condition *must* be added to a RCV RR, RCV RNR, RCV REJ, or RCV SREJ condition. (In a DTE or DCE supervisory condition, the address is optional.) The address may be entered as in DTE/DCE conditions:

CONDITIONS: RCV RR ADR= 01

Or the address may be entered simply as COMMAND or RESP—RCV RR RESP, for example. COMMAND and RESP conditions will look for a specific address, 03 or 01, depending on the selection in the *Emulate* field on the X.25 Layer 2 Setup screen (see Section 36.1(B), above). A logical DTE will *receive* commands addressed to 03, and it will receive responses that have the address 01. A logical DCE receives commands addressed to 01 and responses that use 03.

CMND and RESP softkeys are shown in Figure 36-14.

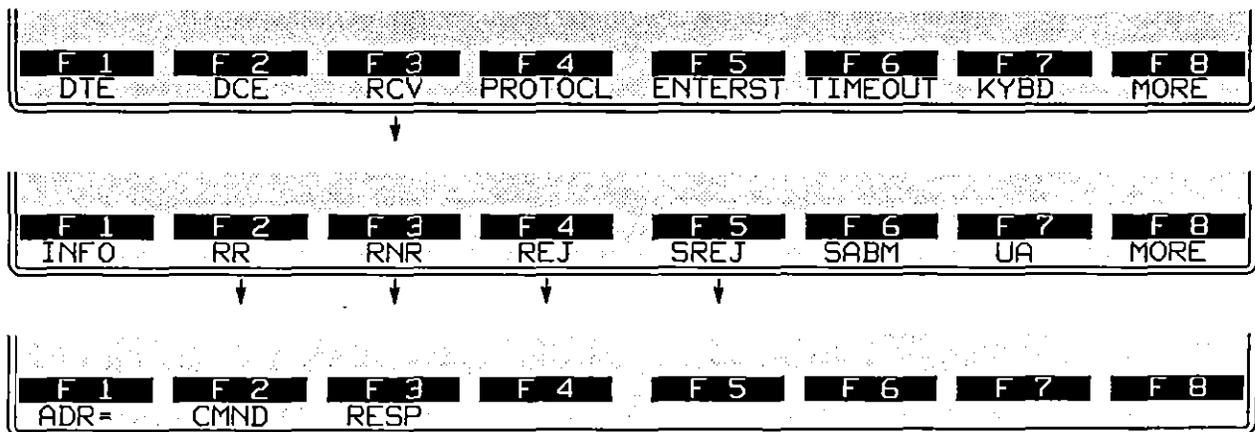


Figure 36-14 Addresses are required in RCV RR, RCV RNR, RCV REJ, and RCV SREJ conditions.

4. *Type invalid.* RCV conditions can detect frames that are invalid “types”—the control field is missing, for example, or the I-field is missing in an I-frame. The Protocol Spreadsheet entry for this condition is RCV INVALID, and the softkey sequence is illustrated in Figure 36-15.

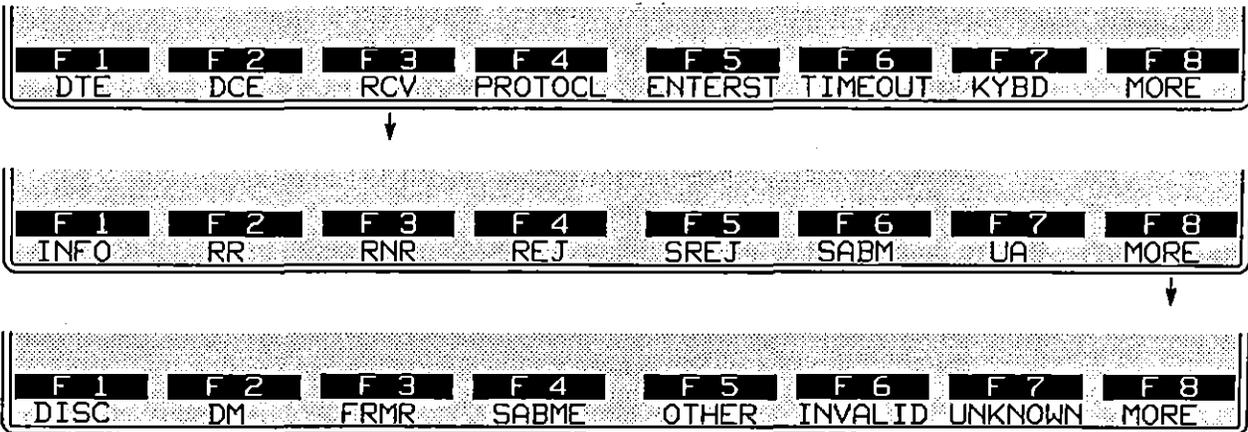


Figure 36-15 *INVALID* and *UNKNOWN* are frame types for *RCV* conditions.

5. *Type unknown.* A frame may be valid in all respects but have a control field that indicates a nonstandard frame type. Such a frame may be matched by a *RCV UNKNOWN* condition (Figure 36-15).

(B) N(S) Error

As a Layer 2 emulator, you do respond to *INFO* frames that have *N(S)* errors. These are detected as *NS_ERR* conditions, not as *RCV INFO* conditions.

*NS_ERR*s apply only to frames received when you are emulating. The same frame that triggers an *NS_ERR* condition also may satisfy a *DTE INFO* or *DCE INFO* condition—but not a *RCV INFO* condition.

NS_ERR will come true for any received frame whose *N(S)* value is not one higher than the previous *N(S)*.

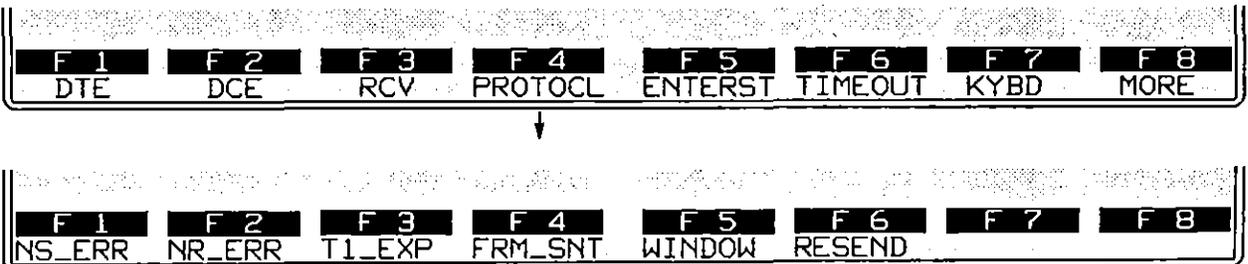


Figure 36-16 The *PROTOCL* key brings up six X.25 emulate conditions.

In the first rack of condition softkeys at Layer 2, press *PROTOCL*. Then press the softkey for *NS_ERR*. See Figure 36-16.

(C) N(R) Error

Received INFO or supervisory frames may have N(R) errors. Such errors are detected as NR_ERR conditions, not as RCV INFO or RR (or RNR or REJ) conditions.

A valid N(R) is any value that (1) acknowledges a frame that is *outstanding* (waiting for acknowledgment); or (2) repeats the last acknowledgment. Any other N(R) value is detected as an error.

(D) T1 Expired

This condition detects the expiration of the T1 timeout-timer that is regulated on the X.25 Frame Level Setup screen. See Section 36.1(A), above.

(E) Frame Sent

This condition is true when, as a result of a SEND or RESEND action, a frame has been passed down to Layer 1. Note that merely SENDING a frame *does not actually transmit the frame onto the line* if, for example, Layer 1 is the X.21 package in call-setup phase.

(F) Window Conditions

The size of the Layer 2 retransmit window is configured on the X.25 Frame Level Setup screen. See Section 36.1(D). There are four conditions that test the current status of this window. They are WINDOW FULL, WINDOW EMPTY, WINDOW NOT_FULL, and WINDOW NOT_EMPTY. The softkey sequence for the WINDOW options is shown in Figure 36-17.

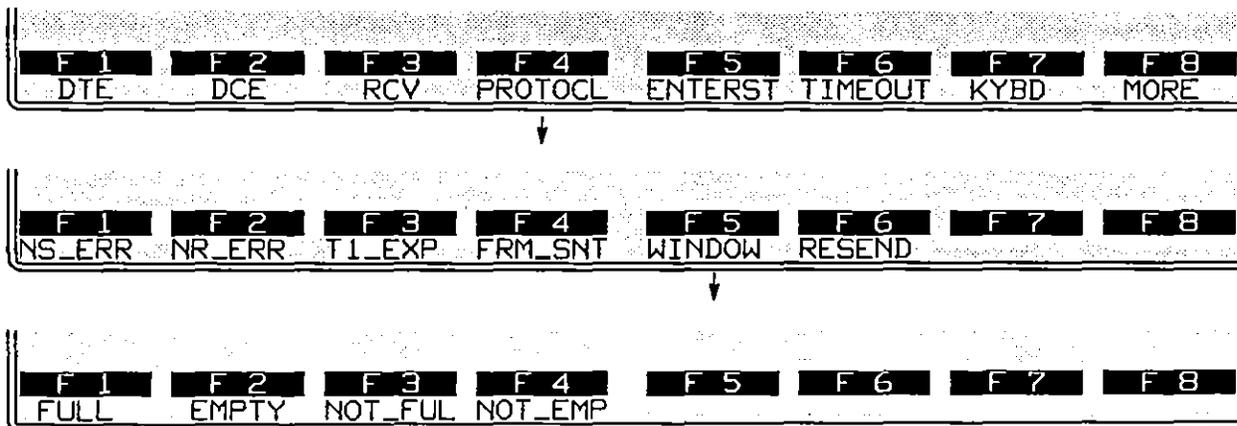


Figure 36-17 When the retransmit window fills, Layer 2 stops buffering frames for retransmission.

WINDOW FULL is true when the window is full of unacknowledged frames and the Layer 2 protocol package will not buffer additional frames until some acknowledgment is received.

Each time an acknowledgment is received, the window is flushed to the extent of the acknowledgment. WINDOW EMPTY means that the latest acknowledgment was complete and left no frames outstanding (unacknowledged). If an RR response is received and the acknowledgment is only partial, this condition will be true:

```
CONDITIONS: RCV RR RESP
            WINDOW NOT_EMPTY
```

CAUTION: Window conditions are status conditions (see Section 30.2) and must always be used in combination with a transitional condition such as a RCV condition.

(G) More to Resend

Frames in the window may have to be resent, usually as the result of a T1 timeout or a Reject frame. One RESEND action retransmits one frame in the window, beginning with the earliest. Subsequent RESEND actions retransmit subsequent frames. The MORE_TO_RESEND and NO_MORE_TO_RESEND conditions allow you to retransmit the entire window, as in the "recover" state in this example:

```
CONDITIONS: RCV REJ RESP
NEXT_ST: recover
STATE: recover
CONDITIONS: ENTER STATE
ACTIONS: RESEND FIRST
CONDITIONS: FRAME_SENT
        MORE_TO_RESEND
ACTIONS: RESEND NEXT
CONDITIONS: FRAME_SENT
        NO_MORE_TO_RESEND
NEXT_ST: xfer
```

MORE_TO_RESEND and NO_MORE_TO_RESEND conditions may be written to the Protocol Spreadsheet by the softkeys shown in Figure 36-18.

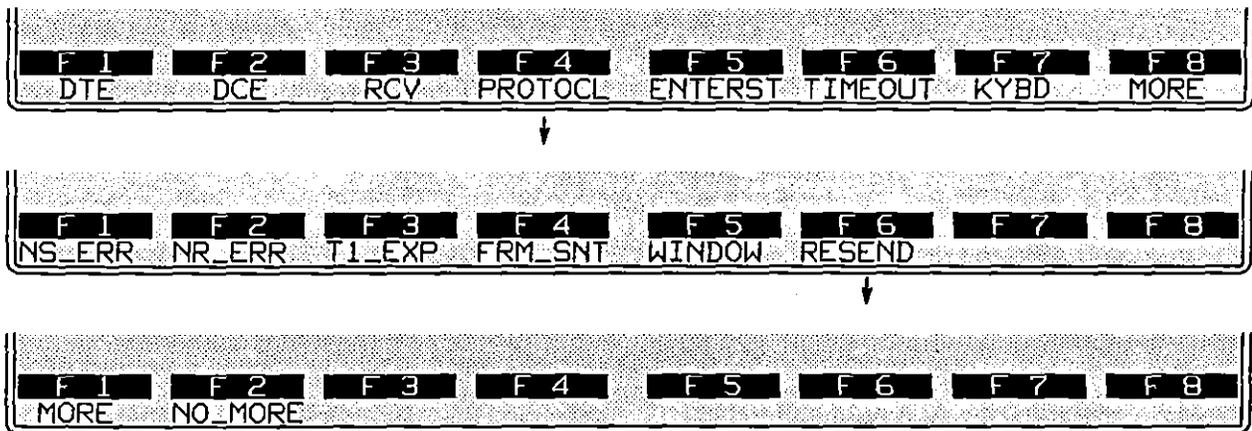


Figure 36-18 The *MORE_TO_RESEND* condition allows you to resend the entire window of frames and then stop when there are *NO_MORE_TO_RESEND*.

CAUTION: MORE_TO_RESEND and NO_MORE_TO_RESEND are status conditions (see Section 30.2) and must always be used in combination with a transitional condition such as FRAME_SENT.

36.5 Emulate Actions

When you have completed a block of conditions in a Protocol Spreadsheet test at Layer 2, press **DONE** to access the set of actions that can be taken as a result of the block of conditions coming true. The set of actions that are specific to the X.25 Layer 2 personality package are shown in the racks of softkeys in Figure 36-19. Except for **ENHANCE** and **SUPPRES**, the actions shown have meaning only when the **INTERVIEW** is emulating DTE or DCE, and not when it is monitoring the line passively.

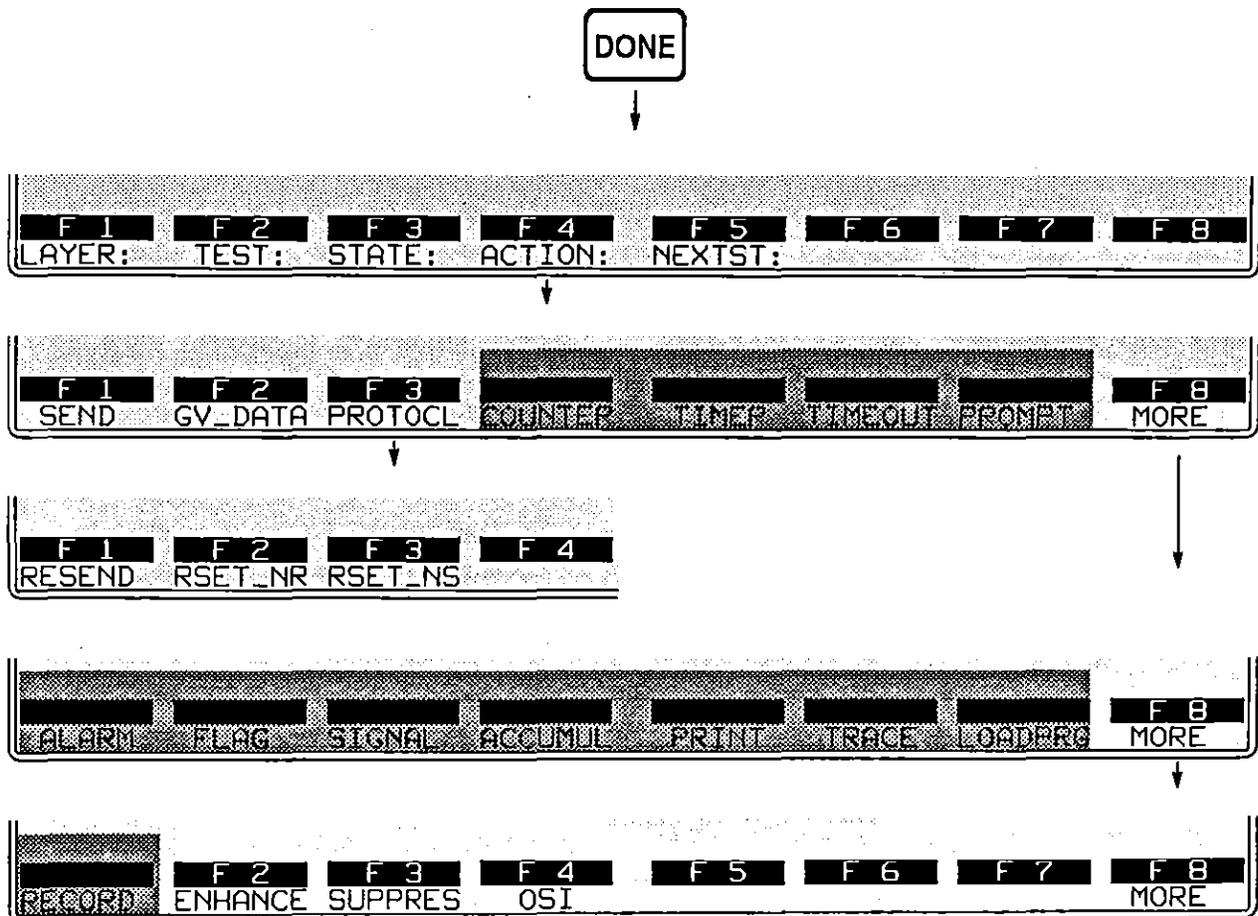


Figure 36-19 Action softkeys specific to X.25 Layer 2.

(A) Send Actions

Press the softkey for **SEND** to access two racks of softkeys with names of frame types that may be named in **SEND** actions. All data generated by the Layer 2 X.25 package must be enclosed in a frame that is identified in a **SEND** action by

type. (Only at Layer 1 can data be generated as a simple character string without any protocol building blocks.) The complete set of frame types is given in Figure 36-20.

When conditions are true for a SEND action, frames are sent immediately down to Layer 1 to be transmitted there. (Note that when the X.21 Layer 1 package is loaded, the sending down of INFO frames will be conditional on data-transfer phase being active at Layer 1.)

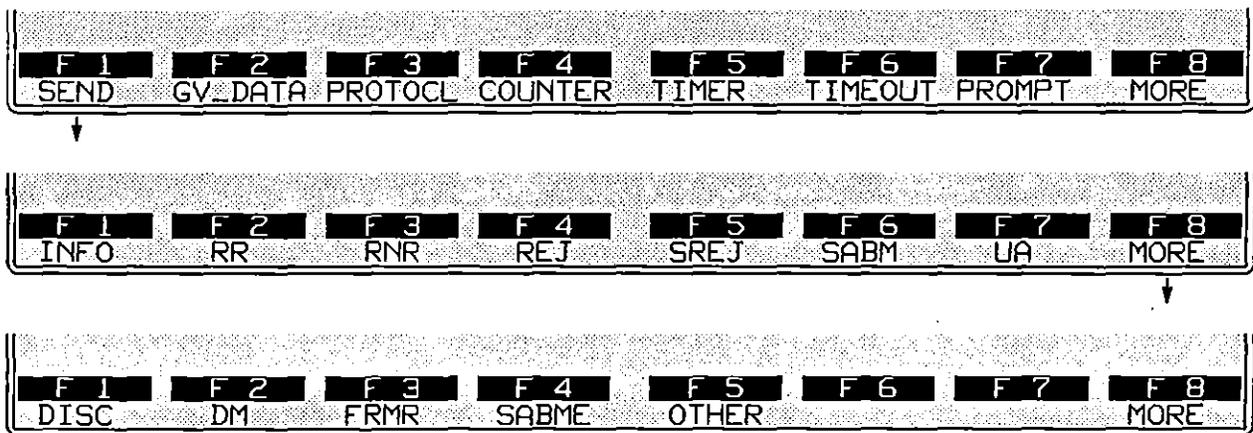


Figure 36-20 SEND actions always specify a frame type.

1. *INFO frames.* SEND INFO is a complete action-entry. Address, poll-bit, N(R), N(S), string, and BCC parameters may be added to an INFO frame, but they are optional. If no optional parameters are entered, the INFO frame will default to the following parameters:

- The address will be appropriate for an INFO frame sent by the "logical emulator" selected on the Frame Level Setup screen. See Section 36.1(B).
- The poll bit will be set to 1.
- The N(R) will increment to the "automatic" value, one higher than the last valid N(S) received.
- The N(S) will increment to the "automatic" value, one higher than the last valid N(S) sent.
- Since there is no default data-string, the I-field will be empty.
- The BCC will be good.

The default parameters for INFO and other frames are given in Table 36-1.

If a Layer 3 package is installed and Layer 3 data is being handed down to Layer 2, the following condition-and-action trigger will accept this data and convey it properly to Layer 1:

CONDITIONS: DL_DATA REQ
ACTIONS: SEND INFO "«DL_DATA»"

SEND INFO actions pass the INFO frame immediately to the next layer down. If the retransmit window is full, the frame is still sent—but it is not buffered in the window and can not be resent.

An INFO frame will be buffered for retransmission regardless of the status of the window if a specific value is entered for the NS= parameter (see "N(S)," below). The specific N(S) value will clear the window and the INFO frame will be buffered in the first window position.

Table 36-1
Default Parameters in SEND Actions

SEND	ADR= logical DTE	logical DCE	P/F	NR=	NS=	STRING	BCC
INFO	01	03	1	AUTO	AUTO	none	GDBCC
SABM	01	03	1	N/A	N/A	N/A	GDBCC
UA	03	01	L	N/A	N/A	N/A	GDBCC
DISC	01	03	1	N/A	N/A	N/A	GDBCC
DM	03	01	L	N/A	N/A	N/A	GDBCC
FRMR	03	01	L	N/A	N/A	none	GDBCC
SABME	01	03	1	N/A	N/A	N/A	GDBCC
RR CMND	01	03	1	AUTO	N/A	N/A	GDBCC
RR RESP	03	01	L	AUTO	N/A	N/A	GDBCC
RNR CMND	01	03	1	LAST_NR	N/A	N/A	GDBCC
RNR RESP	03	01	L	LAST_NR	N/A	N/A	GDBCC
REJ CMND	01	03	1	LAST_NR	N/A	N/A	GDBCC
REJ RESP	03	01	L	LAST_NR	N/A	N/A	GDBCC
SREJ CMND	01	03	1	LAST_NR	N/A	N/A	GDBCC
SREJ RESP	03	01	L	LAST_NR	N/A	N/A	GDBCC
OTHER	01	03	N/A	N/A	N/A	none	GDBCC

2. *Unnumbered frames.* SEND SABM, SEND UA, and so forth, are complete action-entries. Address, P/F-bit, string, and BCC parameters may be added to the SEND action, but they are optional. Default values are sent in the absence of specific optional entries: see Table 36-1.
3. *Supervisory frames.* An address value must be added to SEND RR, SEND RNR, SEND REJ, and SEND SREJ actions. The address may be entered as a specific value.

ACTIONS: SEND RR ADR= 03

Or the address may be entered simply as CMND or RESP—SEND RR RESP, for example. CMND and RESP frames will carry address 01 or 03, depending on the selection in the **Emulate** field on the X.25 Frame Level Setup screen . See Section 36.1(B). Refer to Table 36-1 for the address values sent by the two different logical emulators.

Figure 36-21 shows the address selections for all supervisory frames.

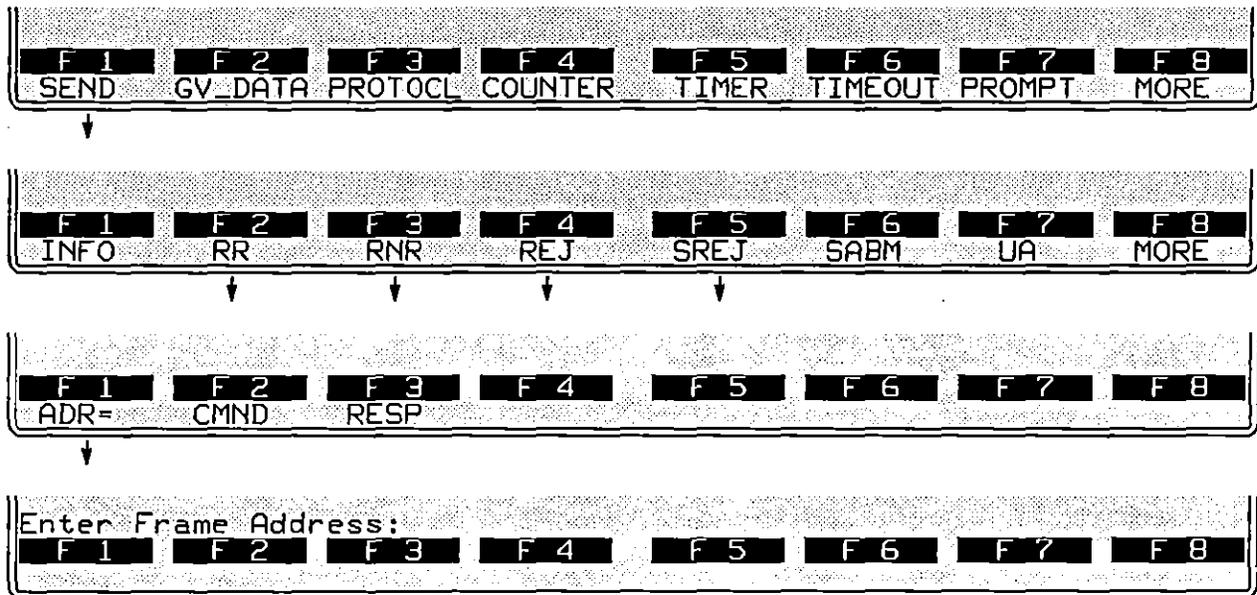


Figure 36-21 An address value must be added to *SEND RR*, *SEND RNR*, *SEND REJ*, and *SEND SREJ* actions.

4. *Other frames.* Any frame type may be entered in a *SEND* action as a hexadecimal value instead of by name. Press the softkey for **OTHER**, on the bottom rack in Figure 36-20. Enter the hex value in the form of two alphanumerics. Then press the softkey for **ADR=** (Figure 36-22) and enter an address value. Here is a **DISconnect** command entered as a *SEND OTHER* action:

ACTIONS: SEND OTHER 43 ADR= 03

P/F, N(R), and N(S) fields are implied in the user-entered hexadecimal control field. An address *should* be added to a *SEND OTHER* action, but if it is not, the default is the (CMND) address 01 when the **Emulate** field on the X.25 Frame Level Setup screen shows **LOGICAL DTE**. The address defaults to 03 for **LOGICAL DCE**. The other default parameter is a good BCC. (In MOD 128, P/F is not included in the hex entry and is a valid optional entry.)

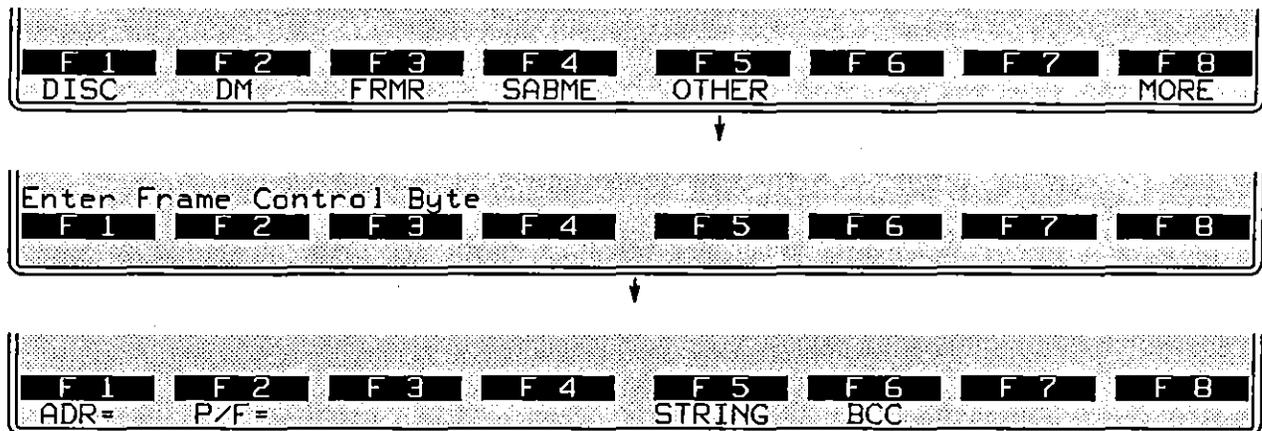


Figure 36-22 *SEND OTHER* actions always specify a *type* value.

5. *Address*. An address may be specified for INFO, unnumbered, and OTHER frames. It must be specified for supervisory frames. There are three softkeys pertaining to address in supervisory frames, ADR=, CMND, and RESP. See under "*Supervisory frames*," above.

The ADR= entry is always followed by the hexadecimal address octet typed as two alphanumeric characters. The address field 03, for example, appears as follows:

ACTIONS: SEND RR ADR= 03

6. *Poll/final bit*. The P/F bit is an optional entry in all SEND actions. P/F values of 0, 1, or LOOPBAK are entered by the softkeys in Figure 36-23. If P/F= LOOPBACK, the bit will echo the last P/F bit received. (Looping the P/F bit is appropriate for UAs and supervisory frames.) Default P/F values are given in Table 36-1.

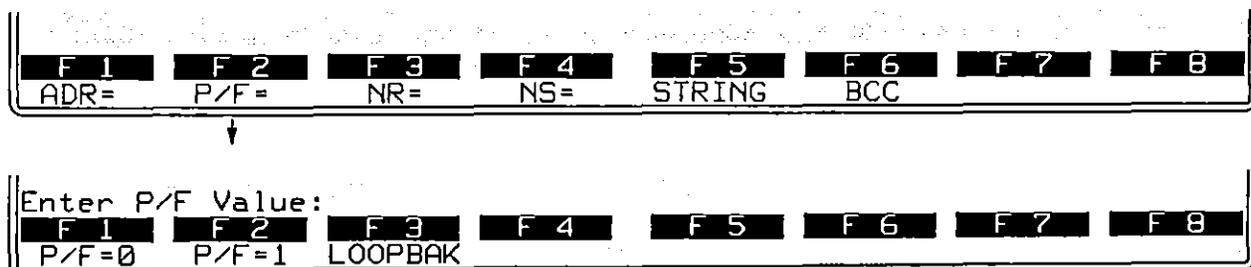


Figure 36-23 A P/F value is optional in all *SEND* entries.

7. *N(R)*. *N(R)* fields are transmitted in INFO and supervisory frames.

To specify an *N(R)* value, press the softkey for NR= (see Figure 36-24). Enter a hexadecimal value written as one or two alphanumeric digits. For example, an entry that represented the highest valid *N(R)* in MOD 8 would be NR= 7. The highest valid entry in MOD 128 would be NR= 7F.

Other *N(R)* options are ACK_NS, LAST_NR, and AUTO. (See Figure 36-24.) ACK_NS means that your *N(R)* will acknowledge (that is, it will be *one higher than*) the last *N(S)* value you received. Normally this will be the correct *N(R)*, except in cases where the last *N(S)* received was erroneous. The NR= ACK_NS selection allows you to overlook *N(S)* errors.

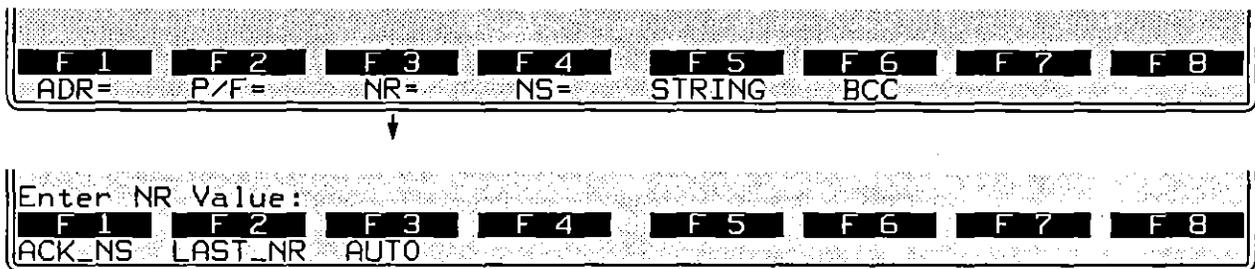


Figure 36-24 The *N(R)* field may be specified in INFO and supervisory frames to be sent.

LAST_NR means that you simply repeat the last *N(R)* you sent. Normally this is the correct *N(R)* following a bad *N(S)*. The NR= LAST_NR option allows you to force the other side to initiate recovery.

AUTO means that you will behave as a normal X.25 Layer 2 station, acking valid *N(S)* values and repeating your last *N(R)* whenever an invalid *N(S)* is received. AUTO is the default *N(R)* selection in SEND INFO, SEND RR, SEND REJ, and SEND SREJ actions. See Table 36-1.

8. *N(S)*. *N(S)* fields are transmitted in INFO frames only. (See the frame-field diagrams in Figure 36-5.) Entries for *N(S)* in SEND INFO actions are optional. The softkeys that open below NS= are illustrated in Figure 36-25.

To specify an *N(S)* value, press the softkey for NS=, then enter a hexadecimal in the form of one or two alphanumerics. Valid hex entries are the same as for *N(R)*. A SEND INFO action that *specifies* an *N(S)* value—NS= 0, for example—will clear the window so that the INFO frame is buffered immediately.

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
ADR=	P/F=	NR=	NS=	STRING	BCC		

↓

Enter NS Value:							
F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
RCVD_NR	SKIP	AUTO					

Figure 36-25 The N(S) field may be specified in a *SEND INFO* action.

Other N(S) options are RCVD_NR, SKIP, and AUTO. RCVD_NR means that you send the N(S) value that the other side says it is expecting. This is the valid N(S) in most cases, but not when you send two or more I-frames in a row without waiting for acknowledgment.

SKIP means that you *add one* to your correct N(S). This will look to the other side as though the line has taken a "hit" and a frame has been lost. This selection causes the window to be cleared.

NS= AUTO is the default setting for SEND INFO actions. AUTO means that every *new* INFO frame sent will have an N(S) value of one higher than the previous.

9. *String*. Strings are sent at X.25 Layer 2 only as adjuncts to frame-types when they are named in SEND actions. If you want to send a string of raw data without a protocol "envelope," you must go to Layer 1 and send the raw string from there.

Press the SEND softkey followed by the softkey for a frame type. Add any necessary or desired SEND options for the particular frame type. Then press the STRING softkey (Figure 36-25).

There is no spreadsheet keyword that identifies send-strings at any layer. *The spreadsheet compiler identifies strings by the quotation marks surrounding them.* Always enclose strings in quotation marks. (To send an actual "-"character in your string, type "\".) See Section 32 for more information on strings.

Here is a simple SEND action that includes no options besides a string:

```
ACTIONS: SEND FRMR "1,6,0"
```

And here is a SEND action that includes a full complement of optional fields, including a string:

```
ACTIONS: SEND INFO ADR= 03 P/F= 0 NR= AUTO NS= AUTO
        " %2%r This is user data." GDBCC
```

Most ASCII-keyboard, control, and hexadecimal characters are legal in a send-string. Special keys (F1, F2, F3) are not legal. Refer to Table 32-2.

To insert a canned fox message into a transmit string, type FOX inside of double parens, as follows: ((FOX)). Remember that the double parens are special characters produced by the [ctrl]-[9] and [ctrl]-[0] combinations. Constants, counters, and flags can also be embedded in a string. See Section 32, Strings.

10. BCC. There are three BCC options for every SEND action at X.25 Layer 2. One of the options, GDBCC, is the default. Any frame that does not request a bad BCC or an abort will have a good frame-check sequence calculated for it and appended to it. BCC also is an option for SEND actions at Layer 1; but it does not occur at Layer 3 or higher.

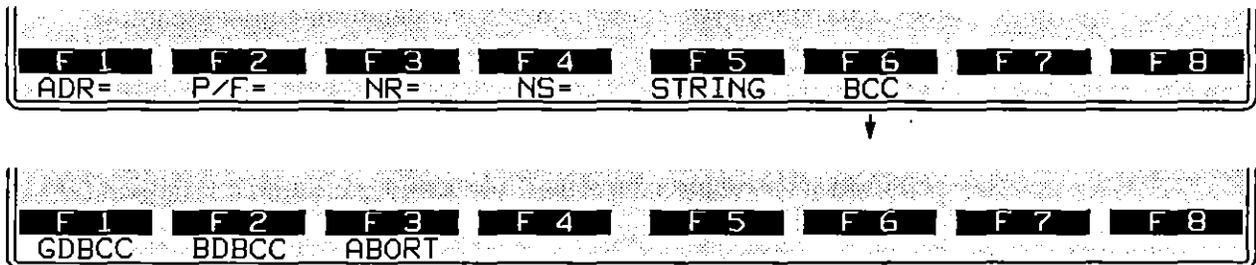


Figure 36-26 Type of BCC is a SEND option for frames at Layer 2.

The three softkey selections for BCC are shown in Figure 36-26. A sixteen-bit CCITT frame check is selected automatically for BOP protocols and cannot be changed or disabled. A bad BCC will be CRC-16 instead of CCITT.

When ABORT is the BCC selection, instead of appending a proper frame check the transmitter will hold the lead at mark for eight bits (or longer if the transmitter is idling r). Inside of a frame, seven 1-bits in a row are sufficient to signal an abort.

(B) Give Data

GIVE_DATA is the [F2] action on the first rack of action softkeys (refer to Figure 36-19). This action takes the I-field from a received INFO frame and passes it up to Layer 3 along with a DL_DATA IND primitive. (See Figure 33-5

in the section, OSI Primitives on the Protocol Spreadsheet.) In an emulate mode, data is delivered up to Layer 3 only by one of two actions at Layer 2: GIVE_DATA, or else a DL_DATA IND primitive followed by the data string.

(C) Resend

The RESEND function is mapped to **[F1]** on the second layer of action softkeys at Layer 2 for X.25. See Figure 36-27. The first RESEND action will resend the first frame in the window. The window is a queue that buffers INFO frames for retransmission in case one or more transmissions are lost or in error.

The first frame in the window always is the *earliest outstanding* (unacknowledged) frame. Every time an acknowledgment is received, the window is cleared to the extent of the acknowledgment and a new "first-frame" position is established. The first RESEND after an acknowledgment always sends the first window frame.

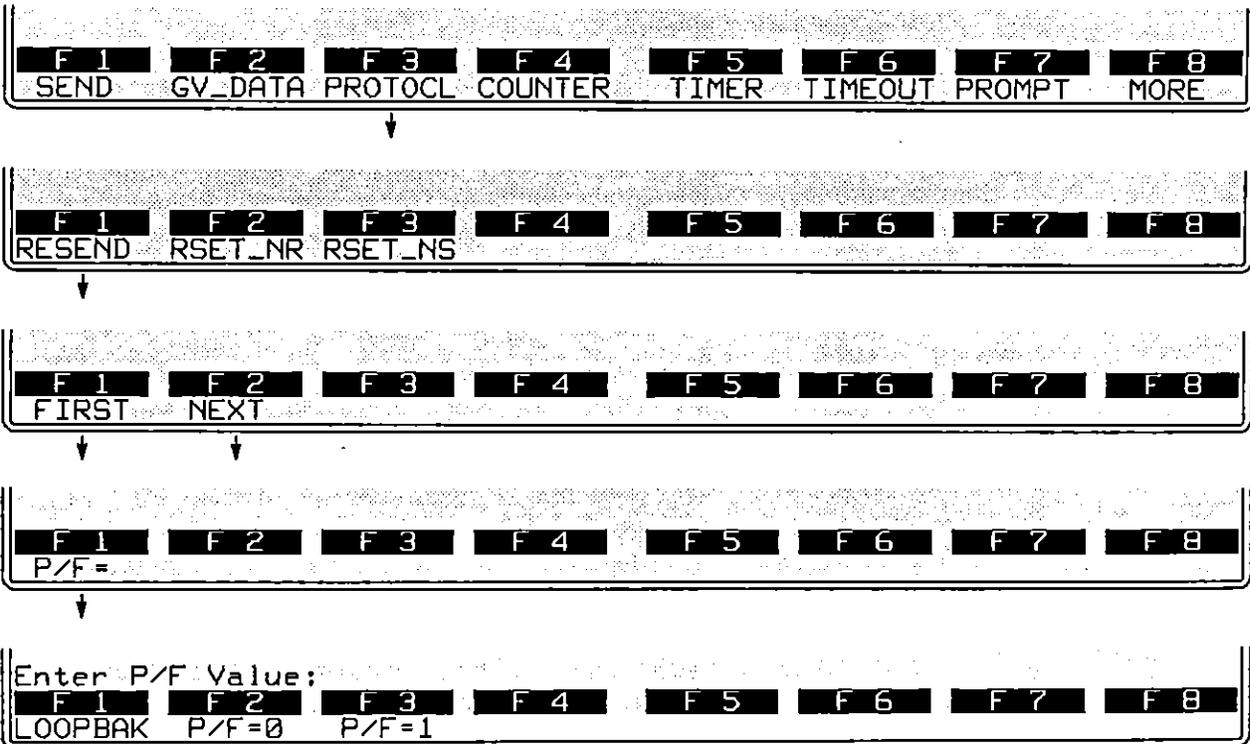


Figure 36-27 The RESEND action allows you to recover from sequence errors.

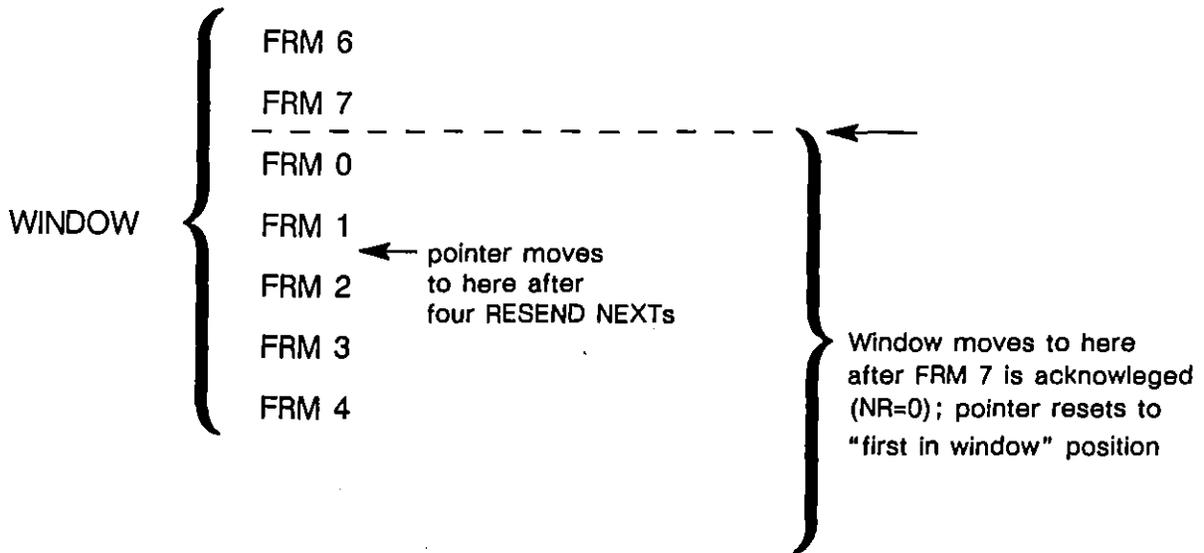


Figure 36-28 Resends cause the pointer to move, while acknowledgments move the pointer and the entire window.

The *second and subsequent* RESENDS following an acknowledgment also will send the first window frame, provided that the keyword **FIRST** is appended directly to the RESEND entry. Otherwise, they send the NEXT (second) and subsequent window frames. Figure 36-28 shows the position of the the resend "pointer" after four consecutive RESEND NEXT actions. RESEND NEXT is the default resend when neither **FIRST** nor **NEXT** is specified.

The resend-pointer is reset to the beginning of the window automatically by any acknowledgment, or by a RESEND FIRST action in the spreadsheet program.

1. *Resend first/next.* RESEND FIRST means that the resend-pointer is reset to the beginning of the window, the first frame in the window is resent, and the pointer is advanced to the second position in the window. The effect of a RESEND FIRST action is illustrated in Figure 36-29.

The RESEND FIRST action makes it possible for you to resend all the frames in the window one by one, and then *resend them again* if necessary.

2. *P/F=loopback/0/1.* The P/F bit in the resend-frame can be set to 0 or 1 by this optional action. If PF= LOOPBACK, the bit will echo the last P/F bit received. (Default is 1 in a RESEND action.)

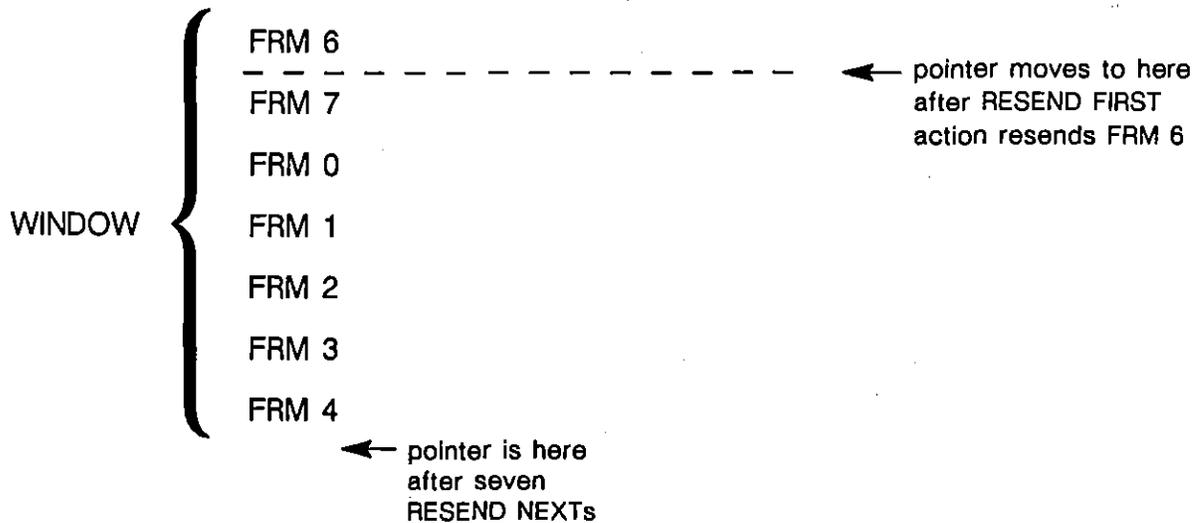


Figure 36-29 *RESEND FIRST* resets the pointer, allowing you to resend the entire window repeatedly.

(D) Reset N(R) and Reset N(S)

RESET_NR and RESET_NS are the **[F2]** and **[F3]** actions on the second rack of action softkeys for X.25 Layer 2. (Refer again to Figure 36-19.) The sequence-number fields in I-frames and supervisory frames can be reset by these two Protocol Spreadsheet actions. Sequence numbers are not reset *automatically* during a test by any frame that is sent or received.

RESET_NS also clears the transmit window.

36.6 Display Actions

ENHANCE and SUPPRESS pertain to lines of data on the Layer 2 protocol trace (see Section 36.2). They do not suppress and enhance data on the raw-data display. Raw data is enhanced and suppressed at Layer 1.

DTE, DCE, and RCV conditions can trigger an ENHANCE or SUPPRESS action. These conditions are active when the INTERVIEW is in monitor mode or in either of the emulate modes.

(A) Enhance

Whenever a DTE, DCE, or RCV condition comes true at Layer 2, the frame that satisfied the condition can be enhanced on the X.25 Layer 2 protocol-trace display, or it can be deleted from the trace completely. In an actions block on the Protocol Spreadsheet, press the ENHANCE softkey—**F2** on the third rack of action softkeys. Figure 36-30 shows the three softkey subselections beneath ENHANCE. They are REVERSE, BLINK, and LOW.

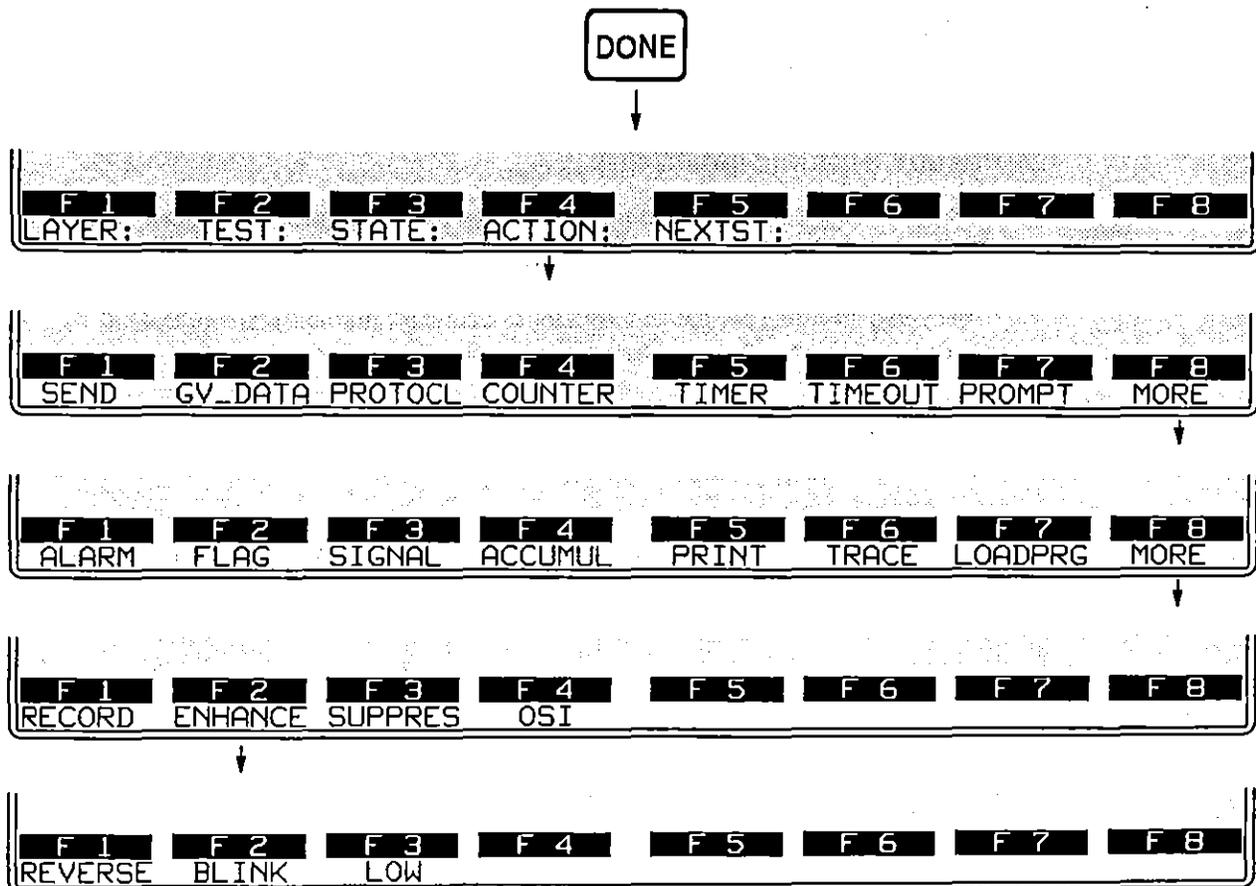


Figure 36-30 Selected frames on the protocol trace may be enhanced or suppressed.

Reverse-image and blink enhancements affect the plasma-display screen. In addition, a low-intensity enhancement can be applied to screens that are transmitted to a black-and-white monitor connected at the RS-170 port at the rear of the INTERVIEW.

Reverse, blink, and low enhancements can be mapped to colors on a color monitor attached at the INTERVIEW's RGB port (Figure 1-6). See Section 17.2 for an explanation of how reverse, blink, and low enhancements relate to character and background colors in the RGB output.

MON/DISK/FD2		BLK-00838 P 06/16/89 17:06							
ASCII/B/NONE/BOP									
SRC	ADDR	TYPE	Nr	Ns	P/F	SIZE	TIME	BCO	
DCE	01	RR	2		0	0002	1706:03.062	0	
DCE	03	INFO	2	0	0	0030	1706:03.179	0	
DTE	03	RR	1		0	0002	1706:03.213	0	
DCE	03	INFO	2	1	0	0008	1706:03.444	0	
DTE	03	RR	2		0	0002	1706:03.485	0	
DTE	01	INFO	2	2	0	0009	1706:03.727	0	
DCE	01	RR	3		0	0002	1706:03.842	0	
DTE	01	INFO	2	3	0	0021	1706:03.910	0	
DTE	01	INFO	2	3	1	0021	1706:04.153	0	
DCE	01	RR	4		1	0002	1706:04.202	0	
DCE	03	INFO	4	2	0	0007	1706:04.384	0	
DTE	03	RR	3		0	0002	1706:04.505	0	
DCE	03	INFO	4	3	0	0018	1706:04.696	0	
DTE	03	RR	4		0	0002	1706:04.741	0	
DTE	01	INFO	4	4	0	0007	1706:04.844	0	
F 1		F 2		F 3		F 4		F 5	
L2TRACE		L3TRACE							

Figure 36-31 A retransmitted DTE I-frame has been enhanced.

Figure 36-31 shows one screen of a Layer 2 protocol trace in which DTE INFO frames with the poll bit set to 1—retransmitted DTE INFO frames, in other words—have been enhanced in reverse video. The trigger that caused this enhancement was as follows:

CONDITIONS: DTE INFO P/F= 1
ACTIONS: ENHANCE REVERSE

(B) Suppress

Individual frames that are suppressed in Layer 2 actions are deleted from the trace display. Figure 36-30 shows the softkey path to SUPPRES.

36.7 Automatic Primitives

A table in a previous section (Table 33-2) listed the OSI service primitives that are monitored at the boundaries of Layer 2 as trigger conditions and sent up to Layer 3 or down to Layer 1 as user-entered spreadsheet actions. These primitives are

layer-specific rather than protocol-specific and are not part of the personality package for X.25 Layer 2; but a few of the primitives are set in motion automatically by X.25 Layer 2 spreadsheet actions. These *automatic* primitives can be thought of as part of the Layer 2 actions themselves, and by extension as part of the X.25 protocol package.

Table 36-2 gives the set of X.25 Layer 2 actions that have action-primitives built into them. For example, whenever a GIVE_DATA action occurs at Layer 2, a DL_DATA IND primitive is forwarded to Layer 3, where a DL_DATA IND condition may be waiting to monitor it.

Whenever a SEND or RESEND action is initiated at Layer 2, a PH_DATA REQ primitive is sent downward along with the PH data (the entire frame).

If a SEND or RESEND action is triggered at Layer 2 while the physical connection at Layer 1 is inactive, Layer 2 will sense the absence of a physical connection and delay the PH_DATA REQ. Instead it will send a PH_ACTIVATE REQ primitive. Only when a PH_ACTIVATE CONF has been returned by Layer 1 will Layer 2 release the data and the data primitive.

NOTE: The RS-232 interface does not distinguish active/inactive status at the physical level. This interface returns PH_ACTIVATE CONF automatically whenever it sees PH_ACTIVATE REQ.

Table 36-2
Automatic Primitives Generated at X.25 Layer 2

X.25 Layer 2 Action	Automatic Primitive	To Layer
GIVE_DATA	DL_DATA IND	3
SEND (TYPE)	(PH_ACTIVATE REQ*) PH_DATA REQ	1 1
RESEND	(PH_ACTIVATE REQ*) PH_DATA REQ	1 1

*Sent if Layer 1 shows inactive status. PH_DATA REQ delayed until PH_ACTIVATE CONF returned by Layer 1.

36.8 Programming Example: Converting Protocol Bytes to Hexadecimal

Listed below is a simple, four-trigger test that enhances the "readability" of the raw display of X.25 data. When this test is entered on the trigger menus or anywhere in the Protocol Spreadsheet program at *Layer 1*, frame-level and packet-level protocol bytes on the raw-data display are converted automatically to hexadecimal, while all user-data is translated into ASCII characters. Figure 36-32 shows the same data both *before* and *after* the test has been applied.



Figure 36-32 In the X.25 data on the right, a simple enhancement program has converted protocol bytes to hexadecimal.

Because it occupies a single state, the protocol-hex test can be entered on the trigger menus; or it may be included in the Protocol Spreadsheet program in this form:

```
LAYER: 1
TEST: ptcl_hex
STATE: enhance
CONDITIONS: DTE STRING "[F][F]([F][A][D][NOT][E][QUAL][F][A][D])((XXXXXXXX0))((XXXXXXXX))[(XXXXXXXX0)]"
ACTIONS: ENHANCE DTE HEX OFF
CONDITIONS: DTE GOOD_BCC
ACTIONS: ENHANCE DTE HEX ON
CONDITIONS: DCE STRING "[F][F]([F][A][D][NOT][E][QUAL][F][A][D])((XXXXXXXX0))((XXXXXXXX))[(XXXXXXXX0)]"
ACTIONS: ENHANCE DCE HEX OFF
CONDITIONS: DCE GOOD_BCC
ACTIONS: ENHANCE DCE HEX ON
```

In the strings, [F] [F] ([F][A][D][NOT][E][QUAL][F][A][D]) looks for the beginning of the frame. The first bit mask—(XXXXXXXX0)—looks for I-frames. The second bit mask looks for a Q ("data-qualified") bit that equals zero. The [(entry ([NOT]) causes the search string to skip one byte, the LCN byte in the packet header. The third bit mask looks for data packets. Only *unqualified* data packets satisfy the string and turn the hex enhancement *off*. The user data that follows is translated into ASCII.

The test in this form makes two assumptions: the numbering mode is MOD 8 and apart from X.29, no protocol is implemented above Layer 3.

36.9 Programming Example: A Simple "Automatic" Layer 2 X.25 Test

Here is a simplified test that makes Layer 2 "automatic" and "transparent" when you are working at Layer 3 or higher.

The test initiates a link-startup sequence when a DL_CONNECT REQ primitive arrives from Layer 3. This primitive will be sent down automatically at Layer 3 as soon as a SEND RESTART or other Layer 3 SEND action is attempted.

Automatic Layer 2 X.25 Test:

LAYER: 2

STATE: Init

CONDITIONS: DL_CONNECT REQ
ACTIONS: SEND SABM P/F=1
TIMEOUT t1 RESTART 3.000

CONDITIONS: RCV UA P/F=1
ACTIONS: DL_CONNECT CONF
TIMEOUT t1 STOP
RESET_NR
RESET_NS
NEXT_STATE: Info_xfr

CONDITIONS: TIMEOUT t1
ACTIONS: SEND SABM P/F=1
TIMEOUT t1 RESTART 3

CONDITIONS: RCV SABM
ACTIONS: DL_CONNECT IND
TIMEOUT lyr3_resp RESTART 1

CONDITIONS: DL_CONNECT RESP
ACTIONS: SEND UA P/F= LOOPBACK
RESET_NR
RESET_NS
NEXT_STATE: Info_xfr

CONDITIONS: TIMEOUT lyr3_resp
ACTIONS: PROMPT "Layer 3 not responding"

STATE: Info_xfr

CONDITIONS: DL_DATA REQ
ACTIONS: SEND INFO "(DL_DATA)"

CONDITIONS: RCV INFO
ACTIONS: SEND RR RESP
GIVE_DATA

CONDITIONS: T1_EXPIRED
ACTIONS: RESEND FIRST P/F= 1

CONDITIONS: RCV REJ RESP
NEXT_STATE: xmt_wndw

```

STATE: xmt_wndw

CONDITIONS: ENTER_STATE
ACTIONS: RESEND FIRST

CONDITIONS: FRAME_SENT
MORE_TO_RESEND
ACTIONS: RESEND NEXT

CONDITIONS: FRAME_SENT
NO_MORE_TO_RESEND
NEXT_STATE: Info_xfr

```

The link startup also can be initiated from "below," by a SABM received from the other side of the link. Note that in this Layer 2 program, the UA response to a SABM is made contingent on a Layer 3 program above that acknowledges the link-startup (DL_CONNECT) indication. The link establishment is designed to fail (while displaying the operator prompt, "Layer 3 not responding") unless the following minimum Layer 3 program is included on the Protocol Spreadsheet:

```

LAYER: 3

STATE: dl_connect

CONDITIONS: DL_CONNECT_IND
ACTIONS: DL_CONNECT_RESP

```

Here is a fuller Layer 3 program that initiates link setup and also handles the transfer of Restart packets that brings the interface to the Ready (for calls) state.

```

LAYER: 3

STATE: dl_connect

CONDITIONS: ENTER_STATE
ACTIONS: DL_CONNECT_REQ

CONDITIONS: DL_CONNECT_CONF
NEXT_STATE: packet_level_ready

CONDITIONS: DL_CONNECT_IND
ACTIONS: DL_CONNECT_RESP
NEXT_STATE: packet_level_ready

STATE: packet_level_ready

CONDITIONS: ENTER_STATE
ACTIONS: SEND_RESTART

CONDITIONS: RCV_RESTART
ACTIONS: RESET_PR_PS
NEXT_STATE: ready_to_call

CONDITIONS: RCV_RESTART_CONF
ACTIONS: RESET_PR_PS
NEXT_STATE: ready_to_call

STATE: ready_to_call

```

In *Info_xfr* state, the test receives *DL_DATA* from Layer 3 and *sends* it down to Layer 2. It *gives* *DL_DATA* up to Layer 3. Recovery actions are taken, as follows: when T1 expires, the test resends the first frame in the window. When a REJ frame is received, the test moves to *xmt_wndw* state and resends the entire window before returning to *Info_xfr* state.

37 X.25 Layer 3

```

** Layer Setup **

DRIVE: HRD   Layer 1 Package: NO PACKAGE   Packages Loaded: NO PACKAGE
DRIVE: FD2   Layer 2 Package: X.25         SDLC             FDI
DRIVE: FD2   Layer 3 Package: X.25         NO PACKAGE
DRIVE: HRD   Layer 4 Package: NO PACKAGE   NO PACKAGE
DRIVE: HRD   Layer 5 Package: NO PACKAGE   NO PACKAGE
DRIVE: HRD   Layer 6 Package: NO PACKAGE   NO PACKAGE
DRIVE: HRD   Layer 7 Package: NO PACKAGE   NO PACKAGE

Depress XEQ Key To Load The Selected Packages

Select Layer
F1 F2 F3 F4 F5 F6 F7 F8
LAYER-1 LAYER-2 LAYER-3 LAYER-4 LAYER-5 LAYER-6 LAYER-7 PROTSEL
    
```

Figure 37-1 The X.25 personality package for Layer 3 is loaded from the Layer Setup screen.

```

** X.25 Packet Level Setup **

Emulate: LOGICAL DTE           Mode of Operation: MOD 8
Window Size: 7
Low Outgoing Channel#: 001   High Outgoing Channel#: FFF

PATH LCN      CALLED          CALLING      FACILITIES   DATA
0 _____
1 _____
2 _____
3 _____
4 _____
5 _____
6 _____
7 _____
8 _____

Enter Mode (MOD 8 Or MOD 128):
F1 F2 F3 F4 F5 F6 F7 F8
MOD 8 MOD 128
    
```

Figure 37-2 Protocol configuration screen for X.25 Layer 3.

37 X.25 Layer 3

Layer 3 X.25 is a "layer personality package" of functions that are loaded into memory from disk via the Layer Setup screen. Figure 37-1 shows the Layer Setup screen configured to load in the Layer 2 and Layer 3 X.25 packages from floppy-disk drive #2. Refer to Section 8 for details on operating the Layer Setup screen.

The Layer 3 X.25 package consists of the following:

- A special X.25 Packet Level Setup screen (shown in Figure 37-2) that controls certain parameters when the unit is tracing or emulating X.25.
- A protocol trace (illustrated in Figure 37-4) that distills from X.25 data the packet-level events that have protocol significance. This trace is accessible by softkey in Run mode at all times.
- A group of conditions and actions at Layer 3 on the Protocol Spreadsheet that facilitate X.25 programming. Figure 37-9 shows the softkey path to the first rack of condition softkeys when the X.25 package is loaded in at Layer 3.

37.1 Packet-Level Setup

The X.25 Packet Level Setup screen must be configured correctly for an accurate trace display and for proper emulation.

To bring up this screen, first go to the Layer Setup screen (press **PROGRAM**, **F5**). Execute the X.25 selection at Layer 3: X.25 should appear in the **Packages Loaded** column. Press **F8** (labeled **PROTSEL**) to bring up the prompt to **Select Protocol Configuration Screen**. Then press **F3** (**LAYER-3**) to call up the X.25 Packet Level Setup screen.

The five parameter fields on this screen as well as the path-data entry fields are shown in Figure 37-2. **Emulate**, **Window Size**, **Low Outgoing Channel #**, **High Outgoing Channel #**, and the entire path-data area apply only to interactive (emulate) tests. **Mode of Operation** must be configured correctly for the protocol trace as well as for proper emulation.

(A) Emulate Logical DTE/DCE

There are two selections in the **Emulate** field on the X.25 Packet Level Setup screen, **LOGICAL DTE** and **LOGICAL DCE**. The entry in this field determines the order of assignment when LCNs are assigned dynamically to call requests during interactive testing.

Configured as a logical DTE, the INTERVIEW assigns LCNs in a descending sequence beginning with the High Outgoing Channel #. See Section (D), below. Configured as a logical DCE, the INTERVIEW assigns LCNs in ascending sequence beginning with the Low Outgoing Channel #.

(B) Mode of Operation

The **Mode of Operation** field selects the mode of numbering DATA and supervisory packets. There are two options, **MOD 8** and **MOD 128**.

MOD 8 uses sequence numbers 0-7. MOD 128 adds an extra byte to the control field in DATA, RR, RNR, and REJ packets. See Figure 37-6. This extra byte allows sequence numbers in a range of 0-127.

The correct "modulus" must be selected in this field in order to conduct interactive communications and also to generate an accurate X.25 Layer 3 trace.

(C) Window Size

Any window size may be entered up to the current modulus minus one: 7 or 127. The window size is the maximum number of unacknowledged data packets that Layer 3 will buffer for retransmission. When the limit is reached, any further data packets that are named in SEND actions triggered at Layer 3 will be passed to Layer 2 but not buffered for resending.

According to CCITT Recommendation X.25, the standard window size is 2. This means that two packets can be outstanding (unacknowledged) at a time.

The window is a queue that buffers packets for retransmission in case one or more transmissions are lost or in error. A RESEND action will resend the first (earliest) packet in the window. Successive RESENDS will send successive packets until there are no more packets to resend; or until the window is reset by an acknowledgment or by a RESEND FIRST action.

(D) Low Outgoing Channel#/High Outgoing Channel#

Logical Channel Numbers (LCNs) may be assigned dynamically on a per-call basis; or they may be reserved during Program mode for a particular call destination (or "path") by means of an entry in the LCN column on the X.25 Packet Level Setup screen.

If the LCN column on the Packet Level Setup screen is left blank with respect to a particular call destination ("path"), an LCN will be assigned dynamically each time the INTERVIEW initiates a call on that network path, as follows:

If the INTERVIEW has been configured as a logical DCE on the Layer 3 Protocol Configuration screen, it will select the Low Outgoing Channel number for the first call to the DTE. Assuming that the first call still is in session, the next call initiated by the INTERVIEW will have the next higher LCN, and so forth, to the upper limit for LCNs established in the High Outgoing Channel# menu field.

If the INTERVIEW has been configured as a logical DTE, the first call initiated by the INTERVIEW will receive the High Outgoing Channel number. If the first call remains in session, the second call will have the second-highest LCN, and so forth, to the lower limit for LCNs set in the Low Outgoing Channel# field.

(E) Path

"Path" is the route of a Call Request packet through the X.25 network, on the way to its destination address. Call Request packets establish the path of a call. When data packets enter the network, they carry the same LCN as the call packet. They use the LCN to identify themselves at each network node, where they are routed along the path already established for the call.

Packets that are programmed to be "sent" on the Protocol Spreadsheet must indicate their destination—that is, they must declare the call that they belong to. In the network, they will use the LCN to identify their call. But the LCN cannot be used for identification on the INTERVIEW's Protocol Spreadsheet, since LCNs usually are assigned dynamically at the time of the call—in which case they cannot be pre-programmed.

Instead, packets on the spreadsheet are provided with a "path" number that ties them, on the X.25 Packet Level Setup screen, to a particular set of Call Request parameter values. Here is an example of how the procedure works:

On the Packet Level Setup screen (see Figure 37-3), the following entry is made in the CALLED column for Path #0: 300170345678. On the Protocol Spreadsheet, this Actions entry is made: SEND CALL PATH= 0.

When the SEND action is triggered, a Call Request packet will be formed with the Called address given for Path #0 on the Packet Level Setup screen: 300170345678. An LCN will be assigned to this call packet dynamically, according to the rules for low and high outgoing channel numbers outlined in Section 37.1(D), above. As long as the call is active, data packets and other packets sent on this *path*—for example, SEND DATA PATH= 0—will carry the same LCN.

(F) LCN

Normally the LCN field for a particular call (or "path") is left blank. During Run mode when the Call Request is created, the LCN is assigned dynamically according to the rules for low and high outgoing channel numbers described above. For the duration of the call, data packets and other packets are constructed with the same LCN.

```

** X.25 Packet Level Setup **

Emulate: LOGICAL DTE           Mode of Operation: MOD 8
Window Size:                    Low Outgoing Channel#: 001  High Outgoing Channel#: 1F
Enter called address for entry #0 (decimal): 300170345678

```

PATH	LCN	CALLED	CALLING	FACILITIES	DATA
0		300170345678		001	0000
1					
2					
3					
4					
5					
6					
7					
8					

F 1 F 2 F 3 F 4 F 5 F 6 F 7 F 8

Figure 37-3 Packet-level setup screen with sample data.

An LCN also can be predefined by the user. This designation is made in the LCN column of the Packet Level Setup screen, not on the Protocol Spreadsheet. The SEND action on the spreadsheet still references only the path number. On the Packet Level Setup screen, that path number is correlated to the LCN that the user enters on the same row.

The LCN field is referred to as a "hex" field. This means that each column (character space) in the data-entry field will equate to one 4-bit, hexadecimal digit on the actual data screen. For example, a data screen may show the two-character sequence %?, where the second, third, and fourth digits represent the LCN. The LCN entry on the Packet Level Setup screen was 123. Or the character data may show this sequence: %t. The LCN on the setup screen was 1E (also 01E).

(G) CALLED

Enter the called address in this field. Addresses are considered "decimal" entries. This means that each column or character space in the data-entry field will equate to a 4-bit, binary-coded decimal (BCD) digit on the actual data screen. Use the numbered keys to make this entry—do not use the [HEX] key to turn on the hex function. A sample address entry is shown in Figure 37-3.

(H) CALLING

Enter the calling address in this field. (Logical DTEs often omit this address: the network knows the addresses of dedicated lines coming into the node and may not require them.)

Addresses are considered "decimal" entries. This means that each column or character space in the data-entry field will equate to a 4-bit, binary-coded decimal (BCD) digit on the actual data screen. Use the numbered keys to make this entry—do not use the **HEX** key to turn on the hex function. A sample address entry is shown in Figure 37-3.

Do not make any allowance in this field for the Address Length byte (see Figure 37-6). That byte is provided automatically.

(I) FACILITIES

Enter the entire Facilities field as it will appear in the Call Request packet. *Omit the Facilities Length byte; that is handled automatically.*

The **FACILITIES** field is referred to as a "character" field. This means that characters—including hexadecimal characters—in the data-entry field will equate one-for-one with the characters on the actual data screen.

The Facilities entry in Figure 37-3 will be transmitted in a Call Request packet exactly as it appears in this field. With a Facilities Length byte preceding it, it will look like this on the data display: %%.

(J) DATA

Enter the Data field as it will appear in the Call Request packet. If you want a Data field that is longer than the ten character spaces in the **DATA** field, you can append a *string* to your Call packet on the Protocol Spreadsheet.

The Data field on the Packet Level Setup screen is a "character" field. This means that the Data entry in Figure 37-3 will be transmitted exactly as it appears in this field. (As in any transmit string on an **INTERVIEW** screen, in normal bit order the *rightmost* bit in the *leftmost* byte will be the first bit transmitted.)

Based on the entries for Path #0 in Figure 37-3, the **INTERVIEW** will send the following Call Request *packet* in a **SEND CALL PATH= 0** action. Frame-level bytes are not shown. Assume an LCN of 01E.

hex:

```
10 1E 00 C3 01 70 34 57 02 00 00 00 00
```

ASCII:

```
0 1E x F 0 34 p 4 V x 5 5 5 5 5 5
```

37.2 Protocol Trace

The Layer 3 X.25 package includes an automatic packet-trace display that summarizes packet-level activity. This trace mode is enabled whenever the unit is in Run mode, both real-time and frozen.

MON/LINE		BLK=00000 S 06/24/89 17:55									
ASCII/8/NONE/BOP											
SRC	LCN	TYPE	Pr	Ps	QDM	MISC	SIZE	TIME	BCC		
DTE	004	CALL				07400000000000	0015	1751:18.233	0		
DCE	004	CALLACC				090112111000	0003	1751:18.454	0		
DCE	004	DATA	0	0	Q		0030	1751:18.484	0		
DTE	004	RR	1				0003	1751:18.608	0		
DCE	004	DATA	0	1			0030	1751:18.629	0		
DCE	004	DATA	0	2	Q		0008	1751:18.728	0		
DTE	004	RR	2				0003	1751:18.754	0		
DTE	004	RR	3				0003	1751:18.804	0		
DTE	004	DATA	3	0	Q		0008	1751:18.834	0		
DTE	004	DATA	3	1			0009	1751:19.047	0		
DCE	004	RR	2				0003	1751:19.178	0		
DCE	004	DATA	2	3	Q		0006	1751:19.319	0		
DTE	004	RR	4				0003	1751:19.359	0		
DCE	004	DATA	2	4			0027	1751:19.384	0		
DTE	004	RR	5				0003	1751:19.494	0		

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
L2TRACE	L3TRACE						

Figure 37-4 Each horizontal row on the trace display represents a packet.

While the unit is in Run mode, press the softkey for PROTOCL (F2) on the primary rack of display-mode softkeys) and then the softkey for L3TRACE (F2) to bring the protocol trace for X.25 Layer 3 to the screen. Figure 37-4 is an example of this trace display. Each horizontal row in the trace represents a packet.

(A) The Protocol Trace in Freeze Mode

Press **FREEZE** to prevent the addition of new data to *all* the display buffers, including the trace buffers. The frozen trace display may be scrolled through or paged through. The top line always is the cursor line (though there is no actual cursor on the trace display). Pressing **DOWN** or **↓** moves the viewing "window" down relative to the data to add one line of fresher data to the bottom of the screen. Pressing **UP** or **↑** moves the viewing window up to add a line of older data to the top of the screen.

Depression of the **FREEZE** key adds fifteen lines—one full page—of newer frames to the frozen trace screen. Depression of **FREEZE** adds fifteen lines of older frames.

The packet displayed on the top line of frozen trace-data will appear in the first frame in the raw-data or data-plus-leads display. To view the character data

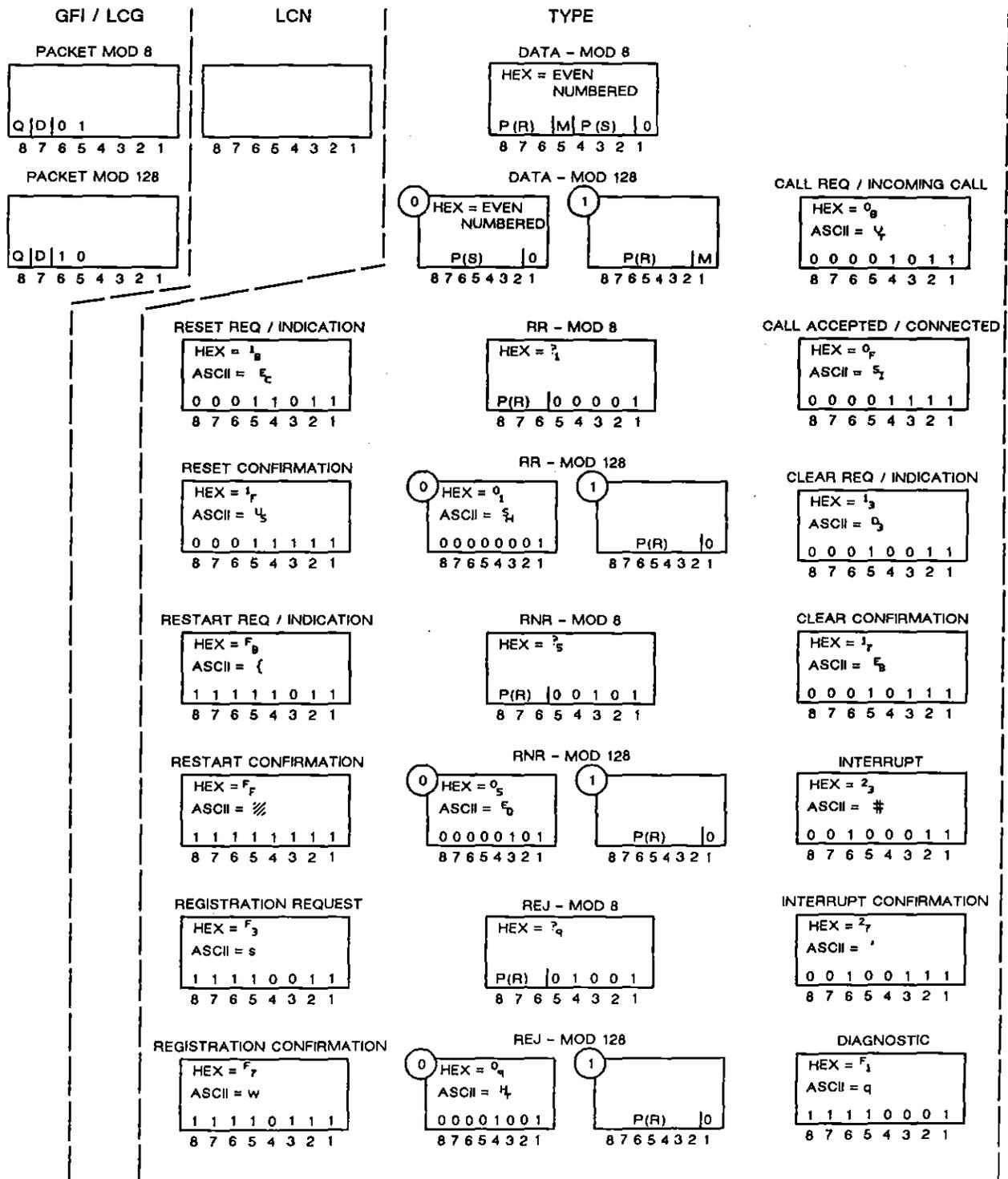
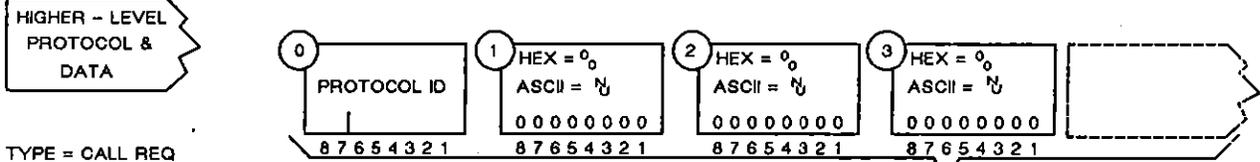


Figure 37-6 X.25 packet fields.

DATA

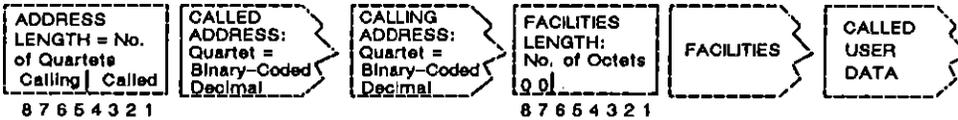
TYPE = DATA



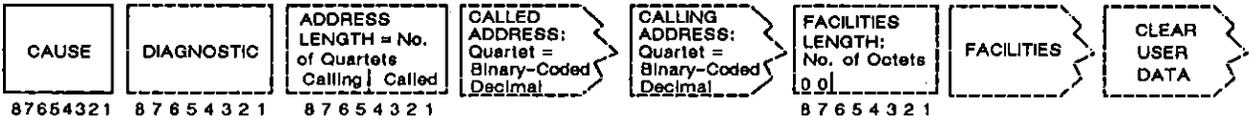
TYPE = CALL REQ



TYPE = CALL ACCEPTED



TYPE = CLEAR REQ



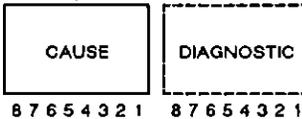
TYPE = CLEAR CONFIRM



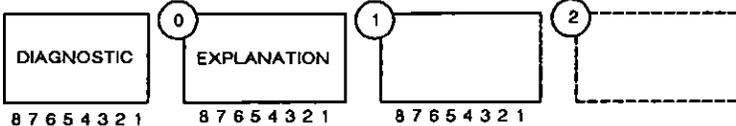
TYPE = INTERRUPT



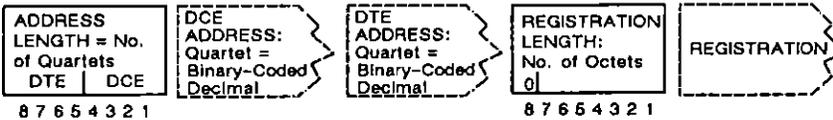
TYPE = RESET or RESTART REQ



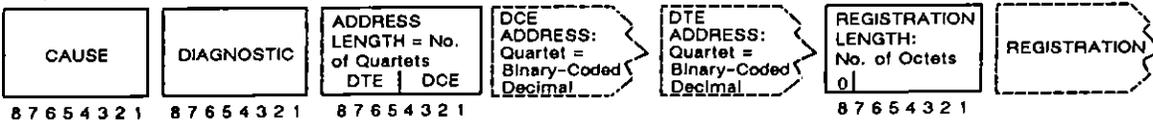
TYPE = DIAGNOSTIC



TYPE = REGISTRATION REQ



TYPE = REGISTRATION CONFIRM



(B) Trace Columns

The columns in the protocol trace for Layer 3 X.25 are explained below.

1. *Source*. The SRC column identifies the lead on which the packet was monitored, TD (DTE) or RD (DCE). This column identifies the *physical* source of the packet, not the *logical* source. The physical DTE uses the TD lead to transmit. The physical DCE uses RD to transmit.

Just as on the data display, RD data is underlined.

2. *LCN*. The LCN for each packet is given in this three-column "hex" field. Each column displays a hexadecimal *digit* (0 through F) that represents four bits out of the twelve-bit LCN.
3. *Type*. The mnemonic (abbreviated) names for seventeen packet types as they appear in the TYPE column of the protocol trace are shown in Figure 37-6 under "TYPE." If a Type octet does not fit any of the patterns in the figure, the packet is listed in the TYPE column as UNKWN followed by the hexadecimal value of the type byte: UNKWN=03.

If the number of bytes in the packet is below the required minimum, the packet is posted as INVALID.

4. *P(R) and P(S)*. One column on the packet-level trace is devoted to P(R) values, and one column to P(S). The packet types that include P(R) or P(S) fields in their control fields are indicated in Figure 37-6. P(R) and P(S) occupy three bits each in modulo 8, seven bits each in modulo 128.

MON/DISK/FD2		BLK=04784	
ASCII/8/NONE/BOP			
SRC	LCN	TYPE	QDM
DCE	004	DATA	6
DCE	004	DATA	7
DCE	004	DATA	8
DTE	004	RR	9
DCE	004	DATA	A
DCE	004	DATA	B
DCE	004	DATA	C
DCE	004	DATA	D
DCE	004	DATA	E
DCE	004	DATA	F
DCE	004	DATA	0
DCE	004	DATA	1
DCE	004	DATA	2
DCE	004	DATA	3

F 1	F 2	F 3	F 4
L2TRACE	L3TRACE		

Figure 37-7 MOD 128 sequence numbers are displayed in two-digit hexadecimal characters.

P(R) and P(S) values are presented in decimal format in modulo-8 traces. Each column displays a single digit that represents a 3-bit binary value (0 through 7). For modulo 128, the values 0 to 7 are given in "character" format, where the columns contain a two-digit hexadecimal character (see Figure 37-7).

Note that the Pr and Ps columns on the trace are staggered to suggest four columns. The two outside columns, comprised of the DTE's P(R) value and the DCE's P(S) value, form a numbering sequence for DCE data packets. The arrows in Figure 37-8 indicate the sequence: the DCE sends packet 4, the DTE acknowledges 4 by returning P(R) 5; the DCE sends 5, the DTE expects 6; the DCE sends 6, the DTE expects 7; the DCE sends 7, the DTE expects 0; the DCE sends 0.

MON/LINE		BLK=0000 5				
ASCII/8/NONE/BOP						
SRC	LCN	TYPE	Pr	Ps	QDM	MI
DCE	004	DATA	2	4		
DTE	004	RR	5			
DCE	004	DATA	2	5		
DTE	004	RR	6			
DTE	004	DATA	6	2		
DCE	004	RR	3			
DCE	004	DATA	3	6		
DTE	004	RR	7			
DTE	004	DATA	7	3		
DCE	004	RR	4			
DCE	004	DATA	4	7		
DTE	004	RR	0			
DTE	004	DATA	0	4		
DCE	004	RR	5			
DCE	004	DATA	5	0		

F 1
F 2
F 3
F 4

L2TRACE L3TRACE

Figure 37-8 Pr and Ps columns are staggered, with the outside columns representing the sequence of DCE numbered data packets.

The two inside columns reveal a similar pattern for DTE data packets (and DCE acknowledgements).

5. *Q, D, and M.* QDM is a three-column field. If the Q (data-qualified) bit is set in a data packet, a Q will appear in the Q column for that row. See Figure 37-4 for examples of this letter-Q display. The position of the Q bit in the first packet byte is indicated at the top left of Figure 37-6.

When the D (delivery) bit is set in a Call, Call Accept/Connect, or Data packet, the letter D appears in the D column. The position of the D bit in the first packet byte is indicated in Figure 37-6.

When the M (*more*) bit is set in a data packet, the letter M appears in the M column on the trace display. The position of the M bit in the Type field of data packets is shown in Figure 37-6.

6. *Misc.* The MISC field presents up to sixteen bytes of character data (decoded in hex) for all packet types *other than data packets* that contain data beyond the Type octet. All such packets and their "miscellaneous" fields are indicated on the right half of Figure 37-6.

Twelve bytes of "miscellaneous" data were expanded for the Call packet in the trace in Figure 37-4. The data in this example includes the address-length byte, four called-address bytes, the facilities-length byte, two facilities bytes, and four bytes of call-user data.

7. *Size.* The number of bytes in each packet is given in this field in four decimal digits.
8. *Time.* The time of the arrival of the *end of the frame containing the packet* at the DTE or DCE monitor is provided by a 24-hour clock and posted to the trace display. The clock is accurate to the millisecond.

When **Time Ticks: ON** is selected on the Front-End Buffer Setup screen (see Section 9), time values are incorporated into the data itself. As a result, times posted to the trace display will not be affected when data captured to disk is played back, even at varying speeds.

If **Time Ticks: OFF** is selected instead, times on the trace during playback will be influenced by "local conditions" such as playback speed, idle suppression, etc.

9. *Frame checking.* An X.25 frame ends as soon as a flag or seven 1-bits in a row are detected. If a flag ends the frame, a frame check is performed and the result is posted both to the data display and to the **BCC** column of the trace display. The symbol denotes a good frame check, while symbolizes a bad frame.

for abort is posted to the trace row when a frame is ended by seven 1-bits.

37.3 Monitor Conditions

When the Layer 3 X.25 personality package is loaded in (via the Layer Setup screen), a set of conditions checks DTE and DCE leads both in monitor and emulate modes. This set of conditions is accessed by the DTE and DCE selections on the first rack of condition softkeys at Layer 3. See Figure 37-9.

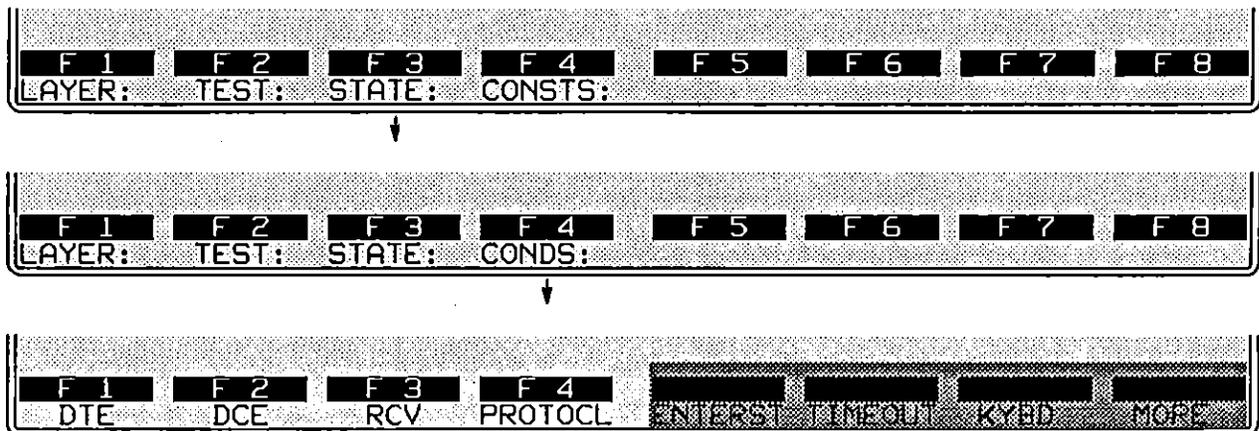


Figure 37-9 Unlike RCV conditions, the softkeys for DTE and DCE are valid when the INTERVIEW is monitoring the line passively.

After the keyword DTE (or DCE) is written to the spreadsheet, a rack of softkeys appears that represents *types* of packets: DATA, RR, RNR, REJ, CALL, and so forth.

(A) Packet Type

The softkeys for data, supervisory, unnumbered, and “other” packets are illustrated in Figure 37-10.

Press a softkey to write one of these packet types to the Layer 3 spreadsheet. DTE or DCE followed by a packet-type mnemonic—DTE DATA, for example, or DCE CLEAR—is a complete condition and will come true if a matching packet is monitored. An LCN condition may be added to the simple packet mnemonic, but it is optional. Other optional conditions that may apply are Q-bit value, D-bit value, M-bit value, cause code, and diagnostic code.

NOTE: A packet-type condition will not come true with respect to a packet that is inside of a frame with a bad frame check, or inside of an aborted frame.

1. *Data packets.* Data packets differ for MOD 8 and MOD 128 numbering schemes. (See Figure 37-6.) For spreadsheet conditions to match data

packets accurately, the correct numbering system ("Mode of Operation") should be selected on the Packet Level Setup screen.

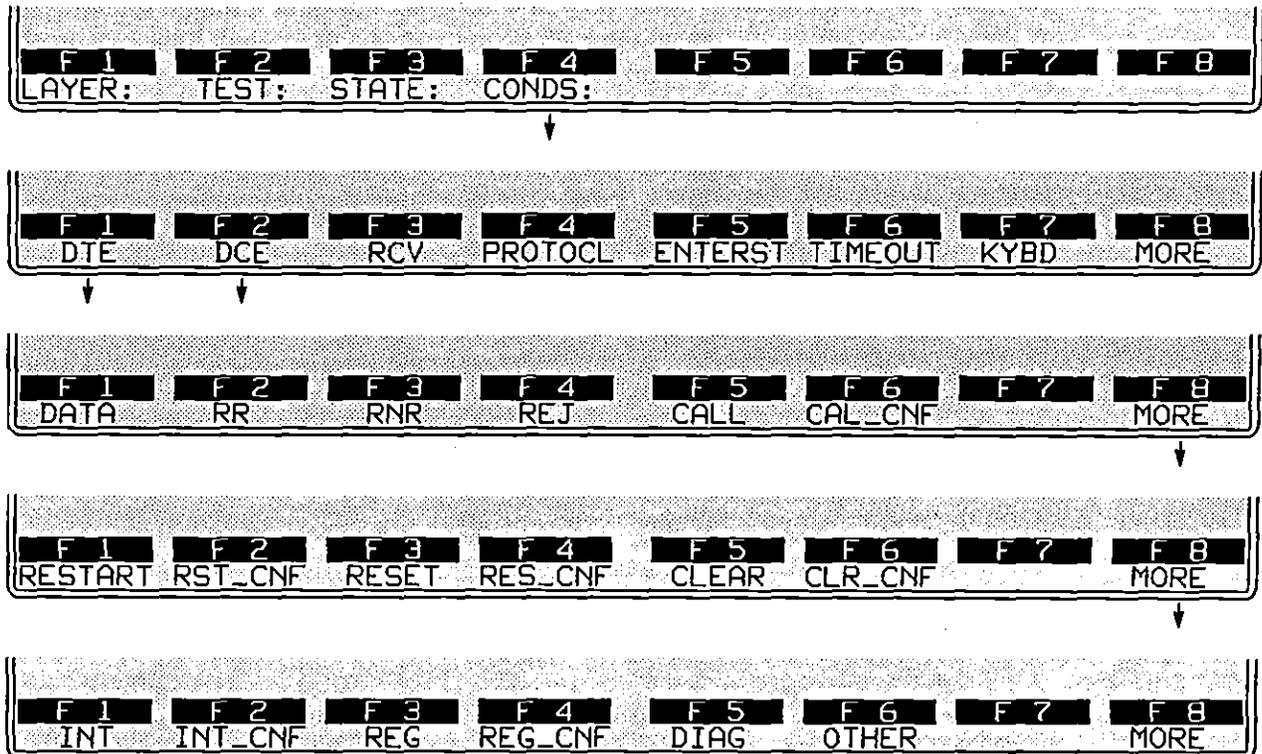


Figure 37-10 Packet types.

When DTE or DCE is monitored for a data packet, a specific LCN may be specified in the spreadsheet condition. A specific value for the Q, D, or M bit also may be indicated in the rack of spreadsheet softkeys just below DATA. (Refer to Figure 37-12.)

2. *Supervisory packets.* The three supervisory-packet types that can be searched for on the data leads are RR (Receive Ready), RNR (Receive Not Ready), and REJect. These packets always contain P(R) fields (see Figure 37-6) and serve mainly to acknowledge or reject data packets.

Like data packets, supervisory packets are constructed differently according to the numbering scheme, MOD 8 or MOD 128.

When DTE or DCE is monitored for a supervisory packet, a specific LCN may be specified in the spreadsheet condition. See Figure 37-11.

3. *Unnumbered packets.* Unnumbered packets generally assist in call setup, call management, call clearing, and subscription services.

The thirteen unnumbered packet types are laid out consecutively from CALL to DIAG on the softkey racks in Figure 37-10. Because these packets lack P(R) and P(S) sequence numbers, they are constructed identically for MOD 8 and MOD 128.

All unnumbered-packet conditions may be made specific to a particular LCN. Call and Call Confirm conditions may specify a D-bit value (Figure 37-13). Restart, Reset, Clear, and Registration Confirm conditions may optionally test for causes and diagnostic codes (see Figure 37-14, Figure 37-15, and Figure 37-16). Diagnostic packets (F5 on on the bottom rack of softkeys in Figure 37-10) also may specify a diagnostic code.

4. *Other packets.* Any packet type may be entered as a hexadecimal value instead of by name. Press the softkey for OTHER. (See F6 in the bottom rack of softkeys in Figure 37-10.) Then enter the hex byte in the form of two alphanumerics. Here, for example, is a Clear Request entered as a hexadecimal:

CONDITIONS: DCE OTHER 13

(B) LCN

All DTE and DCE conditions that name a packet type may specify one particular LCN (logical channel number) as an added condition. For example, a spreadsheet condition may be satisfied by *any* DTE Clear Request packet:

CONDITIONS: DTE CLEAR

Or it may be satisfied by a DTE Clear Request packet only if it carries an LCN of, say, 005:

CONDITIONS: DTE CLEAR LCN= 005

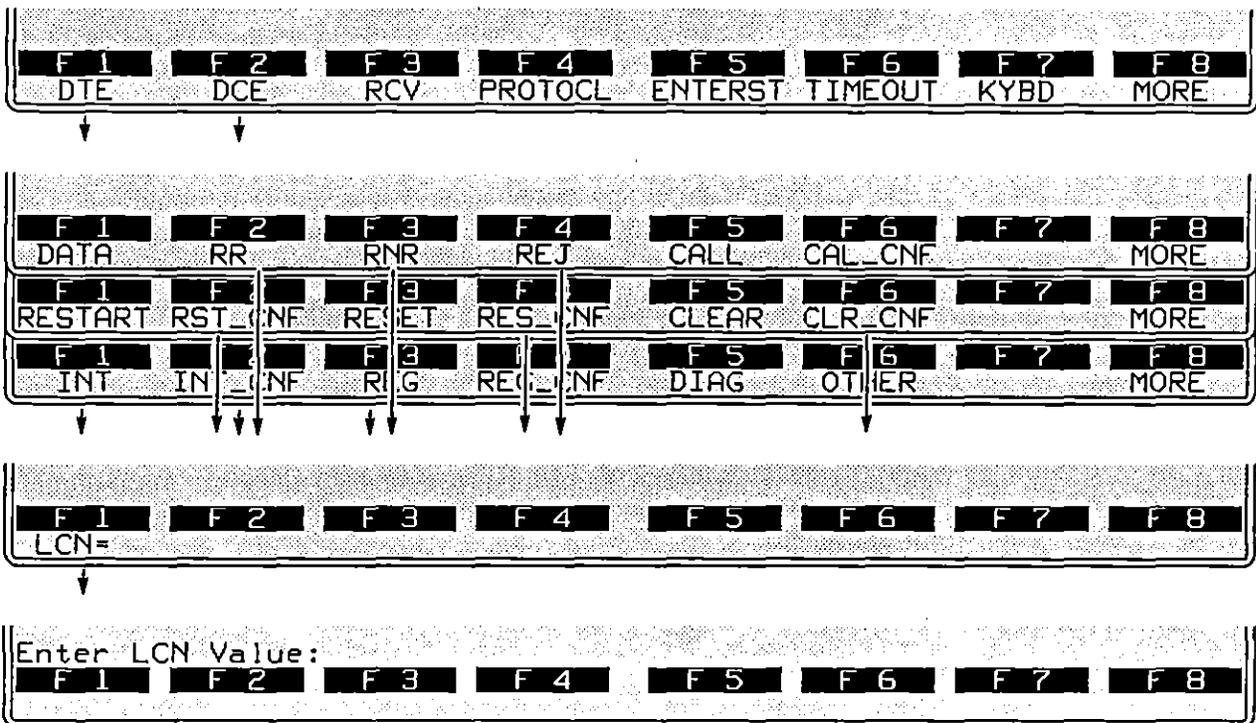


Figure 37-11 LCN is the Condition option available for these nine packet types.

Figure 37-11 indicates the packet types that offer LCN as their only Condition option.

Enter the LCN as one, two, or three hexadecimal digits. Type each digit as an alphanumeric in the range 0-9 and A-F (or a-f): do not use the **HEX** key. Each digit will represent four bits of the twelve-bit LCN. A single-digit or two-digit entry will represent the low-order bits, with the high-order bits zero-filled. Thus LCN= 005 is the same entry as LCN= 05 or LCN= 5.

(C) Q, D, and M Bits

Q-, D-, and M-bit values of 0 or 1 may be specified in Layer 3 conditions that search for DATA packets. See Figure 37-12.

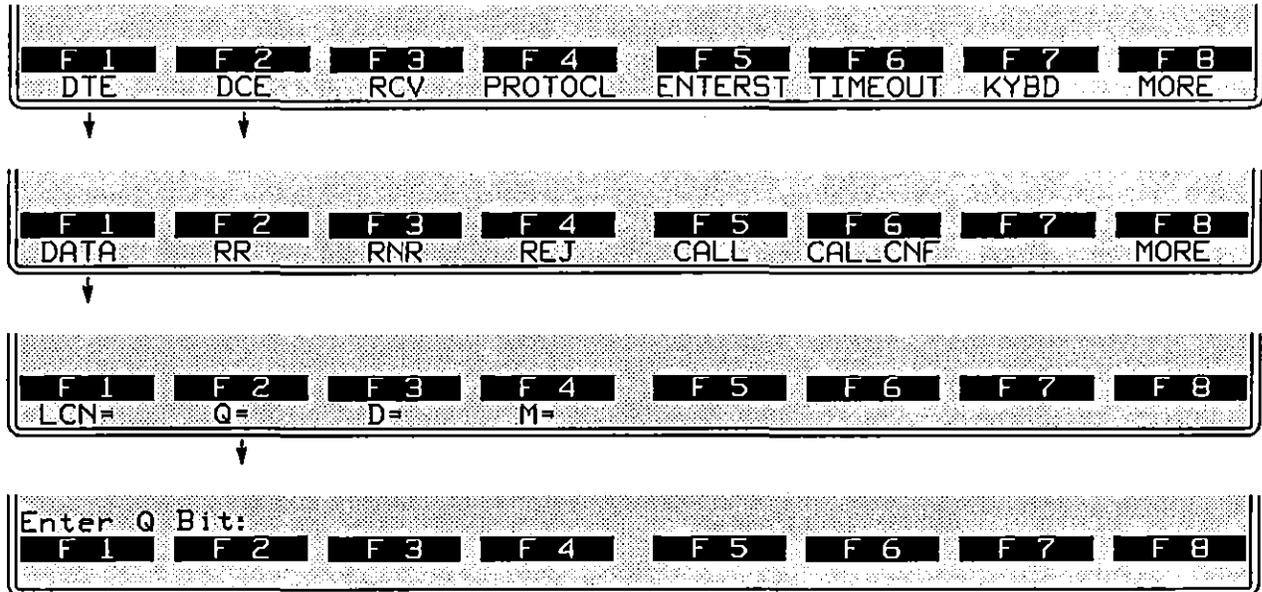


Figure 37-12 When data packets are monitored, Q, D, and M values of 1 (along with LCN values) may be specified.

A D-bit value also may be specified for Call and Call Confirm packets: see Figure 37-13.

The positions of the Q, D, and M bits in the packet header are illustrated in Figure 37-6.

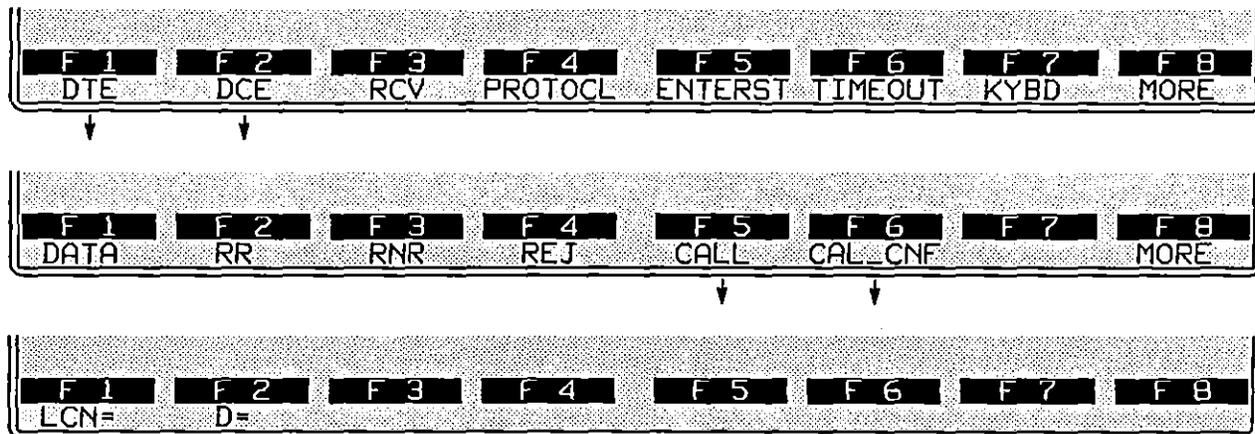


Figure 37-13 Conditions that look for Call and Call Confirm packets also can test for LCN and D-bit values.

(D) Cause and Diagnostic Value

Conditions that look for Restart, Reset, Clear, and Registration Confirm packets may be refined further to test for a particular cause code and/or diagnostic code.

1. **Cause.** The names of causes as well as their hexadecimal values are indicated on the softkey-prompt line near the bottom of the Protocol Spreadsheet screen. To specify a particular cause, the user does not have to memorize cause codes or consult a table. He simply presses [F2], ROLL, and repeats the keystroke to cycle through the list of cause names for a given packet type. Figure 37-14 shows the cycle of causes that pertain to Restart packets. The user presses [F1], SELECT, when the right cause has "rolled" up onto the prompt line. The SELECT softkey writes the current cause onto the spreadsheet screen.

Here is an example of a cause-code entry on the Protocol Spreadsheet:

CONDITIONS: DCE CLEAR CAUSE= NOT_OBTAINABLE

Causes also may be entered into the spreadsheet test as two hexadecimal digits, as in this example:

CONDITIONS: DCE CLEAR CAUSE= 0D

Notice that each digit is an alphanumeric in the range 0-9 and A-F (or a-f): do not use the [HEX] key.

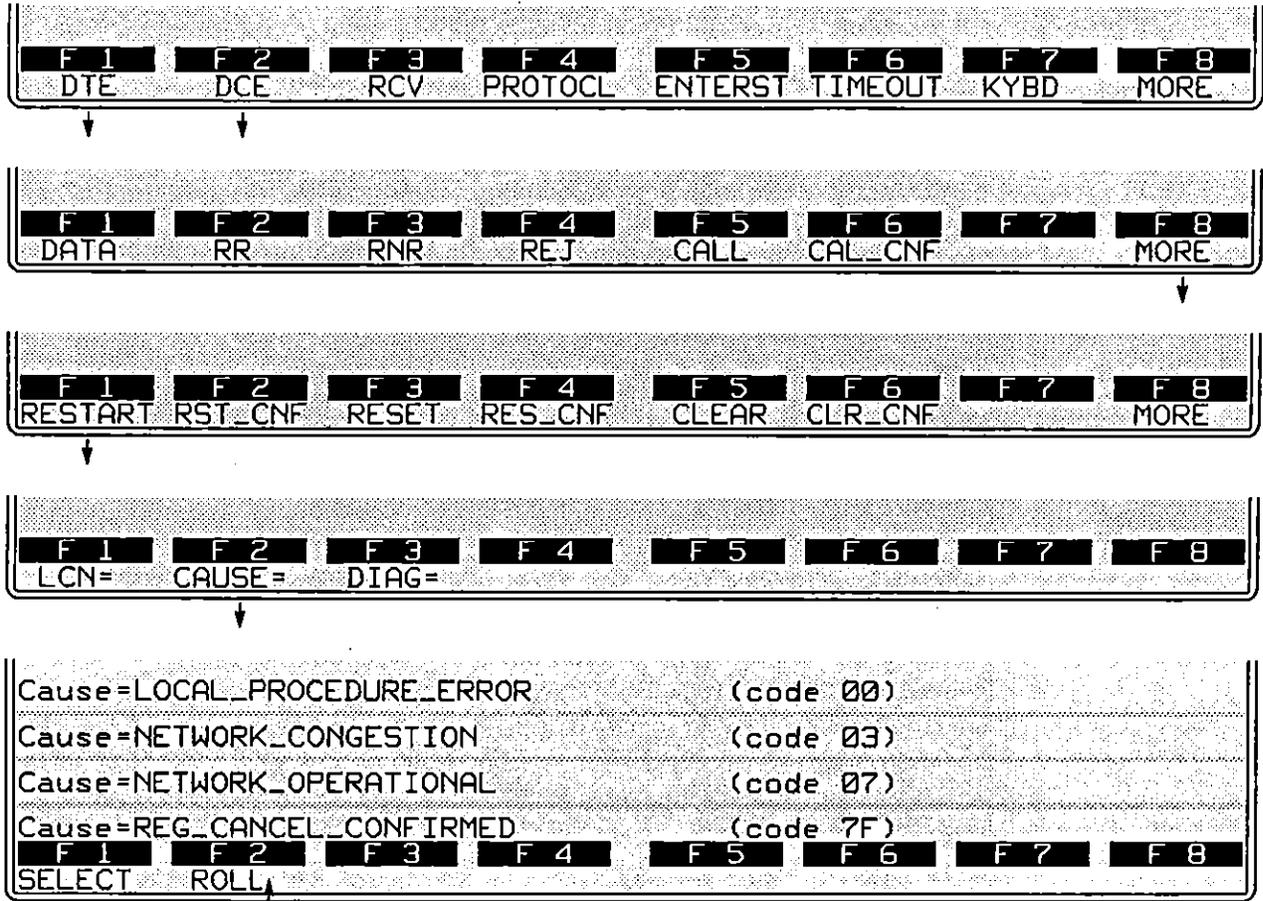


Figure 37-14 A Restart packet may be tested for one of these four causes.

(a) Reset causes

Cause=DTE_ORIGINATED	(code 00)
Cause=OUT_OF_ORDER	(code 01)
Cause=REMOTE_PROCEDURE_ERROR	(code 03)
Cause=LOCAL_PROCEDURE_ERROR	(code 05)
Cause=NETWORK_CONGESTION	(code 07)
Cause=REMOTE_DTE_OPERATIONAL	(code 09)
Cause=NETWORK_OPERATIONAL	(code 0F)
Cause=INCOMPATIBLE_DESTINATION	(code 11)
Cause=NETWORK_OUT_OF_ORDER	(code 1D)
F1 F2 F3 F4 F5 F6 F7 F8	
SELECT ROLL	

(b) Clear causes

Cause=DTE_ORIGINATED	(code 00)
Cause=NUMBER_BUSY	(code 01)
Cause=OUT_OF_ORDER	(code 09)
Cause=REMOTE_PROCEDURE_ERROR	(code 11)
Cause=REVERSE_CHARGE_NOT_ACCEPTED	(code 19)
Cause=INCOMPATIBLE_DESTINATION	(code 21)
Cause=FAST_SELECT_NOT_ACCEPTED	(code 29)
Cause=SHIP_ABSENT	(code 39)
Cause=INVALID_FACILITY_REQUEST	(code 03)
Cause=ACCESS_BARRED	(code 0B)
Cause=LOCAL_PROCEDURE_ERROR	(code 13)
Cause=NETWORK_CONGESTION	(code 05)
Cause=NOT_OBTAINABLE	(code 0D)
Cause=RPOA_OUT_OF_ORDER	(code 15)
F1 F2 F3 F4 F5 F6 F7 F8	
SELECT ROLL	

(c) Registration Confirmation causes

Cause=REG_CANCEL_CONFIRMED	(code 7F)
Cause=INVALID_FACILITY_REQUEST	(code 03)
Cause=LOCAL_PROCEDURE_ERROR	(code 13)
Cause=NETWORK_OPERATIONAL	(code 05)
F1 F2 F3 F4 F5 F6 F7 F8	
SELECT ROLL	

Figure 37-15 The various causes available for (a) Reset, (b) Clear, and (c) Registration Confirm packets.

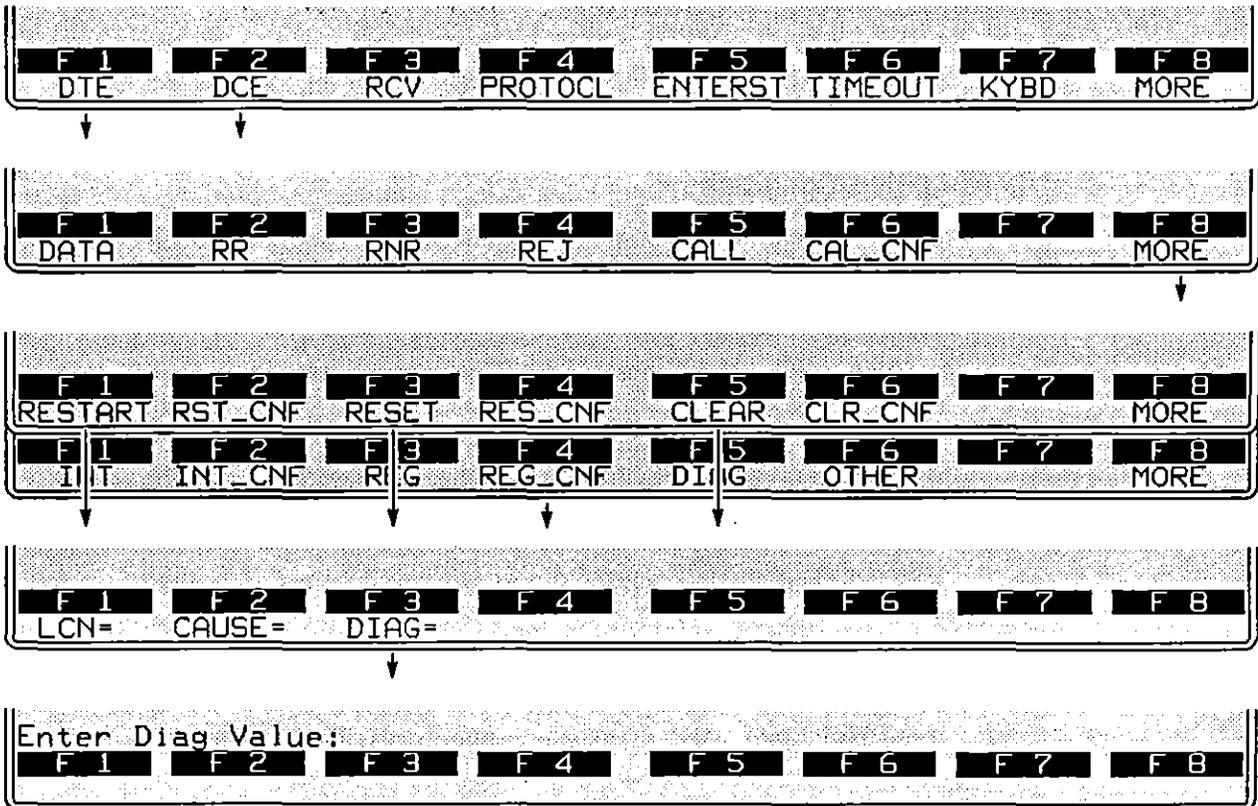


Figure 37-16 These four packet types (plus *Diagnostic*) allow you to enter a diagnostic-code value as a spreadsheet condition.

2. *Diagnostic code.* Diagnostic-code values are optional conditions for the following packet types: Restart, Reset, Clear, Diagnostic, and Registration Confirm. Figure 37-16 shows the softkey sequences that branch down to the diagnostic-code condition for most of these packet types.

Enter the diagnostic code as two hexadecimal digits. Type each digit as an alphanumeric in the range 0-9 and A-F (or a-f): do not use the **RED** key. Here is an example of a spreadsheet condition that specifies both a cause and a diagnostic code:

```
DCE RESET CAUSE= LOCAL_PROCEDURE_ERROR DIAGNOSTIC= 01
```

37.4 Emulate-Mode Conditions

The remaining conditions are functional only when the Line Setup menu is configured for Mode: `EMULATE DTE` or `EMULATE DCE`.

(A) Receive Conditions

Like DTE and DCE conditions, RCV conditions monitor a data path for X.25 packet types. RCV conditions operate only in emulate modes, and they check only the data path that the INTERVIEW is not using to transmit. While a RCV condition may look like a DTE or DCE condition—RCV DATA Q looks the same as DCE DATA Q—there are important differences that are noted below.

1. *Access to the data interface.* The INTERVIEW is a *layered* emulator. The significance of this is that Layer 3 and higher layers have no direct access (in Emulate modes) to the physical layer, Layer 1. In practice this means that a RCV condition at Layer 3 *does not see packets on the line*. It only sees packets that are delivered up from Layer 2 by a *user program* at that layer.

(Similarly, a SEND action at Layer 3 does not in itself send a packet out onto the line. A SEND action merely delivers the packet to Layer 2—provided that Layer 2 has indicated its readiness to receive data from above.)

The following program is not any sort of complete Layer 2 emulation. It is the *minimum* program that must be entered at spreadsheet Layer 2 (with the X.25 personality package installed) in order for a Layer 3 program to have access to the data line. Once this Layer 2 program is entered, Layer 3 can send packets out onto the line and receive packets from the line.

```
LAYER: 2
STATE: datalink
CONDITIONS: DL_CONNECT REQ
ACTIONS: DL_CONNECT CONF
CONDITIONS: DL_DATA REQ
ACTIONS: SEND INFO "(DL_DATA)"
CONDITIONS: RCV INFO
ACTIONS: GIVE_DATA
```

The elements of this program are discussed in Section 33 (OSI Primitives on the Protocol Spreadsheet) and the programming example in Section 36.9.

2. *Valid packet sequencing.* To satisfy RCV conditions, numbered packets must have correct P(R) and P(S) sequencing.
3. *Path.* All RCV conditions that name a packet type may specify one particular "path" as an added condition. Like LCN in DTE and DCE conditions, this path number serves to associate a packet with a particular

call. On the X.25 Packet Level Setup screen, up to nine path numbers may be tied to individual sets of Call Request parameter values, including packet-network "phone" numbers. Refer back to Figure 37-3 for the Packet Level Setup screen.

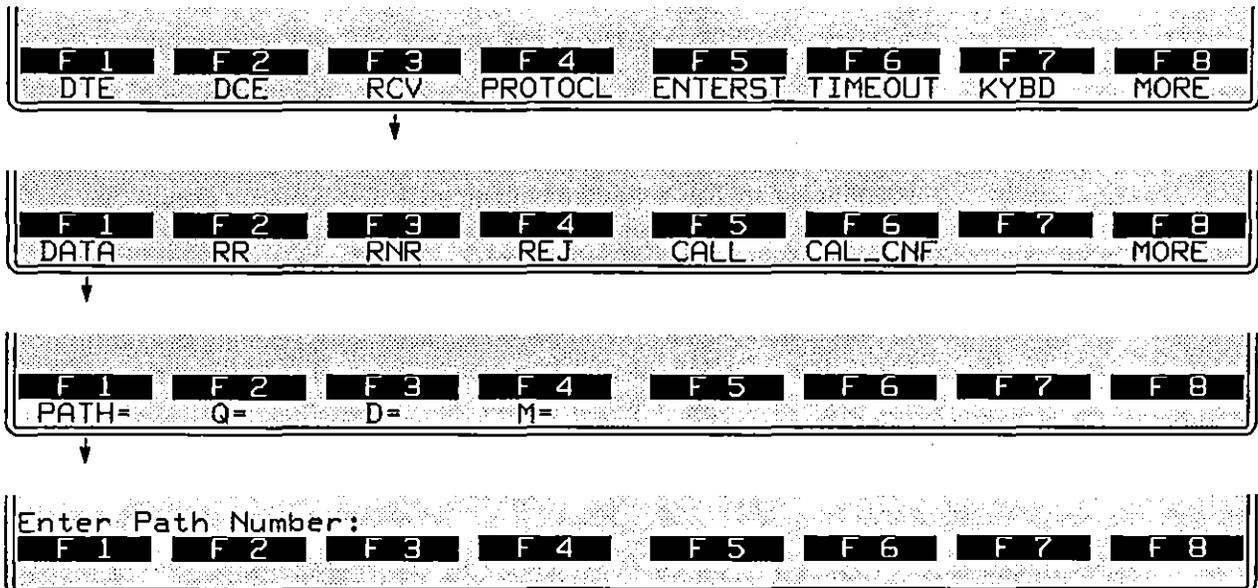


Figure 37-17 *PATH=* replaces *LCN=* as an added condition when the lead is identified as *RCV* rather than *DTE* or *DCE*.

As a packet identifier, the *PATH=* condition in *RCV* conditions is more programmable than the *LCN=* conditions that are available inside of *DTE* and *DCE* conditions. *LCNs* usually are assigned dynamically, by the *INTERVIEW* as well as by other devices, at the time of the call request. By then the test has started running, and it is too late to specify the *LCN* in the spreadsheet program.

The path number, by contrast, may be pre-programmed on the Protocol Spreadsheet. When the call request is sent or received by the *INTERVIEW*, the call parameters are correlated to the Packet Level Setup screen. If the *INTERVIEW* sends a call request that specifies a path number, or if the *INTERVIEW* receives a call request that matches one of the path entries on the setup screen, the *LCN* of the call request is tied to the path number (path #3, say), and any subsequent packets with the same *LCN* will satisfy packet-type conditions that stipulate *PATH= 3*.

A *RCV* condition that specified a path number as a further condition might be the following:

CONDITIONS: RCV DATA PATH= 3

A data packet would satisfy this condition if (1) it had the same LCN as a call request packet with the Calling, Called, Facilities, and Data fields that are entered across from Path 3 on the setup screen; and (2) the call still was active. The call-request parameters on the setup screen may refer to calls that originate at the INTERVIEW or to call requests that are incoming.

The PATH= condition is shown in the racks of softkeys in Figure 37-17.

4. *Type invalid or unknown.* RCV conditions can detect packets that are invalid "types"—the packet header is missing, for example, or the LCN is 000 for anything other than a Restart, Registration, or Diagnostic packet. The Protocol Spreadsheet entry for this condition is RCV INVALID, and the softkey sequence is illustrated in Figure 37-18.

A packet may be valid in all respects but have a packet-type field that indicates a nonstandard packet type. Such a packet may be matched by a RCV UNKNOWN condition (Figure 37-18).

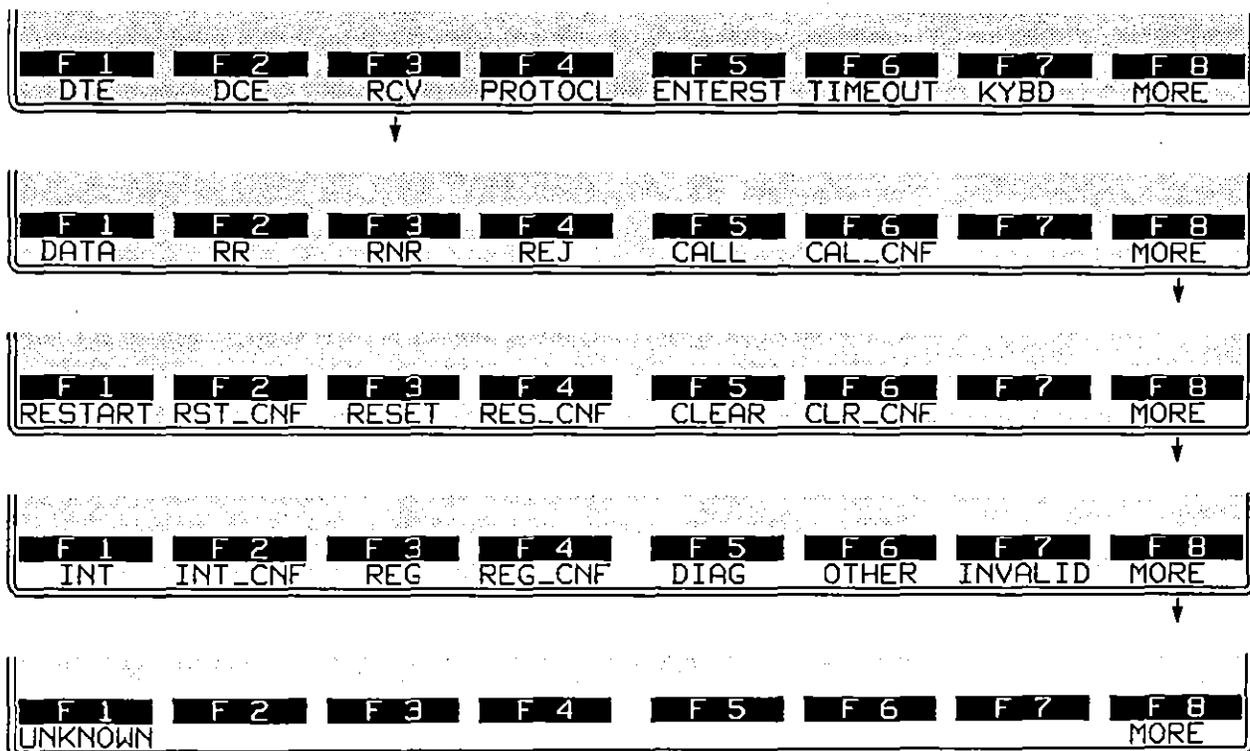


Figure 37-18 *INVALID* and *UNKNOWN* appear on the bottom two racks of RCV packet-type softkeys.

(B) P(S) Error

When you emulate at Layer 3, data packets with P(S) errors are detected as PS_ERR conditions, not as RCV DATA conditions.

PS_ERR applies only to packets received when you are emulating. The same packet that triggers a PS_ERR condition also may satisfy a DTE DATA or DCE DATA condition—but not a RCV DATA condition.

PS_ERR will come true for any received data packet whose P(S) value is not one higher than the previous P(S).

In the first rack of condition softkeys at Layer 3, press PROTOCL. Then press the softkey for PS_ERR. See Figure 37-19.

(C) P(R) Error

Received data or supervisory packets may have P(R) errors. Such errors are detected as PR_ERR conditions, not as RCV DATA or RR (or RNR or REJ) conditions.

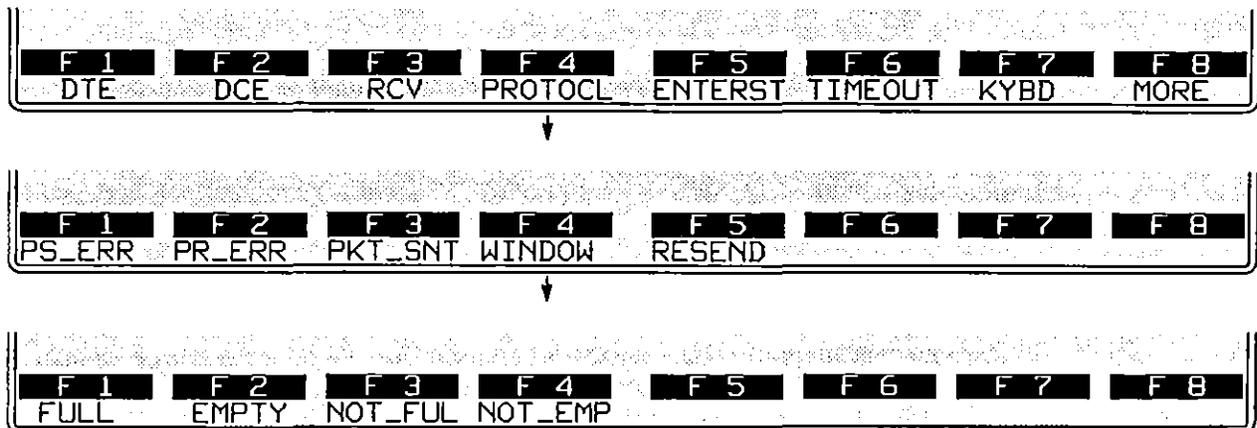


Figure 37-19 The PROTOCL key brings up five X.25 Layer 3 emulate conditions.

A valid P(R) is any value that (1) acknowledges a packet that is *outstanding* (waiting for acknowledgment); or (2) repeats the last acknowledgment. Any other P(R) value is detected as an error.

(D) Packet Sent

This condition is true when, as a result of a SEND or RESEND action, a packet has been passed down to Layer 2. This condition may be useful for certain timing measurements, since merely SENDING a packet *does not actually pass the packet down to the next layer* if, for example, the link is not established at Layer 2.

(E) Window Conditions

The size of the Layer 3 window is configured on the X.25 Packet Level Setup screen; see Section 37.1(C). There are four conditions that test the current status of this window. They are WINDOW FULL, WINDOW EMPTY, WINDOW NOT_FULL, and WINDOW NOT_EMPTY. The softkey sequence for the WINDOW options is shown in Figure 37-19.

WINDOW FULL is true when the window is full of unacknowledged packets and the Layer 3 personality package will not buffer additional packets until some acknowledgment is received.

Each time an acknowledgment is received, the window is flushed to the extent of the acknowledgment. WINDOW EMPTY means that the latest acknowledgment was complete and left no packets outstanding (unacknowledged). If an RR response is received and the acknowledgment is only partial, this condition will be true:

```
CONDITIONS: RCV RR
            WINDOW NOT_EMPTY
```

CAUTION: Window conditions are status conditions (see Section 30.2) and must always be used in combination with a transitional condition such as a RCV condition.

(F) More to Resend

Packets in the window may have to be resent, usually as the result of a timeout or a Reject packet. One RESEND action retransmits one packet in the window, beginning with the earliest. Subsequent RESEND actions retransmit subsequent packets. The MORE_TO_RESEND and NO_MORE_TO_RESEND conditions allow you to retransmit the entire window, as in the "recover" state in this example:

```
CONDITIONS: RCV REJ
NEXT_ST: recover
STATE: recover
CONDITIONS: ENTER_STATE
ACTIONS: RESEND FIRST
CONDITIONS: PACKET_SENT
        MORE_TO_RESEND
ACTIONS: RESEND NEXT
CONDITIONS: PACKET_SENT
        NO_MORE_TO_RESEND
NEXT_ST: xfer
```

MORE_TO_RESEND and NO_MORE_TO_RESEND conditions may be written to the Protocol Spreadsheet by the softkeys shown in Figure 37-20.

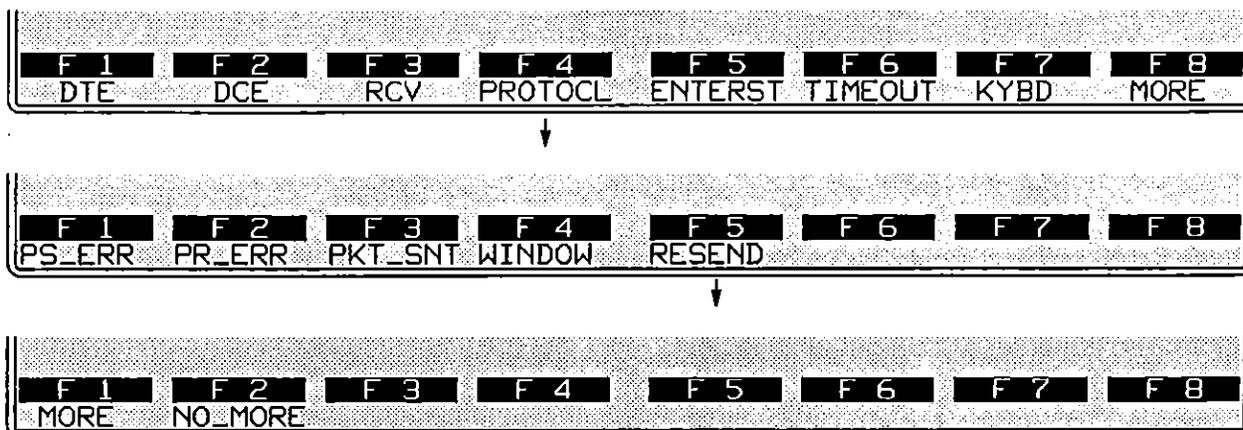


Figure 37-20 The *MORE_TO_RESEND* condition allows you to resend the entire window of packets and then stop when there are *NO_MORE_TO_RESEND*.

CAUTION: MORE_TO_RESEND and NO_MORE_TO_RESEND are status conditions (see Section 30.2) and must always be used in combination with a transitional condition such as PACKET_SENT.

37.5 Emulate Actions

When you have completed a block of conditions in a Protocol Spreadsheet test at Layer 3, press **[Ctrl]**, then **[F4]** (ACTION:), to access the set of actions that can be taken as a result of the block of conditions coming true. The set of actions that are specific to the X.25 Layer 3 personality package are shown in the four lower racks of softkeys in Figure 37-21. Except for ENHANCE and SUPPRES, the actions shown have meaning only when the INTERVIEW is emulating DTE or DCE, and not when it is monitoring the line passively.

(A) Send Actions

Press the softkey for SEND to access three racks of softkeys with names of packet types that may be named in SEND actions. All data generated by the Layer 3 X.25 package must be enclosed in a packet that is identified in a SEND action by *type*. (Only at Layer 1 can data be generated as a simple character string without any protocol building blocks.) The complete set of packet types is given in Figure 37-22.

When conditions are true for a SEND action, packets are sent immediately down to Layer 2 to be processed there.

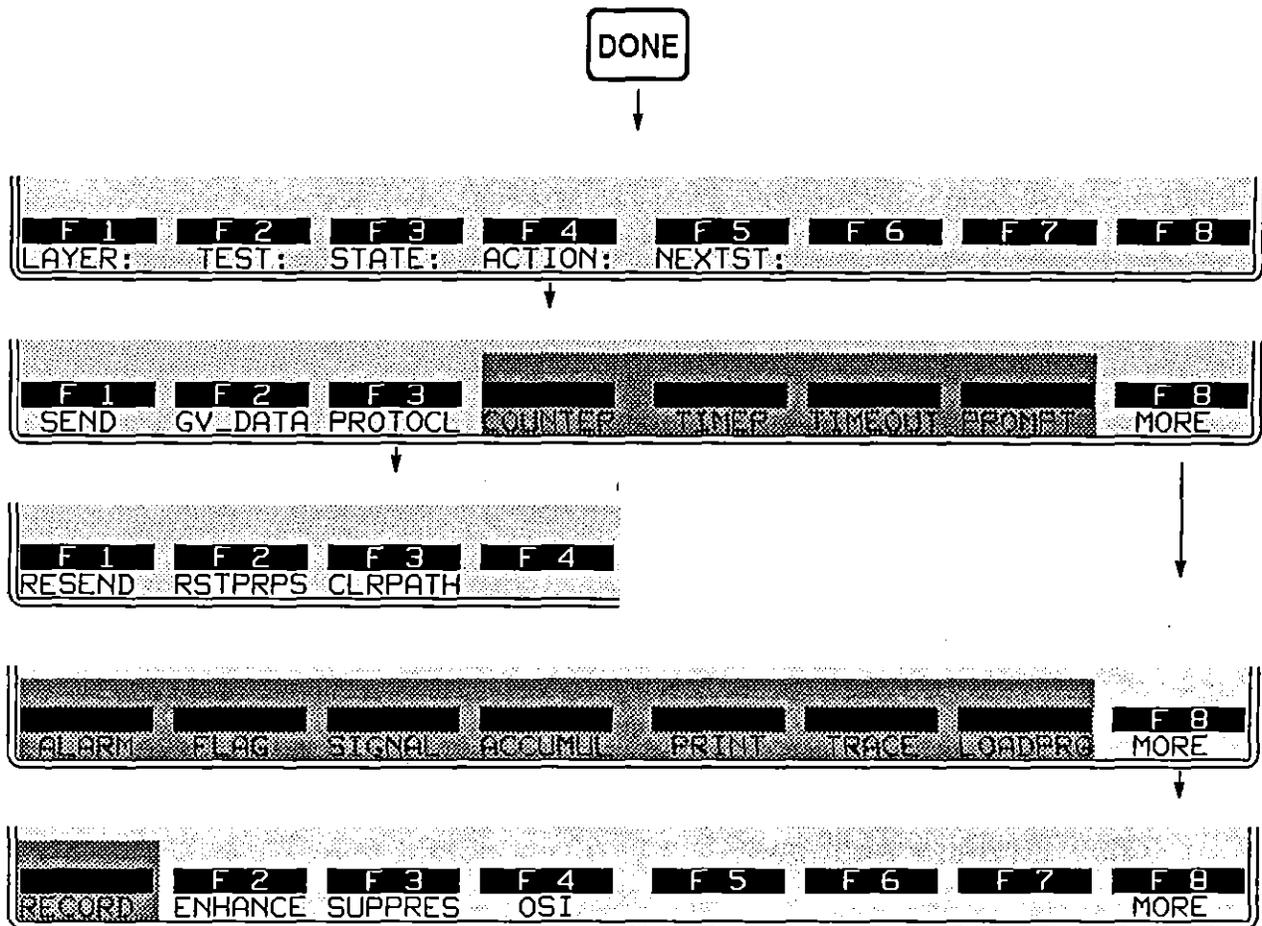


Figure 37-21 Action softkeys specific to X.25 Layer 3.

NOTE: The INTERVIEW is a *layered* emulator. The significance of this is that Layer 3 and higher layers have no direct access (in Emulate modes) to the physical layer, Layer 1. In practice this means that a SEND action at Layer 3 does not in itself send a packet out onto the line. A SEND action merely delivers the packet to Layer 2—provided that Layer 2 has indicated its readiness to receive data from above.

Refer to the Layer 2 program in Section 37.4(A)1. This is the *minimum* program that must be entered at spreadsheet Layer 2 (with the X.25 personality package installed) in order for a Layer 3 program to have access to the data line. Once this Layer 2 program is entered, Layer 3 can send packets directly out onto the line (and receive packets from the line).



Figure 37-22 *SEND* actions always specify a packet type.

1. *Data packets.* *SEND DATA* is a complete action-entry. Path, P(S), P(R), Q, D, M, and string parameters may be added to a data packet, but they are optional.

SEND DATA actions pass the data packet immediately to the next layer down. If the retransmit window is full, the packet is still sent—but it is not buffered in the window and can not be resent.

A data packet will be buffered for retransmission regardless of the status of the window if a specific value is entered for the PS= parameter. See "P(S)," below. The specific P(S) value will clear the window so that the data packet will be buffered in the first window position.

2. *Supervisory packets.* *SEND RR*, *SEND RNR*, and *SEND REJ* are complete action-entries. Path and P(R) parameters may be added to a supervisory packet, but they are optional.

Figure 37-23 shows the parameter options for supervisory *SEND* packets.

3. *Call Request packets.* *SEND CALL* and *SEND CALL_CONF* are complete action-entries. Normally a Call Request packet will be entered with additional parameters. Parameters that are available are PATH=, D, FAC=, and STRING.

When a *SEND CALL* action does not specify a path, it yields a packet with the LCN that is next in the assignment series: see Section 37.1(D). This is true

also when a Call Request specifies a path but the LCN column is blank for that path on the X.25 Packet Level Setup screen.

When a PATH= value is included in a SEND CALL or SEND CALL_CONF packet, the packet will be sent with the LCN, Called Address, Calling Address, Facilities, and Data entries that the operator has provided for that path number on the Packet Level Setup screen. A SEND CALL action that is not linked by a PATH= number to a set of call parameters on the Packet Level Setup screen cannot yield a valid call request no matter what string is added to it, since the address-length field (Figure 37-6) in such a packet will be fixed automatically at %.

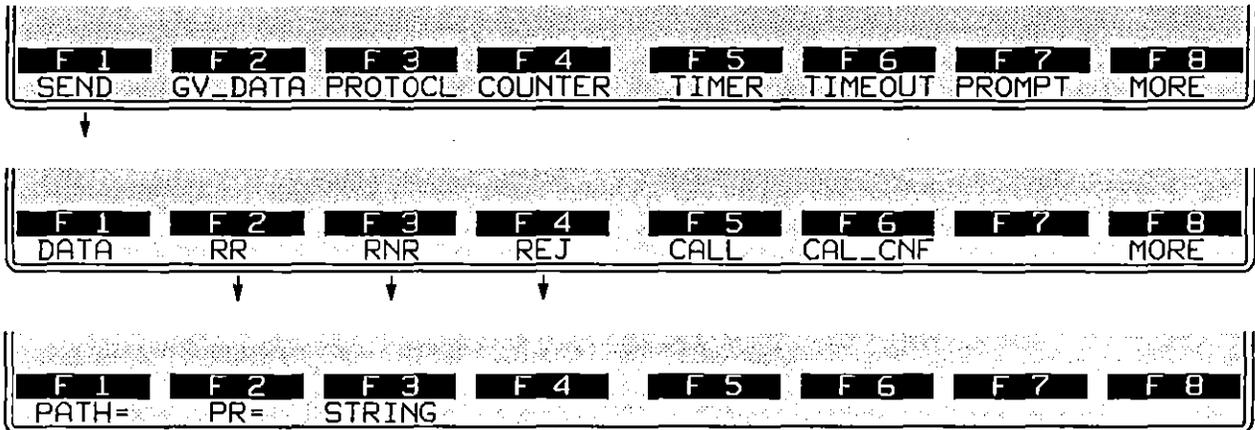


Figure 37-23 Path and P(R) components may be selected for SEND RR, SEND RNR, and SEND REJ actions.

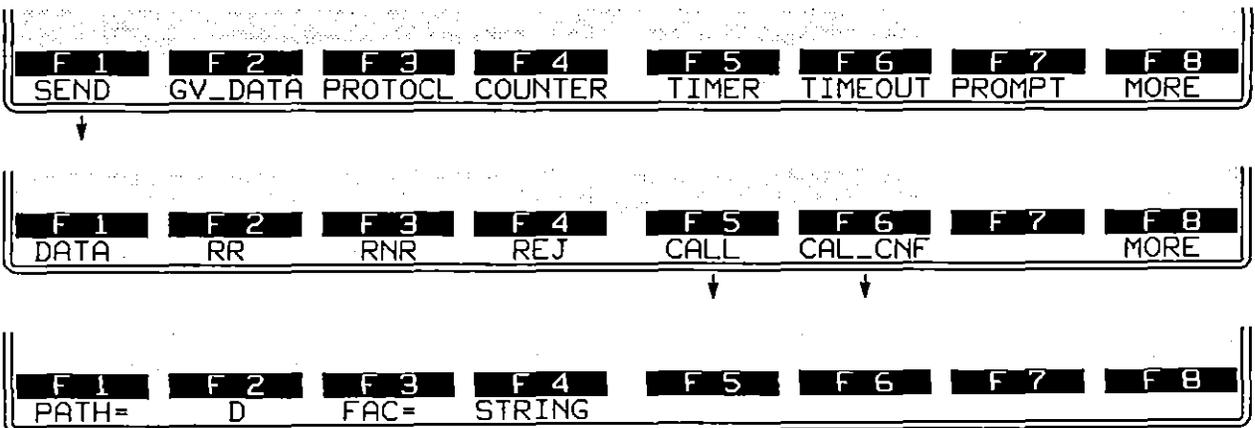


Figure 37-24 SEND CALL actions should include a path number and may set the D bit; they also may append facilities and a data string to the parameters already listed on the Packet Level Setup menu.

The FAC= option provided on the Protocol Spreadsheet is intended to supplement the FACILITIES field on the Packet Level Setup screen. A FAC= string in a spreadsheet action will be *appended* to the Facilities string on the setup screen. This is in case the facilities entry must be longer than the ten bytes permitted on the setup screen. Do not *repeat* your setup-screen Facilities entry on the Protocol Spreadsheet.

Similarly, any STRING entry on the spreadsheet will be appended to the string in the DATA field on the Packet Level Setup screen.

4. *Other unnumbered packets.* The rest of the unnumbered-packet types have softkey options appropriate to their protocol fields (see Figure 37-6). Available softkey parameters for these packet types are PATH=, CAUSE=, DIAG=, and STRING.

Figure 37-25 shows the softkey rack under Registration Confirm, an unnumbered-packet type with the four possible softkey parameters.

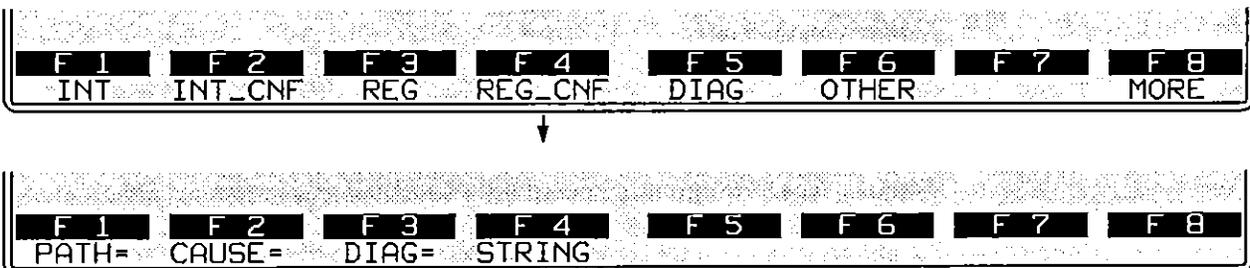


Figure 37-25 Four optional parameters may be added to a SEND REG_CONF action.

5. *"Other" packets.* Any packet type may be entered in a SEND action as a hexadecimal value instead of by name. Press the softkey for OTHER, on the bottom rack in Figure 37-22. Enter the hex value in the form of two alphanumerics. Here is a Call Confirm packet entered as a SEND OTHER action:

ACTIONS: SEND OTHER OF

This SEND OTHER OF action is a good way to send a "stripped down" Call Accepted packet that does not include the additional address and facilities parameters that you may have entered on the Packet Level Setup screen. These parameters sometimes are not used in Call Accepted packets in specific network implementations of X.25.

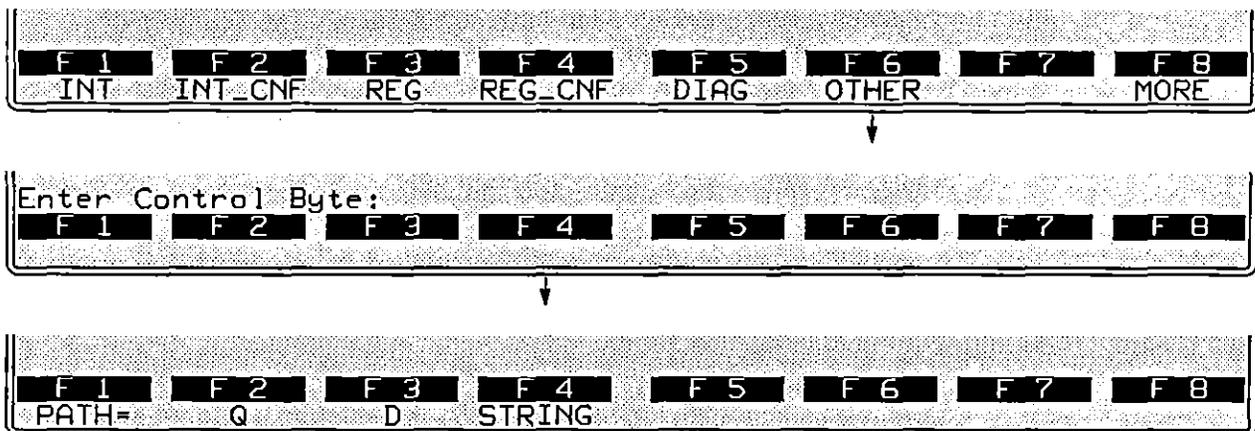


Figure 37-26 Use the *SEND OTHER* action to customize a packet type.

Additional parameters for *SEND OTHER* actions are *PATH=*, *Q*, *D*, and *STRING*. (See Figure 37-26.) Since *M*, *P(R)*, and *P(S)* fields are embraced already in the user-entered hexadecimal control field, these fields are not given as softkey parameters.

6. *Path*. The addition of a *PATH=* entry in a *SEND* action will insure that the packet receives the same LCN as the Call Request with the same *PATH=* value. The LCN itself is not used for identification in *SEND* actions, since LCNs usually are assigned dynamically at the time of the call—too late to be programmed on the Protocol Spreadsheet.

Each path number is tied to a particular set of Call Request parameter values on the X.25 Packet Level Setup screen. See Figure 37-3.

All packet types permit *SEND* actions that have *PATH=* options except Restart, Diagnostic, and Registration. These packets do not refer to a specific call or path. They always receive LCN 000.

As a general rule, path numbers are used at a given layer in the INTERVIEW if (1) the protocol at that layer is multiaddress or multichannel; or (2) the protocol at a layer *below* the given layer is multiaddress or multichannel. Use the same path number at each layer for a given call.

7. *P(S)*. *P(S)* fields are transmitted in data packets only. (See the packet-field diagrams in Figure 37-6.) The softkeys that open below *PS=* are illustrated in Figure 37-27.

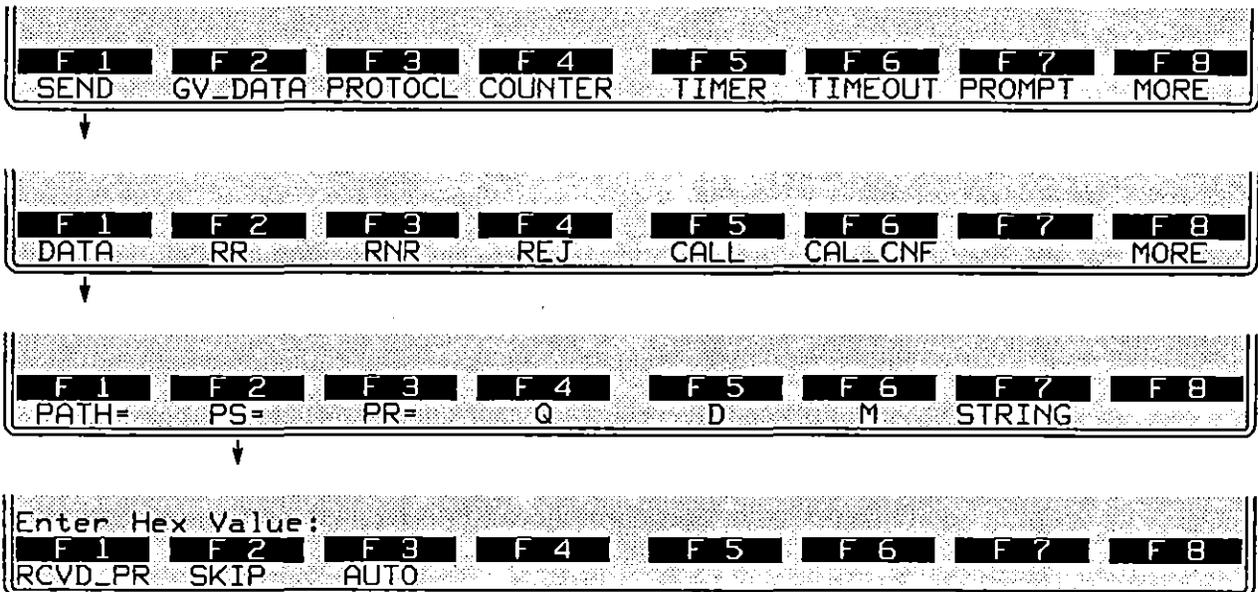


Figure 37-27 The P(S) field may be specified in a *SEND DATA* action.

To specify a P(S) value, press the softkey for PS=, then enter a hexadecimal in the form of one or two alphanumerics. An entry that represented the highest valid P(S) in MOD 8 would be PS= 7. The highest valid entry in MOD 128 is PS= 7F. A SEND DATA action that *specifies* a P(S) value—PS= 0, for example—will clear the window so that the data packet is passed immediately to Layer 2.

Other P(S) options are RCVD_PR, SKIP, and AUTO. RCVD_PR means that you send the P(S) value that the other side says it is expecting. This is the valid P(S) in most cases, but not when you send two or more data packets in a row without waiting for acknowledgment.

SKIP means that you *add one* to your correct P(S). This will look to the other side as though a packet has been lost in transmission. This selection causes the window to be cleared.

PS_AUTO is the default setting for SEND DATA actions. AUTO means that every *new* data packet sent will have a P(S) value of one higher than the previous.

8. *P(R)*. To specify a P(R) value, press the softkey for PR= (see Figure 37-28). Enter a hexadecimal value written as one or two alphanumeric digits. Valid hex entries are the same as for P(S).

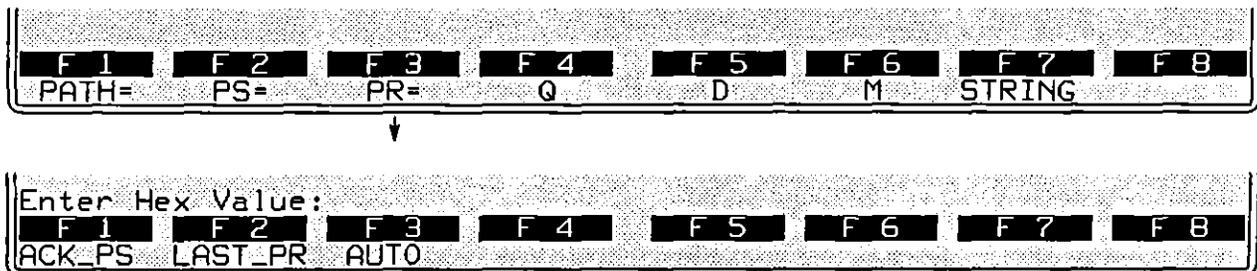


Figure 37-28 The P(R) field may be specified in data and supervisory packets to be sent.

Other P(R) options are ACK_PS, LAST_PR, and AUTO. (See Figure 37-28.) ACK_PS means that your P(R) will acknowledge (that is, it will be *one higher than*) the last P(S) value you received. Normally this will be the correct P(R), except in cases where the last P(S) received was erroneous. The PR=ACK_PS selection allows you to overlook P(S) errors.

LAST_PR means that you simply repeat the last P(R) you sent. Normally this is the correct P(R) following a bad P(S). The PR=LAST_PR option allows you to force the other side to initiate recovery.

AUTO means that you will behave as a normal X.25 Layer 3 PAD, acking valid P(S) values and repeating your last P(R) whenever an invalid P(S) is received.

9. *Q, D, and M.* Softkeys for Q, D, and M are included in the full set of optional parameters for a SEND DATA action in the top rack of Figure 37-28.

Q and D bits also may be set in SEND OTHER actions. The D bit alone is selectable in Call Request and Call Accepted packet types.

10. *Cause.* Actions that send Restart, Reset, Clear, and Registration Confirmation packets may be refined to send a particular cause code and/or diagnostic code.

Press the softkey for one of these SEND actions, then press **[F2]**, CAUSE. See Figure 37-29. The names of the causes as well as their hexadecimal values are indicated on the softkey-prompt line near the bottom of the screen.

To specify a particular cause, the user does not have to memorize cause codes or consult a table. Instead he presses **[F2]**, ROLL, and repeats the keystroke to cycle through the list of cause names for a given packet type. Figure 37-29 shows the cycle of causes that pertain to Clear Request packets. The user presses **[F1]**, SELECT, when the right cause has "rolled" onto the prompt line. The SELECT softkey writes the current cause onto the spreadsheet screen.

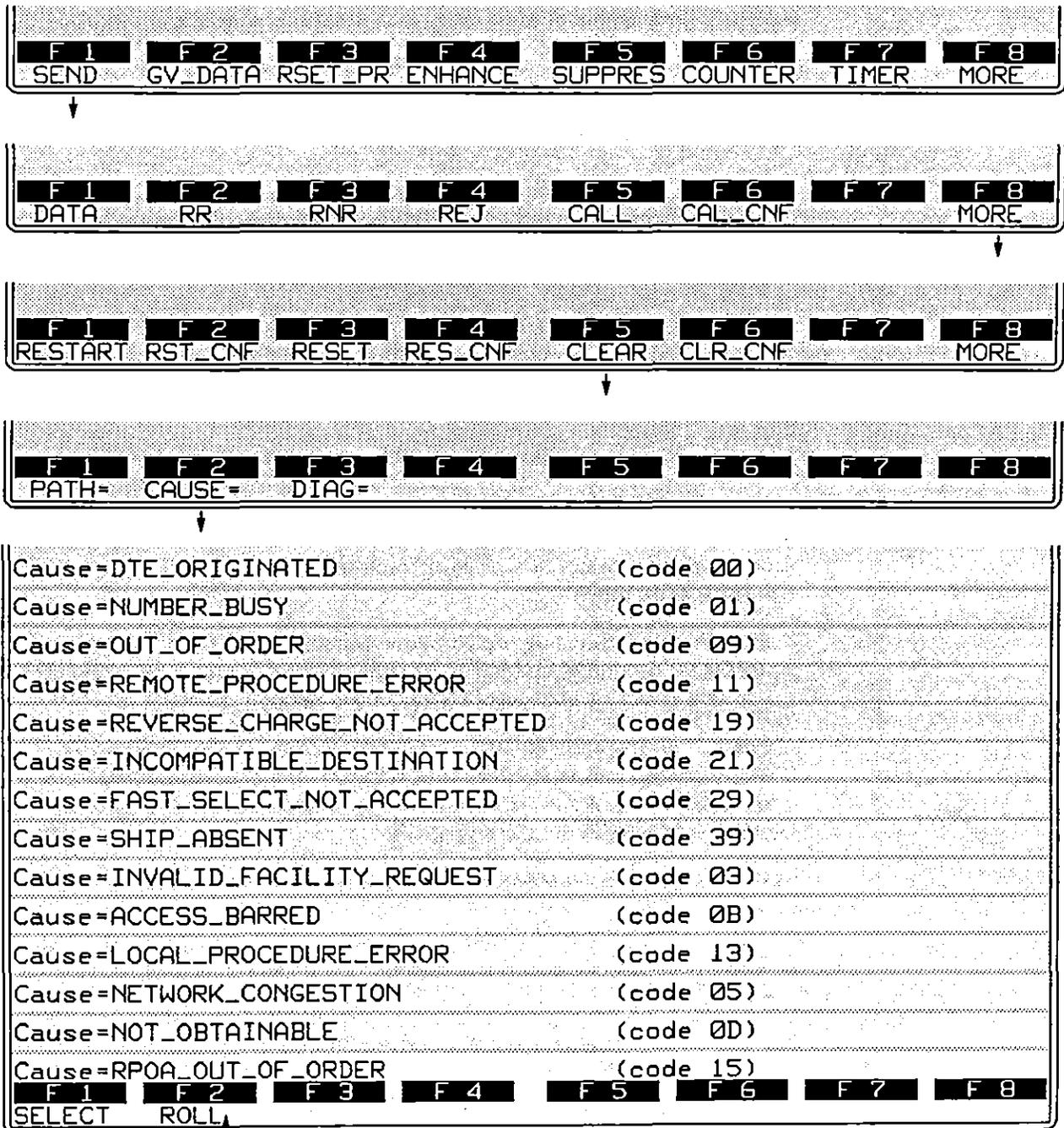


Figure 37-29 These causes are available for a *SEND CLEAR* action.

Causes for Restart, Reset, and Registration Confirmation packets were listed in Figure 37-14 and Figure 37-15.

Here is an example of a cause entry in a SEND action on the Protocol Spreadsheet.

ACTIONS: SEND RESTART CAUSE= NETWORK_OPERATIONAL

Causes may be entered into the spreadsheet test as numeric values.

11. *Diagnostic.* Two digits representing the one-byte diagnostic-code field (Figure 37-6) may be added to a SEND action for Restart, Reset, Clear, Diagnostic, and Registration Confirmation packets. Refer to [F3], DIAG=, in Figure 37-29 or in the lower rack of Figure 37-25.

Enter the diagnostic code as two hexadecimal digits. Type each digit as an alphanumeric in the range 0-9 and A-F (or a-f); do not use the [HEX] key. Here is an example of a SEND action that specifies both a cause and a diagnostic code:

ACTIONS: SEND RESET CAUSE= LOCAL_PROCEDURE_ERROR DIAG= 01

12. *String.* Strings are sent at X.25 Layer 3 only as adjuncts to packet-types when they are named in SEND actions. If you want to send a string of raw data without a protocol "envelope," you must go to Layer 1 and send the raw string from there.

Press the SEND softkey followed by the softkey for a packet type. Add any necessary or desired SEND options for the particular packet type. Then press the STRING softkey (see Figure 37-28, for example).

There is no spreadsheet keyword that identifies send-strings at any layer. *The spreadsheet compiler identifies strings by the quotation marks surrounding them.* Always enclose strings in quotation marks. (To send an actual " character in your string, type \".)

Here is a simple SEND action that includes no options besides a string:

ACTIONS: SEND INT "%"

And here is a SEND action that includes a full complement of optional fields, including a string:

ACTIONS: SEND DATA PATH= 4 PS= AUTO PR= AUTO "57 This is user data."

Most ASCII-keyboard, control, and hexadecimal characters are legal in a send-string. Special keys ([F4], [PORT], [SET MASK]) are not legal. Refer to Table 32-2.

To insert a canned fox message into a transmit string, type FOX inside of double parens, as follows: ((FOX)). Remember that the double parens are special characters produced by the [CTRL]-[9] and [CTRL]-[0] combinations. Constants, counters, and flags can also be embedded in a string. See Section 32, Strings.

(B) Give Data

GIVE_DATA is the **F2** action on the first rack of action softkeys (refer to Figure 37-21). This action takes the data field from a received data packet and passes it up to Layer 4 along with an N_DATA IND primitive. (See Section 33, OSI Primitives on the Protocol Spreadsheet.) In an emulate mode, data normally arrives up at Layer 4 via GIVE_DATA actions at Layer 3.

(C) Resend

The RESEND function is mapped to **F1** on the second layer of action softkeys at Layer 3 for X.25 (below the PROTOCL softkey: see Figure 37-30). A RESEND action will resend the first packet in the window. The window is a queue that buffers data packets for retransmission in case one or more transmissions are lost or in error.

The first packet in the window always is the *earliest outstanding* (unacknowledged) packet. Every time an acknowledgment is received, the window is cleared to the extent of the acknowledgment and a new "first-packet" position is established. The first RESEND after an acknowledgment always sends the first window packet.

The *second and subsequent* RESENDS following an acknowledgment also will send the first window packet, provided that the keyword FIRST is appended directly to the RESEND entry. Otherwise, they send the NEXT (second) and subsequent window packets. Figure 37-31 shows the position of the the resend "pointer" after four consecutive RESEND NEXT actions. RESEND NEXT is the default resend when neither FIRST nor NEXT is specified.

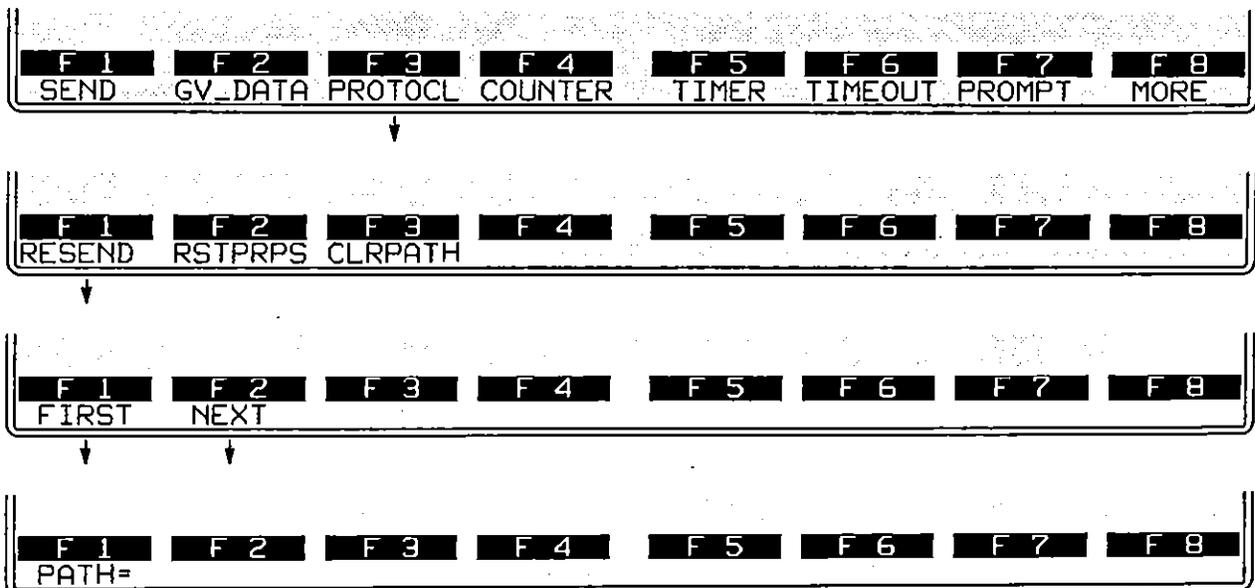


Figure 37-30 The RESEND action allows you to recover from sequence errors.

The resend-pointer is reset to the beginning of the window automatically by any acknowledgment, or by a RESEND FIRST action in the spreadsheet program.

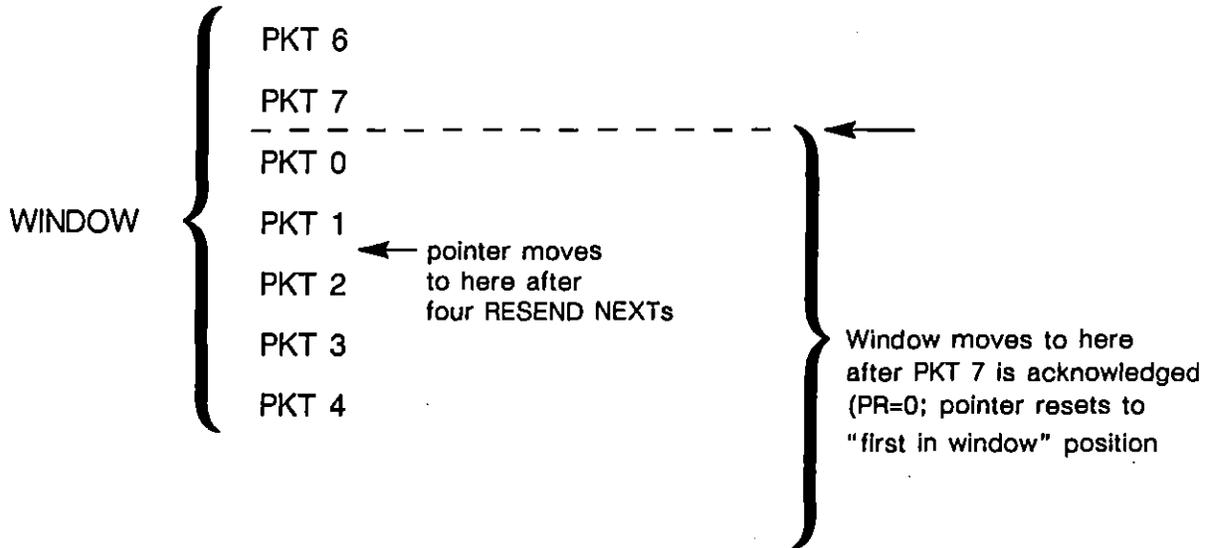


Figure 37-31 Resends cause the pointer to move, while acknowledgments move the pointer and the entire window.

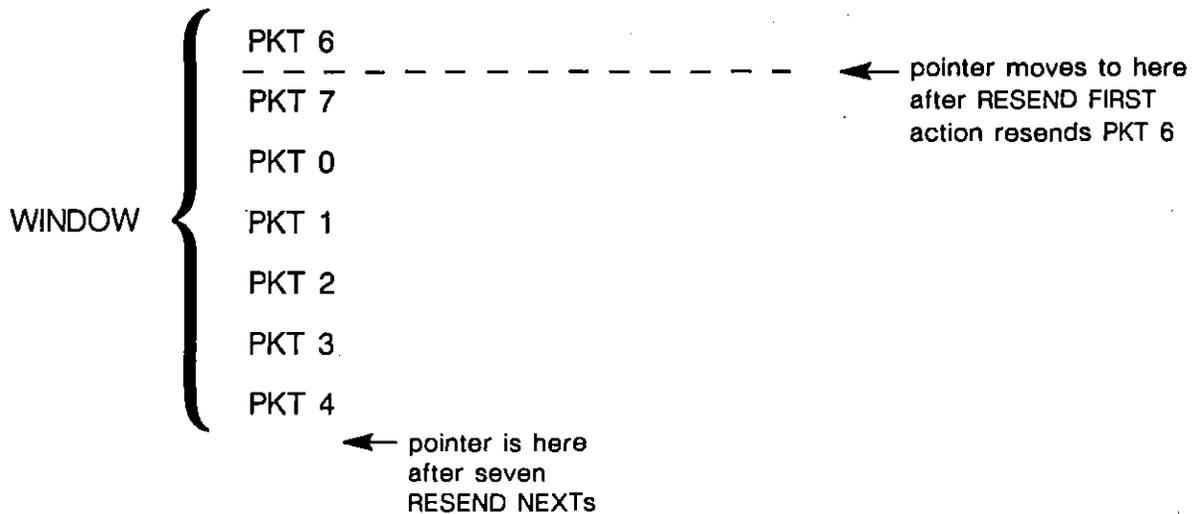


Figure 37-32 RESEND FIRST resets the pointer, allowing you to resend the entire window repeatedly.

1. *Resend first/next.* RESEND FIRST means that the resend-pointer is reset to the beginning of the window, the first packet in the window is resent, and the pointer is advanced to the second position in the window. The effect of a RESEND FIRST action is illustrated in Figure 37-32.

The RESEND FIRST action makes it possible for you to resend all the packets in the window one by one, and then *resend them again* if necessary.

2. *Path=* . The path in the resend packet can be set by this optional action. (Default is 0 in a RESEND action.)

(D) Reset P(R) and P(S)

RSTPRPS is the **F2** action on the rack of action softkeys below PROTOCL. (Refer again to Figure 37-21.) The sequence-number fields in data packets and supervisory packets can be reset by this Protocol Spreadsheet action. Sequence numbers are not reset *automatically* during a test by any packet that is sent or received.

The path number can be set by an optional PATH= selection.

RSTPRPS also clears the transmit window.

(E) Clear Path

Each call that is established in emulated mode is assigned to one of nine independent "paths," each with its own P(R) and P(S) numbering. Thus nine LCNs may be active at once. The CLRPATH action (Figure 37-21) allows you to return a path to the pool to be used again.

In the example below, a Clear request is expected. The actions that result will be to send a Clear confirmation and to clear the path.

```
LAYER: 3
STATE: clearing
CONDITIONS: RCV CLEAR
ACTIONS: SEND CLEAR_CONF
CLRPATH
```

The path number can be set by an optional PATH= selection. Without this selection, the path that is cleared will be that of the most recent packet received.

37.6 Display Actions

ENHANCE and SUPPRESS pertain to lines of data on the Layer 3 protocol trace (see Section 37.2). They do not suppress and enhance data on the raw-data display. Raw data is enhanced and suppressed at Layer 1.

DTE, DCE, and RCV conditions can trigger an ENHANCE or SUPPRESS action. These conditions are active when the INTERVIEW is in monitor mode or in either of the emulate modes.

(A) Enhance

Whenever a DTE, DCE, or RCV condition comes true at Layer 3, the packet that satisfied the condition can be enhanced on the X.25 Layer 3 protocol-trace display, or it can be deleted from the trace completely. In an actions block on the Protocol Spreadsheet, press the ENHANCE softkey. Figure 37-33 shows the three softkey subselections beneath ENHANCE. They are REVERSE, BLINK, and LOW.

Reverse-image and blink enhancements affect the plasma-display screen. In addition, a low-intensity enhancement can be applied to screens that are transmitted to a black-and-white monitor connected at the RS-170 port at the rear of the INTERVIEW.

Reverse, blink, and low enhancements can be mapped to colors on a color monitor attached at the INTERVIEW's RGB port (Figure 1-6). See Section 17.2 for an explanation of how reverse, blink, and low enhancements relate to character and background colors in the RGB output.

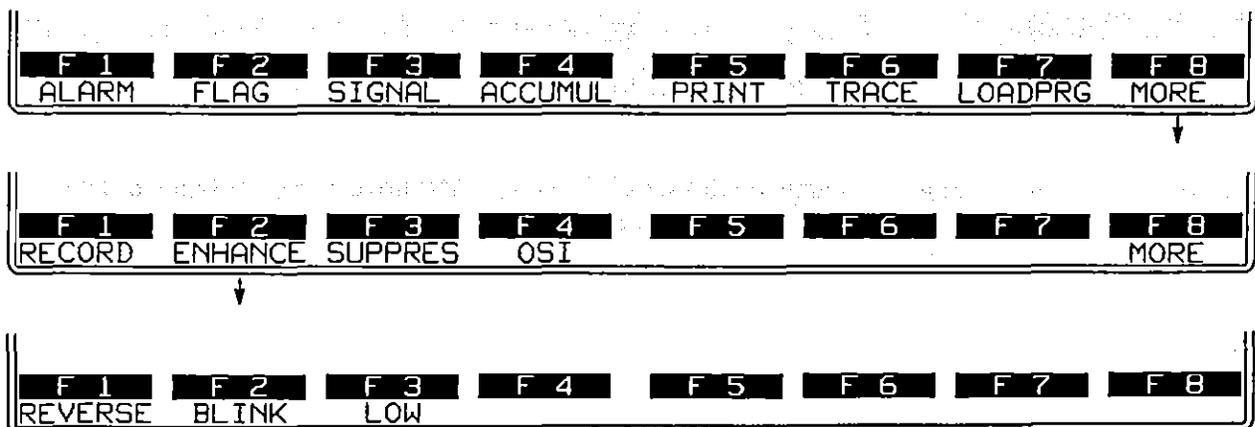


Figure 37-33 Selected packets on the protocol trace may be enhanced or suppressed.

Figure 37-34 shows one screen of a Layer 3 protocol trace in which an Interrupt packet has been enhanced in reverse video. The trigger that caused this enhancement was as follows:

CONDITIONS: DTE INT
ACTIONS: ENHANCE REVERSE

MON/DISK/FD1		BLK=00169 P 06/29/89 16:14							
ASCII/8/NONE/BOP									
SRC	LCN	TYPE	Pr	Ps	QDM	MISC	SIZE	TIME	BCC
DTE	004	RR	6				0003	1614:50.300	0
DCE	004	DATA	7	6			0073	1614:50.504	0
DTE	004	RR	7				0003	1614:50.563	0
DCE	004	DATA	7	7			0029	1614:50.621	0
DTE	004	RR	0				0003	1614:50.680	0
DCE	004	DATA	7	0			0010	1614:50.708	0
DTE	004	RR	1				0003	1614:50.764	0
DTE	004	INT					0004	1614:50.960	0
DTE	004	DATA	1	7	Q		0006	1614:51.000	0
DCE	004	INTCONF					0003	1614:51.091	0
DCE	004	DATA	0	1	Q		0042	1614:51.376	0
DTE	004	RR	2				0003	1614:51.434	0
DCE	004	DATA	0	2			0012	1614:51.436	0
DTE	004	DATA	2	0	Q		0042	1614:51.594	0
DTE	004	RR	3				0003	1614:51.625	0
F 1		F 2		F 3		F 4		F 5	
F 6		F 7		F 8		F 9		F 0	
L2TRACE		L3TRACE							

Figure 37-34 An interrupt packet has been highlighted.

(B) Suppress

Individual packets that are suppressed in Layer 3 actions are deleted from the trace display. Figure 37-33 shows the softkey path to SUPPRES.

37.7 Automatic Primitives

A table in a previous section (Table 33-2) listed the OSI service primitives that are monitored at the boundaries of Layer 3 as trigger conditions and sent up to Layer 4 or down to Layer 2 as user-entered spreadsheet actions. These primitives are layer-specific rather than protocol-specific and are not part of the personality package for X.25 Layer 3; but a few of the primitives are set in motion automatically by X.25 Layer 3 spreadsheet actions. These *automatic* primitives can be thought of as part of the Layer 3 actions themselves, and by extension as part of the X.25 protocol package.

Table 37-1 gives the set of X.25 Layer 3 actions that have action-primitives built into them. For example, whenever a GIVE_DATA action occurs at Layer 3, an N_DATA IND primitive is forwarded to Layer 4, where an N_DATA IND condition may be waiting to monitor it.

Table 37-1
Automatic Primitives Generated at X.25 Layer 3

X.25 Layer 3 Action	Automatic Primitive	To Layer
GIVE_DATA	N_DATA IND	4
SEND (TYPE)	(DL_CONNECT REQ*) DL_DATA REQ	2 2
RESEND	(DL_CONNECT REQ*) DL_DATA REQ	2 2

*Sent If Layer 2 shows Inactive status. DL_DATA REQ delayed until DL_CONNECT CONF returned by Layer 2.

Whenever a SEND or RESEND action is initiated at Layer 3, a DL_DATA REQ primitive is sent downward along with the DL data (the entire packet). This automatic primitive, which was nowhere entered by the user as an action at Layer 3, still will cause a DL_DATA REQ condition to be true at Layer 2.

If a SEND or RESEND action is triggered at Layer 3 while the data link at Layer 2 is inactive, Layer 3 will sense the absence of a link and delay the DL_DATA REQ. Instead it will send a DL_CONNECT REQ primitive. Only when a DL_CONNECT CONF has been returned by Layer 2 will Layer 3 release the data and the data primitive.

37.8 Programming Example: Forcing Data Packets Out on the Line

This program is constructed around the "line-access" program that was given at the beginning of Section 37.4(A). It has elements in common with the Layer 2 emulation in Section 36.9.

The program allows you to send data packets containing fox messages out onto the line interface (and up on the display) *even when you are not connected to another device*. In other words, it allows you to get the feel of layered programming before you attempt a live emulation.

The bulk of the program is entered at Layer 2. Personality packages for X.25 must be loaded in at Layers 2 and 3.

Sample Test: Force Data-Packet Transmit:

LAYER: 3

STATE: fox

CONDITIONS: KEYBOARD " Ff"
ACTIONS: SEND DATA " ((FOX)) "

LAYER: 2

STATE: LINK

CONDITIONS: DL_CONNECT REQ
ACTIONS: DL_CONNECT CONF
NEXT_STATE: Info_xfr

STATE: Info_xfr

CONDITIONS: DL_DATA REQ
ACTIONS: SEND INFO " ((DL_DATA)) "

CONDITIONS: RCV INFO
ACTIONS: SEND RR RESP
GIVE_DATA

CONDITIONS: T1_EXPIRED
NEXT_STATE: xmt_wndw

STATE: xmt_wndw

CONDITIONS: ENTER_STATE
ACTIONS: RESEND FIRST

CONDITIONS: FRAME_SENT
MORE_TO_RESEND
ACTIONS: RESEND NEXT

CONDITIONS: FRAME_SENT
NO_MORE_TO_RESEND
ACTIONS: ALARM
NEXT_STATE: Info_xfr

At Layer 3, you simply enter a KEYBOARD condition and a SEND action. During Run mode, you will press the [F] key in order to send a fox message inside a data packet.

The DL_CONNECT REQ primitive is sent automatically by Layer 3 before he hands the first data packet down to Layer 2. The DL_CONNECT CONF action-primitive is entered "manually" at Layer 2. It is meant to fool Layer 3 into thinking that there is a link.

When Layer 2 does not receive an acknowledgment to his first INFO frame before a T1 timeout expires, he resends the INFO frames containing the data packet (containing the fox message). The RESEND action restarts the T1 timer automatically. Subsequent timeouts will cause additional resends.

Each time the user presses the [F] key, a new data packet is added to the retransmit window at Layer 2. With each T1 timeout, the entire window is resent.

38 SDLC

```

** Layer Setup **

DRIVE: HRD   Layer 1 Package: NO PACKAGE   Packages Loaded
DRIVE: FD2   Layer 2 Package: SDLC         NO PACKAGE
DRIVE: HRD   Layer 3 Package: NO PACKAGE   NO PACKAGE
DRIVE: HRD   Layer 4 Package: NO PACKAGE   NO PACKAGE
DRIVE: HRD   Layer 5 Package: NO PACKAGE   NO PACKAGE
DRIVE: HRD   Layer 6 Package: NO PACKAGE   NO PACKAGE
DRIVE: HRD   Layer 7 Package: NO PACKAGE   NO PACKAGE

Depress XEQ Key To Load The Selected Packages

Select Layer
F1 F2 F3 F4 F5 F6 F7 F8
LAYER-1 LAYER-2 LAYER-3 LAYER-4 LAYER-5 LAYER-6 LAYER-7 PROTSEL
    
```

Figure 38-1 The SDLC personality package for Layer 2 is loaded from the Layer Setup screen.

```

** SDLC Frame Level Setup **

Idle Timeout:      1.0 sec
Emulate:           PRIMARY
Mode of Operation: MOD 8
Window Size:       7
Emulation Addressing: MULTI-DROP

DROP  ADDR      DROP  ADDR      DROP  ADDR      DROP  ADDR
0     ---      4     ---      8     ---      12    ---
1     ---      5     ---      9     ---      13    ---
2     ---      6     ---     10    ---      14    ---
3     ---      7     ---     11    ---      15    ---

Select Emulation Addressing
F1 F2 F3 F4 F5 F6 F7 F8
PT-PT MULTI
    
```

Figure 38-2 Protocol Configuration screen for SDLC.

38 SDLC

SDLC is a "layer personality package" of Layer 2 functions that are loaded into memory from disk via the Layer Setup screen. Figure 38-1 shows the Layer Setup screen configured to load in the SDLC package from floppy-disk drive #2. Refer to Section 8 for details on operating the Layer Setup screen.

The SDLC package consists of the following:

- A special SDLC Frame Level Setup screen that controls certain parameters when the unit is tracing or emulating SDLC.
- Multi-drop or point-to-point operation.
- A protocol trace (illustrated in Figure 38-3) that displays significant SDLC events. This trace is accessible by softkey in Run mode at all times.
- A group of conditions and actions at Layer 2 on the Protocol Spreadsheet that facilitate SDLC programming. Figure 38-8 shows the softkey path to the first rack of condition softkeys when the SDLC package is loaded in at Layer 2.

38.1 Frame-Level Setup

The parameters on the SDLC Frame Level Setup screen must be configured correctly for an accurate trace display and for proper emulation. Use this screen also to enable multi-drop operation.

To bring up this screen, first go to the Layer Setup screen (press **PROGRAM**, **F5**). Execute the SDLC selection at Layer 2: SDLC should appear in the **Packages Loaded** column. Press **F8** (labeled **PROTSEL**) to bring up a prompt to **Select Protocol Configuration Screen**. Then press **F2** (**LAYER-2**) to call up the SDLC Frame Level Setup screen.

The parameter fields on this screen are shown in Figure 38-2. **Idle Timeout**, **Emulate**, **Window Size**, **Emulation Addressing**, and **ADDR** apply to interactive (emulate) tests only. **Mode of Operation** must be configured correctly for the protocol trace as well as for proper emulation.

(A) Idle Timeout

Enter a four-digit (including decimal point) timeout value in this field. The largest valid entry is 65.5 seconds. The smallest entry is .001 second, or 1 millisecond.

Idle timer is the retransmission timer for SDLC INFO and supervisory command frames. When Emulate: PRIMARY is selected on the SDLC Frame Level Setup screen and a value is entered in the Idle Timeout field on this menu, the layer 2 package will handle timings as follows:

- Whenever the INTERVIEW sends a command INFO or supervisory frame with the P bit set and there are no previous polling frames sent by the INTERVIEW currently outstanding (unacknowledged) to the same address, the timer starts timing down from the value entered on the Frame Level Setup screen.
- An acknowledgment by the secondary of the most recent polling INFO or supervisory frame transmitted by the INTERVIEW stops the timer (so that it does not expire). This acknowledgment *must* occur in a frame with the F bit set.
- If F = 0 in the acknowledgment by the secondary of the most recent polling frame sent by the primary, the idle timer restarts at the value selected on the configuration screen.
- An acknowledgment by the secondary of a frame that is not the most recent polling frame transmitted by the INTERVIEW is an incomplete acknowledgment, even if F = 1. This acknowledgment also will restart the idle timer.

Expiration of this Frame Level Setup timeout can only be detected by a T1_EXPIRED condition on the Protocol Spreadsheet at Layer 2.

(B) Emulate Primary/Secondary

There are two selections in the Emulate field on the SDLC Frame Level Setup screen, PRIMARY and SECONDARY. The difference between these two modes is that the primary device makes use of the idle timer. The secondary does not.

(C) Mode of Operation

The Mode of Operation field refers to the mode of numbering INFO and supervisory frames. There are two options, MOD 8 and MOD 128.

MOD 8 uses sequence numbers 0-7. MOD 128 adds an extra byte to the control field in INFO, RR, RNR, REJ, and SREJ frames. See Figure 38-5. This extra byte allows sequence numbers in a range of 0-127.

The correct "modulus" must be selected in this field in order to conduct interactive communications and also to generate an accurate SDLC Layer 2 trace.

(D) Window Size

Any window size may be entered up to the current modulus minus one: 7 or 127. The window size is the maximum number of unacknowledged I-frames that Layer 2 will buffer for retransmission. When the limit is reached, any further INFO frames that are named in SEND actions triggered at Layer 2 will be passed to Layer 1 for transmission but not buffered for retransmission.

The window is a queue that buffers frames for retransmission in case one or more transmissions are lost or in error. A RESEND action will resend the first (earliest) frame in the window. Successive RESENDS will send successive frames until there are no more frames to resend; or until the window is reset by an acknowledgment or by a RESEND FIRST action.

(E) Emulation Addressing

Indicate in this field whether the link is POINT-TO-POINT or MULTI-DROP. POINT-TO-POINT is the default selection. For the INTERVIEW to track N(R) and N(S) for multiple addresses, you must choose MULTI-DROP. If you select MULTI-DROP, an additional field, ADDR, will appear.

(F) ADDR

Specify the addresses of particular controllers in the ADDR table. You may enter up to 16 addresses in the table. The INTERVIEW uses the drop number associated with each listed address to track N(R) and N(S) for resend purposes. All other INFO and supervisory frames with addresses not included in this table will be tracked as a single group. All frames will be displayed on the SDLC protocol trace.

Enter each address as a two-digit hexadecimal value. Use the numbered keys only (not with the HEX key) to make these entries. Addresses in the range 00 through FF are valid.

NOTE: If you enter multiple addresses in this table, consider increasing the number IL buffers. (The default number of buffers is sixteen.) Use the following formula to determine the number of IL buffers you may need:

$$\text{number of buffers} = (\text{number of addresses}) * (\text{window size}) + 3.$$

See Sections 27.5 and 66.1 for information on changing the number of IL buffers.

38.2 Protocol Trace

The SDLC package includes an automatic frame-trace display that summarizes link-level activity. This trace mode is enabled whenever the unit is in Run mode, both real-time and frozen.

While the unit is in Run mode, press the softkey for L2TRACE (F2) on the primary rack of display-mode softkeys) to bring the protocol trace for SDLC Layer 2 to the screen. Figure 38-3 is an example of this trace display. Each horizontal row in the trace represents a frame.

(A) The Protocol Trace in Freeze Mode

Press **FREEZE** to prevent the addition of new data to *all* the display buffers, including the trace buffers. The frozen trace display may be scrolled through or paged through. The top line always is the cursor line (though there is no actual cursor on the trace display). Pressing **DOWN** or **↓** moves the viewing "window" down relative to the data to add one line of fresher data to the bottom of the screen. Pressing **UP** or **↑** moves the viewing window up to add a line of older data to the top of the screen.

Depression of the **FREEZE** key adds fifteen lines—one full page—of newer frames to the frozen trace screen. Depression of **UNFREEZE** adds fifteen lines of older frames.

MON/LINE		BLK=00000 S 04/21/89 22:22							
ASCII/B/NONE/BOP									
SRC	ADDR	TYPE	Nr	Ns	P/F	SIZE	TIME	BCC	
DTE	C1	INFO	7	3	0	0264	2220:06.206	ⓐ	
DTE	C2	SNRM			1	0002	2220:06.230	ⓐ	
DTE	C1	INFO	7	0	0	0011	2220:06.276	ⓐ	
DTE	C1	INFO	7	1	0	0046	2220:06.441	ⓐ	
DTE	C1	INFO	7	2	0	0236	2220:07.247	ⓐ	
DTE	C1	INFO	7	3	0	0264	2220:08.151	ⓐ	
DTE	C1	INFO	7	4	0	0264	2220:09.059	ⓐ	
DTE	C1	INFO	7	5	0	0264	2220:09.976	ⓐ	
DTE	C1	INFO	7	6	0	0136	2220:10.445	ⓐ	
DTE	C1	RR	7		1	0002	2220:10.466	ⓐ	
DCE	C1	INFO	7	7	1	0012	2220:10.788	ⓐ	
DTE	C1	RR	0		1	0002	2220:10.812	ⓐ	
DTE	C1	INFO	0	7	0	0015	2220:10.872	ⓐ	
DTE	C1	INFO	0	0	0	0041	2220:11.021	ⓐ	
DCE	C1	INFO	7	0	1	0011	2220:11.115	ⓐ	

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
DATA	L2TRACE			STATS		NO DISP	

Figure 38-3 Each horizontal row on the trace display represents a frame.

The frame displayed on the top line of frozen trace-data will appear as the first frame in the raw-data or data-plus-leads display. To view the raw data that generated a particular line in the trace display, use **RAW** or **RAW** (or **↑** or **↓**) to

move the line in question to the top of the screen. Then press one of the data softkeys. Figure 38-4 shows part of a dual-line data screen in Freeze mode. The first frame in the display is the same one that is traced at the top of Figure 38-3.

```
*MON/LINE*          BLK=00000 S 04/21/89 22:22
OFFSET=22481 68% DTE=11100110 DCE=
A%YPF2%F%-Choose another schedule%J%YPF3%.0%-Cho
ose a conference room%<%YPF4%<W%-Change to the next day%(1%YPF5
%6%-Change to the previous day%A%YPF6%F%-Look at the whole mo
nth%J%YPF7%&0%-Schedule a meeting%J/%YPF8%JW%-Print a schedule%
```

Figure 38-4 Data-display of Protocol Trace shown in Figure 38-3.

(B) Trace Columns

The columns in the protocol trace for SDLC are explained below.

1. *Source.* The SRC column identifies the lead on which the frame was monitored, TD (DTE) or RD (DCE).
Just as on the data display, RD data is underlined.
2. *Address.* The address byte (see Figure 38-5) is given in the ADDR column, with its two hexadecimal digits presented as full-size alphanumerics.
The address in SDLC always belongs to the secondary.
3. *Type.* The mnemonic (abbreviated) names for twenty frame types as they appear in the TYPE column of the protocol trace are shown in Figure 38-5 under "CONTROL." The control field, therefore, indicates the frame type. If a control octet does not fit any of the patterns in the figure, the frame is listed in the TYPE column as UNKWN followed by the hexadecimal value of the control byte: UNKWN=3F.
If the number of bytes in the frame is below the required minimum, the frame is posted as INVALID.
4. *N(R) and N(S).* One column on the frame-level trace is devoted to N(R) values, and one column to N(S). The frame types that include N(R) or N(S) fields in their control fields are indicated in Figure 38-5.

In multi-drop operation, each address listed in the table on the Frame Level Setup screen (Section 38.1) has N(R) and N(S) tracked. All other addresses not included in the table are tracked as a single group.

N(R) and N(S) occupy three bits each in modulo 8, seven bits each in modulo 128.

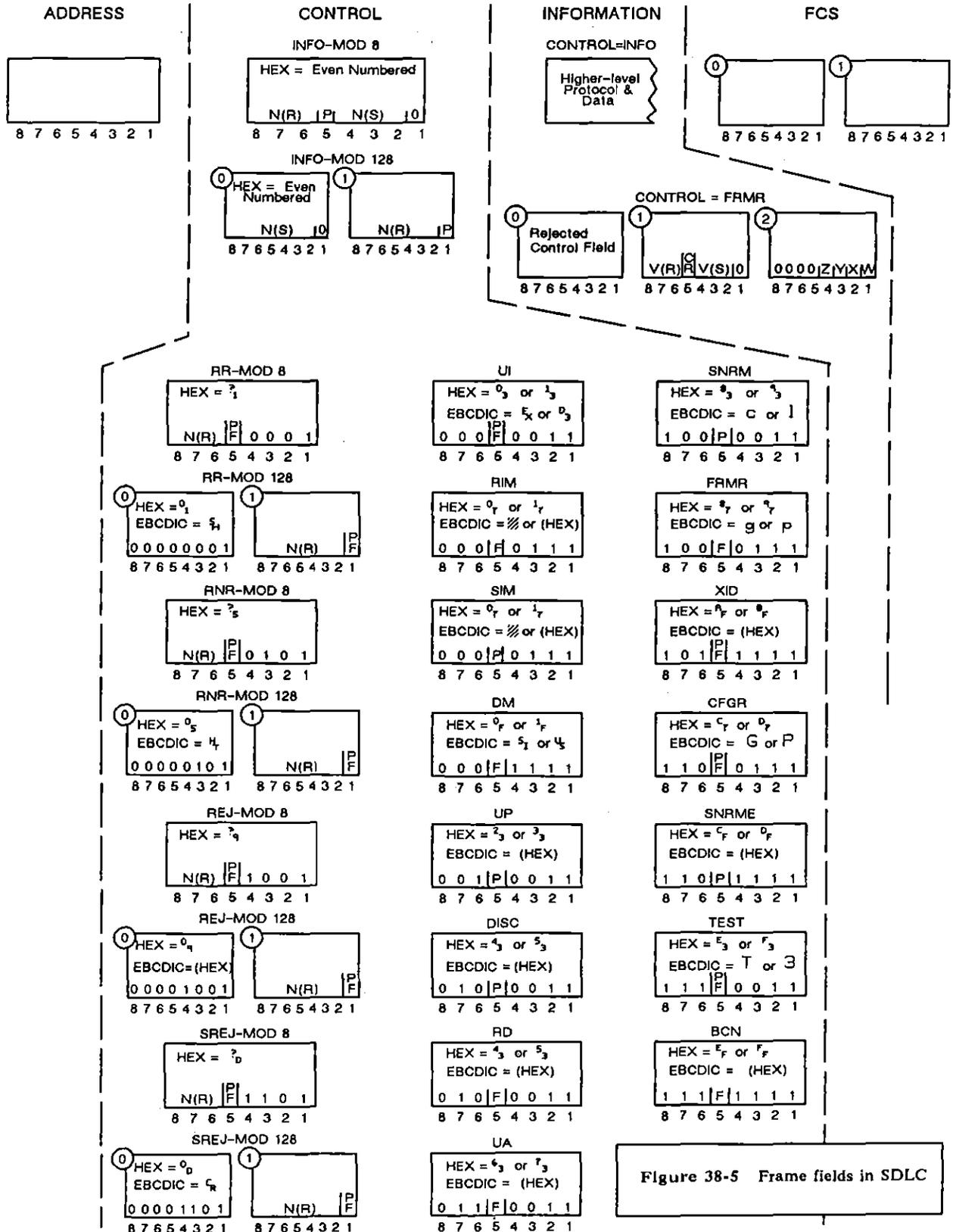


Figure 38-5 Frame fields in SDLC

frames, 0 through 6; the DCE acknowledges frames through 6 (NR=7); the DTE begins a new window with frame 7; and so on.

The two outside columns reveal a similar pattern for DCE I-frames.

5. *P and F.* The status of the poll or the final bit is given in the **P/F** column. Whether this bit is the P or F bit is indicated for most frame types in Figure 38-5 (under "CONTROL").
6. *Size.* The number of bytes in each frame is given in this column in four decimal digits. The count begins with the address byte and excludes the two-byte FCS. Frames without I-fields show a count of two.
7. *Time.* The time of the arrival of the *end of the frame* at the DTE or DCE monitor is provided by a 24-hour clock and posted to the trace display. The clock is accurate to the millisecond.

When **Time Ticks: ON** is selected on the Front-End Buffer Setup screen (see Section 9), time values are incorporated into the data itself. As a result, times posted to the trace display will not be affected when recorded data is played back, even at varying speeds.

If **Time Ticks: OFF** was selected instead during live recording, times on the trace during playback will be influenced by "local conditions" such as playback speed, idle suppression, etc.

8. *Frame checking.* An SDLC frame ends as soon as a flag or seven 1-bits in a row are detected. If a flag ends the frame, a frame check is performed and the result is posted both to the data display and to the **BCC** column of the trace display. The symbol **☐** denotes a good frame check, while **■** symbolizes a bad frame.

■ for abort is posted to the displays when a frame is ended by seven 1-bits.

38.3 Monitor Conditions

When the SDLC personality package is loaded in (via the Layer Setup screen), a set of conditions checks DTE and DCE leads both in monitor and emulate modes. This set of conditions is accessed by the DTE and DCE selections on the first rack of condition softkeys at Layer 2. See Figure 38-8.

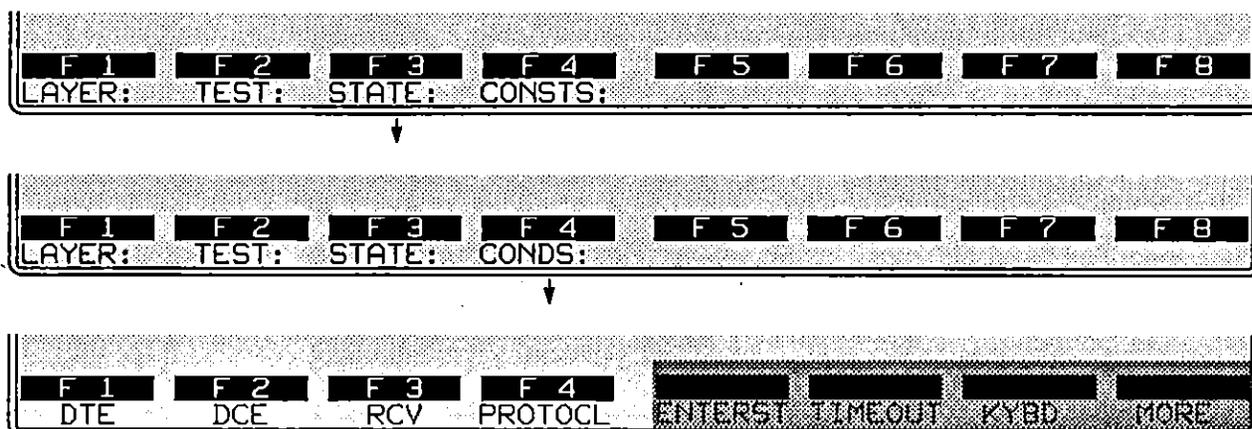


Figure 38-8 Unlike RCV conditions, the softkeys for DTE and DCE are valid when the INTERVIEW is monitoring the line passively.

After the keyword DTE (or DCE) is written to the spreadsheet, a rack of softkeys appears that represent *types* of frames: INFO, SNRM, UA, and so forth.

(A) Frame Types

The softkeys for INFO, supervisory, unnumbered, and "other" frames are illustrated in Figure 38-9. Press a softkey to write one of these frame types to the Layer 2 spreadsheet. DTE or DCE followed by a frame-type mnemonic—DCE INFO, for example, or DTE SNRM—is a complete condition and will come true if a matching frame is monitored. Address, poll/final, and BCC conditions may be added to the simple frame mnemonic, but they are optional.

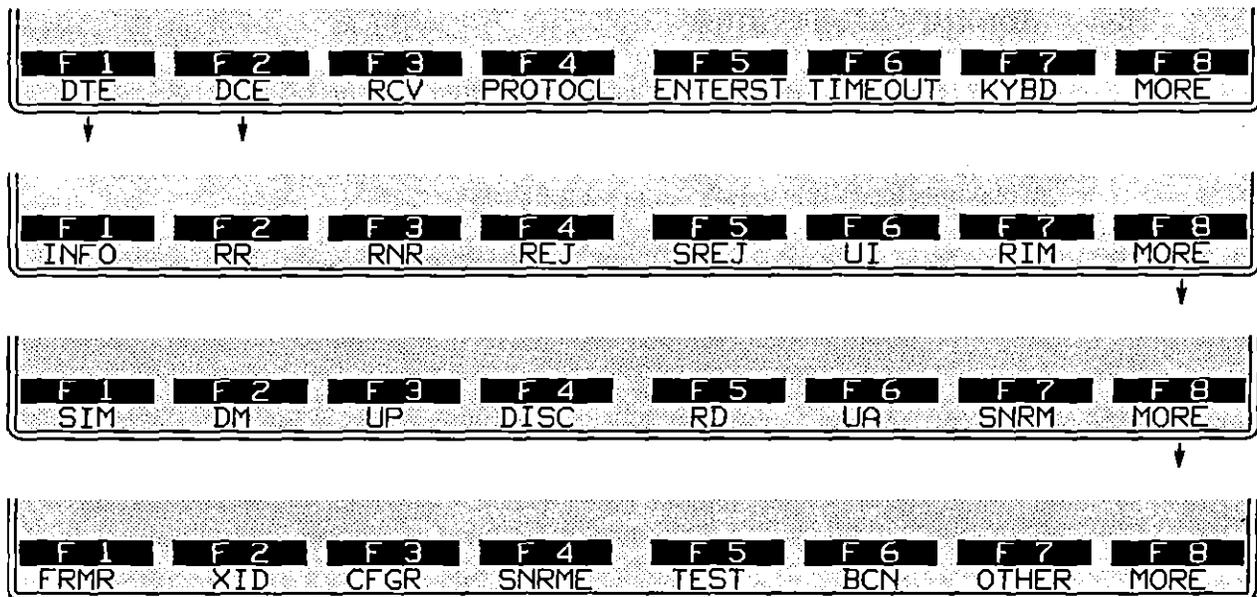


Figure 38-9 Frame types.

1. *Info frames.* INFO frames differ for MOD 8 and MOD 128 numbering schemes. (See Figure 38-5.) For spreadsheet conditions to match I-frames accurately, the correct numbering system ("Mode of Operation") should be selected on the Frame Level Setup screen.
2. *Supervisory frames.* The four supervisory-frame types that can be searched for on the data leads are RR (Receive Ready), RNR (Receive Not Ready), REJECT, and SREJ (Selective Reject). These frames always contain N(R) fields (see Figure 38-5) and serve mainly to acknowledge or reject INFO frames.

Like INFO frames, supervisory frames are constructed differently according to the numbering scheme, MOD 8 or MOD 128.
3. *Unnumbered frames.* Unnumbered frames generally assist in link-setup and takedown. These contain neither N(R) nor N(S) fields. Fifteen unnumbered-frame types are shown in Figure 38-9, from UI in the second rack of softkeys through BCN in the bottom rack.
4. *Other frames.* Any frame type may be entered as a hexadecimal value instead of by name. Press the softkey for OTHER. See Figure 38-10. Then enter the hex byte in the form of two alphanumerics. Here, for example, is a SNRM (with the P bit set) entered as a hexadecimal:

CONDITIONS: DCE OTHER 93

Address, poll/final, and BCC conditions may be appended to OTHER conditions. In MOD 8, the P/F bit is already specified in the hex entry, and a P/F condition will be ignored.

Figure 38-10 The hex value of any frame may be specified under *OTHER*.

5. *Address*. An address condition may be added to INFO, supervisory, unnumbered, and OTHER conditions. Press the softkey for ADR=, shown in Figure 38-11. Then enter the hexadecimal address octet as two alphanumerics. The address octet 5, for example, appears as follows:

CONDITIONS: DTE INFO ADR= C1

Figure 38-11 The hex value of the address byte is entered as two alphanumerics for *all* frame types.

To bypass the ADR= selection (as well as the other options on the same rack of softkeys in Figure 38-11) press **done**.

6. *Poll/final bit.* P/F conditions are optional for all frame types. P/F values of 0 or 1 are entered by the softkey sequence in Figure 38-12.

Press **Done** to bypass the P/F= condition and the other conditions on the same softkey level in Figure 38-12.

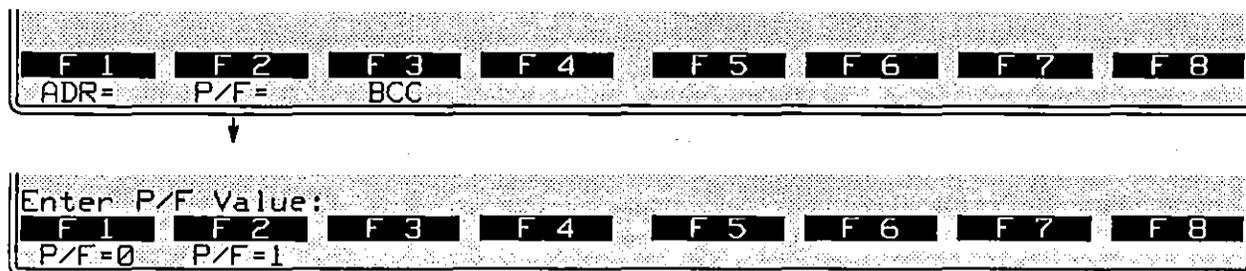


Figure 38-12 The value of the P/F bit may be chosen as a condition.

(B) BCC Conditions

DTE and DCE frames may be monitored for good and bad frame checks and for aborts. All DTE or DCE frames may be monitored with respect to frame checking, as in this example:

CONDITIONS: DTE BDBCC

The softkey sequence for this spreadsheet entry is given in Figure 38-13.

Or a particular *type* of frame may have a BCC or abort condition appended to it:

CONDITIONS: DCE INFO ABORT

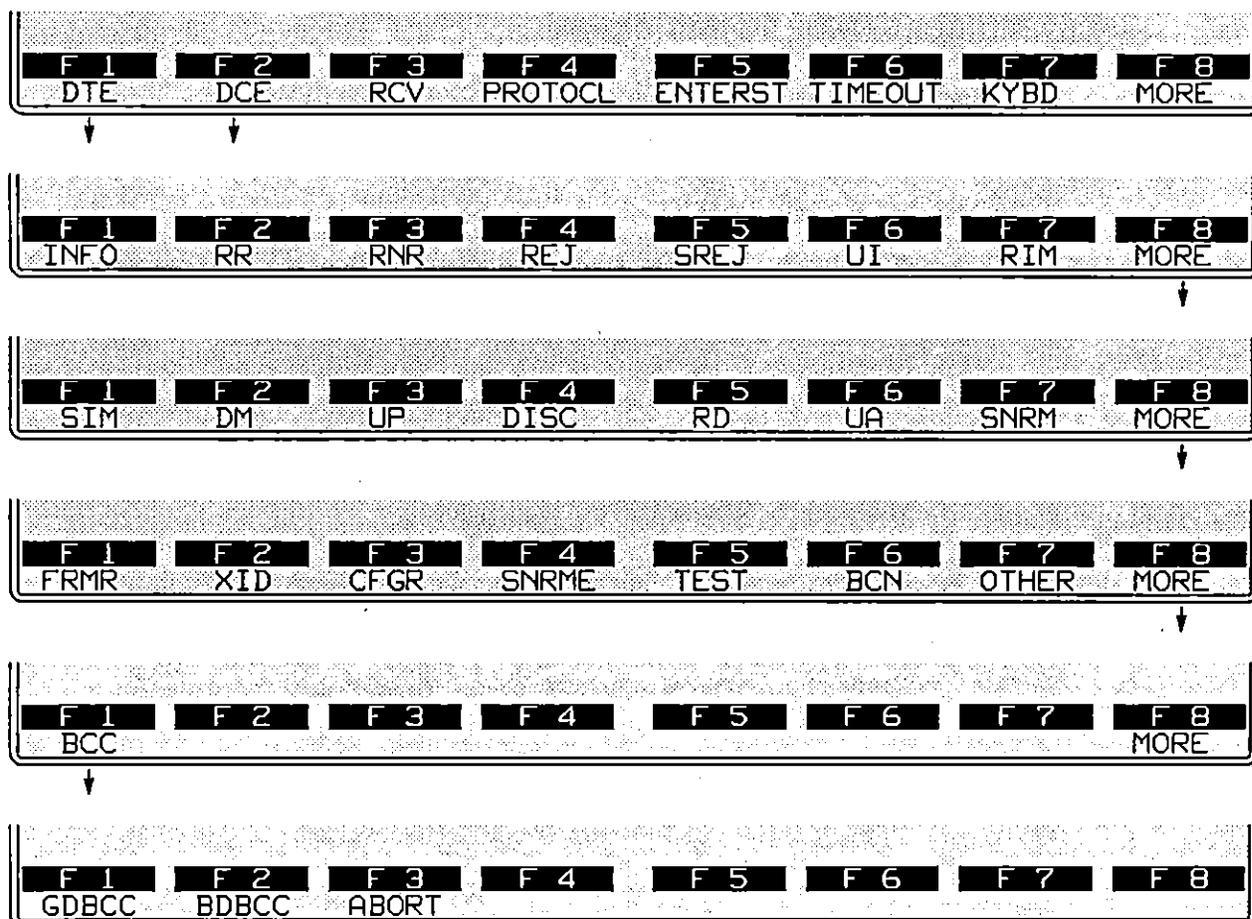


Figure 38-13 A condition may search for *all* good, bad, or aborted frames.

38.4 Emulate-Mode Conditions

The remaining conditions are functional only when the Line Setup menu is configured for Mode: `EMULATE DTE` or `EMULATE DCE`.

(A) Receive Conditions

Like DTE and DCE conditions, RCV conditions monitor a data lead for SDLC frame types. RCV conditions operate only in emulate modes, and they check only the data lead that the INTERVIEW is not using to transmit. While a RCV condition may look like a DTE or DCE condition—RCV INFO P/F=1 looks the same as DCE INFO P/F=1—there are important differences that are noted below.

1. *Valid frame sequencing.* To satisfy RCV conditions, numbered frames must have correct N(R) and N(S) sequencing.
2. *Good BCC.* RCV conditions cannot match frames with bad frame checks, nor can they match aborted frames. (Emulate-mode conditions are designed for ease of programming, and the assumption is that as an SDLC emulator, you are not required to acknowledge—or negative-acknowledge—bad or aborted frames.)

If you wish to count bad BCCs or aborts, use DTE or DCE conditions instead of RCV conditions.

3. *Type invalid or unknown.* RCV conditions can detect frames that are invalid “types”—the control field is missing, for example, or the I-field is missing in an I-frame. The Protocol Spreadsheet entry for this condition is RCV INVALID, and the softkey sequence is illustrated in Figure 38-14.

A frame may be valid in all respects but have a control field that indicates a nonstandard frame type. Such a frame may be matched by a RCV UNKNOWN condition (Figure 38-14).

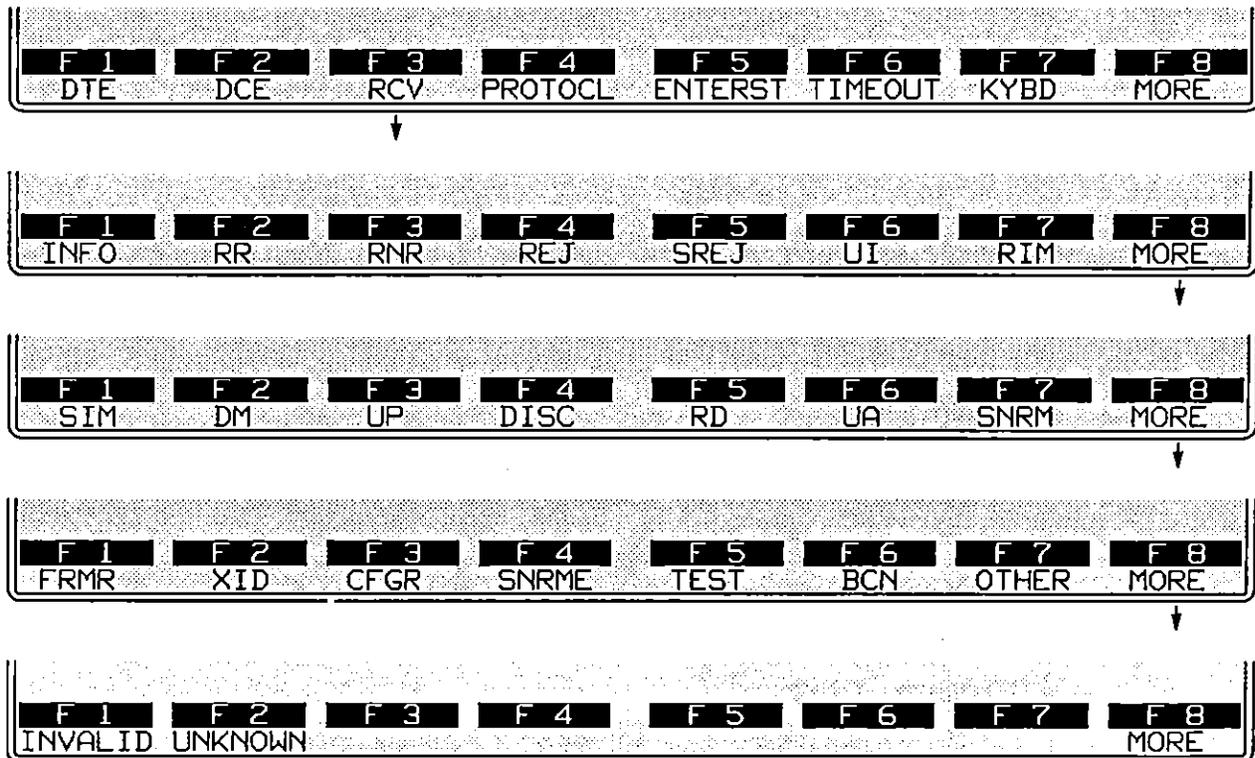


Figure 38-14 *INVALID* and *UNKNOWN* are frame types for *RCY* conditions.

(B) N(S) Error

As a Layer 2 emulator, you do respond to *INFO* frames that have *N(S)* errors. These are detected as *NS_ERR* conditions, not as *RCY INFO* conditions.

NS_ERRs apply only to frames received when you are emulating. The same frame that triggers an *NS_ERR* condition also may satisfy a *DTE INFO* or *DCE INFO* condition—but not a *RCY INFO* condition.

NS_ERR will come true for any received frame whose *N(S)* value is not one higher than the previous *N(S)*.

In the first rack of condition softkeys at Layer 2, press *PROTOCL*. Then press the softkey for *NS_ERR*. See Figure 38-15.

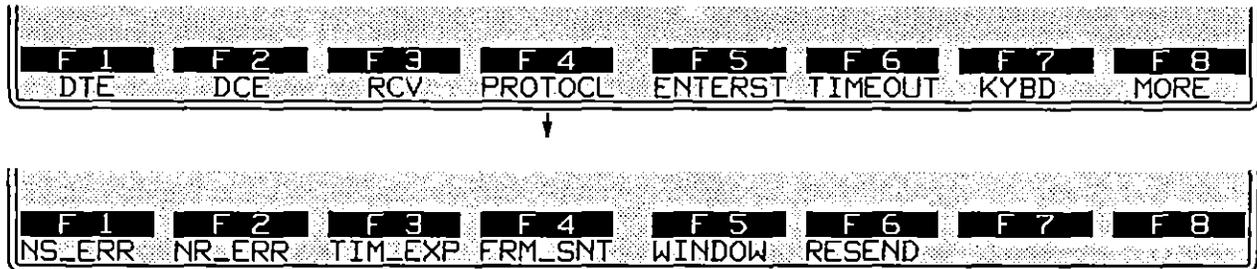


Figure 38-15 The *PROTOCL* key brings up six SDLC emulate conditions.

(C) N(R) Error

Received INFO or supervisory frames may have N(R) errors. Such errors are detected as NR_ERR conditions, not as RCV INFO or RR (or RNR or REJ) conditions.

A valid N(R) is any value that (1) acknowledges a frame that is *outstanding* (waiting for acknowledgment); or (2) repeats the last acknowledgment. Any other N(R) value is detected as an error.

(D) Timeout Expired

This condition detects the expiration of the idle timeout that is regulated on the SDLC Frame Level Setup screen. See Section 38.1(A), above.

(E) Frame Sent

This condition is true when, as a result of a SEND or RESEND action, a frame has been passed down to Layer 1.

(F) Window Conditions

The size of the Layer 2 retransmit window is configured on the SDLC Frame Level Setup screen. See Section 38.1(D). There are four conditions that test the current status of this window. They are WINDOW FULL, WINDOW EMPTY, WINDOW NOT_FULL, and WINDOW NOT_EMPTY. The softkey sequence for the WINDOW options is shown in Figure 38-16.

WINDOW FULL is true when the window is full of unacknowledged frames and the Layer 2 protocol package will not buffer additional frames until some acknowledgment is received.

Each time an acknowledgment is received, the window is flushed to the extent of the acknowledgment. WINDOW EMPTY means that the latest acknowledgment was complete and left no frames outstanding (unacknowledged). If an RR response is received and the acknowledgment is only partial, this condition will be true:

CONDITIONS: RCV RR
WINDOW NOT_EMPTY

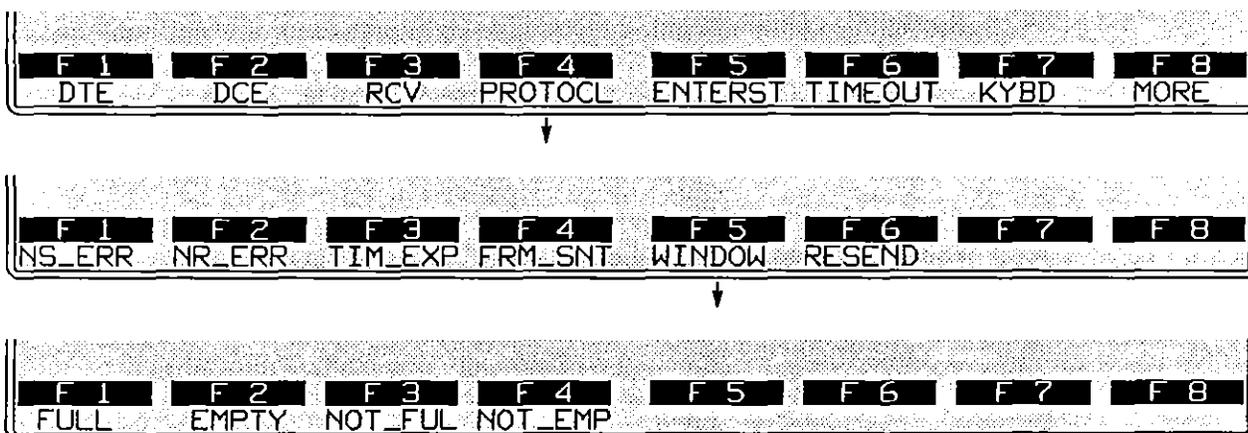


Figure 38-16 When the retransmit window fills, Layer 2 stops passing frames down to Layer 1.

If you select a window condition when the INTERVIEW is configured for multi-drop operation, an ADR= softkey selection appears. During multi-drop emulation, there may be several transmit windows—one for each controller address listed in the ADDR table on the SDLC Frame Level Setup screen. (See Section 38.1.) The INTERVIEW can check window conditions for any of these addresses, but only if you complete the ADR= entry.

CAUTION: Window conditions are status conditions (see Section 30.2) and must always be used in combination with a transitional condition such as a RCV condition.

(G) More to Resend

Frames in the window may have to be resent, usually as the result of an idle-timer timeout or a Reject frame. One RESEND action retransmits one frame in the window, beginning with the earliest. Subsequent RESEND actions retransmit subsequent frames. The MORE_TO_RESEND and NO_MORE_TO_RESEND conditions allow you to retransmit the entire window, as in the “recover” state in this example:

```

CONDITIONS: RCV REJ
NEXT_ST: recover
STATE: recover
CONDITIONS: ENTER_STATE
ACTIONS: RESEND
CONDITIONS: FRAME_SENT
MORE_TO_RESEND
ACTIONS: RESEND
CONDITIONS: FRAME_SENT
NO_MORE_TO_RESEND
NEXT_ST: xfer

```

During multi-drop emulation, there may be several transmit windows—one for each controller address listed in the ADDR table on the SDLC Frame Level Setup screen. (See Section 38.1.) For MORE_TO_RESEND and NO_MORE_TO_RESEND conditions, the INTERVIEW will check the transmit window of the address specified in the last RESEND action.

MORE_TO_RESEND and NO_MORE_TO_RESEND conditions may be written to the Protocol Spreadsheet by the softkeys shown in Figure 38-17.

CAUTION: MORE_TO_RESEND and NO_MORE_TO_RESEND are status conditions (see Section 30.2) and must always be used in combination with a transitional condition such as FRAME_SENT.

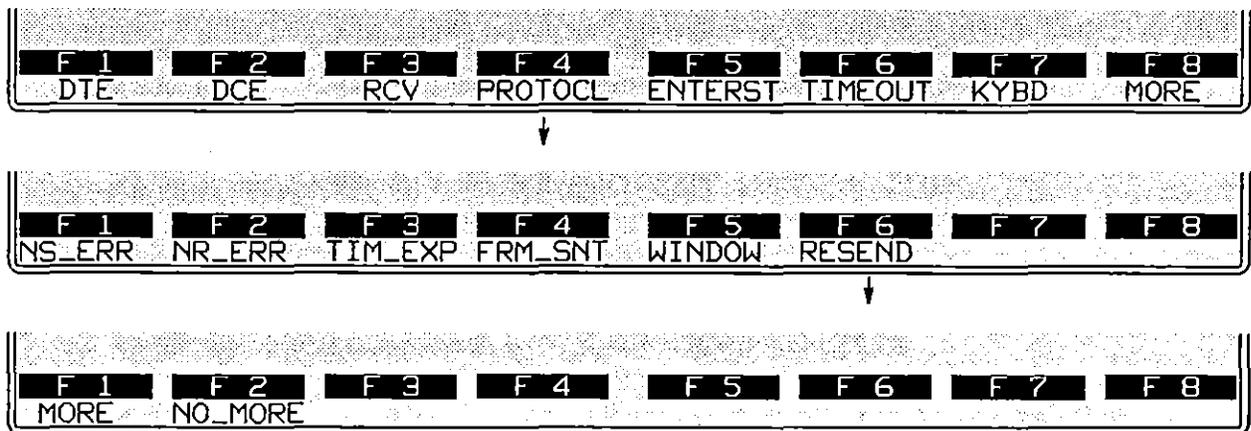


Figure 38-17 The *MORE_TO_RESEND* condition allows you to resend the entire window of frames and then stop when there are *NO_MORE_TO_RESEND*.

38.5 Emulate Actions

When you have completed a block of conditions in a Protocol Spreadsheet test at Layer 2, press **DONE** to access the set of actions that can be taken as a result of the block of conditions coming true. The set of actions that are specific to the SDLC personality package are shown in the racks of softkeys in Figure 38-18. Except for **ENHANCE** and **SUPPRES**, the actions shown have meaning only when the **INTERVIEW** is emulating DTE or DCE, and not when it is monitoring the line passively.

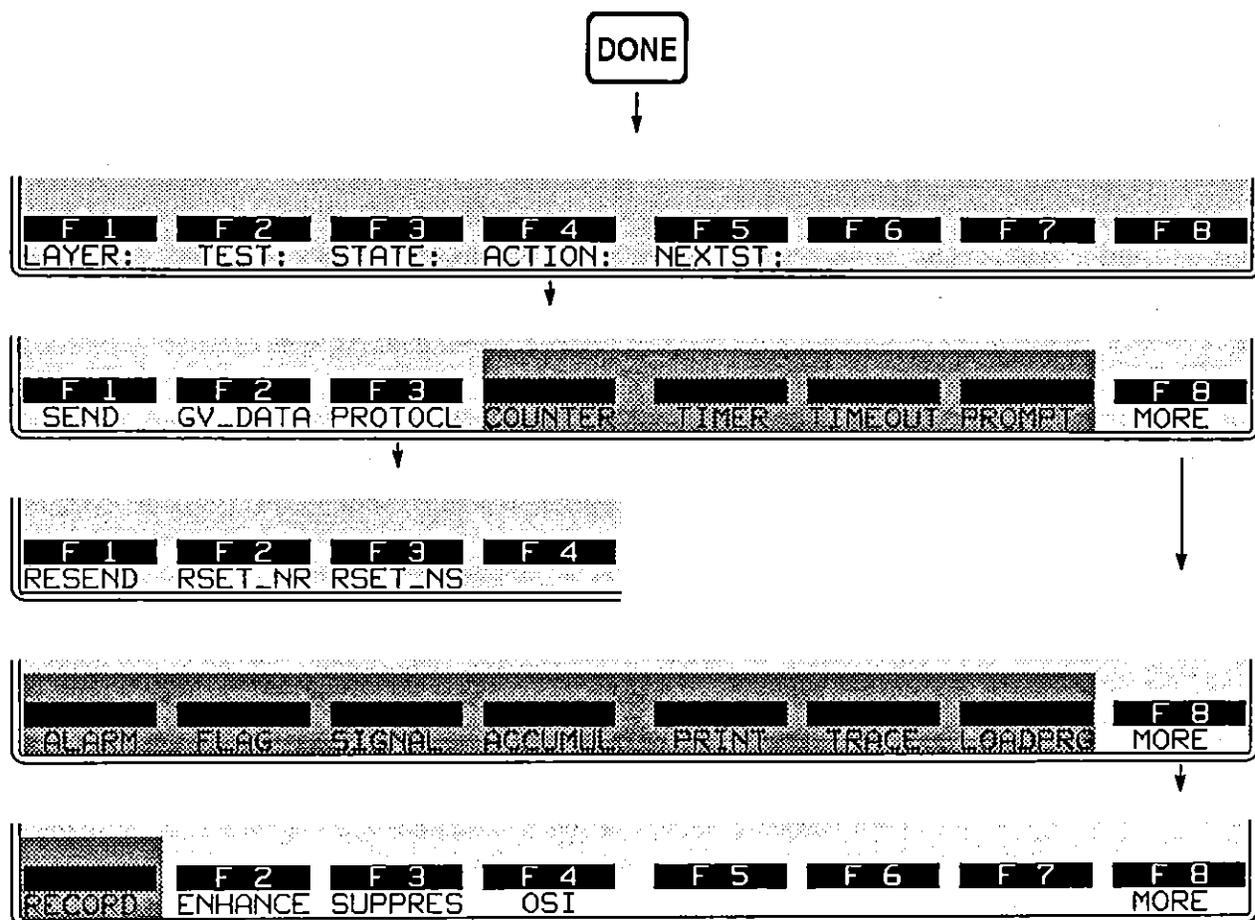


Figure 38-18 Action softkeys specific to SDLC.

(A) Send Actions

Press the softkey for **SEND** to access three racks of softkeys with names of frame types that may be named in **SEND** actions. All data generated by the SDLC package must be enclosed in a frame that is identified in a **SEND** action by *type*.

(Only at Layer 1 can data be generated as a simple character string without any protocol building blocks.) The complete set of frame types is given in Figure 38-19.

When conditions are true for a SEND action, frames are sent immediately down to Layer 1 to be transmitted there.

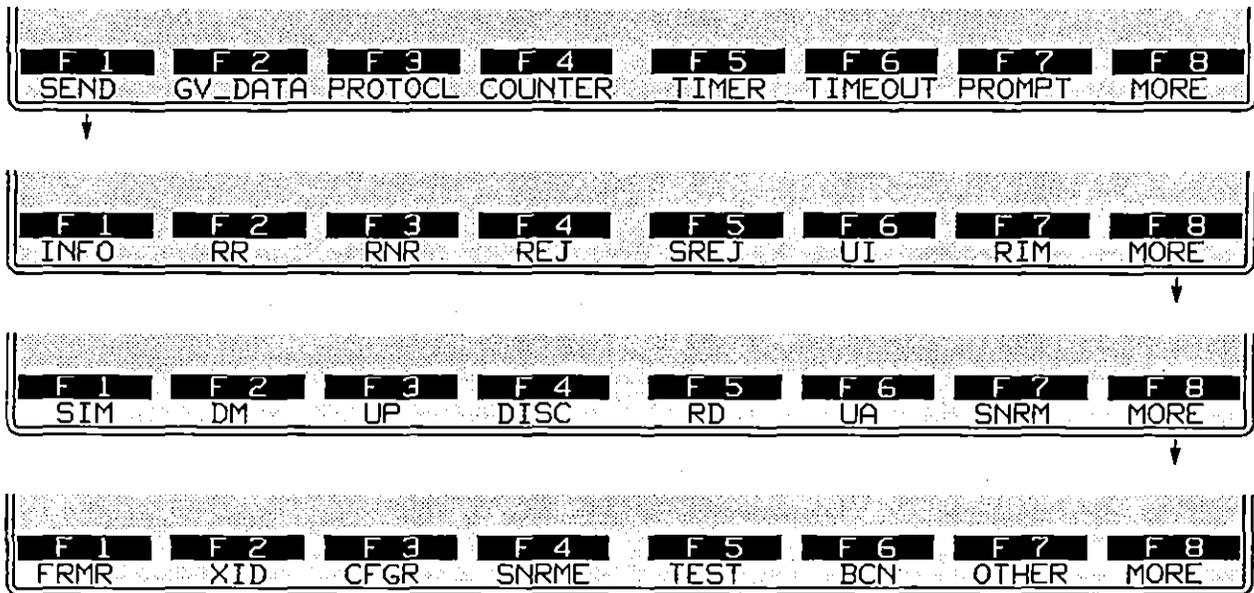


Figure 38-19 SEND actions always specify a frame type.

1. *INFO frames.* SEND INFO is a complete action-entry. Poll-bit, N(R), N(S), string, and BCC parameters may be added to an INFO frame, but they are optional. Although not strictly required, you should enter an address.

SEND INFO actions pass the INFO frame immediately to the next layer down. If the retransmit window is full, the frame is still sent—but it is not buffered in the window and can not be *resent*.

An INFO frame will be buffered for retransmission regardless of the status of the window if a specific value is entered for the NS= parameter (see "N(S)," below). The specific N(S) value will clear the window and the INFO frame will be buffered in the first window position.

2. *Unnumbered frames.* SEND SNRM, SEND UA, and so forth, are complete action-entries. P/F-bit, string, and BCC parameters may be added to the SEND action, but they are optional. Although not strictly required, you should enter an address.

3. *Supervisory frames.* SEND RR, SEND RNR, SEND REJ, and SEND SREJ are complete action-entries. P/F-bit, string, and BCC parameters may be added to the SEND action, but they are optional. Although not strictly required, you should enter an address.

The address in SDLC always belongs to the secondary.

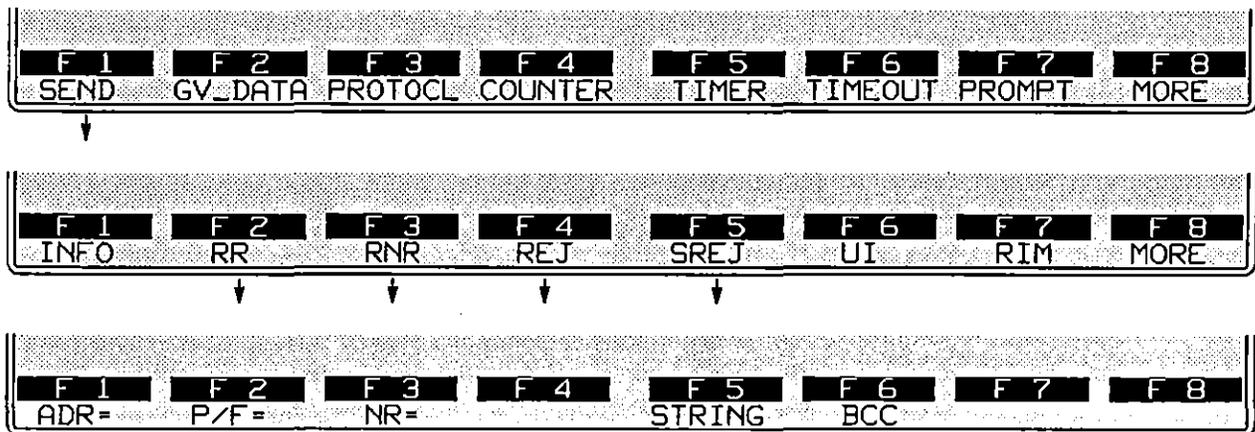


Figure 38-20 Address, P/F, N(R), and block-check parameters as well as data strings may be added to SEND SUPERVISORY actions.

4. *Other frames.* Any frame type may be entered in a SEND action as a hexadecimal value instead of by name. Press the softkey for OTHER, on the bottom rack in Figure 38-19. Enter the hex value in the form of two alphanumerics. Here is a DISconnect command entered as a SEND OTHER action:

ACTIONS: SEND OTHER 43 ADR= C3

Since P/F, N(R), and N(S) fields are implied already in the user-entered hexadecimal control field, BCC is a *valid* optional parameters in a SEND OTHER action. (In MOD 128, P/F is not included in the hex entry and is a valid optional entry.) Although not strictly required, you should enter an address.

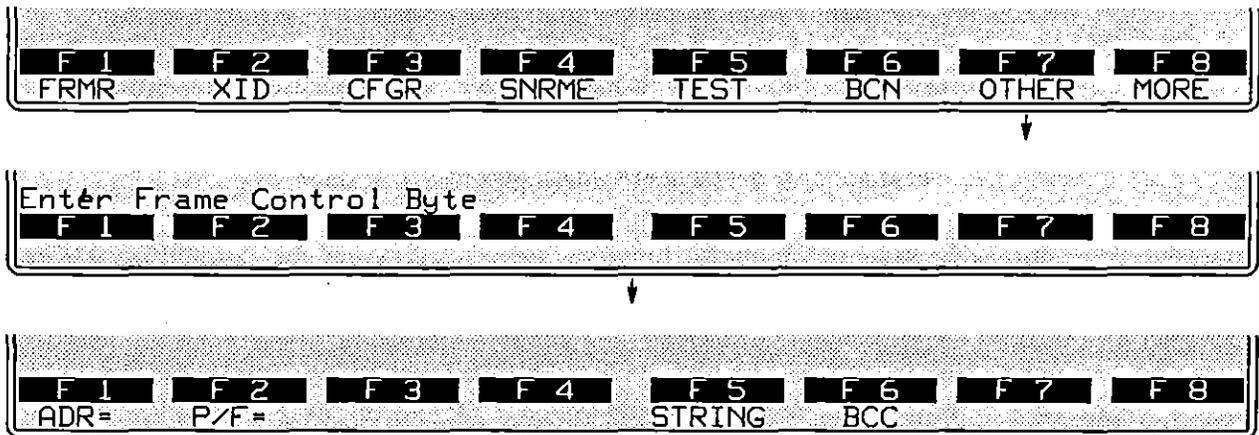


Figure 38-21 *SEND OTHER* actions always specify a *type* value in hex.

5. *Address*. Although not strictly required, an address should be specified for all frame types. The ADR= entry is followed by the hexadecimal address octet typed as two alphanumeric characters. The address field ζ , for example, appears as follows:

ACTIONS: SEND RR ADR= C3

There is also a LOOPBAK softkey selection for the address field. This selection causes the SEND action to refer to the address of the most recently received frame.

If you selected **Emulation Addressing: MULTI-DROP** on the SDLC Frame Level Setup screen, you also should have entered controller addresses in the table immediately below. (See Section 38.1.) The INTERVIEW tracks N(R) and N(S) for each address on the table. It can send frames to any of these addresses, but only if you complete the ADR= entry. Then, the INTERVIEW will send the appropriate frame from that address' transmit window. If you bypass the ADR= softkey selection or specify an address which does not appear in the table, the INTERVIEW will not send to any address.

When **Emulation Addressing: POINT-TO-POINT** is the selection, you should still enter an address. The INTERVIEW will not necessarily default to the appropriate address.

6. *Poll/final bit*. The P/F bit is an optional entry in all SEND actions. A P/F value of LOOPBAK, 0, or 1 are entered by the softkeys in Figure 38-22. If P/F= LOOPBACK, the bit will echo the last P/F bit received. (Looping the P/F bit is appropriate for UAs and supervisory frames.)

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
ADR=	P/F=	NR=	NS=	STRING	BCC		

↓

Enter P/F Value:							
F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
LOOPBAK	P/F=0	P/F=1					

Figure 38-22 A P/F value is optional in all *SEND* entries.

7. *N(R)*. *N(R)* fields are transmitted in INFO and supervisory frames.

To specify an *N(R)* value, press the softkey for NR= (see Figure 38-23). Enter a hexadecimal value written as one or two alphanumeric digits. For example, an entry that represented the highest valid *N(R)* in MOD 8 would be NR= 7. The highest valid entry in MOD 128 would be NR= 7F.

Other *N(R)* options are ACK_NS, LAST_NR, and AUTO. (See Figure 38-23.) ACK_NS means that your *N(R)* will acknowledge (that is, it will be *one higher than*) the last *N(S)* value you received *for the same frame address*. Normally this will be the correct *N(R)*, except in cases where the last *N(S)* received was erroneous. The NR= ACK_NS selection allows you to overlook *N(S)* errors.

LAST_NR means that you simply repeat the last *N(R)* you sent to the same address. Normally this is the correct *N(R)* following a bad *N(S)*. The NR= LAST_NR option allows you to force the other side to initiate recovery.

AUTO means that you will behave as a normal SDLC station, acking valid *N(S)* values and repeating your last *N(R)* whenever an invalid *N(S)* is received. AUTO is the default *N(R)* selection in SEND INFO, SEND RR, SEND REJ, and SEND SREJ actions.

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
ADR=	P/F=	NR=	NS=	STRING	BCC		

↓

Enter NR Value:							
F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
ACK_NS	LAST_NR	AUTO					

Figure 38-23 The *N(R)* field may be specified in INFO and supervisory frames to be sent.

8. *N(S)*. *N(S)* fields are transmitted in INFO frames only. (See the frame-field diagrams in Figure 38-5.) Entries for *N(S)* in SEND INFO actions are optional. The softkeys that open below NS= are illustrated in Figure 38-24.

To specify an *N(S)* value, press the softkey for NS=, then enter a hexadecimal in the form of one or two alphanumerics. Valid hex entries are the same as for *N(R)*. A SEND INFO action that *specifies* an *N(S)* value—NS= 0, for example—will clear the window so that the INFO frame is passed immediately to Layer 1.

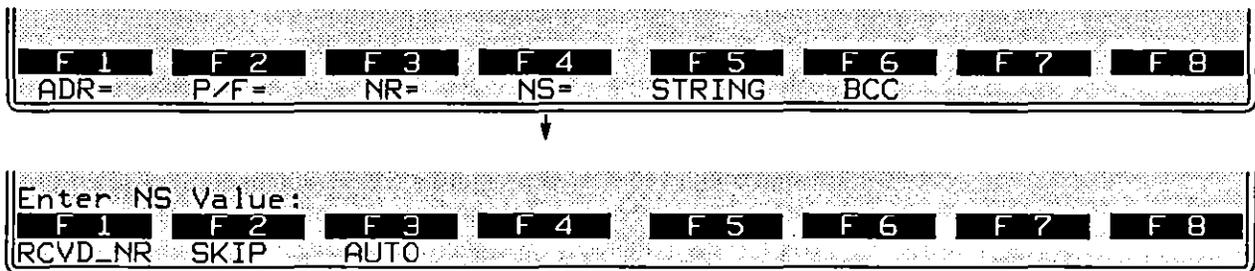


Figure 38-24 The *N(S)* field may be specified in a SEND INFO action.

Other *N(S)* options are RCVD_NR, SKIP, and AUTO. RCVD_NR means that you send the *N(S)* value that the other side says it is expecting. This is the valid *N(S)* in many cases, but not when you send two or more I-frames in a row without waiting for acknowledgment.

SKIP means that you *add one* to your correct *N(S)*. This will look to the other side as though the line has taken a "hit" and a frame has been lost. This selection causes the window to be cleared.

NS= AUTO is the default setting for SEND INFO actions. AUTO means that every *new* INFO frame sent will have an *N(S)* value of one higher than the previous *N(S)* to the same frame address.

9. *String*. Strings are sent at Layer 2 only as adjuncts to frame-types when they are named in SEND actions. If you want to send a string of raw data without a protocol "envelope," you must go to Layer 1 and send the raw string from there.

Press the SEND softkey followed by the softkey for a frame type. Add any necessary or desired SEND options for the particular frame type. Then press the STRING softkey (Figure 38-24).

There is no spreadsheet keyword that identifies send-strings at any layer. *The spreadsheet compiler identifies strings by the quotation marks surrounding them.* Always enclose strings in quotation marks. (To send an actual " -character in your string, type \".)

in the section, OSI Primitives on the Protocol Spreadsheet.) In an emulate mode, data is delivered up to Layer 3 only by one of two actions at Layer 2: GIVE_DATA, or else a DL_DATA IND primitive followed by the data string.

(C) Resend

The RESEND function is mapped to **F1** on the second layer of action softkeys at Layer 2 for SDLC. See Figure 38-26. A RESEND action will resend the first frame in the window. The window is a queue that buffers INFO frames for retransmission in case one or more transmissions are lost or in error.

The first frame in the window always is the *earliest outstanding* (unacknowledged) frame. Every time an acknowledgment is received, the window is cleared to the extent of the acknowledgment and a new "first-frame" position is established. The first RESEND after an acknowledgment always sends the first window frame.

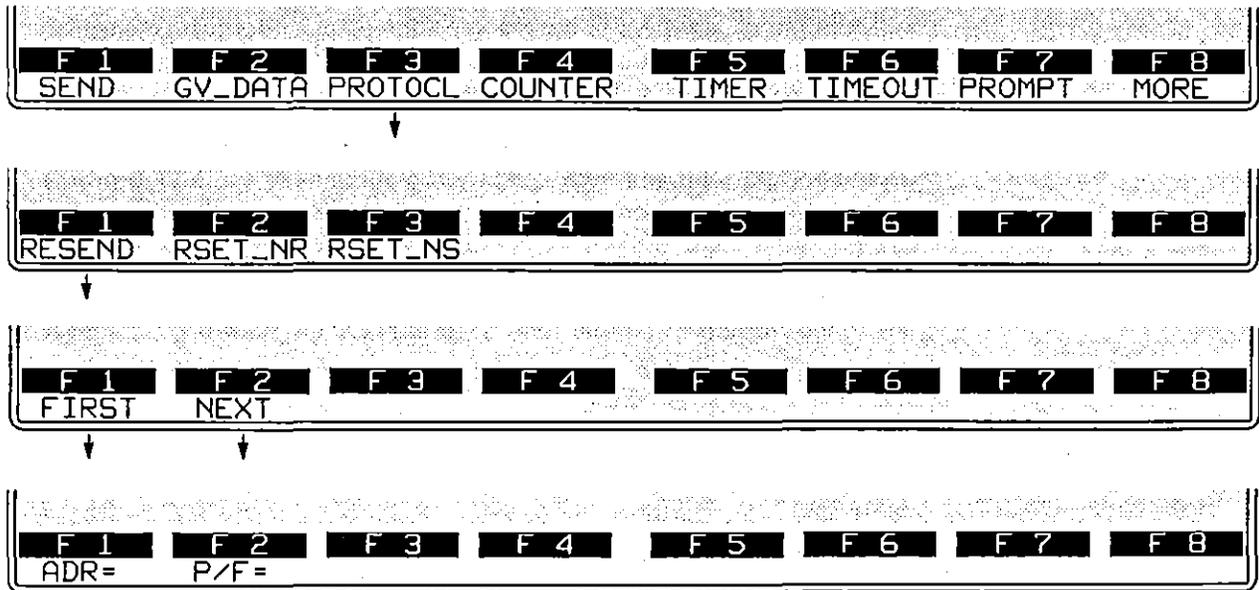


Figure 38-26 The RESEND action allows you to recover from sequence errors.

The *second and subsequent* RESENDS following an acknowledgment also will send the first window frame, provided that the keyword FIRST is appended directly to the RESEND entry. Otherwise, they send the NEXT (second) and subsequent window frames. Figure 38-27 shows the position of the the resend "pointer" after four consecutive RESEND actions. RESEND NEXT is the default resend when neither FIRST nor NEXT is specified.

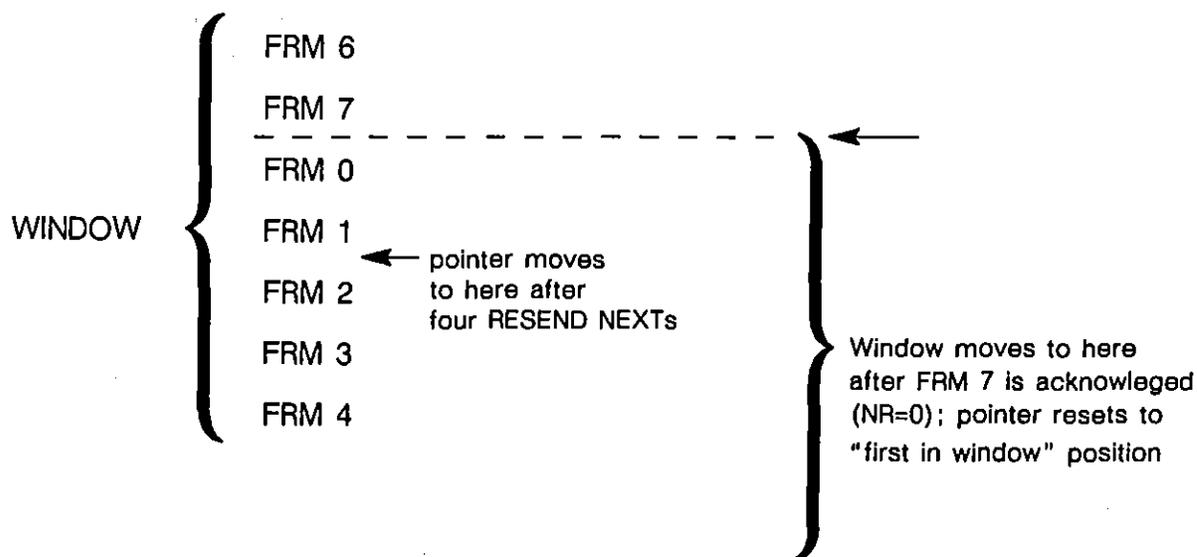


Figure 38-27 Resends cause the pointer to move, while acknowledgments move the pointer and the entire window.

The resend-pointer is reset to the beginning of the window automatically by any acknowledgment, or by a RESEND FIRST action in the spreadsheet program.

1. *Resend first/next.* RESEND FIRST means that the resend-pointer is reset to the beginning of the window, the first frame in the window is resent, and the pointer is advanced to the second position in the window. The effect of a RESEND FIRST action is illustrated in Figure 38-28.

The RESEND FIRST action makes it possible for you to resend all the frames in the window one by one, and then *resend them again* if necessary.

2. *ADR=.* Although not strictly required, you should designate the controller address of the resend frame. Two-digit hexadecimal values in the range 00 through FF are valid.

There is also a LOOPBAK softkey selection for the address field. This selection causes the RESEND action to refer to the address of the most recently received frame.

If you selected **Emulation Addressing: MULTI-DROP** on the SDLC Frame Level Setup screen, you also should have entered controller addresses in the table immediately below. (See Section 38.1.) The INTERVIEW tracks N(R) and N(S) for each address on the table. It can resend frames to any of these addresses, but only if you complete the ADR= entry. Then, the INTERVIEW will resend the appropriate frame from that address' transmit window. If you bypass the ADR= softkey selection or specify an address which does not appear in the table, the INTERVIEW will not resend to any address.

When Emulation Addressing: ~~POINT-TO-POINT~~ is the selection, you should still enter an address. The INTERVIEW will not necessarily default to the appropriate address.

3. $P/F=$. The P/F bit in the resend-frame can be set to zero or one by this optional action. (Default is 1 in a RESEND action.)

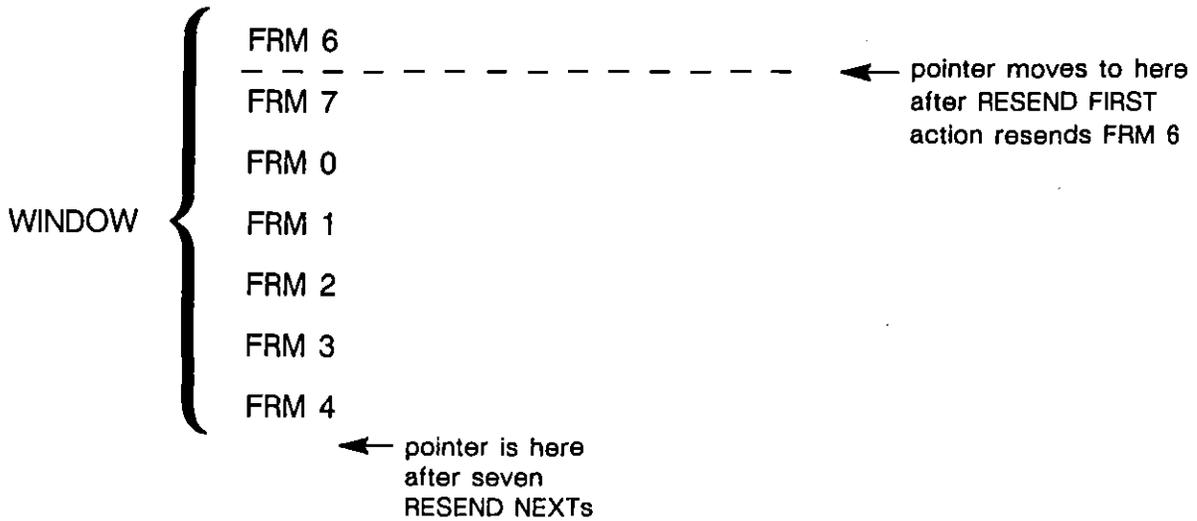


Figure 38-28 RESEND FIRST resets the pointer, allowing you to resend the entire window repeatedly.

(D) Reset N(R) and Reset N(S)

RESET_NR and RESET_NS are the **F2** and **F3** actions on the second rack of action softkeys for SDLC. (Refer to Figure 38-26.) The sequence-number fields in I-frames and supervisory frames can be reset by these two Protocol Spreadsheet actions. Sequence numbers are not reset *automatically* during a test by any frame that is sent or received.

RESET_NS also clears the transmit window.

If you press RSET_NR or RSET_NS, another softkey rack will be presented.

1. $ADR=$. Although not strictly required, you should designate a specific controller address. Two-digit hexadecimal values in the range 00 through FF are valid.

There is also a LOOPBAK softkey selection for the address field. This selection causes the RESET_NR and RESET_NS actions to refer to the address of the most recently received frame.

If you selected **Emulation Addressing: MULTI-DROP** on the SDLC Frame Level Setup screen, you also should have entered controller addresses in the table immediately below. (See Section 38.1.) The INTERVIEW tracks N(R) and N(S) for each address on the table. It can reset N(R) or N(S) for any of these addresses, but only if you complete the ADR= entry. Then, the INTERVIEW will reset N(R)—or N(S)—for that address only. If you bypass the ADR= softkey selection or specify an address which does not appear in the table, the INTERVIEW will not reset N(R)—or N(S)—for any address.

When **Emulation Addressing: POINT-TO-POINT** is the selection, you should still enter an address. The INTERVIEW will not necessarily default to the appropriate address.

38.6 Display Actions

ENHANCE and SUPPRESS pertain to lines of data on the Layer 2 protocol trace (see Section 38.2). They do not suppress and enhance data on the raw-data display. Raw data is enhanced and suppressed at Layer 1.

DTE, DCE, and RCV conditions can trigger an ENHANCE or SUPPRESS action. These conditions are active when the INTERVIEW is in monitor mode or in either of the emulate modes.

(A) Enhance

Whenever a DTE, DCE, or RCV condition comes true at Layer 2, the frame that satisfied the condition can be enhanced on the SDLC protocol-trace display, or it can be deleted from the trace completely. In an actions block on the Protocol Spreadsheet, press the ENHANCE softkey—**F2**—on the third rack of action softkeys. Figure 38-29 shows the three softkey subselections beneath ENHANCE. They are REVERSE, BLINK, and LOW.

Reverse-image and blink enhancements affect the plasma-display screen. In addition, a low-intensity enhancement can be applied to screens that are transmitted to a black-and-white monitor connected at the RS-170 port at the rear of the INTERVIEW.

Reverse, blink, and low enhancements can be mapped to colors on a color monitor attached at the INTERVIEW's RGB port (Figure 1-6). See Section 17.2 for an explanation of how reverse, blink, and low enhancements relate to character and background colors in the RGB output.

Figure 38-30 shows one screen of a Layer 2 protocol trace in which DTE SNRM frames have been enhanced in reverse video. The trigger that caused this enhancement was as follows:

CONDITIONS: DTE SNRM
ACTIONS: ENHANCE REVERSE

(B) Suppress

Individual frames that are suppressed in Layer 2 actions are deleted from the trace display. Figure 38-29 shows the softkey path to SUPPRES.

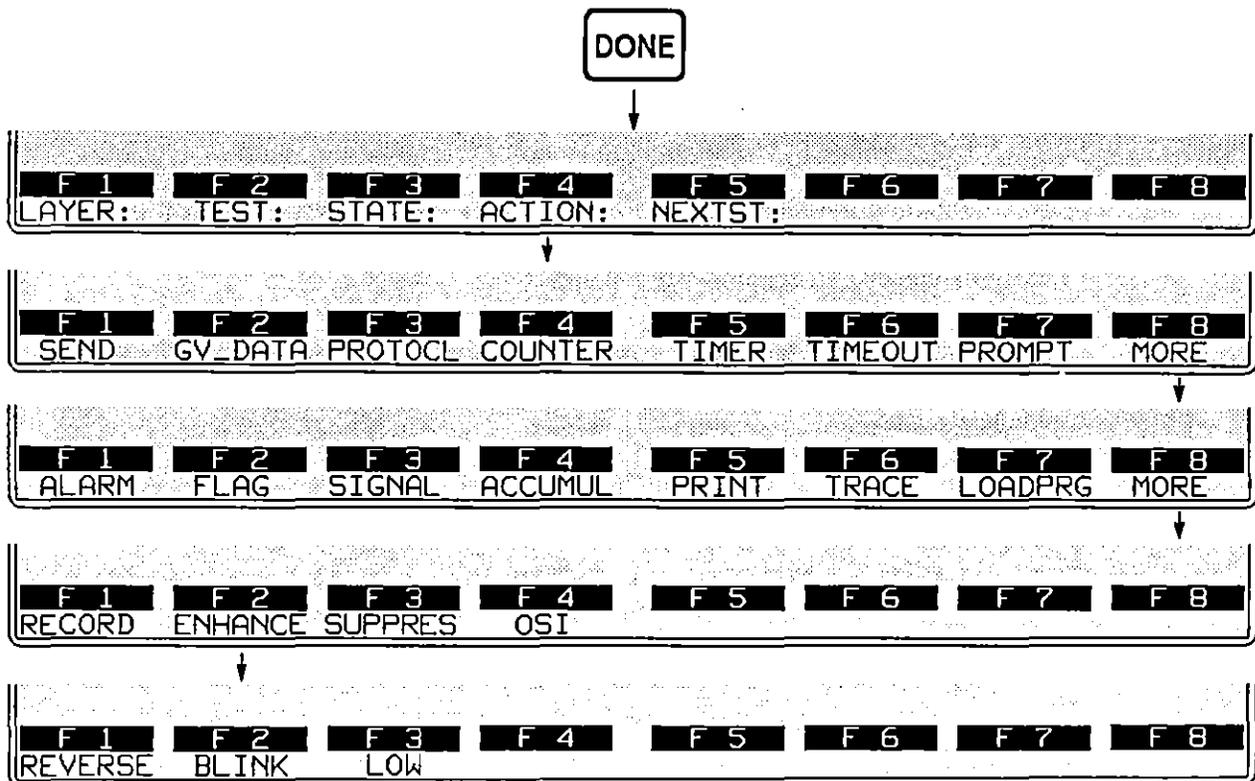


Figure 38-29 Selected frames on the protocol trace may be enhanced (or suppressed).

MON/LINE		BLK=00000 S 04/21/89 22:22						
EBCDIC/B/NONE/BOP								
SRC	ADDR	TYPE	Nr	Ns	P/F	SIZE	TIME	BCC
DTE	C1	INFO	7	3	0	0264	2220:06.206	0
DTE	C2	SNRM			1	0002	2220:06.230	0
DTE	C1	INFO	7	0	0	0011	2220:06.276	0
DTE	C1	INFO	7	1	0	0046	2220:06.441	0
DTE	C1	INFO	7	2	0	0236	2220:07.247	0
DTE	C1	INFO	7	3	0	0264	2220:08.151	0
DTE	C1	INFO	7	4	0	0264	2220:09.059	0
DTE	C1	INFO	7	5	0	0264	2220:09.976	0
DTE	C1	INFO	7	6	0	0136	2220:10.445	0
DTE	C1	RR	7		1	0002	2220:10.466	0
DCE	C1	INFO	7	7	1	0012	2220:10.788	0
DTE	C1	RR	0		1	0002	2220:10.812	0
DTE	C1	INFO	0	7	0	0015	2220:10.872	0
DTE	C1	INFO	0	0	0	0041	2220:11.021	0
DCE	C1	INFO	7	0	1	0011	2220:11.115	0

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
DATA	L2TRACE			STATS		NO DISP	

Figure 38-30 Set Normal Response Mode (SNRM) commands have been enhanced on the DTE side.

38.7 Automatic Primitives

A table in a previous section (Table 33-2) listed the OSI service primitives that are monitored at the boundaries of Layer 2 as trigger conditions and sent up to Layer 3 or down to Layer 1 as user-entered spreadsheet actions. These primitives are layer-specific rather than protocol-specific and are not part of the personality package for SDLC; but a few of the primitives are set in motion automatically by SDLC spreadsheet actions at Layer 2. These *automatic* primitives can be thought of as part of the Layer 2 actions themselves, and by extension as part of the SDLC protocol package.

Table 38-1 gives the set of SDLC actions that have action-primitives built into them. For example, whenever a GIVE_DATA action occurs at Layer 2, a DL_DATA IND primitive is forwarded to Layer 3, where a DL_DATA IND condition may be waiting to monitor it.

Whenever a SEND or RESEND action is initiated at Layer 2, a PH_DATA REQ primitive is sent downward along with the PH data (the entire frame).

If a SEND or RESEND action is triggered at Layer 2 while the physical connection at Layer 1 is inactive, Layer 2 will sense the absence of a physical connection and delay the PH_DATA REQ. Instead it will send a PH_ACTIVATE REQ primitive. Only when a PH_ACTIVATE CONF has been returned by Layer 1 will Layer 2 release the data and the data primitive.

NOTE: The RS-232 interface does not distinguish active/inactive status at the physical level. This interface returns PH_ACTIVATE CONF automatically whenever it sees PH_ACTIVATE REQ.

Table 38-1
Automatic Primitives Generated at Layer 2 SDLC

SDLC Layer 2 Action	Automatic Primitive	To Layer
GIVE_DATA	DL_DATA IND	3
SEND {TYPE}	(PH_ACTIVATE REQ*) PH_DATA REQ	1 1
RESEND	(PH_ACTIVATE REQ*) PH_DATA REQ	1 1

* Sent if Layer 1 shows inactive status. PH_DATA REQ delayed until PH_ACTIVATE CONF returned by Layer 1.