AX.25 Data Link State Machine

Eric L. Scace K3NA
10701 Five Forks Road
Frederick MD 21701
USA

0. Summary

This paper is part of a series of papers which provide extended finite state machine representations for AX.25 and related protocols. The state machines are depicted using state description language (SDL) graphic conventions from the 2.100 series of Recommendations developed by the International Telegraph and Telephone Consultative Committee (CCITT) of the International Telecommunications Union (ITU). An extended finite state machine representation of a communications protocol such as AX.25 avoids the ambiguities associated with prose descriptions. These descriptions also compel the protocol designer to confront many of the error scenarios which arise on a communications path, and simplify the implementor's task of producing correct solutions which will interwork with solutions created by others.

This particular paper describes an extended finite state machine which executes the data link procedures between two stations.

1. Status of Proposal

The data link SDL description here is a draft. It borrows heavily from the SDL description of LAPD, developed by the CCITT during 1983-1988 and found in Recommendation Q.921. The data link SDL description was validated by conducting a complete, paragraph-by-paragraph review of the AX.25 prose description and by including annotated cross-references the original working draft. [For simplicity, cross references and other working annotations have not been carried forward into this paper.] The ARRL Digital Committee working group has reviewed this description in detail, and intends to include this machine as an Annex of the upcoming publication of AX.25 Revision 2.1.

Since the working group's review this past July, I have taken the liberty of removing the capability to transmit FRMR frames. This was done for five reasons:

a) According to the present prose description of AX.25, UI frame transmission and reception are prohibited when a station is undergoing FRMR recovery. This seemed to be an unnecessary encumberence to stations which use both connection-oriented and connections data transfer capabilities.

b) During FRMR recovery, the link can not be re-established by the station which has sent the FRMR and is awaiting a mode-setting command.

c) Whilst neither of the preceeding two items are fatal flaws, they do introduce delays in the transfer of information while recovering from very rare error conditions. Recovery from these infrequent errors can be done more simply by just resetting the link with an SABM + UA exchange.

d) An implementation which does not *send* FRMR, but which still receives *and processses* FRMRs, is fully compatible with existing AX.25 Revision 2.0 implementations.

e) Both the SDL description and implementation are simplified. In the SDL description, for example, an entire state is eliminated (plus other ancillary simplifications are made in other states).

Nevertheless, the SDL description to support transmission of FRMR is available, and can be restored to the overall data link SDL machine if desired

The entire data link SDL machine description is still considered to be in a draft state. You are invited to review and comment on this material. Comments are desired so that the final publication is as useful as possible to its readers.

2. Location in the Overall Model

The data link SDL machine forms the heart of the overall AX.25 system. The data link SDL machine resides within the data link layer of the Open Systems Interconnection Reference Model. This SDL machine interacts with the AX.25 user above (through the segmentor SDL machine), and with the link multiplexor SDL machine below.

A data link SDL machine exists for each ongoing communication with a remote station. This communication may use connection-oriented information transfer (via I frames), connectionless information transfer (via UI frames), or both (simultaneously or alternately).

A summary of primitives (signals), flags, error codes, timers, and queues has been compiled on the first page of the SDL diagrams.

2.1 Interaction with the AX.25 User

The AX.25 user directs the operation of the data link SDL machine through the primitives (signals) described below. "DL" in some primitive names stands for "data link".

The DL Establish Request primitive is used by the AX.25 user to request establishment of an AX.25 connection. When the connection has been established (SABM + UA exchanged), the data link SDL machine informs the AX.25 user with a **DL Establish Confirm** primitive. The data link SDL machine uses the **DL Establish Indication** primitive to inform the AX.25 user that connection establishment has occured (triggered by receipt of an SABM from the remote station), and also to inform the AX.25 user that a link reset has occurred.

The DL Unit Data Request primitive can be used by the AX.25 user at any time to transmit information in a connectionless manner via UI frames. The data link SDL machine will deliver any information received via UI frames through a **DL Unit Data Indication** primitive to the AX.25 user.

The AX.25 user submits a **DL Data Request** primitive to the data link SDL machine to cause information to be transmitted in a connection-oriented manner via I frames. The data link SDL machine employs the **DL Data Indication** primitive to deliver information received via I frames.

The Set Own Receiver Busy and **Clear Own Receiver Busy** primitives are sent by the AX.25 user to cause the data link SDL machine to temporarily suspend, and then resume, the flow of incoming information. These primitives affect connection-oriented procedures only. Connectionless information transfer (via UI frames) can not be suspended; if the AX.25 user is not prepared to accept incoming connectionless information, it must discard the information itself.

The data link SDL machine uses the **DL Error Indication** primitive to inform the AX.25 user when frames have been received which are inconsistent with the protocol definition. Error codes are provided with each DL Error Indication to explain the exact type of protocol error which occured.

2.2 Passing Primitives through the Segmentor

The segmentor SDL machine (described elsewheres) passes all primitives through transparently, except for DL Unit Data Request, DL Unit Data Indication, DL Data Request, and DL Data Indication primitives.

The DL Unit Data Request and DL Data Request primitives, received from the AX.25 user, will be segmented into multipled Request primitives (and then sent down to the data link SDL machine) when the amount of data to be transferred exceeds the capabilities of the data link SDL machine (i.e., is greated than N1 octets).

A sequence of DL Unit Data Indication and DL Data Indication primitives, received from the data link SDL machine, will be assembled in a single Indication primitive (and then sent up to the AX.25 user) when the contents indicate that segmentation had been performed.

The details of the segmentor SDL machine are laid out in a separate companion paper. Unfortunately, due to the brief time between the Committee's last working group meeting and the publication deadline for these proceedings, the segmentor paper was not completed. I hope to have it available as a handout at the Conference.

2.3 Interaction with the Link Multiplexor

The data link SDL machine directs the operation of the link multiplexor SDL machine below. "LM" in some primitive names stands for "link multiplexor".

DM, UA, SABM, DISC, UI, I, RR, RNR, and **RE J** primitives are used to convey outgoing frames of these types from the data link SDL machine to the link multiplexor for transmission, and. (with the addition of **FRMR) to** convey incoming received frames of these types from the link multiplexor SDL machine to the data link SDL machine for protocol processing.

The data link SDL machine uses **LM Seize Request** when transmission of an acknowledgement is desired. When the lower layer link multiplexor SDL machine has obtained a transmission opportunity, the multiplexor provides the **LM Seize Confirm** primitive. The data link SDL machine will then generate the correct acknowledgement (e.g., RR, RNR, I frame), as appropriate for that moment, send it to the link multiplexor SDL machine in a primitive, and then conclude with **the LM Release Request** primitive.

3. Overview of States

Unlike the earlier state tables found at the end of the AX.25 Revision 2.0 description, the use of an extended finite state machine allows an effective description to be provided with far fewer states. There are only five states in the **data** link SDL

machine.

The Disconnected State (state 0) is the initial state. It is also the state entered when the DISC + **UA** protocol exchange occurs.

The Awaiting Connection State (state 1) is entered when the station transmits **a** SABM command and is waiting for the UA response. When the flag "layer 3 initiated" is set, the **SABM** was transmitted because the AX.25 user had requested connection establishment (or re-establishment); if the flag is clear, the SABM was transmitted as part of an internal data link resetting procedure.

The Connected State (state 3) is reached when the **SABM** + UA protocol exchange has been completed. Transfer of information using **I** frames is now available. Flags are used to implement local and remote busy (Receiver Not Ready) conditions. **A** remote busy condition is periodically revalidated by interrogation of the remote station. An overall limit is imposed on the maximum number of times a remote station can signal busy without acknowledging additional outstanding I frames, to avoid "hangs" caused by infinite **RNRs.** During quiet periods on the link, the remote station is occasionally interrogated to ensure that both parties are still present.

The Timer Recovery State (state 4) is entered when a **T1** timeout has **occured** on a previously-transmitted I **frame,** and the remote station is being interrogated to determine if retransmission is required. Flags continue to implement local and remote busy conditions, as described above.

The Awaiting Release State (state 2) is entered when the station transmits the DISC command **and** is waiting for the UA response.

4. Internal Queue

An internal queue is used for information to be transmitted using the connection-oriented procedures (via I frames).

5. Timers

AX.25 timers **T1** and T3 are implemented with the data link SDL machine. **A** smoothed round trip time (from transmission of **a** frame to receipt of its acknowledgement) is also maintained and used for calculating the appropriate value of TI.

6. Other Parameters Associated with AX.25 Procedures

Present TNC implementations use a variety of parameters in their **AX.25** implementations. Not all of these parameters are found in the data link **SDL** machine. For convenience, many of the popular ones are listed alphabetically below along with their location within the overall family of SDL machines described in this series of papers.

AXDELAY -- time window allowed for an intervening repeater to start operating -- physical (same as **T104).**

AXHANG -- time window allowed for an intervening repeater to stop operating -- physical (same as **T100).**

BEACON -- timer which triggers periodic transmission of a UI **frame** containing a user-specified announcement -- **This** process is considered an **AX.25** user.

CHECK -- timer which triggers interrogation of a quiet AX.25 connection -- data link (same as T3).

DWAIT -- time window allowed for digipeaters to seize a simplex radio channel and begin digipeating -- physical (same as **T101).**

FRACK -- timer supervising the receipt of **acknowledgements** for outstanding I, SABM, and DISC frames -- data link (same as **T1).** Note that this timer has been improved by incorporating a smoothed round trip time calculation, and by setting it to successively larger multiples of the smoothed round trip time as multiple retries occur.

HID -- timer which triggers period transmission of a UI frame containing digipeater identification -- This process is considered an AX.25 user.

MAXFRAME -- maximum number of1 **frames** which can be sent before awaiting acknowledgement -- data link (same as the window size parameter **"k").**

MYALIAS -- **callsign** used by the digipeat function -- link multiplexor (implied in **the** address check).

MYCALL -- **callsign** used for frames sent and received -- link multiplexor (implied in the address check).

PACLEN -- maximum length of the information field of a UI or I frame -- data link (same as N1).

PACTIME -- a timer which triggers the transmission of a packet containing all untransmitted keyboard input received by the TNC to date -- This timer is considered part of an AX.25 user process.

RESPTIME -- time delay between receipt of a frame before transmission of the response; intended to prevent collision between an acknowledgement and subsequent I frames -- embedded in data link SDL machine, but not explicit. The data link SDL machine instead uses the LM Seize Request and LM Seize Confirm primitives to decide when the opportunity has come: to transmit an acknowledgement. This strategy results in improved acknowledgements, since the acknowledgement sent reflects the exact situation at the time of transmission (rather than the situation at the time that RESPTIME expired).

RETRY -- number of times retransmission is attempted -- data link (same as N2).

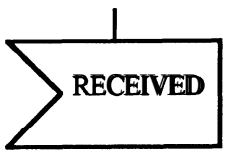
TXDELAY -- time window allowed for remote station to synchronize on incoming flag fill -- physical (same as T105).

UNPROTO -- triggers transmission of a UI frame -- This process is considered an AX.25 user.

DATA LINK

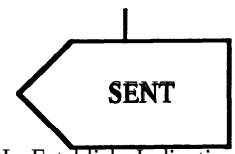
Summary of Primitives, States, Queues, Flags, Errors, and Timers

DL Primitives



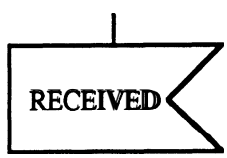
DL Establish Request

DL Release Request
DL Data Request
DL Unit Data Request
Set Own Receiver Busy
Clear Own Receiver Busy



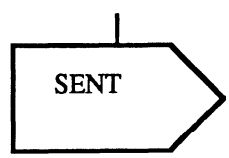
DL Establish Indication
DL Establish Confirm
DL Release Indication
DL Data Indication
DL Unit Data Indication
DL Error Indication

ML Primitives



LM Seize Confirm

DM
UA
SABM
DISC
RR
RNR
REJ
UI
FRMR
I



LM Seize Request
LM Release Request
DM
UA
SABM
DISC
RR
RNR
REJ
UI
I

States

0 -- disconnected.
1 -- awaiting connection.
2 -- awaiting release.
3 -- connected.
4 -- timer recovery

Queues

I frame queue -- queue of information to be transmitted in I-frames.

Error Codes

A -- F=1 received but P=1 not outstanding.
B -- Unexpected DM with F=1 in states 3, 4, and 5.
C -- Unexpected UA in states 3, 4, and 5.
D -- UA received without F=1 when SABM or DISC was sent with P=1.
E -- DM received in states 3, 4, or 5.
F -- Data link reset; i.e., SABM received in state 3 or 4.
J -- N(R) sequence error.
L -- control field invalid or not implemented.
M -- information field was received in a U- or S-type frame.
N -- length of frame incorrect for frame type.
O -- I-frame exceeded maximum allowed length.
P -- N(s) out of the window.
Q -- UI response received, or UI command with P=1 received.
R -- UI frame exceeded maximum allowed length.
S -- I response received.

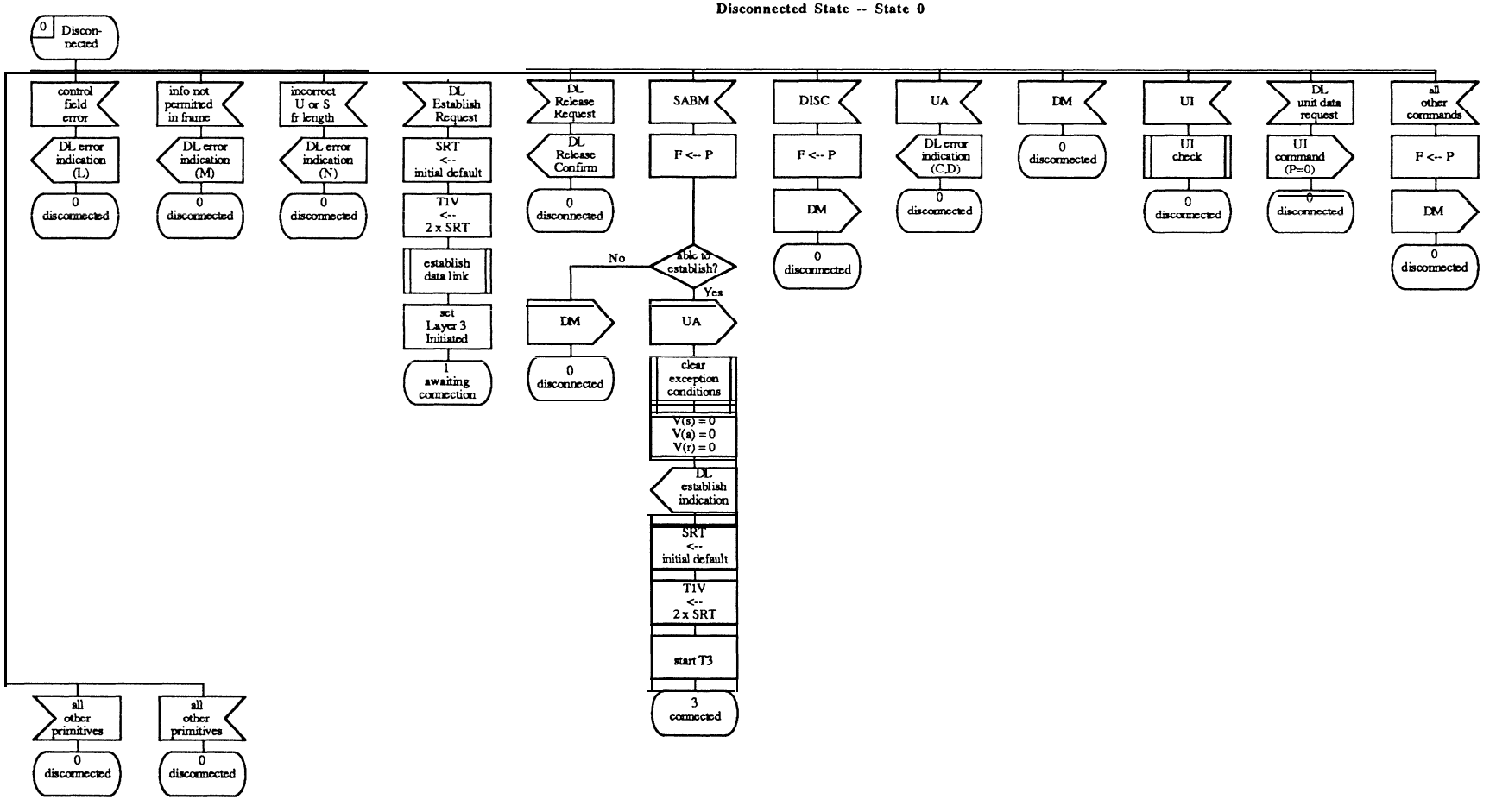
Flags

Layer 3 Initiated -- SABM was sent by request of Layer 3; i.e., DL-Establish-Request primitive.
Peer Receiver Busy -- remote station is busy and can not receive I-frames.
Own Receiver Busy -- Layer 3 is busy and can not receive I-frames.
Reject Exception -- a REJect frame has been sent to the remote station.
Acknowledge Pending -- I-frames have been successfully received but not yet acknowledged to the remote station.

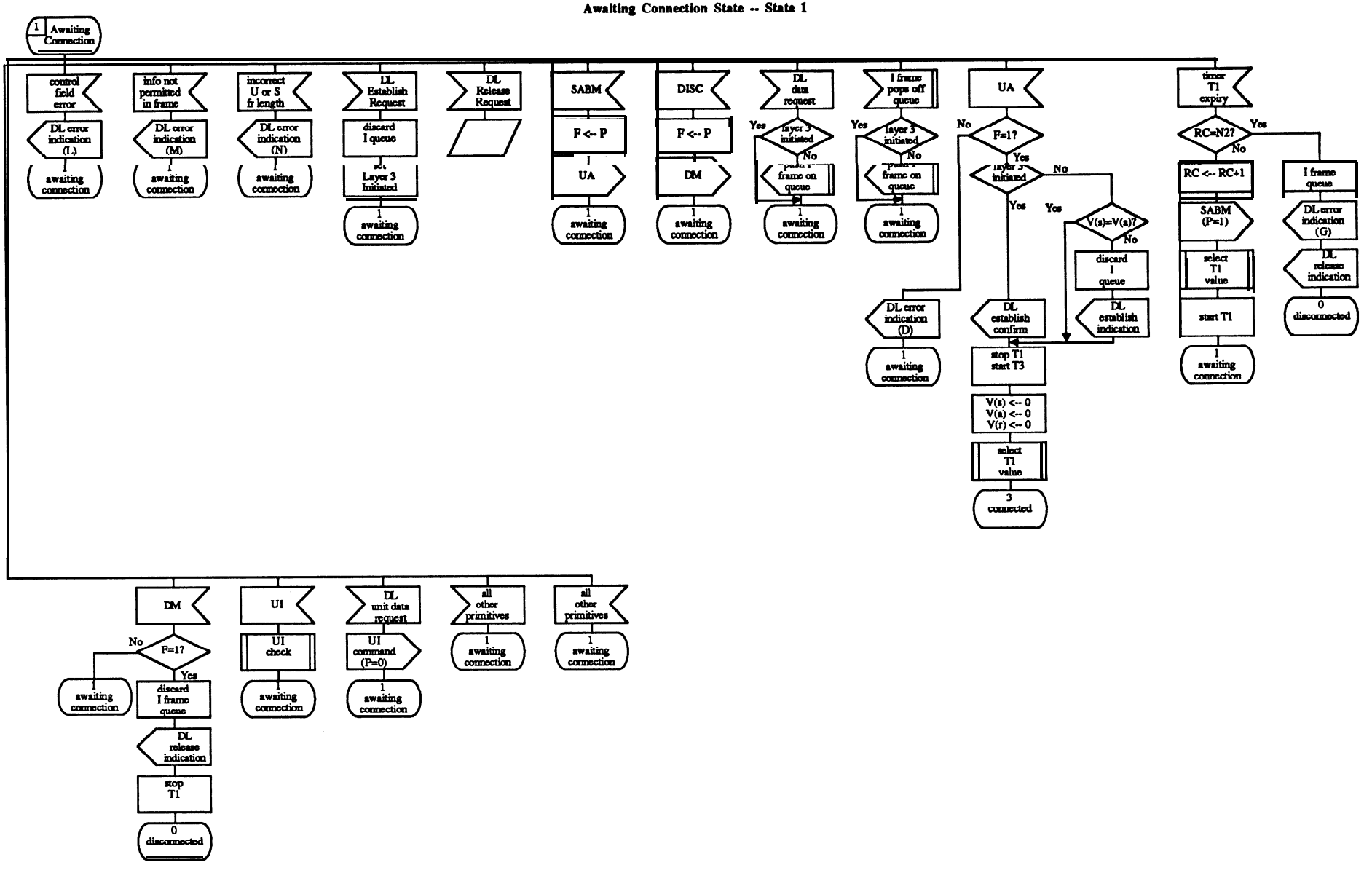
Timers

SRT -- smoothed round trip time.
T1V -- next T1 value; default initial value is initial value of SRT.
T1 -- outstanding I-frame or P-bit.
T3 -- idle supervision (keep alive).

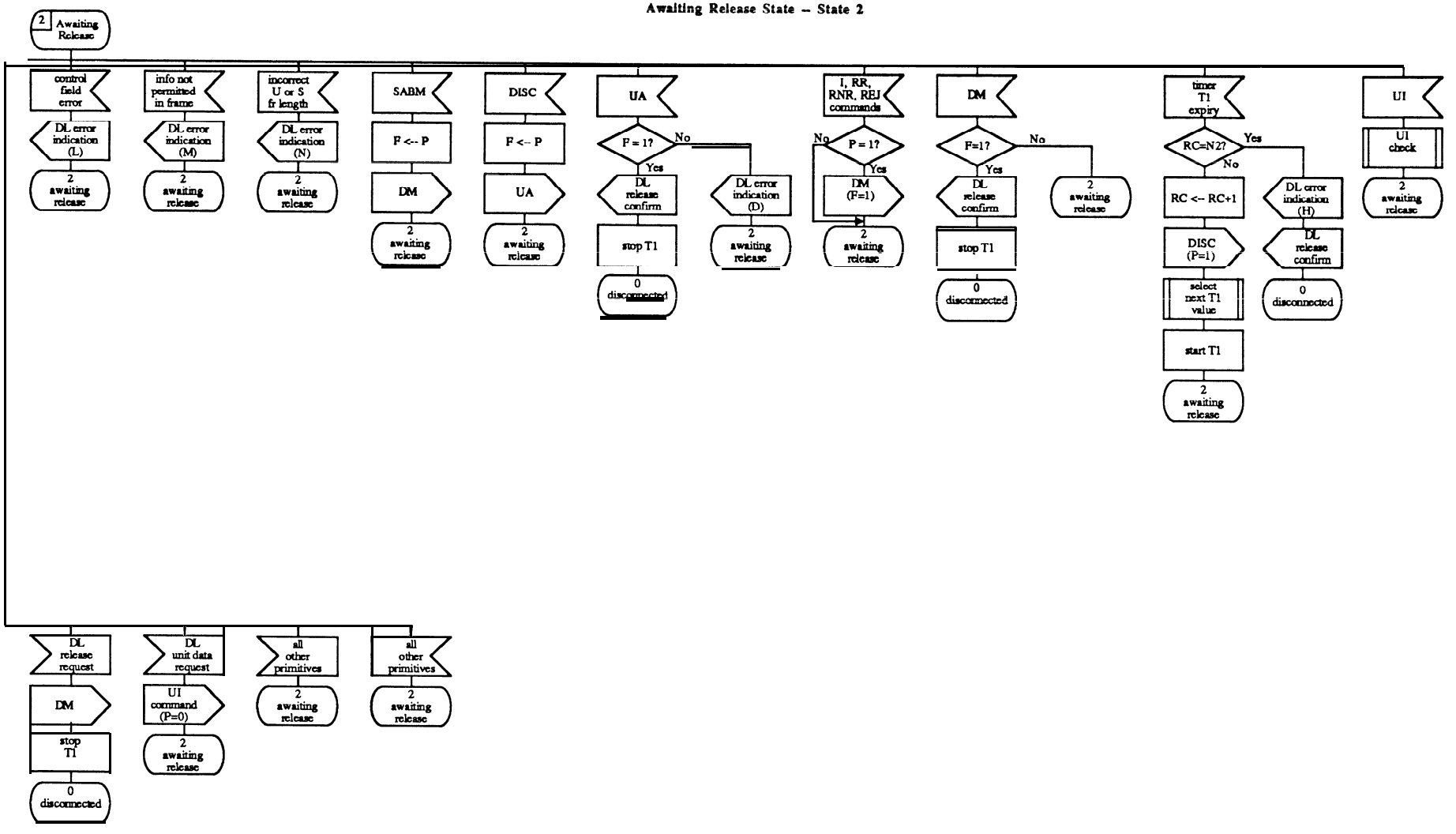
Data Link
Disconnected State -- State 0



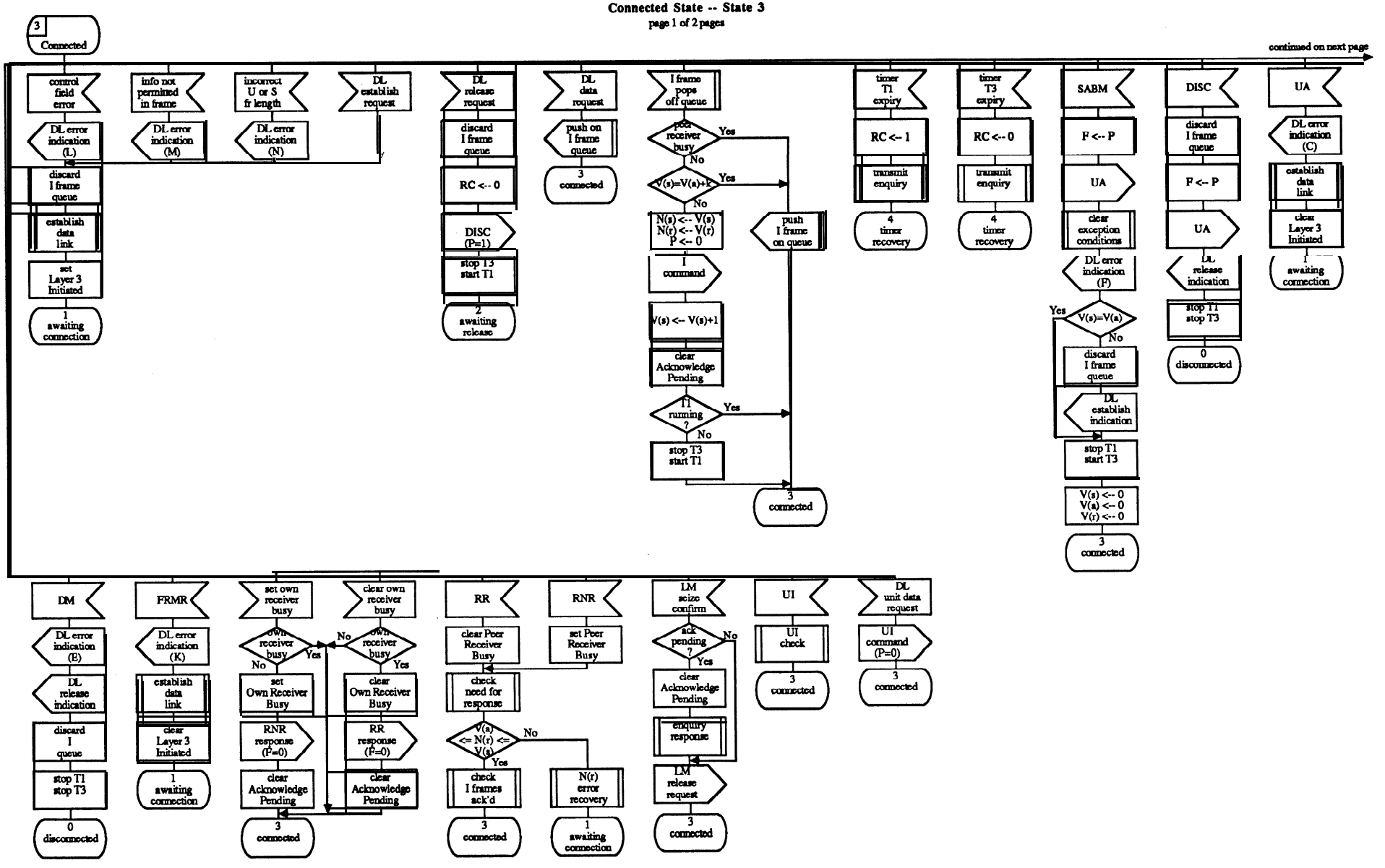
Data Link
Awaiting Connection State -- State 1

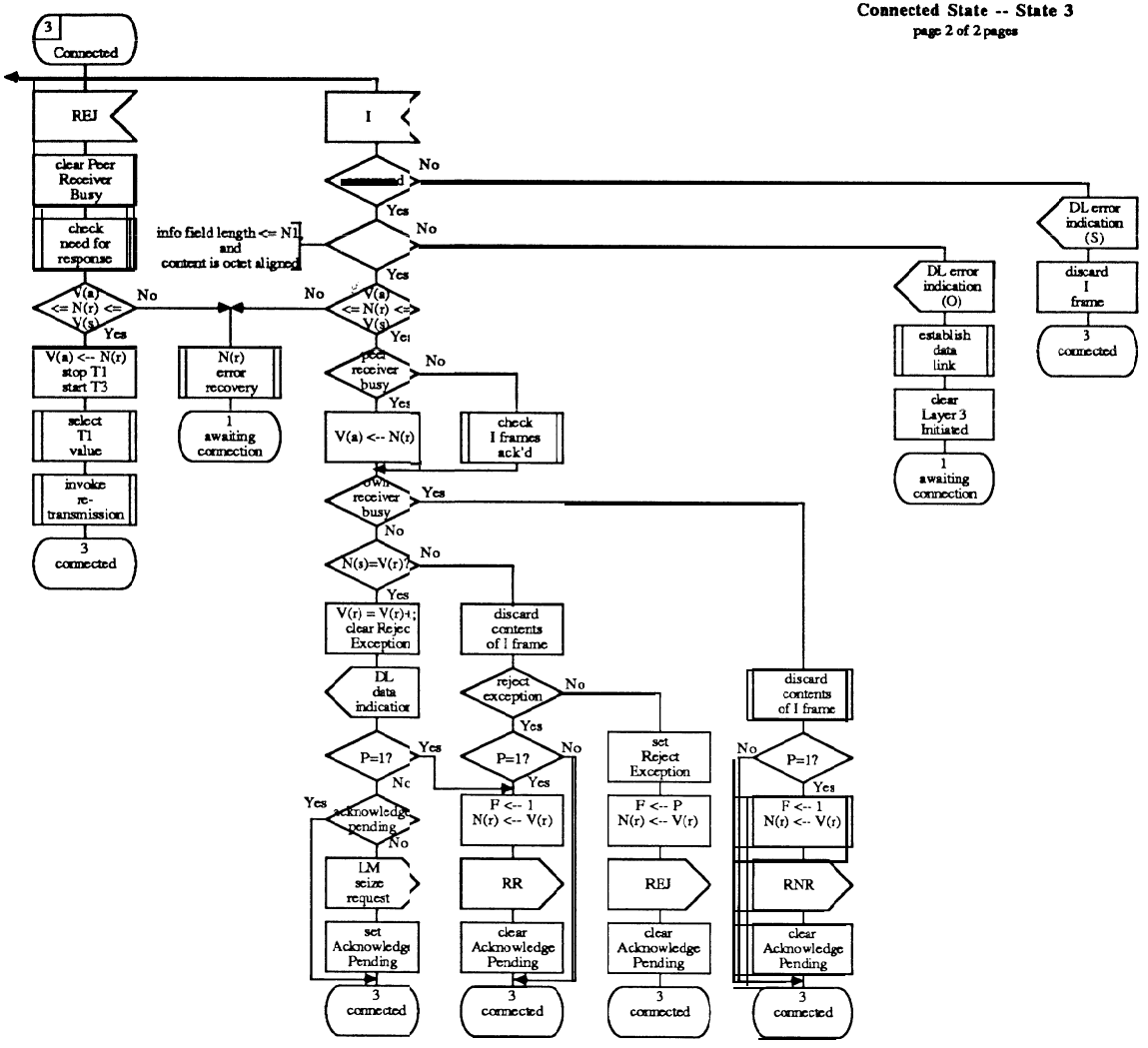


Data Link
Awaiting Release State -- State 2

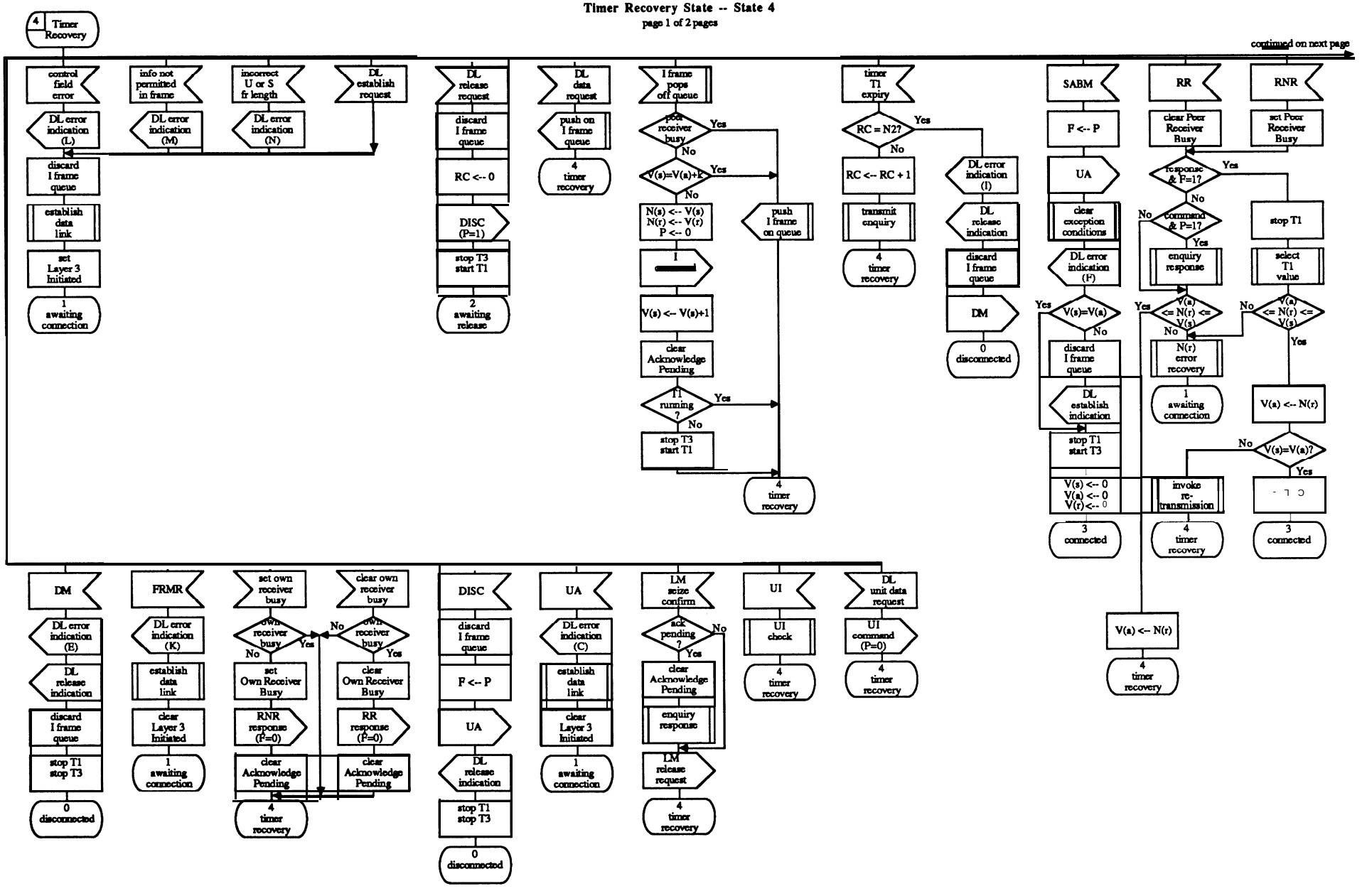


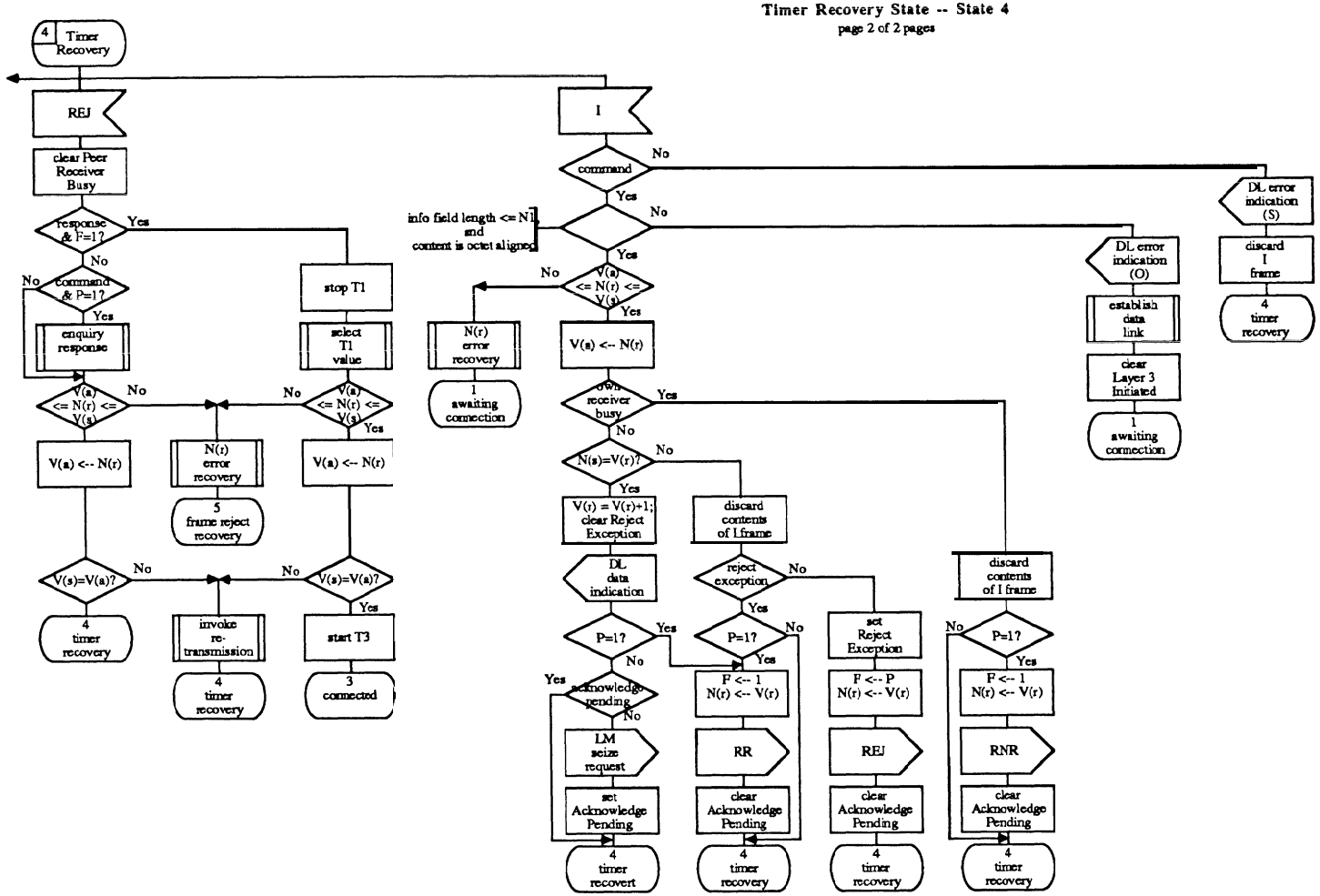
Data Link
Connected State -- State 3
page 1 of 2 pages




3
Connected
REJ
clear Peer Receiver Busy
check need for response
V(a) <= N(r) <= V(s)
No
Yes
V(a) <-- N(r)
stop T1
start T3
select T1 value
invoke re-transmission
3
connected
N(r) error recovery
1
awaiting connection
I
No
Yes
info field length <= N1 and content is octet aligned
No
Yes
V(a) <= N(r) <= V(s)
No
Yes
peer receiver busy
No
Yes
V(a) <-- N(r)
check I frames ack'd
own receiver busy
Yes
No
N(s)=V(r)?
No
Yes
V(r) = V(r)+1; clear Rejec Exception
discard contents of I frame
DL data indicatior
reject exception
No
Yes
P=1?
Yes
No
P=1?
Yes
No
Acknowledge pending
Yes
No
LM seize request
set Acknowledge Pending
3
connected
F <-- 1
N(r) <-- V(r)
RR
clear Acknowledge Pending
3
connected
set Reject Exception
F <-- P
N(r) <-- V(r)
REJ
clear Acknowledge Pending
3
connected
discard contents of I frame
No
P=1?
Yes
F <-- 1
N(r) <-- V(r)
RNR
clear Acknowledge Pending
3
connected
DL error indication (O)
establish data link
clear Layer 3 Initiated
1
awaiting connection
DL error indication (S)
discard I frame
3
connected

Data Link
Timer Recovery State -- State 4
page 1 of 2 pages




4 Timer Recovery
REJ
clear Peer Receiver Busy
response & F=1?
Yes
No
command & P=1?
No
Yes
enquiry response
V(a) <= N(r) <= V(s)
No
V(a) <-- N(r)
V(s)=V(a)?
No
4 timer recovery
N(r) error recovery
5 frame reject recovery
invoke re-transmission
4 timer recovery
stop T1
select T1 value
V(a) <= N(r) <= V(s)
Yes
V(a) <-- N(r)
V(s)=V(a)?
Yes
start T3
3 connected
I
command
No
Yes
info field length <= N1 and content is octet aligned
No
Yes
V(a) <= N(r) <= V(s)
No
Yes
N(r) error recovery
1 awaiting connection
V(a) <-- N(r)
own receiver busy
Yes
No
N(s)=V(r)?
No
Yes
V(r) = V(r)+1; clear Reject Exception
DL data indication
P=1?
Yes
No
acknowledge pending
Yes
No
LM seize request
set Acknowledge Pending
4 timer recovert
discard contents of I frame
reject exception
No
Yes
P=1?
Yes
No
F <-- 1 N(r) <-- V(r)
RR
clear Acknowledge Pending
4 timer recovery
set Reject Exception
F <-- P N(r) <-- V(r)
REJ
clear Acknowledge Pending
4 timer recovery
discard contents of I frame
No
P=1?
Yes
F <-- 1 N(r) <-- V(r)
RNR
clear Acknowledge Pending
4 timer recovery
DL error indication (O)
establish data link
clear Layer 3 Initiated
1 awaiting connection
DL error indication (S)
discard I frame
4 timer recovery

Data Link
Subroutines