AX.25 Link Access Protocol for Amateur Packet Radio

Version 2.2
Revision: 11 November 1997

Portions of this book first appeared as "AX.25" John Ackermann, AG9V. 1992

Authors:
William A. Beech, NJ7P
Douglas E. Nielsen, N7LEM
Jack Taylor, N7OO

Edited by:
Lee Knoper, N7CUU
73223.262@compuserve.com

Production Editors:
Greg Jones, WD5IVD, wd5ivd@tapr.org
Paul Rinaldo, W4RI, prinaldo@arrl.org

Foreward

Packet radio has linked many thousands of amateur radio stations together directly and by the packet network. The packet network has grown from a series of digipeaters to a sophisticated global network consisting of several types of nodes, including HF gateways, Internet gateways and satellite links. This progress has happened since the publication of version 2.0 of the AX.25 protocol in 1984 Terry L. Fox, WB4JFI.

A major effort towards updating version 2.0 was published in the 7th Computer Networking Conference by Eric Scace, K3NA, in 1988.

Additional portions of this update were handed out at the conference. Eric's work is included in this update of the standard, together with protocol improvements that will aid networking and HF users.

This document is a revision of the AX.25 Version 2.0 Protocol Standard found on the Internet and available from the American Radio Relay League. The authors of this new version took exception to the use of AX.25; none of the Layer 3 protocol has been used. We have changed these references to LAPA to signify the Link Access Protocol - Amateur, a data-link (Layer 2) protocol.

The authors wish to thank Eric Gustafson, N7CL, and Lyle Johnson, WA7GXD, for providing the material and encouragement during this development.

July 1993

William A. Beech, NJ7P
Douglas E. Nielsen, N7LEM
Jack Taylor, N7OO

Preface

This document is the fourth edition of the AX.25 Amateur Packet Radio Link Layer Protocol (Version 2.2, 1996) published by the American RadioRelay League (ARRL) and the Tucson Amateur Packet Radio Corporation (TAPR) .

In July, 1984, the Administrative Council of the International Amateur Radio Union (IARU) met in Paris and designated the ARRL as the international clearinghouse for information relating to packet radio, with a view towards encouraging common standards and regulations.

This document defines a protocol used between two amateur radio stations in a point-to-point or networked communications environment. The protocol specifies only link layer and physical layer functions. It is not intended to specify any upper-layer protocol other than certain interface requirements to and from other layers.

This protocol recognizes and accomodates the uniqueness of the amateur radio operating environment. In the interval since the publication of the first edition of the standard, an amateur radio digital network has evolved. Because this development has negated the need for the digipeater mode of operation, the proposed new specification limits digipeating to a maximum of two hops or separate radio links.

This document goes a step beyond most international standards by making the System Description Language (SDL), included in Appendix C, the basis for the standard. The SDL takes precedence over the text of this document and should be used to resolve any apparent discrepancies between the two. The SDL is a much clearer description of the protocol than the verbal text.

A version of this protocol developed from the SDL in the "C" programming language is available from the authors.

Contents

Forward iii
Preface iv

1. Abstract 1
1.1. General 1

2. Concepts and Terminology 2
2.1. Basic Concepts 2
2.2. AX.25 Model 2
2.3. Data-Link Service Access Point 3
2.4. Segmenter 4
2.5. Data Link 4
2.6. Management Data Link 5
2.7. Link Multiplexer 5
2.8. Physical 5
2.9. System Description Language 5

3. Frame Structure 6
3.1. Flag Field 6
3.2. Address Field 7
3.3. Control Field 7
3.4. PID Field 7
3.5. Information Field 8
3.6. Bit Stuffing 8
3.7. Frame-Check Sequence 8
3.8. Order of Bit Transmission 8
3.9. Invalid Frames 8
3.10. Frame Abort 9
3.11. Inter-Frame Time Fill 9
3.12. Address-Field Encoding 9
3.12.1. Non-repeater Address-Field Encoding 9
3.12.2. Destination Subfield Encoding 11
3.12.3. Source Subfield Encoding 12
3.12.4. Layer 2 Repeater Address Encoding 13
3.12.5. Multiple Repeater Operation 15

4. Elements of Procedure and Formats of Fields 16
4.1. General 16
4.2. Control Fields 16
4.2.1. Control-Field Formats 16
4.2.1.1. Information-Transfer Format 17
4.2.1.2. Supervisory Format 17
4.2.1.3. Unnumbered Format 17
4.2.2. Control-Field Parameters 18
4.2.3. Sequence Numbers 18
4.2.4. Frame Variables and Sequence Numbers 18
4.2.4.1. Send State Variable V(S) 18
4.2.4.2. Send Sequence Number N(S) 18
4.2.4.3. Receive State Variable V(R) 18
4.2.4.4. Received Sequence Number N(R) 18
4.2.4.5. Acknowledge State Variable V(A) 18
4.3. Control-Field Coding for Commands and Responses 19
4.3.1. Information Command Frame Control Field 19
4.3.2. Supervisory Frame Control Field 20
4.3.2.1. Receive Ready (RR) Command and Response 21
4.3.2.2. Receive Not Ready (RNR) Command and Response 21
4.3.2.3. Reject (REJ) Command and Response 21
4.3.2.4. Selective Reject (SREJ) Command and Response 21
4.3.3. Unnumbered Frame Control Fields 22
4.3.3.1. Set Asynchronous Balanced Mode (SABM) Command 23
4.3.3.2. Set Asynchronous Balanced Mode Extended (SABME) Command . . . 23
4.3.3.3. Disconnect (DISC) Command 23
4.3.3.4. Unnumbered Acknowledge (UA) Response 23
4.3.3.5. Disconnected Mode (DM) Response 23
4.3.3.6. Unnumbered Information (UI) Frame 24
4.3.3.7. Exchange Identification (XID) Frame 24
4.3.3.8. Test (TEST) Frame 28
4.3.3.9. FRMR Response Frame 28
4.4. Link Error Reporting and Recovery 28
4.4.1. TNC Busy Condition 28
4.4.2. Send Sequence Number Error 29
4.4.3. Reject (REJ) Recovery 29
4.4.4. Selective Reject (SREJ) Recovery 29
4.4.5. Timeout Error Recovery 30
4.4.5.1. T1 Timer Recovery 30
4.4.5.2. Timer T3 Recovery 30
4.4.6. Invalid Frame or FCS Error 30

5. Elements for Layer-to-Layer Communication 31
5.1. Layer 3 Entity <—> Management Data-link State Machine 31
5.2. Management Data-Link State Machine <-> Link Multiplexer State Machine 31
5.3. Layer 3 Entity <—> Data-Link State Machine 32
5.4. Data-Link State Machine <—> Link Multiplexer State Machine 33
5.5. Link Multiplexer State Machine <—> Physical State Machine 34
5.6. Physical State Machine <—> Hardware 34

6. Description of AX.25 Procedures 35
6.1. Address Field Operation 35
6.1.1. Address Information 35
6.1.2. Command/Response Procedure 35
6.2. Poll/Final (P/F) Bit Procedures 36
6.3. Procedures For Link Set-Up and Disconnection 36
6.3.1. AX.25 Link Connection Establishment 36
6.3.2. Parameter Negotiation Phase 37
6.3.3. Information-Transfer Phase 38
6.3.4. Link Disconnection 38
6.3.5. Disconnected State 38
6.3.6. Collision Recovery 39
6.3.6.1. Collisions in a Half-Duplex Environment 39
6.3.6.2. Collisions of Unnumbered Commands. 39
6.3.6.3. Collision of a DM with a SABM(E) or DISC 39
6.3.7. Connectionless Operation 39
6.4. Procedures for Information Transfer 40
6.4.1. Sending I Frames 40
6.4.2. Receiving I Frames 40
6.4.2.1. Not Busy 40
6.4.2.2. Busy 40
6.4.3. Priority Acknowledge 40
6.4.4. Reception of Out-of-Sequence Frames 41
6.4.4.1. Implicit Reject (REJ) 41
6.4.4.2. Selective Reject (SREJ) 41
6.4.4.3. Selective Reject-Reject (SREJ/REJ) 41
6.4.5. Reception of Incorrect Frames 41
6.4.6. Receiving Acknowledgement 41
6.4.7. Receiving REJ 42
6.4.8. Receiving an SREJ 42
6.4.9. Receiving an RNR Frame 42
6.4.10. Sending a Busy Indication 42
6.4.11. Waiting Acknowledgement 43

6.5. Resetting Procedure 43
6.6. Disassembler/Reassembler 44
6.7. List of System Defined Parameters 44
6.7.1. Timers 44
6.7.1.1. Acknowledgment Timer T1 44
6.7.1.2. Response Delay Timer T2 45
6.7.1.3. Inactive Link Timer T3 45
6.7.1.4. Repeater Hang Timer T100 (AXHANG) 45
6.7.1.5. Priority Window Timer T101 (PRIACK) 45
6.7.1.6. Slot Time Timer T102 (p-persistence) 45
6.7.1.7. Transmitter Startup Timer T103 (TXDELAY) 45
6.7.1.8. Repeater Startup Timer T104 (AXDELAY) 45
6.7.1.9. Remote Receiver Sync Timer T105 45
6.7.1.10. Ten Minute Transmission Limit Timer T106 45
6.7.1.11. Anti-Hogging Limit Timer T107 46
6.7.1.12. Receiver Startup Timer T108 46
6.7.1.13. Next Segment Timer TR210 46
6.7.2. Parameters 46
6.7.2.1. Maximum Number of Octets in an I Field (N1) 46
6.7.2.2. Maximum Number of Retries (N2) 46
6.7.2.3. Maximum Number of I Frames Outstanding (k) 46

Appendix A: Glossary 47

Appendix B: References 48

Appendix C1: Introduction to System Description Language 49
C1.1. Principles of Extended Finite State Machines 49
C1.2. SDL Symbol Definition 49

Appendix C2a: Simplex Physical Layer State Machines 52
C2a.1. Interaction with the Link Multiplexer 52
C2a.2. Interface to the Hardware 53
C2a.3. Internal Operation of the Machine 53

Appendix C2b: Duplex Physical Layer State Machines 63
C2b.1. Interaction with the Link Multiplexer 63
C2b.2. Interface to the Hardware 64
C2b.3. Internal Operation of the Machine 64

Appendix C3: Link Multiplexer State Machine 70
C3.1. Interaction with the Data-Link State Machine 70
C3.2. Interaction with the Physical Layer State Machine 70
C3.3. Internal Operation of the Machine 71

Appendix C4: Data-Link State Machine 79
C4.1. Interaction with the Data-Link Service Access Point 79
C4.2. Interaction with the Link Multiplexer State Machine 80
C4.3. Internal Operation of the Machine 80

Appendix C5: Management Data-Link State Machine 107
C5.1. Interaction with the Data-Link Service Access Point 107
C5.2. Interaction with the Link Multiplexer State Machine 107
C5.3. Internal Operation of the Machine 107

Appendix C6: Segmenter/Reassembler 117
C6.1. Segmenter State Machine 117
C6.2. Reassembler State Machine 117
C6.3. Internal Operation of the Machine 118
C6.3.1. Internal Operation of the Segmenter State Machine 118
C6.3.2. Internal Operation of the Reassembler State Machine 118
C6.4. Final Observations 119

Appendix D: Data Link Service Access Point and Primitives 126
D.1. Model of a Data-Link Connection 126
D.2. Queue Model Concepts 127
D.3. DLC Establishment 128
D.4. Data Transfer 128
D.5. DLC Release 129
D.6. Relationship of Primitives at the Two DLC Endpoints 129

List of Figures xi

List of Figures

Figure 2.1. Seven layer OSI reference model. 2
Figure 2.2. AX.25 finite state machine model (single link). 2
Figure 2.3. AX.25 finite state machine model (multiple stream). 3
Figure 2.4. Example use of primitive types. 4

Figure 3.1a. U and S frame construction. 6
Figure 3.1b. Information frame construction. 6
Figure 3.2. PID definitions. 7
Figure 3.3. Non-repeater address-field encoding. 9
Figure 3.4. Non-repeater AX.25 frame. 10
Figure 3.5. Destination field encoding. 11
Figure 3.6. Source field encoding. 12
Figure 3.7. Repeater address encoding. 13

Figure 4.1a. Control-field formats (modulo 8). 16
Figure 4.1b. Control-field formats (modulo 128). 17
Figure 4.2a. I frame control field (modulo 8). 19
Figure 4.2b. I frame control field (modulo 128). 19
Figure 4.3a. S frame control fields (modulo 8). 20
Figure 4.3b. S frame control fields (modulo 128). 20
Figure 4.4. U frame control fields. 22
Figure 4.5. Parameter negotiation - parameter field elements. 25

Figure 6.2 Segment header format. 44

Figure C1.1. SDL examples C1-C4. 51

Figure C2a.1. Summary of primitives, states, queues, flags, errors and timers. 55
Figure C2a.2. Simplex physical ready state. 56
Figure C2a.3. Simplex physical receiving state. 57
Figure C2a.4. Simplex physical transmitter suppression state. 58
Figure C2a.5. Simplex physical transmitter start state. 59
Figure C2a.6. Simplex physical transmitting state. 60
Figure C2a.7. Simplex physical digipeating state. 61
Figure C2a.9. Simplex physical subroutines C-2-A-12. 62
Figure C2a.8. Simplex physical receiver start state. 62
Figure C2b.1. Summary of primitives, states, queues, flags, errors and timers. 66
Figure C2b.3. Duplex physical receiving state. 67
Figure C2b.2. Duplex physical receiver ready state. 67
Figure C2b.5. Duplex physical transmitter start state. 68
Figure C2b.4. Duplex physical transmitter ready state. 68
Figure C2b.6. Duplex physical transmitting state. 69

Figure C3.1. Summary of primitives, states, flags, errors and timers. 74
Figure C3.1. Link Multiplexer idle state. 75
Figure C3.2. Link Multiplexer seize pending state. 76
Figure C3.3. Link Multiplexer seized state. 77
Figure C3.4. Link Multiplexer subroutines. 78

Figure C4.1. Summary of primitives, states, flags, errors and timers. 83
Figure C4.2. Data-link disconnected state. (Pages 84-85) 84
Figure C4.3. Data-link awaiting connection state. (Pages 86-88) 86
Figure C4.4. Data-link awaiting release state. (Pages 89-91) 89
Figure C4.5. Data-link connected state. (Pages 92-97) 92
Figure C4.6. Data-link timer recovery state. (Pages 98-102) 98
Figure C4.7. Data-link subroutines state. (Pages 103-106) 103

Figure C5.1. Management Data-link ready state. 109
Figure C5.2. Management Data-link negotiating state. 110
Figure C5.4. Management Data-link window notification subroutine. 112
Figure C5.5. Management Data-link T1 negotiation subroutine. 113
Figure C5.6. Management Data-link retry notification subroutine. 114
Figure C5.7. Management Data-link optional functions negotiation subroutine. 115
Figure C5.8. Management Data-link classes of procedure negotiation subroutines C-5-1. 116

Figure C-6.1. Primitives, States, Queues, Flags, Parameters, Errors and Timers. 121
Figure C-6.2. Segmenter Ready State. 122
Figure C-6.3. Reassembler Ready State. 123
Figure C-6.4. Reassembler Assembling Data State. 124
Figure C-6.5. Reassembler Assembling Unit Data State. 125

Figure D.1. Queue model of a data-link connection. 127
Figure D.2. Relationships between queue model objects. 129
Figure D.3. Example of a connection-oriented data exchange. 130

1. Abstract

This document details AX.25 version 2.2 digital communication standard. The objective of this standard is to ensure link-layer compatibility between stations. This document is intended to assist the designers and users of amateur packet radio equipment by providing a high-level common reference publication. However, the existence of this protocol is not intended to disparage anyone from designing, marketing or using products, processes or procedures not conforming to the protocol.

As with any evolving technical standard, this protocol is subject to periodic review. Interested parties are encouraged to use the latest edition.

1.1. General

The amateur radio community has expressed the need and desire to define a protocol that can accept and reliably deliver data over a variety of communications links between two signaling terminals. The AX.25 version 2.2 Link-Layer Protocol provides this service, independent of the existence of any upper layer.

This protocol conforms to International Standards Organization (ISO) Information Standards (IS) 3309, 4335 and 7809 High-level Data Link Control (HDLC) and uses terminology found in these documents. It also follows the principles of Consultative Committee in International Telegraph and Telephone (CCITT) Recommendation Q.920 and Q.921 (LAP-D) in the use of multiple links, distinguished by the address field, on a single shared channel. Parameter negotiation was extracted from ISO IS 8885. The data-link service definitions were extracted from ISO IS 8886.

As defined, this protocol works equally well in either half- or full-duplex amateur radio environments, and has been improved for operation over partially impaired HF circuits.

It works equally well for direct connections between two individual amateur packet radio stations, or between an individual station and a multi-port controller.

It permits the establishment of more than one link-layer connection per device, if the device is so capable.

It also permits self-connections. A self-connection occurs when a device establishes a link to itself using its own address for both the source and destination of the frame.

Most link-layer protocols assume that one primary (or master) device (generally called a Data Circuit Terminating Equipment, or DCE), is connected to one or more secondary (or slave) device(s) (usually called a Data Terminating Equipment, or DTE). This type of unbalanced operation is not practical in a shared RF amateur radio environment. Instead, AX.25 assumes that both ends of the link are of the same class, thereby eliminating the two different classes of devices.

In this protocol specification, the phrase Terminal Node Controller (TNC) refers to the balanced type of device found in amateur packet radio. Other standards refer to these peer entities as DXEs.

2. Concepts and Terminology

2.1. Basic Concepts

ISO has developed a reference model for Open Systems Interconnection (OSI) to better facilitate the interconnection of different types of computing systems. The basic structuring technique in this reference model is known as layering. According to this technique, communication among application processes is viewed as being logically partitioned into an ordered set of layers represented in a vertical sequence as shown in Figure 2.1. Each layer provides a Service Access Point (SAP) for interface to the next higher layer. Note that any layer may be a null, where no function or code is provided. Such is the case with the current TAPR TNC-2 equipment, where only Layers 1, 2 and 7 are provided; these comprise the minimum configuration for reliable communications.

Layer	Function
7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

Figure 2.1. Seven layer OSI reference model.

2.2. AX.25 Model

The two lower layers, data link and physical, can be further subdivided into several distinct finite state machines as shown in Figure 2.2. This example shows a single link to the radio port.

<table>
<tr><th>Layer</th><th colspan="2">Function(s)</th></tr>
<tr><td rowspan="3">Data Link (2)</td><td>Segmenter</td><td rowspan="2">Management Data Link</td></tr>
<tr><td>Data Link</td></tr>
<tr><td colspan="2">Link Multiplexer</td></tr>
<tr><td rowspan="2">Physical (1)</td><td colspan="2">Physical</td></tr>
<tr><td colspan="2">Silicon/Radio</td></tr>
</table>

Figure 2.2. AX.25 finite state machine model (single link).

Figure 2.3 shows an example of multiple links to the radio port. The link multiplexer described in this standard multiplexes multiple data-link connections into one physical connection. A separate data-link machine must be provided for each connection allowed by the implementation.

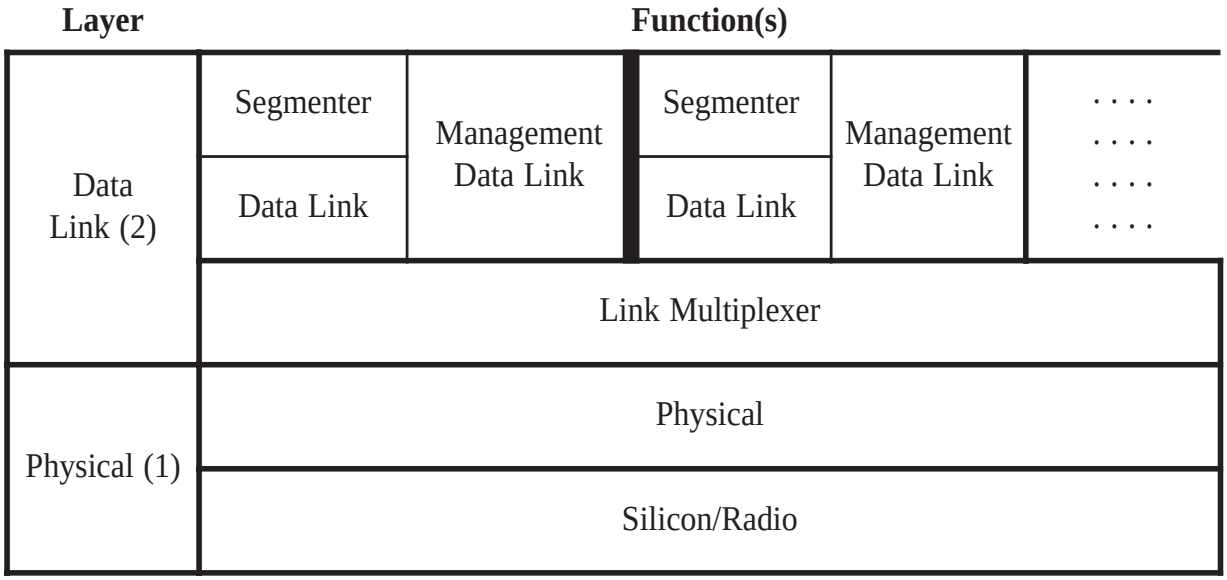


Figure 2.3. AX.25 finite state machine model (multiple stream).

2.3. Data-Link Service Access Point

Figures 2.2 and 2.3 indicate a Data-Link Service Access Point (DLSAP) at the upper boundary of Layer 2. This DLSAP is the point at which the data-link layer provides services to Layer 3. Associated with each DLSAP is one or more data-link connection endpoint(s).

Entities exist in each layer. Entities may be the Link Multiplexer, Data Link, Management Data Link or Segmenter. Entities in the same layer, but in different systems that must exchange information to achieve a common objective, are called "peer entities." Entities in adjacent layers interact through their common boundary. The services provided by the data-link layer are the combination of the services and functions provided by both the data-link layer and the physical layer.

Cooperation between data-link layer entities is governed by a peer-to-peer protocol specific to the layer. For example, when information is to be exchanged between two Layer 3 entities, an association must be established between the entities through the data-link layer using the AX.25 protocol. This association is called a data-link connection. Data-link connections are provided by the data-link layer between two or more DLSAPs.

Layer 3 requests services from the data-link layer via command/responseinteractions known as service "primitives." (Similarly, the interactionbetween the data-link layer and the physical layer also occurs via serviceprimitives.) Primitives are discussed in greater detail in Section 5.

The primitives that are exchanged between the data-link layer and adjacent layers are of the following four types:

a) REQUEST primitive type: used by a higher layer to request a service from the next lower layer;

b) INDICATION primitive type: used by a layer to provide a service to notify the next higher layer of any specific activity that is service related. The INDICATION primitive may be the result of an activity of the lower layer related to the primitive type REQUEST at the peer entity;

c) RESPONSE primitive type: used by a layer to acknowledge receipt from a lower layer of the primitive type INDICATION. AX.25 does not use the RESPONSE primitive; and

d) CONFIRM primitive type: used by a layer to provide the requested service to confirm that the activity has been completed.

Figure 2.4 illustrates the use of the four primitive types in conjunction with the connect primitive.

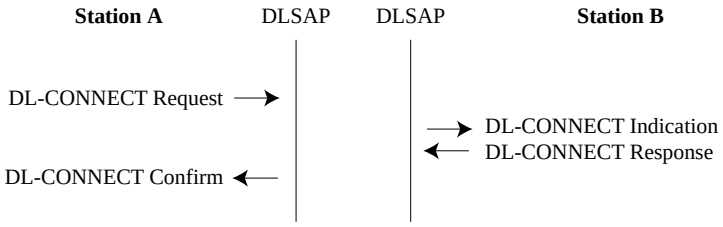


Figure 2.4. Example use of primitive types.

2.4. Segmenter

The Segmenter State Machine accepts input from the higher layer through the DLSAP. If the unit of data to be sent exceeds the limits of a AX.25 Information (I) frame (see Section 4.3.1) or Unnumbered Information (UI) frame (see Section 4.3.3.6), the segmenter breaks the unit down into smaller segments for transmission. Incoming segments are reassembled for delivery to the higher layer and passed through the DLSAP. The segmenter passes all other signals unchanged.

One segmenter exists per data link. Because a single piece of equipment may have multiple data links in operation simultaneously (e.g., to support multiple higher-layer applications), there can be multiple, independently operating segmenters within the equipment.

2.5. Data Link

The Data-link State Machine is the heart of the AX.25 protocol. The Data-link State Machine provides all logic necessary to establish and release connections between two stations and to exchange information in a connectionless (i.e., via UI frames) and connection-oriented (i.e., via I frames with recovery procedures) manner.

One Data-link State Machine exists per data link. Because a single piece of equipment may have multiple data links in operation simultaneously (e.g., to support multiple higher layer applications), there can be multiple, independently operating data-link machines within the equipment.

2.6. Management Data Link

The Management Data-link State Machine provides for the parameter negotiation of the AX.25 protocol. The Management Data-link State Machine provides all logic necessary to negotiate operating parameters between two stations.

One Management Data-link State Machine exists per data link. Because a single piece of equipment may have multiple data links in operation simultaneously (e.g., to support multiple higher layer applications), there can be multiple, independently operating management data-link machines within the equipment.

2.7. Link Multiplexer

The Link Multiplexer State Machine allows one or more data links to share the same physical (radio) channel. The Link Multiplexer State Machine provides the logic necessary to give each data link an opportunity to use the channel, according to the rotation algorithm embedded within the link multiplexer.

One Link Multiplexer State Machine exists per physical channel. If a single piece of equipment has multiple physical channels operating simultaneously, then an independently operating Link Multiplexer State Machine exists for each channel.

2.8. Physical

The Physical State Machine manipulates the radio transmitter and receiver. One Physical State Machine exists per physical channel.

Because different types of radio channel operations are used, the Physical State Machine exists in different forms. Each form hides the peculiar characteristics of each radio channel from the higher layer state machines. Two Physical State Machines have been defined in this standard: simplex and full duplex Physical State Machines.

2.9. System Description Language

Each of the above finite state machines is described in the System Description Language in Appendix C.

3. Frame Structure

Link layer packet radio transmissions are sent in small blocks of data, called frames.

There are three general types of AX.25 frames:

a) Information frame (I frame);
b) Supervisory frame (S frame); and
c) Unnumbered frame (U frame).

Each frame is made up of several smaller groups, called fields. Figures 3.1a and 3.1b illustrate the three basic types of frames. Note that the first bit to be transmitted is on the left side.

Flag	Address	Control	Info	FCS	Flag
01111110	112/224 Bits	8/16 Bits	N*8 Bits	16 Bits	01111110

Figure 3.1a. U and S frame construction.

Flag	Address	Control	PID	Info	FCS	Flag
01111110	112/224 Bits	8/16 Bits	8 Bits	N*8 Bits	16 Bits	01111110

Figure 3.1b. Information frame construction.

Notes:

- The Info field exists only in certain frames (Section 4.4.3)
- FCS is the Frame Check Sequence field (Section 4.4.6)
- PID is the Protocol Identifier field (Section 3.4)

Each field is made up of an integral number of octets (8-bit byte of binary data) and serves the specific function outlined below.

All fields except the Frame Check Sequence (FCS) are transmitted low-order bit first. FCS is transmitted bit 15 first.

3.1. Flag Field

The flag field is one octet long. Because the flag delimits frames, it occurs at both the beginning and end of each frame. Two frames may share one flag, which would denote the end of the first frame and the start of the next frame. A flag consists of a zero followed by six ones followed by another zero, or 01111110 (7E hex). As a result of bit stuffing (see Section 3.6), this sequence is not allowed to occur anywhere else inside a complete frame.

3.2. Address Field

The address field identifies both the source of the frame and its destination. In addition, the address field contains the command/response information and facilities for Layer 2 repeater operation.

The encoding of the address field is described in Section 3.12.

3.3. Control Field

The control field identifies the type of frame being passed and controls several attributes of the Layer 2 connection. It is one or two octets in length; its encoding is discussed in Section 4.2.

3.4. PID Field

The Protocol Identifier (PID) field appears in information frames (I and UI) only. It identifies which kind of Layer 3 protocol, if any, is in use.

The PID itself is not included as part of the octet count of the information field. The encoding of the PID is as follows:

HEX	MSB LSB	Translation
**	yy01yyyy	AX.25 layer 3 implemented.
**	yy10yyyy	AX.25 layer 3 implemented.
0x01	00000001	ISO 8208/CCITT X.25 PLP
0x06	00000110	Compressed TCP/IP packet. Van Jacobson (RFC 1144)
0x07	00000111	Uncompressed TCP/IP packet. Van Jacobson (RFC 1144)
0x08	00001000	Segmentation fragment
0xC3	11000011	TEXNET datagram protocol
0xC4	11000100	Link Quality Protocol
0xCA	11001010	Appletalk
0xCB	11001011	Appletalk ARP
0xCC	11001100	ARPA Internet Protocol
0xCD	11001101	ARPA Address resolution
0xCE	11001110	FlexNet
0xCF	11001111	NET/ROM
0xF0	11110000	No layer 3 protocol implemented.
0xFF	11111111	Escape character. Next octet contains more Level 3 protocol information.
Escape character. Next octet contains more Level 3 protocol information.	00001000	

Figure 3.2. PID definitions.

Where:

An “X” indicates all combinations used.

Note: All forms of XX11XXXX and XX00XXXX other than those listed above are reserved at this time for future Layer 3 protocols. The assignment of these formats is subject to mutual agreement among amateur radio operators. It is recommended that the creators of Layer 3 protocols contact the ARRL for suggested encodings.

3.5. Information Field

The information (I) field conveys user data from one end of the link to the other.

The I fields are allowed in only five types of frames:

a) The I frame;
b) The UI frame;
c) The XID frame;
d) The TEST frame; and
e) The FRMR frame.

The I field defaults to a length of 256 octets and contains an integral number of octets. These constraints apply prior to the insertion of zero bits as specified in Section 3.6. Any information in the I field is passed along the link transparently, except for the zero-bit insertion (see Section 3.6) necessary to prevent flags from accidentally appearing in the I field.

3.6. Bit Stuffing

In order to ensure that the flag bit sequence mentioned above does not appear accidentally anywhere else in a frame, the sending station monitors the bit sequence for a group of five or more contiguous "1" bits. Any time five contiguous "1" bits are sent, the sending station inserts a "0" bit after the fifth "1" bit. During frame reception, any time five contiguous "1" bits are received, a "0" bit immediately following five "1" bits is discarded.

3.7. Frame-Check Sequence

The Frame-Check Sequence (FCS) is a sixteen-bit number calculated by both the sender and the receiver of a frame. It ensures that the frame was not corrupted by the transmission medium. The Frame-Check Sequence is calculated in accordance with recommendations in the HDLC reference document, ISO 3309.

3.8. Order of Bit Transmission

The FCS field of an AX.25 frame is sent most-significant bit first. All other fields are sent with each octet's least-significant bit first.

3.9. Invalid Frames

A frame is considered by the link layer to be an invalid frame if it:

a) consists of less than 136 bits (including the opening and closing flags);
b) is not bounded by opening and closing flags; or
c) is not octet aligned (an integral number of octets).

3.10. Frame Abort

If a frame must be prematurely aborted, at least fifteen contiguous "1"s are sent without bit stuffing added.

3.11. Inter-Frame Time Fill

Whenever it is necessary for a TNC to keep its transmitter on while not actually sending frames, the time between frames should be filled with contiguous flags.

3.12. Address-Field Encoding

The address field of all frames consists of a destination, source and (optionally) two Layer 2 repeater subfields. Each subfield consists of an amateur callsign and a Secondary Station Identifier (SSID). The callsign is made up of upper-case alpha and numeric ASCII characters only. The SSID is a four-bit integer that uniquely identifies multiple stations using the same amateur callsign.

The HDLC address field is extended beyond one octet by assigning the least-significant bit of each octet to be an "extension bit." The extension bit of each octet is set to "0" to indicate the next octet contains more address information, or to "1", to indicate that this is the last octet of the HDLC address field. To make room for this extension bit, the amateur radio call- sign information is shifted one bit left.

3.12.1. Non-repeater Address-Field Encoding

If Layer 2 repeaters are not being used, the address field is encoded as shown in Figure 3.3. The destination address is the callsign and SSID of the amateur radio station to which the frame is addressed. The source address contains the amateur callsign and SSID of the station that sent the frame. These callsigns are the callsigns of the two ends of a Layer 2 AX.25 link only.

First Octet Sent

Address Field of Frame	
Destination Address Subfield	Source Address Subfield
A1 A2 A3 A4 A5 A6 A7	A8 A9 A10 A11 A12 A13 A14

Figure 3.3. Non-repeater address-field encoding.

A1 through A14, above, are the fourteen octets that make up the two address subfields of the address field. The destination subfield is seven octets long (A1 through A7), and is sent first. This address sequence provides the receivers of frames time to check the destination address subfield to see if the frame is addressed to them while the rest of the frame is being received. The source address subfield is then sent in octets A8 through A14. Both of these subfields are encoded in the same manner, except that the last octet of the address field has the HDLC address extension bit set.

The SSID octet at the end of each address subfield (A7 and A14) contains the SSID and the "C" bit. The C bits identify command and response frames (see Section 6.1.2). The SSID octet at the end of each optional Layer 2 repeater address subfield (A21 and A28) contains the SSID and the "H" bit ("[H]as-been-repeated"). The H bits indicate that the Layer 2 repeater station has repeated the frame (see Section 3.12.3). Each SSID octet contains two bits that are reserved for future use.

Figure 3.4 shows a typical AX.25 frame in the non-repeater mode of operation.

Octet	ASCII	Bin Data	Hex Data
Flag		01111110	7E
A1	N	10011000	98
A2	J	10010100	94
A3	7	01101110	6E
A4	P	10100000	A0
A5	space	01000000	40
A6	space	01000000	40
A7	SSID	11100000	E0
A8	N	10011000	98
A9	7	01101110	6E
A10	L	10011000	98
A11	E	10001010	8A
A12	M	10011010	9A
A13	space	01000000	40
A14	SSID	01100001	61
Control	I	00111110	3E
PID	none	11110000	F0
FCS	part1	XXXXXXXX	HH
FCS	part2	XXXXXXXX	HH
Flag		01111110	7E
	Bit position	76543210	

Figure 3.4. Non-repeater AX.25 frame.

The frame shown is an I frame, not going through a Layer 2 repeater, from N7LEM (SSID=0) to NJ7P (SSID=0), without a Layer 3 protocol. The P/F bit is set; the receive sequence number [N(R)] is 1; the send sequence number [N(S)] is 7.

3.12.2. Destination Subfield Encoding

Figure 3.5 shows how an amateur callsign is placed in the destination address subfield, occupying octets A1 through A7.

Octet	ASCII	Bin Data	Hex Data
A1	N	10011000	98
A2	J	10010100	94
A3	7	01101110	6E
A4	P	10100000	A0
A5	space	01000000	40
A6	space	01000000	40
A7	SSID	11100000	E0
A7	SSID	CRRSSID0	
	Bit position	76543210	

Figure 3.5. Destination field encoding.

Where:

a) The top octet (A1) is the first octet sent, with bit 0 of each octet being the first bit sent, and bit 7 being the last bit sent.

b) The first (low-order or bit 0) bit of each octet is the HDLC address extension bit, set to zero on all but the last octet in the address field, where it is set to one.

c) The bits marked “R” are reserved bits. They may be used in an agreed-upon manner in individual networks. When not implemented, they are set to one.

d) The bit marked “C” is the command/response bit of an AX.25 frame, as outlined in Section 6.1.2.

e) The characters of the callsign are standard seven-bit ASCII (upper case only) placed in the left-most seven bits of the octet to make room for the address extension bit. If the callsign contains fewer than six characters, it is padded with ASCII spaces between the last call sign character and the SSID octet.

f) The 0000 SSID is reserved for the first personal AX.25 station. This provision establishes one standard SSID for “normal” stations to use for the first station.

3.12.3. Source Subfield Encoding

Figure 3.6 shows how an amateur callsign is placed in the destination address subfield, occupying octets A1 through A7.

Octet	ASCII	Bin Data	Hex Data
A8	N	10011000	98
A9	7	01101110	6E
A10	L	10011000	98
A11	E	10001010	8A
A12	M	10011010	9A
A13	space	01000000	40
A14	SSID	CRRSSID0	
	Bit position	76543210	

Figure 3.6. Source field encoding.

Where:

a) The top octet (A8) is the first octet sent, with bit 0 of each octet being the first bit sent, and bit 7 being the last bit sent.

b) The first (low-order or bit 0) bit of each octet is the HDLC address extension bit, set to zero on all but the last octet in the address field, where it is set to one.

c) The bits marked “R” are reserved bits. They may be used in an agreed-upon manner in individual networks. When not implemented, they are set to one.

d) The bit marked “C” is the command/response bit of an LA PA frame, as outlined in Section 6.1.2.

e) The characters of the callsign are standard seven-bit ASCII (upper case only) placed in the leftmost seven bits of the octet to make room for the address extension bit. If the callsign contains fewer than six characters, it is padded with ASCII spaces between the last callsign character and the SSID octet.

f) The 0000 SSID is reserved for the first personal AX.25 station. This provision establishes one standard SSID for “normal” stations to use for the first station.

3.12.4. Layer 2 Repeater Address Encoding

Evolving consensus opinion is that repeater chaining belongs to a higher protocol layer. Consequently, it is being phased out of Layer 2, although backward compatibility is being maintained with a limit of two repeaters.

If a frame is to go through Layer 2 amateur packet repeater(s), an additional address subfield is appended to the end of the address field. This additional subfield contains the callsign(s) of the repeater(s) to be used. This allows more than one repeater to share the same RF channel. If this subfield exists, the last octet of the source subfield has its address extension bit set to "0", indicating that more address-field data follows. The repeater address subfield is encoded in the same manner as the destination and source address subfields, except for the most-significant bit in the last octet, called the H bit. As discussed in Section 3.12.1, the H bit indicates whether a frame has been repeated or not.

The H bit is set to "0" on frames going to a repeater. The repeater changes the H bit to "1" before it retransmits the frame. Stations monitor and repeat frames that meet the following conditions:

a) the frame is addressed to this station in a repeater address subfield;
b) the H bit in its repeater address subfield is 0; or
c) all previous H bits are set to one.

Figure 3.7 shows how the repeater address subfield is encoded. Figure 3.8 is an example of a complete frame after being repeated.

Octet	ASCII	Bin Data	Hex Data
A15	N	10011000	98
A16	J	10010100	94
A17	7	01101110	6E
A18	P	10100000	A0
A19	space	01000000	40
A20	space	01000000	40
A21	SSID	HRRSSID1	
	Bit position	76543210	

Figure 3.7. Repeater address encoding.

Where:

a) The top octet is the first octet sent, with bit 0 being sent first and bit 7 sent last of each octet.

b) As with the source and destination address subfields discussed above, bit 0 of each octet is the HDLC address extension bit, is set to "0" on all but the last address octet, where it is set to "1".

c) The "R" bits are reserved in the same manner as in the source and destination subfields.

d) The "H" bit is the has-been-repeated bit. It is set to "0" when a frame has not been repeated, and set to "1" by the repeating station when repeated.

Octet	ASCII	Bin Data	Hex Data
Flag		01111110	7E
A1	N	10011000	98
A2	J	10010100	94
A3	7	01101110	6E
A4	P	10100000	A0
A5	space	01000000	40
A6	space	01000000	40
A7	SSID	11100000	E0
A8	N	10011000	98
A9	7	01101110	6E
A10	L	10011000	98
A11	E	10001010	8A
A12	M	10011010	9A
A13	space	01000000	40
A14	SSID	01100000	60
A15	N	10011000	98
A16	7	01101110	6E
A17	O	10011110	9E
A18	O	10011110	9E
A19	space	01000000	40
A20	space	01000000	40
A21	SSID	11100011	E3
Control	I	00111110	3E
PID	none	11110000	F0
FCS	part 1	XXXXXXXX	HH
FCS	part 2	XXXXXXXX	HH
Flag		01111110	7E
	Bit position	76543210	

Figure 3.8. AX.25 frame in repeater mode.

The above frame is the same as shown in Figure 3.3, except for the addition of a repeater address subfield (N7OO, SSID=1). The H bit is set, indicating this frame is from the output of the repeater.

3.12.5. Multiple Repeater Operation

The link-layer AX.25 protocol allows operation through more than one repeater. Up to two repeaters may be used by extending the repeater address subfield. When there is more than one repeater address, the repeater address immediately following the source address subfield will be considered the address of the first repeater of a multiple-repeater chain. As a frame progresses through a chain of repeaters, each successive repeater will set the H bit in its SSID octet, indicating that the frame has been successfully repeated through it. No other changes to the frame are made (except for the necessary recalculation of the FCS). The destination station can determine the route the frame took to reach it by examining the address field and use this path to return frames.

The number of repeater addresses is variable. The last repeater address will have the address extension bit of the SSID octet set to "1" indicating the end of the address field. All other address octets will have their address extension bit set to "0".

Note that various timers (see Section 6.6.1) may require adjustment to accommodate the additional delays encountered when a frame must pass through a multiple-repeater chain.

4. Elements of Procedure and Formats of Fields

4.1. General

The elements of procedure define the command and response frames used on the AX.25 link.

Procedures are built from these elements and are described in Section 6.

4.2. Control Fields

The control field identifies the type of frame being sent. The control fields in AX.25 are modeled after the ISO HDLC balanced operation control fields.

4.2.1. Control-Field Formats

The three formats of control fields used in AX.25 are the:

a) Information frame (I frame);

b) Supervisory frame (S frame); and

c) Unnumbered frame (U frame).

Figures 4.1a and 4.1b illustrate the basic format of the control field associated with each of these three types of frames.

The control field can be one or two octets long and may use sequence numbers to maintain link integrity. These sequence numbers may be three-bit (modulo 8) or seven-bit (modulo 128) integers.

Control Field Type	Control-Field Bits							
	7	6	5	4	3	2	1	0
I Frame	N(R)			P	N(S)			0
S Frame	N(R)			P/F	S	S	0	1
U Frame	M	M	M	P/F	M	M	1	1

Figure 4.1a. Control-field formats (modulo 8).

Control Field Type	Control-Field Bits															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I Frame	N(R)							P	N(S)							0
S Frame	N(R)							P/F	0	0	0	0	S	S	0	1

Figure 4.1b. Control-field formats (modulo 128).

Where:

a) Bit 0 is the first bit sent and bit 7 (or bit 15 for modulo 128) is the last bit sent of the control field.

b) N(S) is the send sequence number (bit 1 is the LSB).

c) N(R) is the receive sequence number [bit 5 (or bit 9 for modulo 128) is the LSB].

d) The "S" bits are the supervisory function bits; their encoding is discussed in Section 4.2.1.2.

e) The "M" bits are the unnumbered frame modifier bits; their encoding is discussed in Section 4.2.1.3.

f) The P/F bit is the Poll/Final bit. The P/F bit is used in all types of frames. The P/F bit is also used in a command (poll) mode to request an immediate reply to a frame. The reply to this poll is indicated by setting the response (final) bit in the appropriate frame. Only one outstanding poll condition per direction is allowed at a time. The procedure for P/F bit operation is described in Section 6.2.

4.2.1.1. Information-Transfer Format

All I frames have bit 0 of the control field set to "0". N(S) is the sender's send sequence number (the send sequence number of this frame). N(R) is the sender's receive sequence number (the sequence number of the next expected receive frame). These numbers are described in Section 4.2.4.

4.2.1.2. Supervisory Format

Supervisory frames have bit 0 of the control field set to "1", and bit 1 of the control field set to "0". S frames provide supervisory link control such as acknowledging or requesting retransmission of I frames, and link-layer window control. Because S frames do not have an information field, the sender's send variable and the receiver's receive variable are not incremented for S frames.

4.2.1.3. Unnumbered Format

Unnumbered frames have both bits 0 and 1 of the control field set to "1". U frames are responsible for maintaining additional control over the link beyond what is accomplished with S frames. U frames are responsible for establishing and terminating link connections. U frames also allow for the transmission and reception of information outside of the normal flow control. Some U frames may contain both information and PID fields.

4.2.2. Control-Field Parameters

4.2.3. Sequence Numbers

If modulo 8 operation is in effect (the default), an I frame is assigned a sequential number from 0 to 7. This step allows up to seven outstanding I frames per Layer 2 connection at one time.

If modulo 128 operation is in effect, an I frame is assigned a sequential number between 0 and 127. This step allows up to 127 outstanding I frames per Layer 2 connection at one time.

4.2.4. Frame Variables and Sequence Numbers

4.2.4.1. Send State Variable V(S)

The send state variable exists within the TNC and is never sent. It contains the next sequential number to be assigned to the next transmitted I frame. This variable is updated with the transmission of each I frame.

4.2.4.2. Send Sequence Number N(S)

The send sequence number is found in the control field of all I frames. It contains the sequence number of the I frame being sent. Just prior to the transmission of the I frame, N(S) is updated to equal the send state variable.

4.2.4.3. Receive State Variable V(R)

The receive state variable exists within the TNC. It contains the sequence number of the next expected received I frame. This variable is updated upon the reception of an error-free I frame whose send sequence number equals the present received state variable value.

4.2.4.4. Received Sequence Number N(R)

The received sequence number exists in both I and S frames. Prior to sending an I or S frame, this variable is updated to equal that of the received state variable, thus implicitly acknowledging the proper reception of all frames up to and including N(R)-1.

4.2.4.5. Acknowledge State Variable V(A)

The acknowledge state variable exists within the TNC and is never sent. It contains the sequence number of the last frame acknowledged by its peer [V(A)-1 equals the N(S) of the last acknowledged I frame].

4.3. Control-Field Coding for Commands and Responses

4.3.1. Information Command Frame Control Field

The information (I) command transfers sequentially-numbered frames containing an information field across a data link.

The information-frame control field is encoded as shown in Figures 4.2a and 4.2b. These frames are sequentially numbered by the N(S) subfield to maintain control of their passage over the link-layer connection.

Control Field Type	Control-Field Bits							
	7	6	5	4	3	2	1	0
Information	N(R)			P	N(S)			0

Figure 4.2a. I frame control field (modulo 8).

Control Field Type	Control-Field Bits															
	Second Octet								First Octet							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I Frame	N(R)							P	N(S)							0

Figure 4.2b. I frame control field (modulo 128).

4.3.2. Supervisory Frame Control Field

The supervisory frame control fields are encoded as shown in Figures 4.3a and 4.3b.

Control Field Type		Control-Field Bits			
		7 6 5	4	3 2	1 0
Receive Ready	RR	N(R)	P/F	0 0	0 1
Receive Not Ready	RNR	N(R)	P/F	0 1	0 1
Reject	REJ	N(R)	P/F	1 0	0 1
Selective Reject	SREJ	N(R)	P/F	1 1	0 1

Figure 4.3a. S frame control fields (modulo 8).

Control Field Type	Control-Field Bits			
	Second Octet		First Octet	
	15 14 13 12 11 10 9	8	7 6 5 4 3 2 1 0	
RR	N(R)	P/F	0 0 0 0 0 0 0 1	
RNR	N(R)	P/F	0 0 0 0 0 1 0 1	
REJ	N(R)	P/F	0 0 0 0 1 0 0 1	
SREJ	N(R)	P/F	0 0 0 0 1 1 0 1	

Figure 4.3b. S frame control fields (modulo 128).

Where:

Acronym	Description of Frame Identifiers
RR	Receive Ready - System Ready To Receive.
RNR	Receive Not Ready - TNC Buffer Full.
REJ	Reject Frame - Out of Sequence or Duplicate.
SREJ	Selective Reject - Request single frame repeat.

4.3.2.1. Receive Ready (RR) Command and Response

Receive Ready accomplishes the following:

a) indicates that the sender of the RR is now able to receive more I frames;
b) acknowledges properly received I frames up to, and including N(R)-1;and
c) clears a previously-set busy condition created by an RNR command having been sent.

The status of the TNC at the other end of the link can be requested by sending an RR command frame with the P-bit set to one.

4.3.2.2. Receive Not Ready (RNR) Command and Response

Receive Not Ready indicates to the sender of I frames that the receiving TNC is temporarily busy and cannot accept any more I frames. Frames up to N(R)-1 are acknowledged. Frames N(R) and above that may have been transmitted are discarded and must be retransmitted when the busy condition clears.

The RNR condition is cleared by the sending of a UA, RR, REJ or SABM(E) frame.

The status of the TNC at the other end of the link is requested by sending an RNR command frame with the P bit set to one.

4.3.2.3. Reject (REJ) Command and Response

The reject frame requests retransmission of I frames starting with N(R). Any frames sent with a sequence number of N(R)-1 or less are acknowledged. Additional I frames which may exist may be appended to the retransmission of the N(R) frame.

Only one reject frame condition is allowed in each direction at a time. The reject condition is cleared by the proper reception of I frames up to the I frame that caused the reject condition to be initiated.

The status of the TNC at the other end of the link is requested by sending a REJ command frame with the P bit set to one.

4.3.2.4. Selective Reject (SREJ) Command and Response

The selective reject, SREJ, frame is used by the receiving TNC to request retransmission of the single I frame numbered N(R). If the P/F bit in the SREJ frame is set to “1”, then I frames numbered up to N(R)-1 inclusive are considered as acknowledged. However, if the P/F bit in the SREJ frame is set to “0”, then the N(R) of the SREJ frame does not indicate acknowledgement of I frames.

Each SREJ exception condition is cleared (reset) upon receipt of the I frame with an N(S) equal to the N(R) of the SREJ frame.

A receiving TNC may transmit one or more SREJ frames, each containing a different N(R) with the P bit set to “0”, before one or more earlier SREJ exception conditions have been cleared. However, a SREJ is not transmitted if an earlier REJ exception condition has not been cleared as indicated in Section 4.5.4. (To do so would request retransmission of an I frame that would be retransmitted by the REJ operation.) Likewise, a REJ frame is not transmitted if one or more earlier SREJ exception conditions have not been cleared as indicated in Section 4.5.4.

I frames transmitted following the I frame indicated by the SREJ frame are not retransmitted as the result of receiving a SREJ frame. Additional I frames awaiting initial transmission may be transmitted following the retransmission of the specific I frame requested by the SREJ frame.

4.3.3. Unnumbered Frame Control Fields

Unnumbered frame control fields are either commands or responses.

Figure 4.4 shows the layout of U frames implemented within this protocol.

Control Field Type		Type	Control-Field Bits			
			7 6 5	4	3 2	1 0
Set Async Balanced Mode	SABME	Cmd	0 1 1	P	1 1	1 1
Set Async Balanced Mode	SABM	Cmd	0 0 1	P	1 1	1 1
Disconnect	DISC	Cmd	0 1 0	P	0 0	1 1
Disconnect Mode	DM	Res	0 0 0	F	1 1	1 1
Unnumbered Acknowledge	UA	Res	0 1 1	F	0 0	1 1
Frame Reject	FRMR	Res	1 0 0	F	0 1	1 1
Unnumbered Information	UI	Either	0 0 0	P/F	0 0	1 1
Exchange Identification	XID	Either	1 0 1	P/F	1 1	1 1
Test	TEST	Either	1 1 1	P/F	0 0	1 1

Figure 4.4. U frame control fields.

Where:

Acronym	Description of Frame Identifiers
SABM	Connect Request
SABME	Connect Request Extended (modulo 128)
DISC	Disconnect request
FRMR	Frame Reject
UI	Unnumbered Information Frame
DM	Disconnect Mode - System Busy or Disconnected.
XID	Exchange Identifications - Negotiate features.
UA	Unnumbered Acknowledge
TEST	Test

4.3.3.1. Set Asynchronous Balanced Mode (SABM) Command

The SABM command places two Terminal Node Comtrollers (TNC) in the asynchronous balanced mode (modulo 8). This a balanced mode of operation in which both devices are treated as equals or peers.

Information fields are not allowed in SABM commands. Any outstanding I frames left when the SABM command is issued remain unacknowledged.

The TNC confirms reception and acceptance of a SABM command by sending a UA response frame at the earliest opportunity. If the TNC is not capable of accepting a SABM command, it responds with a DM frame if possible.

4.3.3.2. Set Asynchronous Balanced Mode Extended (SABME) Command

The SABME command places two TNCs in the asynchronous balanced mode extended (modulo 128). This is a balanced mode of operation in which both devices are treated as equals or peers.

Information fields are not allowed in SABME commands. Any outstanding I frames left when the SABME command is issued remains unacknowledged.

The TNC confirms reception and acceptance of a SABME command by sending a UA response frame at the earliest opportunity. If the TNC is not capable of accepting a SABME command, it responds with a DM frame. A TNC that uses a version of AX.25 prior to v2.2 responds with a FRMR.

4.3.3.3. Disconnect (DISC) Command

The DISC command terminates a link session between two stations. An information field is not permitted in a DISC command frame.

Prior to acting on the DISC frame, the receiving TNC confirms acceptance of the DISC by issuing a UA response frame at its earliest opportunity. The TNC sending the DISC enters the disconnected state when it receives the UA response.

Any unacknowledged I frames left when this command is acted upon remain unacknowledged.

4.3.3.4. Unnumbered Acknowledge (UA) Response

The UA response frame acknowledges the reception and acceptance of a SABM(E) or DISC command frame. A received command is not actually processed until the UA response frame is sent. Information fields are not permitted in a UA frame.

4.3.3.5. Disconnected Mode (DM) Response

The disconnected mode response is sent whenever a TNC receives a frame other than a SABM(E) or UI frame while in a disconnected mode. The disconnected mode response also indicates that the TNC cannot accept a connection at the moment. The DM response does not have an information field.

Whenever a SABM(E) frame is received and it is determined that a connection is not possible, a DM frame is sent. This indicates that the called station cannot accept a connection at that time.

While a TNC is in the disconnected mode, it responds to any command other than a SABM(E) or UI frame with a DM response with the P/F bit set to “1”.

4.3.3.6. Unnumbered Information (UI) Frame

The Unnumbered Information frame contains PID and information fields and passes information along the link outside the normal information controls. This allows information fields to be exchanged on the link, bypassing flow control.

Because these frames cannot be acknowledged, if one such frame is obliterated, it cannot be recovered.

A received UI frame with the P bit set causes a response to be transmitted. This response is a DM frame when in the disconnected state, or an RR (or RNR, if appropriate) frame in the information transfer state.

4.3.3.7. Exchange Identification (XID) Frame

The Exchange Identification frame causes the addressed station to identify itself, and to provide its characteristics to the sending station. An information field is optional within the XID frame. A station receiving an XID command returns an XID response unless a UA response to a mode setting command is awaiting transmission, or a FRMR condition exists.

The XID frame complies with ISO 8885. Only those fields applicable to AX.25 are described. All other fields are set to an appropriate value. This implementation is compatible with any implementation which follows ISO 8885. Only the general-purpose XID information field identifier is required in this version of AX.25.

The information field consists of zero or more information elements. The information elements start with a Format Identifier (FI) octet. The second octet is the Group Identifier (GI). The third and forth octets form the Group Length (GL). The rest of the information field contains parameter fields.

The FI takes the value 82 hex for the general-purpose XID information. The GI takes the value 80 hex for the parameter-negotiation identifier. The GL indicates the length of the associated parameter field. This length is expressed as a two-octet binary number representing the length of the associated parameter field in octets. The high-order bits of length value are in the first of the two octets. A group length of zero indicates the lack of an associated parameter field and that all parameters assume their default values. The GL does not include its own length or the length of the GI.

The parameter field contains a series of Parameter Identifier (PI), Parameter Length (PL), and Parameter Value (PV) set structures, in that order. Each PI identifies a parameter and is one octet in length. Each PL indicates the length of the associated PV in octets, and is one octet in length. Each PV contains the parameter value and is PL octets in length. The PL does not include its own length or the length of its associated PI. A PL value of zero indicates that the associated PV is absent; the parameter assumes the default value. A PI/PL/PV set may be omitted if it is not required to convey information, or if present values for the parameter are to be used. The PI/PL/PV fields are placed into the information field of the XID frame in ascending order. There is only one entry for each PI/PL/PV field used. A parameter field containing an unrecognized PI is ignored. An omitted parameter field assumes the currently negotiated value.

The parameter fields described below represent the minimum implementation and do not preclude the negotiation of other parameters between consenting stations.

The encoding of each PI/PL/PV applicable to AX.25 is detailed in Figure 4.5. Some of the fields are defined in this standard. Only the fields discussed below are required in an implementation that complies with this version of AX.25.

Name	PI	PL	Parameter Field Element		Type	Bit	Value
Classes of Procedures	2	2	Balanced-ABM		E	1	1
			Unbalanced-NRM-Pri	*	E	2	0
			Unbalanced-NRM-Sec	*	E	3	0
			Unbalanced-ARM-Pri	*	E	4	0
			Unbalanced-ARM-Sec	*	E	5	0
			Half Duplex		E	6	0/1
			Full Duplex		E	7	0/1
			Reserved	*		8-16	0
HDLC Optional Functions	3	3	1 Reserved	*	E	1	0
			2 REJ cmd/resp		E	2	0/1
			3A SREJ cmd/resp		E	3	0/1
			4 UI cmd/resp	*	E	4	0
			5 SIM cmd/RIM resp	*	E	5	0
			6 UP cmd	*	E	6	0
			7A Basic address	*	E	7	0
			7B Extended address		E	8	1
			8 Delete I resp	*	E	9	0
			9 Delete I cmd	*	E	10	0
			10A Modulo 8		E	11	0/1
			10B Modulo 128		E	12	0/1
			11 RSET cmd	*	E	13	0
			12 TEST cmd/resp		E	14	1
			13 RD resp	*	E	15	0
			14A 16-bit FCS		E	16	1
			14B 32-bit FCS	*	E	17	0
			15A Synchronous Tx		E	18	1
			15B Start/stop Tx	*	E	19	0
			15C Start/Stop Basic Flow Ctl	*	E	20	0
			15D Start/stop Octet Transparent	*	E	21	0
			3B SREJ Multiframe	*	E	22	0
			Reserved	*	E	23-24	0
I Field Length Tx	5	N	Max I field length Tx (bits) N1*8	*	B	NA	B
I Field Length Rx	6	N	Max I field length Rx (bits) N1*8		B	NA	B
Window Size Tx	7	1	Window Size k (frames) TX	*	B B	1-7 8	0-127 0
Window Size Rx	8	1	Window Size k (frames) RX		B B	1-7 8	0-127 0
Ack Timer	9	N	Wait for Ack T1 (Msec)		B	NA	B
Retrys	10	N	Retry Count N2		B	NA	B

Figure 4.5. Parameter negotiation - parameter field elements.

- Note that Type E is a bit field and Type B is a numeric field of N octets.
- Parameter field elements marked * are defined in ISO 8885. They are shown for compatibility purposes only and are not needed to negotiate the features of this version of AX.25.

The Classes of Procedures parameter field (PI=2) serves to negotiate half- or full-duplex:

- Bit 1 is always a 1.
- Bits 2 through 5 and 8 through 16 are always zero.
- Either Bit 6 (half-duplex) or bit 7 (full-duplex), but not both, must be set. If this parameter field is not present, the current values are retained. The default is half-duplex.

The HDLC Optional Functions parameter field (PI=3) allows the negotiation of implicit reject (REJ), selective reject (SREJ), or selective reject-reject (SREJ/REJ) and modulo 8 or 128:

- Bits 1, 4-7, 9, 10, 13, 15, 17 and 19-24 are always zero.
- Bits 8, 14, 16 and 18 are always a one.
- Implicit reject is selected by setting bit 1 and resetting bit 2.
- Selective reject is selected by resetting bit 1 and setting bit 2.
- Selective reject-reject is selected by setting bit 1 and bit 2.
- Clearing both bit 1 and 2 is not allowed.
- Modulo 8 operation is selected by setting bit 11 and resetting bit 12.
- Modulo 128 operation is selected by setting bit 12 and resetting bit 11. If this parameter field is not present, the current values are retained. The default is selective reject-reject and modulo 8.

The I Field Length Receive parameter field (PI=6) allows the sending TNC to notify the receiving TNC of the maximum size of an Information field (N1) it will handle without error. A transmitting TNC may not exceed this size, but may send smaller frames. If this field is not present, the current values are retained. The default is 256 octets (2048 bits).

The Window Size Receive parameter field (PI=8) allows the sending TNC to notify the receiving TNC of the maximum size of the window (k) it will handle without error. If the TNCs are using modulo 128, this allows the negotiation of a window size less than 127 to conserve memory. If the TNCs are using selective reject or selective reject-reject, the receiving TNC is required to buffer k frames at any time. A transmitting TNC may not exceed this size, but may send fewer frames. If this field is not present, the current values are retained. The default is 4 for modulo 8 and 32 for modulo 128.

The Acknowledge Timer parameter field (PI=9) allows the negotiation of the wait for acknowledgement timer (T1). If this field is not present, the current values are retained. The default is 3000 MSec.

The Retries parameter field (PI=10) allows the negotiation of the retry count (N1). If this field is not present, the current values are retained. The default is 10 retries.

A FRMR condition may be established if the received XID command information field exceeds the maximum defined storage capability of the station, or if the receiving station is using AX.25 version 2.0 or earlier versions.

A typical XID frame is shown in Figure 4.6.

Flag	7E	Start Flag
A1	98	From Address (Example Call-SSID: NJ7P-0)
A2	94	
A3	6E	
A4	A0	
A5	40	
A6	40	
A7	E0	
A8	98	To Address (Example Call-SSID: N7LEM-0)
A9	6E	
A10	98	
A11	8A	
A12	9A	
A13	40	
A14	61	
Ctl	AF	Control Field (XID)
FI	82	Format indicator
GI	80	Group Identifier - parameter negotiation
GL	00	Group length - all of the PI/PL/PV fields (2 bytes)
	17	
PI	02	Parameter Indicator - classes of procedures
PL	02	Parameter Length
PV	00	Parameter Variable - Half Duplex, Async Balanced Mode
	20	
PI	03	Parameter Indicator - optional functions
PL	03	Parameter Length
PV	86	Parameter Variable - SREJ/REJ, extended addr 16-bit FCS, TEST cmd/resp, Modulo 128 synchronous transmit
	A8	
	02	
PI	06	Parameter Indicator - Rx I field length (bits)
PL	02	Parameter Length
PV	04	Parameter Variable - 1024 bits (128 octets)
	00	
PI	08	Parameter Indicator - Rx window size
PL	01	Parameter length
PV	02	Parameter Variable - 2 frames
PI	09	Parameter Indicator - Timer T1
PL	02	Parameter Length
PV	10	Parameter Variable - 4096 MSec
	00	
PI	0A	Parameter Indicator - Retries (N1)
PL	01	Parameter Length
PV	03	Parameter Variable - 3 retries
FCS	XX	
FCS	XX	
Flag	7E	

Figure 4.6. Typical XID frame.

4.3.3.8. Test (TEST) Frame

The Test command causes the addressed station to respond with the TEST response at the first respond opportunity; this performs a basic test of the data-link control. An information field is optional with the TEST command. If present, the received information field is returned, if possible, by the addressed station, with the TEST response. The TEST command has no effect on the mode or sequence variables maintained by the station.

A FRMR condition may be established if the received TEST command information field exceeds the maximum defined storage capability of the station. If a FRMR response is not returned for this condition, a TEST response without an information field is returned.

The station considers the data-link layer test terminated on receipt of the TEST response, or when a time-out period has expired. The results of the TEST command/response exchange are made available for interrogation by a higher layer.

4.3.3.9. FRMR Response Frame

The FRMR response is removed from the standard for the following reasons:

a) UI frame transmission was not allowed during FRMR recovery;
b) During FRMR recovery, the link could not be reestablished by the station that sent the FRMR;
c) The above functions are better handled by simply resetting the link with a SABM(E) + UA exchange;
d) An implementation that receives and process FRMRs but does not transmit them is compatible with older versions of the standard;
and e) SDL is simplified and removes the need for one state.

This version of AX.25 operates with previous versions of AX.25. It does not generate a FRMR Response frame, but handles error conditions by resetting the link.

4.4. Link Error Reporting and Recovery

Several link-layer errors can be recovered without terminating the connection. These error situations may occur as a result of transmission errors or malfunctions within the TNC.

4.4.1. TNC Busy Condition

When a TNC is temporarily unable to receive I frames (e.g., when receive buffers are full), it sends a Receive Not Ready (RNR) frame. This informs the sending TNC that the receiving TNC cannot handle any more I frames at the moment. This receiving TNC clears this condition by the sending a UA, RR, REJ or SABM(E) command frame.

4.4.2. Send Sequence Number Error

If the send sequence number, N(S), of an otherwise error-free received frame does not match the receive state variable, V(R), a send sequence error has occurred. If SREJ has been negotiated and the N(s) is in the range "greater-than V(r)" and "less-than V(r)+k," the information field is saved; otherwise it is discarded. The receiver will not acknowledge this frame or any other I frames until N(S) matches V(R).

The control field of the erroneous I frame(s) is accepted so that link supervisory functions such as checking the P/F bit can be performed. Because of this update, the retransmitted I frame may have an updated P bit and N(R).

4.4.3. Reject (REJ) Recovery

The REJ frame requests a retransmission of I frames following the detection of a N(S) sequence error. Only one outstanding "sent REJ" condition is allowed at a time. This condition is cleared when the requested I frame has been received.

A TNC receiving the REJ command clears the condition by resending all outstanding I frames (up to the window size), starting with the frame indicated in N(R) of the REJ frame.

4.4.4. Selective Reject (SREJ) Recovery

The SREJ command/response initiates more-efficient error recovery by requesting the retransmission of a single I frame following the detection of a sequence error. This is an advancement over the earlier versions in which the requested I frame was retransmitted togther with all additional I frames subsequently transmitted and successfully received.

When a TNC sends one or more SREJ commands, each with the P bit set to "0" or "1", or one or more SREJ responses, each with the F bit set to "0", and the "sent SREJ" conditions are not cleared when the TNC is ready to issue the next response frame with the F bit set to "1", the TNC sends a SREJ response with the F bit set to "1", with the same N(R) as the oldest unresolved SREJ frame.

Because an I or S format frame with the F bit set to "1" can cause checkpoint retransmission, a TNC does not send SREJ frames until it receives at least one in-sequence I frame, or it perceives by timeout that the checkpoint retransmission will not be initiated at the remote TNC.

With respect to each direction of transmission on the data link, one or more "sent SREJ" exception conditions from a TNC to another TNC may be established at a time. A "sent SREJ" exception condition is cleared when the requested I frame is received.

The SREJ frame may be repeated when a TNC perceives by timeout that a requested I frame will not be received, because either the requested I frame or the SREJ frame was in error or lost.

When appropriate, a TNC receiving one or more SREJ frames initiates retransmission of the individual I frames indicated by the N(R) contained in each SREJ frame. After having retransmitted the above frames, new I frames are transmitted later if they become available.

When a TNC receives and acts on one or more SREJ commands, each with the P bit set to “0”, or an SREJ command with the P bit set to “1”, or one or more SREJ responses each with the F bit set to “0”, it disables any action on the next SREJ response frame if that SREJ frame has the F bit set to “1” and has the same N(R) (i.e., the same value and the same numbering cycle) as a previously actioned SREJ frame, and if the resultant retransmission was made following the transmission of the P bit set to a “1”.

When the SREJ mechanism is used, the receiving station retains correctly-received I frames and delivers them to the higher layer in sequence number order.

4.4.5. Timeout Error Recovery

4.4.5.1. T1 Timer Recovery

If a transmission error causes a TNC to fail to receive (or to receive and discard) a single I frame, or the last I frame in a sequence of I frames, then the TNC does not detect a send-sequence-number error and consequently does not transmit a REJ/SREJ. The TNC that transmitted the unacknowledged I frame(s) following the completion of timeout period T1, takes appropriate recovery action to determine when I frame retransmission as described in Section 6.4.10 should begin. This condition is cleared by the reception of an acknowledgement for the sent frame(s), or by the link being reset.

4.4.5.2. Timer T3 Recovery

Timer T3 ensures that the link is still functional during periods of low information transfer. When T1 is not running (no outstanding I frames), T3 periodically causes the TNC to poll the other TNC of a link. When T3 times out, an RR or RNR frame is transmitted as a command with the P bit set, and then T1 is started. When a response to this command is received, T1 is stopped and T3 is started. If T1 expires before a response is received, then the waiting acknowledgement procedure (Section 6.4.11) is executed.

4.4.6. Invalid Frame or FCS Error

If an invalid frame is received, or a frame is received with an FCS error, that frame is discarded with no further action taken.

5. Elements for Layer-to-Layer Communication

Communication between layers is accomplished with primitives. In an abstract way, primitives represent the logical exchange of information and control between the data link and adjacent layers; they do not specify or constrain implementations.

Primitives consist of commands and their respective responses associated with the services requested from a lower layer. The general syntax of a primitive is:

XXX - Generic name - Type: Parameters

Where "XXX" designates the interface across which the primitive flows.

For this Standard, "XXX" is:

a) DL for communications between Layer 3 and the data-link layer;
b) LM for communications between the data-link layer and the link multiplexer;
c) PH for communications between the link multiplexer and the physical layer; and
d) MDL for communications between Layer 3 and the layer management.

5.1. Layer 3 Entity <—> Management Data-link State Machine

Communication between the Layer 3 Entity and the Management Data-link State Machine is characterized by three primitives:

- **MDL-NEGOTIATE Request**. The Layer 3 entity uses this primitive to request the Data-link State Machine to notify/negotiate.

- **MDL-NEGOTIATE Confirm**. The Management Data-link State Machine uses this primitive to notify the Layer 3 entity that notification/negotiation is complete.

- **MDL-ERROR Indicate**. The Management Data-link State Machine uses this primitive to notify the Layer 3 entity that notification/negotiation has failed.

5.2. Management Data-Link State Machine <—> Link Multiplexer State Machine

Communication between the Management Data-link State Machine and the Link Multiplexer State Machine is characterized by two primitives:

- **LM-DATA Request**. The Management Data-link State Machine uses this primitive to pass frames of any type (XID, UI, etc.) to the Link Multiplexer State Machine.

- **LM-DATA Indication**. The Link Multiplexer State Machine uses this primitive to pass frames of any type (XID, UI, etc.)to the Management Data-link State Machine.

5.3. Layer 3 Entity <—> Data-Link State Machine

Communication between the Layer 3 Entity and the Data-link State Machine is characterized by thirteen primitives:

- **DL-CONNECT Request**. The Layer 3 entity uses this primitive to request the establishment of a AX.25 connection.

- **DL-CONNECT Indication**. The Data-link State Machine uses this primitive to indicate that a AX.25 connection has been requested.

- **DL-CONNECT Confirm**. The Data-link State Machine uses this primitive to indicate that a AX.25 connection has been made.

- **DL-DISCONNECT Request**. The Layer 3 entity uses this primitive to request the release of a AX.25 connection.

- **DL-DISCONNECT Indication**. The Data-link State Machine uses this primitive to indicate that a AX.25 connection has been released.

- **DL-DISCONNECT Confirm**. The Data-link State Machine uses this primitive to indicate that a AX.25 connection has been released and confirmed.

- **DL-DATA Request**. The Layer 3 entity uses this primitive to request the transmission of data using connection-oriented protocol. If necessary, this frame is examined and acted upon by the segmenter.

- **DL-DATA Indication**. The reassembler uses this primitive to indicate reception of Layer 3 data using connection oriented protocol.

- **DL-UNIT-DATA Request**. The Layer 3 entity uses this primitive to request the transmission of data using connectionless protocol. If necessary, this frame is examined and acted upon by the segmenter.

- **DL-UNIT-DATA Indication**. The reassembler uses this primitive to indicate reception of Layer 3 data using connectionless protocol.

- **DL-ERROR Indication**. The Data-link State Machine uses this primitive to indicate when frames inconsistent with this protocol definition have been received. This includes short frames, frames with inconsistent parameter values, etc. The error indications are discussed in the SDL appendices.

- **DL-FLOW-OFF Request**. The Layer 3 entity uses this primitive to temporarily suspend the flow of incoming information.

- **DL-FLOW-ON Request**. The Layer 3 entity uses this primitive to resume the flow of incoming information.

5.4. Data-Link State Machine <—> Link Multiplexer State Machine

Communication between the Data-link State Machine and the Link Multiplexer State Machine is characterized by six primitives:

- **LM-SEIZE Request**. The Data-link State Machine uses this primitive to request the Link Multiplexer State Machine to arrange for transmission at the next available opportunity. The Data-link State Machine uses this primitive when an acknowledgement must be made; the exact frame in which the acknowledgement is sent will be chosen when the actual time for transmission arrives.

- **LM-SEIZE Confirm**. This primitive indicates to the Data-link State Machine that the transmission opportunity has arrived.

- **LM-RELEASE Request**. The Link Multiplexer State Machine uses this primitive to stop transmission.

- **LM-EXPEDITED-DATA Request**. The data-link machine uses this primitive to pass expedited data to the link multiplexer.

- **LM-DATA Request**. The Data-link State Machine uses this primitive to pass frames of any type (SABM, RR, UI, etc.) to the Link Multiplexer State Machine.

- **LM-DATA Indication**. The Link Multiplexer State Machine uses this primitive to pass frames of any type (SABM, RR, UI, etc.) to the Data-link State Machine.

5.5. Link Multiplexer State Machine <—> Physical State Machine

Communication between the Link Multiplexer State Machine and the Physical State Machine is characterized by eight primitives:

- **PH-SEIZE Request**. The Link Multiplexer State Machine uses this primitive before each transmission to request access to the radio channel.
- **PH-SEIZE Confirm**. The Physical State Machine uses this primitive to confirm that the channel has been seized.
- **PH-RELEASE Request**. The Link Multiplexer State Machine uses this primitive to release the radio channel.
- **PH-QUIET Indication**. The Physical State Machine uses this primitive to indicate that the channel is not busy.
- **PH-BUSY Indication**. The Physical State Machine uses this primitive to indicate that the channel is busy.
- **PH-EXPEDITED-DATA Request**. The Link Multiplexer State Machine uses this primitive to request transmission of each digipeat or expedite data frame.
- **PH-DATA Request**. The Link Multiplexer State Machine uses this primitive to request transmission of each normal frame.
- **PH-DATA Indication**. The Physical State Machine uses this primitive to provide incoming frames to the link multiplexer.

5.6. Physical State Machine <—> Hardware

Communication between the Physical State Machine and the Hardware is characterized by five primitives:

- **Acquisition of Signal**. The hardware uses this primitive to notify the Physical State Machine that modem synchronization, flag fill or frame structure have been detected.
- **Loss of Signal**. The hardware uses this primitive to notify the Physical State Machine that modem synchronization, flag fill or frame structure have been lost.
- **Frame**. The hardware uses this primitive and the Physical State Machine to pass frames to send or that have been received.
- **Turn On Transmitter**. The Physical State Machine uses this primitive to tell the hardware to key the transmitter.
- **Turn Off Transmitter**. The Physical State Machine uses this primitive to tell the hardware to unkey the transmitter.

6. Description of AX.25 Procedures

The following paragraphs describe the procedures involved in setting up, using and disconnecting a balanced link between two TNC stations.

6.1. Address Field Operation

6.1.1. Address Information

All transmitted frames have address fields conforming to Section 3.12. All frames have both the destination device and the source device addresses in the address field, with the destination address coming first. This allows many links to share the same RF channel. The destination address is always the address of the station(s) for which the frame is intended; the source address contains the address of the device that sent the frame.

If point-to-multipoint operation is desired, the destination address can be a group name or club callsign. Operation with destination addresses other than actual amateur callsigns is a subject for further study.

6.1.2. Command/Response Procedure

AX.25 implements the command/response information in the address field. The command/response information is conveyed using two bits to maintain compatibility with previous versions of AX.25.

An upward-compatible AX.25 TNC communicating with a distant TNC determines if the latter is using an older version of this protocol by testing the command/response bit information located in bit 7 of the SSID octets of both the destination and source address subfields. If both C bits are set to “0”, then the distant device is using the older protocol. The newer version of the protocol always has one of these two bits set to “1” and the other bit set to “0”, depending on whether the frame is a command or a response.

The command/response information is encoded into the address field as shown in Figure 6.1. Implementations of AX.25 prior to version 2.0 defined these bits to be either both “0” or both “1”.

Frame TypeDest.	SSID C-Bit	Source SSID C-Bit
Previous versions	0	0
Command (V.2.X)	1	0
Response (V.2.X)	0	1
Previous versions	1	1

Figure 6.1. Command/Response encoding.

Because all frames are considered to be either commands or responses, a device always has one of the bits set to “1” and the other bit set to “0”.

The use of the command/response information in AX.25 allows S frames to be either commands or responses. This arrangement helps maintain proper control over the link during the information transfer state.

6.2. Poll/Final (P/F) Bit Procedures

The response frame returned by a TNC depends on the previous command received, as described in the following paragraphs.

The next response frame returned by the TNC to a SABM(E) or DISC command with the P bit set to "1" is a UA or DM response with the F bit set to "1".

The next response frame returned to an I frame with the P bit set to "1", received during the information transfer state, is an RR, RNR or REJ response with the F bit set to "1".

The next response frame returned to a supervisory command frame with the P bit set to "1", received during the information transfer state, is an RR, RNR or REJ response frame with the F bit set to "1".

The next response frame returned to a S or I command frame with the P bit set to "1", received in the disconnected state, is a DM response frame with the F bit set to "1".

The P bit is used in conjunction with the timeout recovery condition discussed in Section 4.5.5.

When not used, the P/F bit is set to "0".

6.3. Procedures For Link Set-Up and Disconnection

6.3.1. AX.25 Link Connection Establishment

To connect to a distant TNC, the originating TNC sends a SABM command frame to the distant TNC and starts its T1 timer. If the distant TNC exists and accepts the connect request, it responds with a UA response frame and resets all of its internal state variables (V(S), V(A) and V(R)). Reception of the UA response frame by the originating TNC causes it to cancel the T1 timer and set its internal state variables to "0".

If the distant TNC doesn't respond before T1 times out, the originating TNC resends the SABM frame and starts T1 running again. The originating TNC tries to establish a connection until it has tried unsuccessfully N2 times. N2 is defined in Section 6.7.2.3.

If the distant TNC receives a SABM command and cannot enter the indicated state, it sends a DM frame.

When the originating TNC receives a DM response to its SABM(E) frame, it cancels its T1 timer and does not enter the information-transfer state.

The originating TNC sending a SABM(E) command ignores and discards any frames except SABM, DISC, UA and DM frames from the distant TNC.

In response to a received SABM(E), frames other than UA and DM are sent only after the link is set up and if no outstanding SABM(E) exists.

6.3.2. Parameter Negotiation Phase

Parameter negotiation occurs at any time. It is accomplished by sending the XID command frame and receiving the XID response frame. Implementations of AX.25 prior to version 2.2 respond to an XID command frame with a FRMR response frame. The TNC receiving the FRMR uses a default set of parameters compatible with previous versions of AX.25.

The receipt of an XID response from the other station establishes that both stations are using AX.25 version 2.2 or higher and enables the use of the segmenter/reassembler and selective reject.

This version of AX.25 implements the negotiation or notification of six AX.25 parameters. Notification simply tells the distant TNC some limit that cannot be exceeded. The distant TNC can choose to use the limit or some other value that is within the limits. Notification is used with the Window Size Receive (k) and Information Field Length Receive (N1) parameters. Negotiation involves both TNCs choosing a value that is mutually acceptable. The XID command frame contains a set of values acceptable to the originating TNC. The distant TNC chooses to accept the values offered, or other acceptable values, and places these values in the XID response. Both TNCs set themselves up based on the values used in the XID response. Negotiation is used by Classes of Procedures, HDLC Optional Functions, Acknowledge Timer and Retries.

The Classes of Procedure parameter field (PI=2) negotiates half- or full-duplex operation. This reverts to half-duplex if either TNC cannot support full-duplex (i.e., if the XID command requests full-duplex and the receiving TNC can only support half-duplex, it sets the value to half-duplex in the XID response. If this parameter field is not present, the default half-duplex operation is selected.

The HDLC Optional Functions parameter field (PI=3) allows the negotiation of implicit reject (REJ), selective reject (SREJ), or selective reject-reject (SREJ/REJ), and modulo 8 or 128. Function reverts to the lesser of the selection offered in the XID command and XID response frames. Ordering is (highest to lowest): selective reject-reject, selective reject and implicit reject: Modulo 128 and modulo 8. If this parameter field is absent, the default function selective reject and modulo 8 are selected.

The I Field Length Receive parameter field (PI=6) allows the sending TNC to notify the receiving TNC of the maximum size of an Information field (N1) it will handle without error. A transmitting TNC may not exceed this size, but may send smaller frames.

The Window Size Receive parameter field (PI=8) allows the sending TNC to notify the receiving TNC of the maximum size of the window (k) it will handle without error. If the TNCs are using modulo 128, this allows the negotiation of a window size less than 127 to conserve memory. If the TNCs are using selective reject or selective reject-reject, the receiving TNC is required to buffer k frames at any time.

The Acknowledge Timer parameter field (PI=9) allows the negotiation of the “Wait for Acknowledgement” timer (T1). Function reverts to the greater of the values offered in the XID command and XID response frames.

The Retries parameter field (PI=10) allows the negotiation of the retry count (N1). Function reverts to the greater of the values offered in the XID command and XID response frames.

Defaults for the negotiated parameters for use with a previous version of AX.25 are:

Set Half Duplex
Set Implicit Reject
Modulo = 8
I Field Length Receive = 2048 bits
Window Size Receive = 4
Acknowledge Timer = 3000 MSec
Retries = 10

Defaults for the negotiated parameters for use with this version of AX.25 are:

Set Half Duplex
Set Selective Reject
Modulo = 8
I Field Length Receive = 2048 bits
Window Size Receive = 7
Acknowledge Timer = 3000 MSec
Retries = 10

6.3.3. Information-Transfer Phase

After establishing a link connection, the TNC enters the information-transfer state. In this state, the TNC accepts and transmits I and S frames according to the procedure outlined in Section 6.4.

If the TNC receives a SABM(E) command while in the information-transfer state, it follows the resetting procedure outlined in Section 6.5.

6.3.4. Link Disconnection

While in the information-transfer state, either TNC may indicate a request to disconnect the link by transmitting a DISC command frame and starting timer T1.

After receiving a valid DISC command, the TNC sends a UA response frame and enters the disconnected state. After receiving a UA or DM response to a sent DISC command, the TNC cancels timer T1 and enters the disconnected state.

If a UA or DM response is not correctly received before T1 times out, the DISC frame is sent again and T1 is restarted. If this happens N2 times, the TNC enters the disconnected state.

6.3.5. Disconnected State

In the disconnected state, a TNC monitors received commands, reacts to the receipt of a SABM(E) as described in Section 6.3.1, and transmits a DM frame in response to a DISC command.

In the disconnected state, a TNC may initiate a link set up as outlined in connection establishment (Section 6.3.1). It may also respond to the receipt of a SABM(E) command and establish a connection, or it may refuse the SABM(E) and send a DM instead.

Any TNC receiving a command frame other than a SABM(E) or UI frame with the P bit set to "1" responds with a DM frame with the F bit set to "1". The offending frame is ignored.

When the TNC enters the disconnected state after an error condition, or if an internal error has resulted in the TNC being in the disconnected state, the TNC indicates this by sending a DM response rather than a DISC frame and follows the link disconnection procedure outlined in Section 6.3.4. The TNC may then try to reestablish the link using the link set up procedure outlined in Section 6.3.1.

6.3.6. Collision Recovery

6.3.6.1. Collisions in a Half-Duplex Environment

Collisions of frames in a half-duplex environment are taken care of by the retry nature of the T1 timer and retransmission count variable. No other special action is required.

6.3.6.2. Collisions of Unnumbered Commands.

If sent and received SABM(E) or DISC command frames are the same, both TNCs send a UA response at the earliest opportunity, and both devices enter the indicated state.

If sent and received SABM(E) or DISC commands are different, both TNCs enter the disconnected state and transmit a DM frame at the earliest opportunity.

6.3.6.3. Collision of a DM with a SABM(E) or DISC

When an unsolicited DM response frame is sent, a collision between it and a SABM(E) or DISC may occur. In order to prevent this DM from being misinterpreted, all unsolicited DM frames are transmitted with the F bit set to "0". All SABM(E) and DISC frames are sent with the P bit set to "1". This prevents confusion when a DM frame is received.

6.3.7. Connectionless Operation

An additional type of operation exists in amateur radio that is not feasible using Layer 2 connections. This is the "round-table" operation, in which several amateurs may be engaged in one conversation. This type of operation cannot be accommodated by current AX.25 link-layer connections.

The way round-table activity is implemented is technically outside the AX.25 connection, although it still uses the AX.25 frame structure.

AX.25 uses a special frame for this operation, the Unnumbered Information (UI) frame. In this type of operation, the destination address has a code word installed in it that prevents users of that specific round-table from seeing all frames going through the shared RF medium. For example, if a group of amateurs are engaged in a round-table discussion about packet radio, they could put "PACKET" in the destination address; they will receive frames only from others in the same discussion. An added advantage of the use of AX.25 in this manner is that the source of each frame is in the source address subfield; software could be written to automatically display who is making what comments.

Since this mode is connectionless, there are no requests for retransmissions of bad frames. Without the handshaking activity of a point-to-point connection, collisions may also occur, with the potential of losing the frames that collided.

6.4. Procedures for Information Transfer

Once a connection has been established as outlined above, both TNCs can accept I, S and U frames.

6.4.1. Sending I Frames

Whenever a TNC has an I frame to transmit, it sends the I frame with the N(S) of the control field equal to its current send state variable V(S). After the I frame is sent, the send state variable is incremented by one. If timer T1 is not running, it is started. If timer T1 is running, it is restarted.

The TNC does not transmit any more I frames if its send state variable equals the last received N(R) from the other side of the link plus k. If the TNC sent more I frames, the flow control window would be exceeded and errors could result.

If a TNC is in a busy condition, it may still send I frames as long as the distant TNC is not also busy.

6.4.2. Receiving I Frames

The reception of I frames that contain zero-length information fields is reported to the next layer; no information field will be transferred.

6.4.2.1. Not Busy

If a TNC receives a valid I frame (one with a correct FCS and whose send sequence number equals the receiver's receive state variable) and is not in the busy condition, it accepts the received I frame, increments its receive state variable, and acts in one of the following manners:

a) If it has an I frame to send, that I frame may be sent with the transmitted N(R) equal to its receive state variable V(R) (thus acknowledging the received frame). Alternately, the TNC may send an RR frame with N(R) equal to V(R), and then send the I frame.

or b) If there are no outstanding I frames, the receiving TNC sends an RR frame with N(R) equal to V(R). The receiving TNC may wait a small period of time before sending the RR frame to be sure additional I frames are not being transmitted.

6.4.2.2. Busy

If the TNC is in a busy condition, it ignores any received I frames without reporting this condition, other than repeating the indication of the busy condition.

If a busy condition exists, the TNC receiving the busy condition indication polls the sending TNC periodically until the busy condition disappears.

A TNC may poll the busy TNC periodically with RR or RNR frames with the P bit set to "1".

6.4.3. Priority Acknowledge

This version of AX.25 implements the priority acknowledgement procedure. This feature precludes a non-priority frame from being transmitted during slot 0, the time when the TNC receiving the previous frame would be expected to send an acknowledgement.

6.4.4. Reception of Out-of-Sequence Frames

6.4.4.1. Implicit Reject (REJ)

When an I frame is received with a correct FCS but its send sequence number N(S) does not match the current receiver's receive state variable, the frame is discarded. A REJ frame is sent with a receive sequence number equal to one higher than the last correctly received I frame if an uncleared N(S) sequence error condition has not been previously established. The received state variable and poll bit of the discarded frame is checked and acted upon, if necessary.

This mode requires no frame queueing and frame resequencing at the receiver. However, because the mode requires transmission of frames that may not be in error, its throughput is not as high as selective reject. This mode is ineffective on systems with long round-trip delays and high data rates.

6.4.4.2. Selective Reject (SREJ)

When an I frame is received with a correct FCS but its send sequence number N(S) does not match the current receiver's receive state variable, the frame is retained. SREJ frames are sent with a receive sequence number equal to the value N(R) of the missing frame, and P=1 if an uncleared SREJ condition has not been previously established. If an SREJ condition is already pending, an SREJ will be sent with P=0. The received state variable and poll bit of the received frame are checked and acted upon, if necessary.

This mode requires frame queueing and frame resequencing at the receiver. The holding of frames can consume precious buffer space, especially if the user device has limited memory available and several active links are operational.

6.4.4.3. Selective Reject-Reject (SREJ/REJ)

When an I frame is received with a correct FCS but its send sequence number N(S) does not match the current receiver's receive state variable, and if N(S) indicates 2 or more frames are missing, a REJ frame is transmitted. All subsequently received frames are discarded until the lost frame is correctly received. If only one frame is missing, a SREJ frame is sent with a receive sequence number equal to the value N(R) of the missing frame. The received state variable and poll bit of the received frame are checked and acted upon. If another frame error occurs prior to recovery of the SREJ condition, the receiver saves all frames received after the first errored frame and discards frames received after the second errored frame until the first errored frame is recovered. Then, a REJ is issued to recover the second errored frame and all subsequent discarded frames.

6.4.5. Reception of Incorrect Frames

When a TNC receives a frame with an incorrect FCS, an invalid frame, or a frame with an improper address, that frame is discarded.

6.4.6. Receiving Acknowledgement

Whenever an I or S frame is correctly received, even in a busy condition, the N(R) of the received frame is checked to see if it includes an acknowledgement of outstanding sent I frames. The T1 timer is canceled if the received frame actually acknowledges previously unacknowledged frames. If the T1 timer is canceled and there are still some frames that have been sent that are not acknowledged, T1 is started again. If the T1 timer expires before an acknowledgement is received, the TNC proceeds with the retransmission procedure outlined in Section 6.4.11.

6.4.7. Receiving REJ

After receiving a REJ frame, the transmitting TNC sets its send state variable to the same value as the REJ frame's received sequence number in the control field. The TNC then retransmits any I frame(s) outstanding at the next available opportunity in accordance with the following:

a) If the TNC is not transmitting at the time and the channel is open, the TNC may begin retransmission of the I frame(s) immediately.

b) If the TNC is operating on a full-duplex channel transmitting a UI or S frame when it receives a REJ frame, it may finish sending the UI or S frame and then retransmit the I frame(s).

c) If the TNC is operating in a full-duplex channel transmitting another I frame when it receives a REJ frame, it may abort the I frame it was sending and start retransmission of the requested I frames immediately.

d) The TNC may send just the one I frame outstanding, or it may send more than the one indicated if more I frames followed the first unacknowledged frame, provided that the total to be sent does not exceed the flow-control window (k frames).

If the TNC receives a REJ frame with the poll bit set, it responds with either an RR or RNR frame with the final bit set before retransmitting the outstanding I frame(s).

6.4.8. Receiving an SREJ

After receiving a SREJ frame, the transmitting TNC retransmits the individual I frame indicated by the N(R) contained in the SREJ at the next available opportunity. After retransmitting the frame above, new I frames may be retransmitted subsequently if they become available. If the P bit was set, then all frames up to N(R)-1 are acknowledged.

6.4.9. Receiving an RNR Frame

Whenever a TNC receives an RNR frame, it stops transmitting I frames until the busy condition is cleared. If timer T3 expires after the RNR was received, an RR or RNR command with the P bit set is sent to poll the distant TNC of its status; then timer T1 is started. If an RNR frame is received in response to this poll, T1 is stopped and T3 is started again. If no response is received before T1 expires, the waiting acknowledgment procedure (Section 6.4.11) is performed. If an RR frame is received in response to the poll, then T1 is stopped and the busy condition cleared.

6.4.10. Sending a Busy Indication

Whenever a TNC enters a busy condition, it indicates this by sending an RNR response at the next opportunity. While the TNC is in the busy condition, it may receive and process S frames. If a received S frame has the P bit set to "1", the TNC sends an RNR frame with the F bit set to "1" at the next possible opportunity. To clear the busy condition, the TNC sends either an RR or REJ frame with the received sequence number equal to the current receive state variable, depending on whether the last received I frame was properly received or not.

6.4.11. Waiting Acknowledgement

If the originating TNC's timer T1 expires while awaiting the distant TNC's acknowledgement of an I frame transmitted, the originating TNC restarts timer T1 and transmits an appropriate supervisory command frame (RR or RNR) with the P bit set.

If the TNC correctly receives a supervisory response frame with the F bit set and with an N(R) within the range from the last N(R) received to the last N(S) sent plus one, the TNC restarts timer T1 and sets its send state variable V(S) to the received N(R). It may then resume with I frame transmission or retransmission, as appropriate.

If, on the other hand, the TNC correctly receives a supervisory response frame with the F bit not set, or an I frame or supervisory command frame, and with an N(R) within the range from the last N(R) received to the last N(S) sent plus one, the TNC does not restart timer T1; it uses the received N(R) as an indication of acknowledgement of transmitted I frames up to and including I frame numbered N(R)-1.

If timer T1 expires before a supervisory response frame with the F bit set is received, the TNC retransmits an appropriate supervisory command frame (RR or RNR) with the P bit set. After N2 attempts to get a supervisory response frame with the F bit set from the distant TNC, the originating TNC initiates a link resetting procedure as described in Section 6.5.

6.5. Resetting Procedure

The link resetting procedure initializes both directions of data flow after a unrecoverable error has occurred. This resetting procedure is used only in the information-transfer state of an AX.25 link.

A TNC initiates a reset procedure whenever it receives an unexpected UA response frame, or after receipt of a FRMR frame from a TNC using an older version of the protocol.

A TNC resets the link by sending a SABM(E) frame and starting timer T1. After receiving a SABM(E) frame from the TNC to which it was previously connected, the receiver of a SABM(E) frame sends a UA frame back at the earliest opportunity, sets its send and receive state variables V(S) and V(R) to "0" and stops T1, unless it has sent a SABM(E) or DISC itself. If the UA frame is correctly received by the first TNC, it resets its send and receive state variables V(S) and V(R), and stops timer T1. Any busy condition that previously existed is also cleared.

If a DM response is received, the TNC enters the disconnected state and stops timer T1. If timer T1 expires before a UA or DM response frame is received, the SABM(E) is retransmitted and timer T1 restarted. If timer T1 expires N2 times, the TNC enters the disconnected state. Any previously existing link conditions are cleared.

Other commands or responses received by the TNC before completion of the reset procedure are discarded.

One TNC may request that the other TNC reset the link by sending a DM response frame. After the DM frame is sent, the sending TNC then enters the disconnected state.

6.6. Disassembler/Reassembler

The segmenter/reassembler procedure is only enabled if both stations on the link are using AX.25 version 2.2 or higher. The use of the segmenter/reassembler allows the transmission of packets longer than N1 in a simple and clean manner. This adds less than one percent overhead for the standard N1 of 256 bytes. It also adds the ability to send large Level 3 data entities such as IP datagrams as single entities over AX.25.

The segmenter is a simple process that divides long data units into smaller segments for transmission, attaching a two-octet header to each segment. At the receiving end, segments are reassembled into the original data unit. Overhead is kept to a minimum throughout; steps are taken to prevent deadlock situations from arising in the buffer management of both stations on the link. The header is illustrated in Figure 6.2.

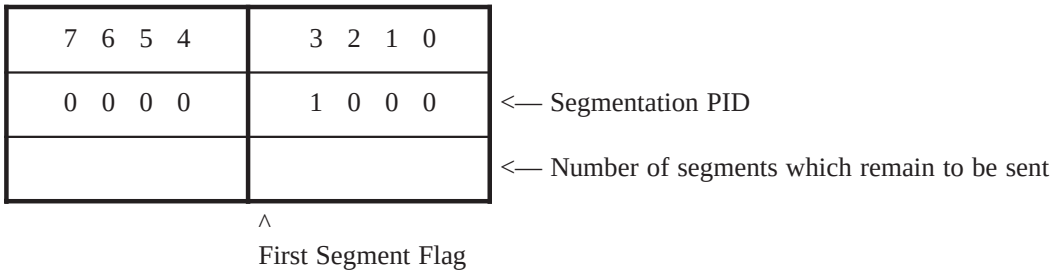


Figure 6.2 Segment header format.

The reassembler can tell when a segmented frame is received by the PID. If the first segment flag is set, then the amount of buffer space required for the entire frame can be calculated and allocated. If using the segmenter over connectionless service and a segment is lost, error recovery is not done by the reassembler. An error is passed to Layer 3; it is up to Layer 3 to recover.

6.7. List of System Defined Parameters

6.7.1. Timers

Thirteen timers maintain the integrity of the AX.25 Layer 2 connection and are discussed in the following subsections.

6.7.1.1. Acknowledgment Timer T1

T1, the Acknowledgement Timer, ensures that a TNC does not wait indefinitely for a response to a frame it sends. This timer cannot be expressed in absolute time; the time required to send frames varies greatly with the signaling rate used at Layer 1. T1 should take at least twice the amount of time it would take to send maximum length frame to the distant TNC and get the proper response frame back from the distant TNC. This allows time for the distant TNC to do some processing before responding.

If Layer 2 repeaters are used, the value of T1 should be adjusted according to the number of repeaters through which the frame is being transferred.

6.7.1.2. Response Delay Timer T2

T2, the Response Delay Timer, may optionally be implemented by the TNC to specify a maximum amount of delay to be introduced between the time an I frame is received and the time the resulting response frame is sent. This delay is introduced to allow a receiving TNC to wait a short period of time to determine if more than one frame is being sent to it. If more frames are received, the TNC can acknowledge them at once (up to seven), rather than acknowledging each individual frame. The use of timer T2 is not required; it is simply recommended to improve channel efficiency. Note that to achieve maximum throughput on full-duplex channels, acknowledgments should not be delayed beyond k/2 frames. The k parameter is defined in Section 6.8.2.3.

6.7.1.3. Inactive Link Timer T3

T3, the Inactive Link Timer, maintains link integrity whenever T1 is not running. It is recommended that whenever there are no outstanding unacknowledged I frames or P-bit frames (during the information-transfer state), an RR or RNR frame with the P bit set to "1" be sent every T3 time units to query the status of the other TNC. The period of T3 is locally defined, and depends greatly on Layer 1 operation. T3 should be greater than T1; it may be very large on channels of high integrity.

6.7.1.4. Repeater Hang Timer T100 (AXHANG)

T100, the Repeater Hang Timer, tracks the amount of time an audio repeater will keep its transmitter keyed after it stops receiving. This timer can increase channel efficiency when an audio repeater is used. If the repeater's transmitter remains keyed, it is not necessary to add AXDELAY to the transmitter key-up time.

6.7.1.5. Priority Window Timer T101 (PRIACK)

T101, the Priority Window Timer, prevents stations from transmitting non-priority frames during the first available transmission time slot. The first transmission time slot is reserved for priority frames (acknowledgments and digipeat frames).

6.7.1.6. Slot Time Timer T102 (p-persistence)

T102, the Slot Time Timer, randomly delays stations before they begin transmitting immediately after the channel becomes clear. This helps prevent several stations from beginning to transmit at the same time and causing collisions.

6.7.1.7. Transmitter Startup Timer T103 (TXDELAY)

T103, the Transmitter Startup Timer, allows time to be reasonably certain that the transmitter has properly ramped up and is ready to transmit after being keyed, before any frames are sent.

6.7.1.8. Repeater Startup Timer T104 (AXDELAY)

T104, the Repeater Startup Timer, allows time to be reasonably certain that audio repeaters have had time to start their transmitters before frames are sent.

6.7.1.9. Remote Receiver Sync Timer T105

T105, the Remote Receiver Sync Timer, introduces additional delay time after TXDELAY, if needed, to allow a remote receiver to sync up before transmitting frames.

6.7.1.10. Ten Minute Transmission Limit Timer T106

T106, the Ten Minute Transmission Limit Timer, ensures that the transmitter is not keyed for more than ten minutes.

6.7.1.11. Anti-Hogging Limit Timer T107

T107, the Anti-Hogging Limit Timer, prevents a station from monopolizing the channel.

6.7.1.12. Receiver Startup Timer T108

T108, the Receiver Startup Timer, allows sufficient time to be reasonably certain that the receiver is monitoring the status of the channel (busy or not) after unkeying the transmitter, before attempting to start transmitting again.

6.7.1.13. Next Segment Timer TR210

T210, the Next Segment Timer, ensures that the reassembler doesn't wait forever for the next segment of a segmented frame.

6.7.2. Parameters

6.7.2.1. Maximum Number of Octets in an I Field (N1)

The default maximum number of octets allowed in the I field is 256. This variable is negotiable between end stations. The I field is an integral number of octets.

6.7.2.2. Maximum Number of Retries (N2)

The maximum number of retries is used in conjunction with the T1 timer.

6.7.2.3. Maximum Number of I Frames Outstanding (k)

The maximum number of I frames outstanding at a time is seven (modulo 8) or 127 (modulo 128).

Appendix A

Glossary

(Note: This appendix is not part of the protocol.)

ARRL	American Radio Relay League
AX.25	Link Access Protocol - Amateur
C/R	Command/Response Bits
CSMA	Carrier Sense Multiple Access
DISC	Disconnect Frame
DL	Communications between Layer 3 entity and data link layer
DLSAP	Data Link Service Access Point
DM	Disconnect Mode Frame
FCS	Frame Check Sequence
HDLC	High Level Data Link Control
I	Information Frame
ISO	International Standards Organization
LM	Communications between link multiplexer and data-link Layer
MDL	Communications between management entity and data-link Layer
N(r)	Receive Sequence Number
N(s)	Send Sequence Number
OSI	Open Systems Interconnect
P/F	Poll/Final bit
PH	Communications between physical Layer and link multiplexer
PID	Protocol Identifier
REJ	Reject Frame
RNR	Receiver Not Ready Frame
RR	Receiver Ready Frame
SABM	Set Asynchronous Balanced Mode Frame
SABME	Set Asynchronous Balanced Mode Extended Frame
SREJ	Selective Reject Frame
SSID	Secondary Station Identifier
TAPR	Tucson Amateur Packet Radio
TEST	Test Frame
TNC	Terminal Node Controller
UA	Unnumbered Acknowledge Frame
UI	Unnumbered Information Frame
V(a)	Acknowledge State Variable
V(r)	Receive State Variable
V(s)	Send State Variable
XID	Exchange Identification Frame

Appendix B

References

(Note: This appendix is not part of the protocol.)

Black, Uyless D., 1993, "Data-Link Protocols."

CCITT Recommendation X.25, "Interface Between Data Terminal Equipment (DTE) and Data-Circuit Terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks."

CCITT Recommendation Q.920/Q.921, Blue Book, 1989, "Digital Subscriber Signaling System No. 1 (DSS 1), Data link Layer."

Fox, Terry L., October 1984, "AX.25 Amateur Packet-Radio Link-Layer Protocol."

ISO 3309, 4th edition, 1 June 91, "Information Technology - Telecommunications and Information Exchange Between Systems — High-level Data-Link Control (HDLC) Procedures - Frame Structure."

ISO 4335, 4th edition, 15 September 91 (with Amendment 4), "Information Technology — Telecommunications and Information Exchange Between Systems — High-level Data-Link Control (HDLC) Procedures - Elements of Procedures."

ISO 7776, 1st edition, 15 December 86, "Information Processing Systems — Data Communication — High-level Data-Link Control Procedures — Description of the X.25 LAPB-Compatible DTE Data-Link Procedures."

ISO 7809, 2nd edition, 15 September 91 (with Amendments 5, 6, and 7), "Information Technology — Telecommunications and Information Exchange Between Systems — High-level Data-Link Control (HDLC) Procedures — Classes of Procedures."

ISO 8885, 2nd edition, 1 June 91 (with Amendment 3, 4 and 5), "Information Technology — Telecommunications and Information Exchange Between Systems — High-level Data-Link Control (HDLC) Procedures — General Purpose XID Frame Information Field Content and Format."

ISO 8886, 1st edition, 15 June 91, "Information Technology — Telecommunications and Information Exchange Between Systems — Data-Link Service Definition for Open Systems Interconnection."

Scace, Eric L., K3NA, "Various AX.25 State Machines — 7th Computer Networking Conference — LAPA Link Access Protocol Specification."

Appendix C1

Introduction to System Description Language

C1.1. Principles of Extended Finite State Machines

An extended finite state machine models the operation of one party on a communications channel. The condition of the machine is described by its state, a resting condition where the machine awaits input signals from either the application or from remote parties on the communications channel.

Whenever an input signal arises, the machine is triggered to execute a series of operations. These operations may include calculations, the generation of signals to the remote parties on the communications channel, and the generation of signals to the application. The sequence of operations concludes with the machine reaching again a resting state (the same state or a different one).

The entire sequence of operations performed by the machine is atomic. For modeling purposes, the sequence is considered to occur instantaneously, and can not be interrupted by any further event. The processing of such further events does not begin until the machine has completed the sequence of operations and has reached a resting state.

Signals may be sent between machines within the same equipment, or via the communications channel(s) to state machines in other equipments. These signals are often called "primitives."

Extended finite state machines differ from their cousins, finite state machines, in three respects relevant to AX.25:

- They can maintain internal variables, such as flags, sequence counters, and lists.

- Timers may be set. The expiration of a timer at a later time generates an input signal to trigger the machine to execute a specific series of operations. Timers may be stopped before they expire.

- Internal queues may be maintained. The queues are used to retain input signals (or other information) for processing at a later, more appropriate time.

C1.2. SDL Symbol Definition

Figure C1.1 defines the symbols used in all of the SDL graphic descriptions. The reader may find it helpful to review the text below and the figure along with an actual SDL description from one of the appendices which follow. The SDL descriptions combine together the operations described by the various symbols into a sequence that is read down the page.

The state symbol denotes the resting states of the extended finite state machine. Each state is numbered and named. The sequence number simply indicates the order in which the states are drawn in the SDL. All the permitted sequences of operations from a given state originate below the corresponding state symbol. For convenience, each SDL machine is accompanied by a summary page that lists, among other things, all of the state names and their corresponding numbers.

Input signal reception (primitive) symbols have notches on either the left or right side. By convention, inputs with the notch on the left are from higher or equal layer state machines; inputs with the notch on the right are from the lower layer state machine. The name of the input primitive is labeled within the symbol. The SDL machine summary page lists all of the input primitives by name and source.

In addition, the left-notch input signal symbol is used for timer expiration. The number of the expired timer is written inside the symbol. All timers are numbered, by convention, with indications beginning "T" and then (usually) a three-digit number. The "hundreds" digit indicates the layer number of the Open Systems Interconnection (OSI) model at which the state machine resides; e.g., T1xx timers are physical layer, T2xx timers are data-link layer, etc. However, to prevent confusion, the present indicators T1 and T3 are used for AX.25 timers. The SDL machine summary page lists all of the timers by their indicator, and gives a brief description of the purpose of each timer.

Similarly, output signal reception (primitive) symbols have pointers on either the left or right side. Output symbols pointing to the left are outputs to the higher or equal layer state machines; outputs pointing to the right are to the lower layer state machine. The name of the output primitive is written within the symbol. The SDL machine summary page lists all of the output primitives by name and destination.

Internal signal symbols are used to post items onto queues (points to left) and to trigger the state machine when something is waiting on the queues to be popped off (notch to left). Each internal signal has a description label identifying which queue is involved, and what material is being posted or popped. The SDL machine summary page describes each internal queue used by the state machine.

The save symbol is used to indicate that a particular input event does not cause operations to be done in the present state. Instead, that particular input event is "saved" until the state machine (triggered by other events) has reached a new and different state.

The processing description symbol contains within it a description of internal action(s) executed by the state machine. Examples of these actions are starting and stopping timers, setting and clearing flags, and setting values into variables.

The test symbol is used for branching. The text written within the symbol is posed as a question, and then the appropriate branch is taken.

The subroutine symbol is used to encapsulate frequently used sequences of steps; the name of the subroutine is written within the symbol. The expansion of the subroutine is listed at the end of the SDL machine description. Subroutine expansions begin with a subroutine start symbol, flow down the page through the specified sequence of operations, and end with the return-from-subroutine symbol. Note that subroutines are not permitted to contain states, nor are they permitted to branch into different return legs. Each subroutine has a single point of return.

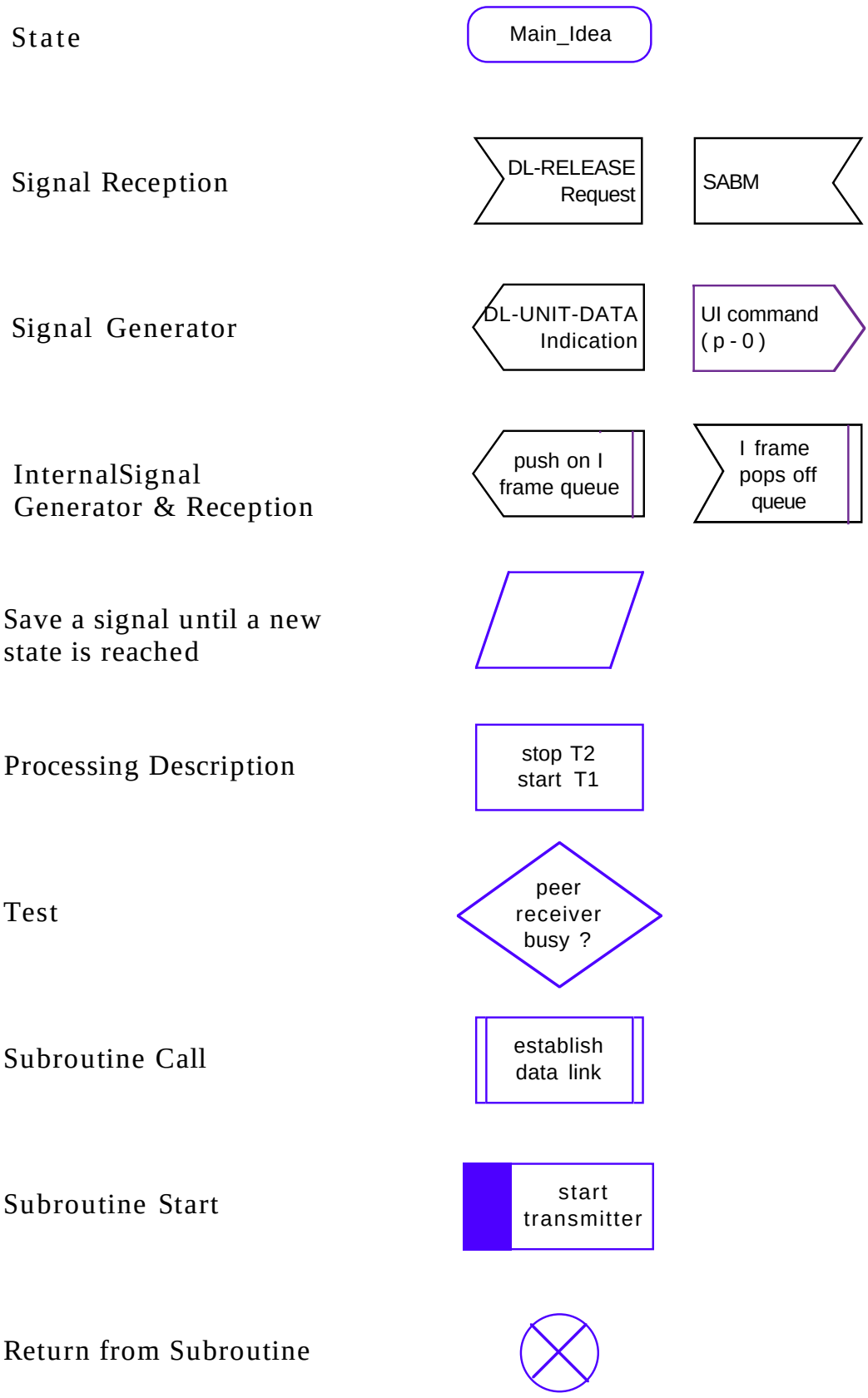


Figure C1.1. SDL examples C1-C4.

Appendix C2a

Simplex Physical Layer State Machines

C2a.1. Interaction with the Link Multiplexer

The Link Multiplexer State Machine directs the operation of the simplex Physical State Machine through the following physical (PH) primitives:

- **PH-SEIZE Request**. This primitive requests the simplex state machine to begin transmitting at the next available opportunity. When that opportunity has been identified (according to the CSMA/p-persistence algorithm included within), the transmitter started, a parameterized window provided for the startup of a conventional repeater (if required), and a parameterized time allowed for the synchronization of the remote station's receiver (known as TXDELAY in most implementations), then a PH-SEIZE Confirm primitive is returned to the link multiplexer.
- **PH-DATA Request**. This primitive from the Link Multiplexer State Machine provides a AX.25 frame of any type (UI, SABM, I, etc.) that is to be transmitted. An unlimited number of frames may be provided. If the transmission exceeds the 10-minute limit or the anti-hogging time limit, the half-duplex Physical State Machine automatically relinquishes the channel for use by the other stations. The transmission is automatically resumed at the next transmission opportunity indicated by the CSMA/p-persistence contention algorithm.
- **PH-RELEASE Request**. The Link Multiplexer State Machine provides this primitive when a submitted sequence of frames to be transmitted on behalf of a particular AX.25 connection has been completed. The simplex Physical State Machine will piggyback any straggling digipeat frames (if time permits) and then relinquish the channel.
- **PH-EXPEDITED-DATA Request**. This primitive from the Link Multiplexer State Machine provides the AX.25 frame that is to be transmitted immediately. The simplex Physical State Machine gives preference to priority frames over normal frames, and will take advantage of the PRIACK window. Priority frames can be provided by the link multiplexer at any time; a PH-SEIZE Request and subsequent PH Release Request are not employed for priority frames.
- **PH-DATA Indication**. During reception, the simplex Physical State Machine provides each AX.25 frame to the link multiplexer in a Frame primitive (Section C2a.2). No analysis is done on the frame by the simplex Physical State Machine; it does not examine lengths, the frame check sequence, the need for digipeating, or any other content of the frame; these responsibilities are carried out by the higher level state machines.
- **PH-BUSY Indication**. The simplex Physical State Machine provides this primitive whenever the channel becomes busy. "Busy" here means the detection of a valid modem synchronization sequence, HDLC Flags or HDLC frames — not the detection of an FM carrier on a two-meter radio. An indication of busy is provided to the higher layer state machines so that various timers that supervise the AX.25 connection can be suspended. This avoids the undesirable situation on a busy channel where AX.25, having sent data and expecting and acknowledgement, times out and attempts retransmissions — and the only reason an acknowledgement was not received was because the remote station did not yet have a chance to make a transmission. Since the channel is simplex, this primitive is also provided when the simplex Physical State Machine starts transmitting.
- **PH-QUIET Indication**. The simplex Physical State Machine provides this primitive whenever the channel becomes quiet. Since the channel is simplex, this primitive is also provided when the simplex Physical State Machine finishes transmitting.

C2a.2. Interface to the Hardware

As the lowest layer state machine in the standard, the physical layer state machine is envisioned to manipulate a typical radio tranceiver. It employs a number of primitives, described below:

• **Turn On Transmitter**. This primitive keys the transceiver's PTT line.

• **Turn Off Transmitter**. This primitive unkeys the transceiver's PTT line.

• **Frame**. This primitive passes data for actual transmission of a frame. Although SDL representation of bit-by-bit transmission of the contents of a frame are possible, they are not used here because the additional complexity was not required. The Frame primitive differs, however, from all other primitives used in the state machines in one respect: it is not atomic. In this model, the Frame primitive occupies time; this allows the simplex Physical State Machine to consume time associated with transmission, and to trigger the 10 minute transmitter protection and anti-hogging timers. This primitive also passes data from an actual reception of a frame.

• **Acquisition of Signal**. This primitive indicates the presence of modem synchronization, flag fill or frame structure.

• **Loss of Signal**. This primitive indicates the loss of modem synchronization, flag fill or frame structure.

C2a.3. Internal Operation of the Machine

The internal states, queues, flags and timers are summarized in Figure C2a.1. All queues are first-in, first-out. These items are used in a straightforward manner; no further explanation is deemed necessary here.

Note that the anti-hogging time limit is not applied to the digipeating function. However, the 10-minute transmitter timer is enforced while digipeating. In the unlikely event that the 10-minute limit is exceeded, the transmission of digipeated frames is temporarily suspended and the channel is relinquished. After other stations have had the opportunity to digipeat frames (i.e., PRIACK expires), but before the p-persistence algorithm takes effect, the state machine jumps back on the channel to resume transmission of those frames still in the priority queue. While this logic is provided in the state diagrams for completeness, it seems unlikely that it would ever be used.

PH Primitives (Received from the Link Multiplexer):

PH-SEIZE Request
PH-RELEASE Request
PH-EXPEDITED-DATA Request
PH-DATA Request

PH Primitives (Sent to the Link Multiplexer):

PH-SEIZE Confirm
PH-BUSY Indication
PH-QUIET Indication
PH-DATA Indication

PH Primitives (Received from the Radio):

Acquisition of Signal
Loss of Signal
Frame

PH Primitives (Sent to the Radio):

Turn on Transmitter
Turn Off Transmitter
Frame

States:

0 — Ready
1 — Receiving
2 — Transmitter Suppression
3 — Transmitter Start
4 — Transmitting
5 — Digipeating
6 — Receiver Start

Error Codes:

No Error Codes Used.

Queues:

Priority Queue — holds all expedited data frames to be transmitted in the order in which they arrived from the higher layer.
Normal Queue — holds all normal frames, plus Seize and Release Requests, in the order in which they arrived from the higher level.

Flags and Parameters:

Digipeating — set when this transmission is for digipeating frames.
Cleared when this transmission is for normal frames.
Repeater Up — set when repeater is expected to still be transmitting.
Cleared when repeater carrier is expected to have dropped.
Interrupted — set when anti-hogging or 10-minute transmitter limits have interrupted the transmission of normal frames.
p — p-persistence value, in the range 0-1.

Timers:

T100 — Repeater Hang (AXHANG)
T101 — Priority Window (PRIACK)
T102 — Slot Time (p-persistence)
T103 — Transmitter Startup (TXDELAY)
T104 — Repeater Startup (AXDELAY)
T105 — Remote Receiver Sync
T106 — Ten Minute Transmission Limit
T107 — Anti-Hogging Limit
T108 — Receiver Startup

Figure C2a.1. Summary of primitives, states, queues, flags, errors and timers.

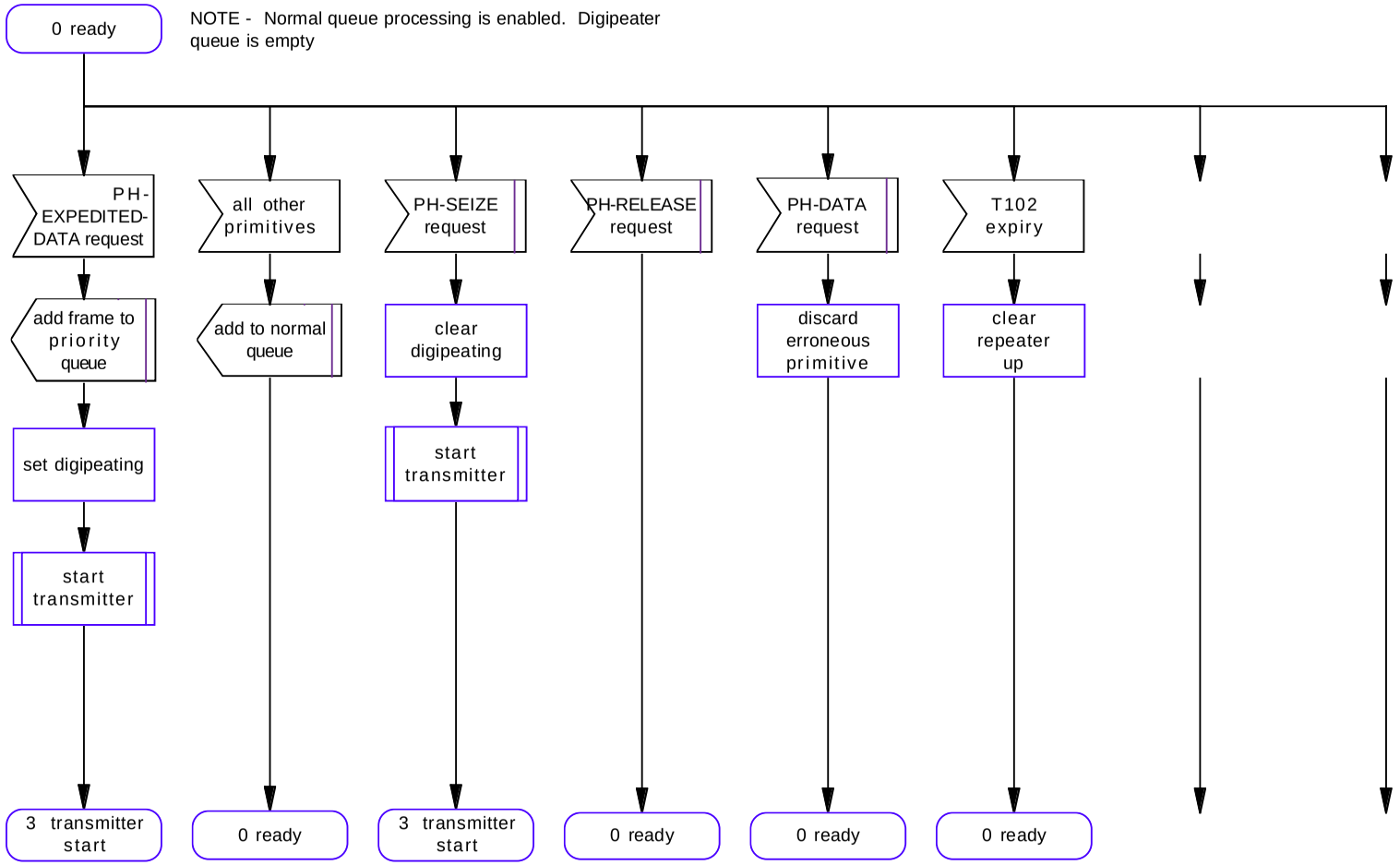


Figure C2a.2. Simplex physical ready state.

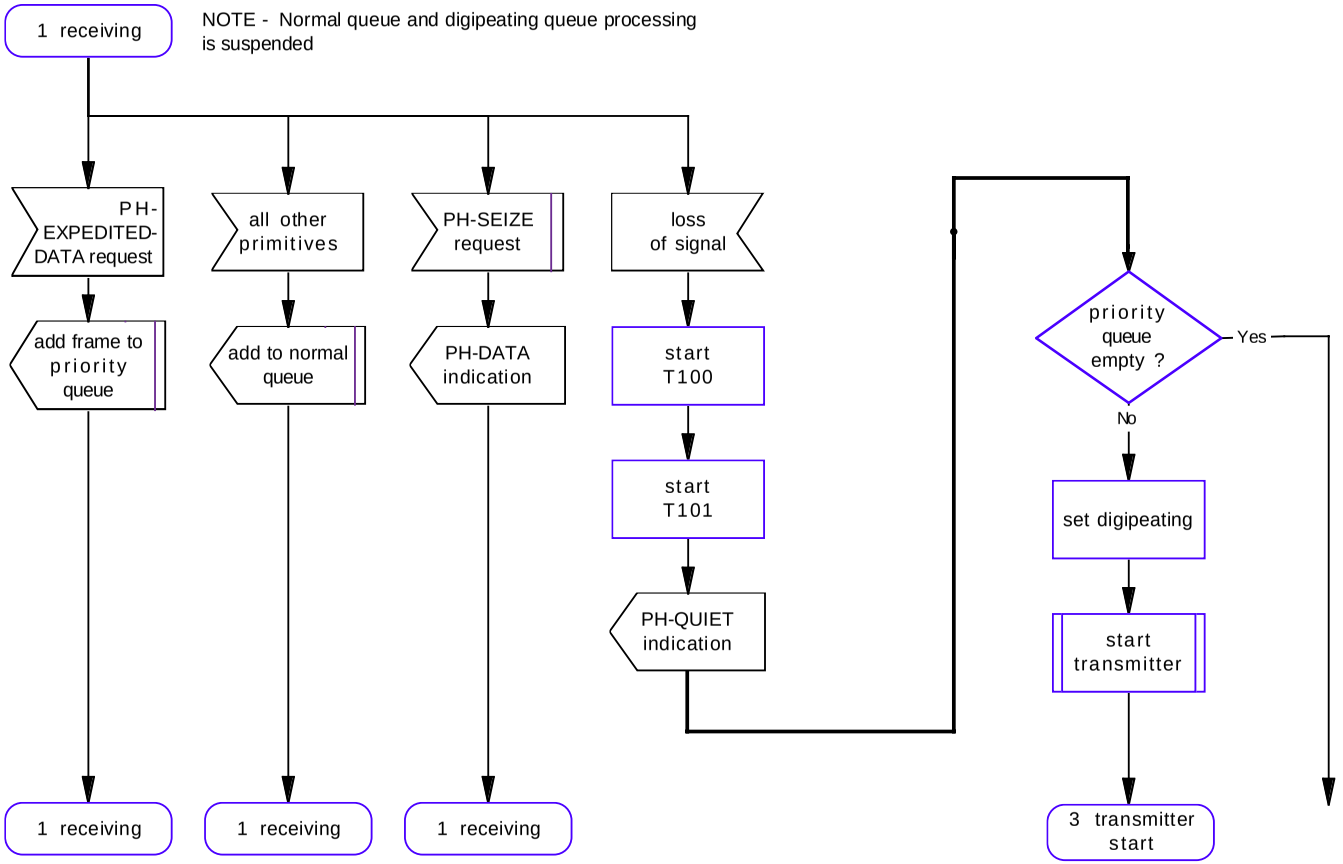


Figure C2a.3. Simplex physical receiving state.

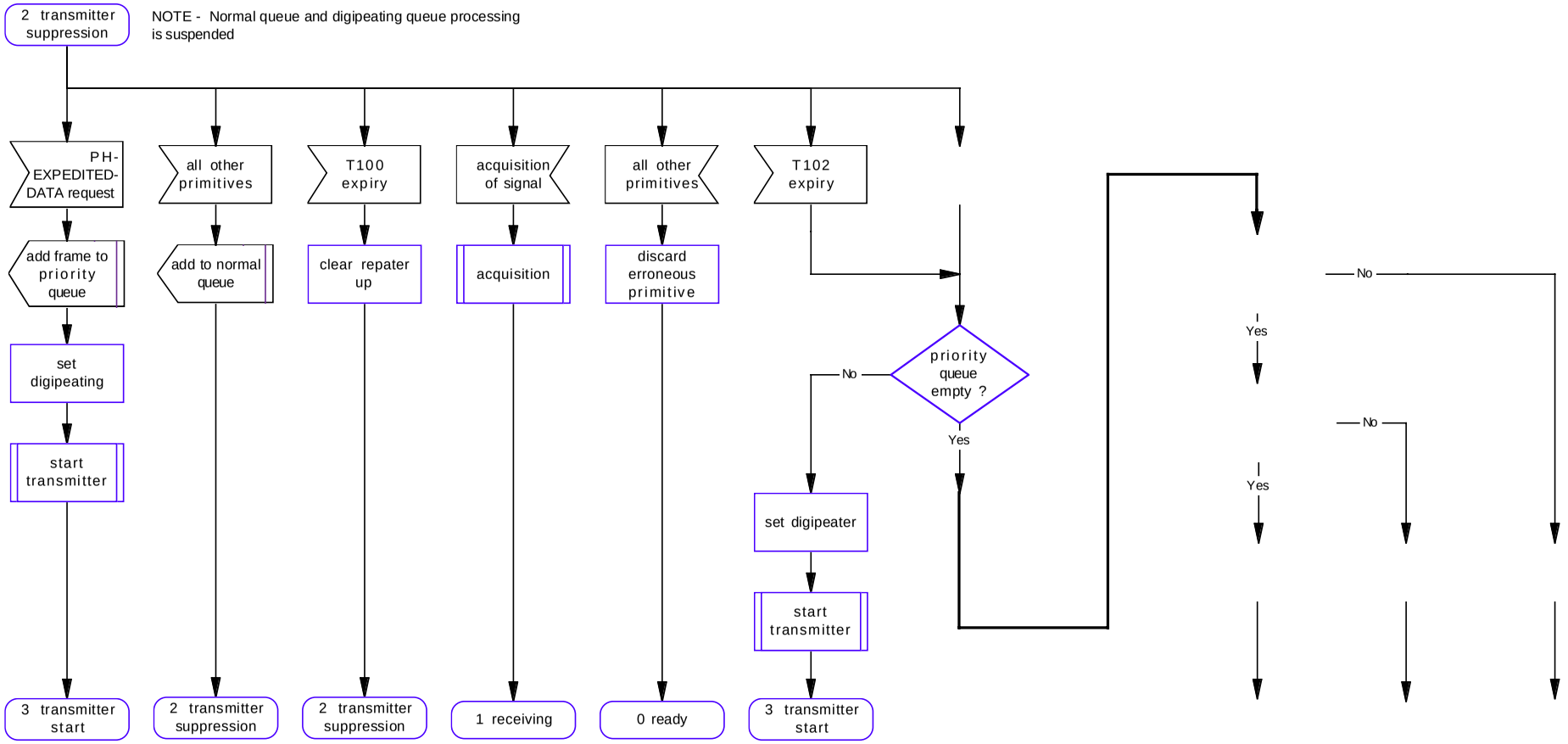


Figure C2a.4. Simplex physical transmitter suppression state.

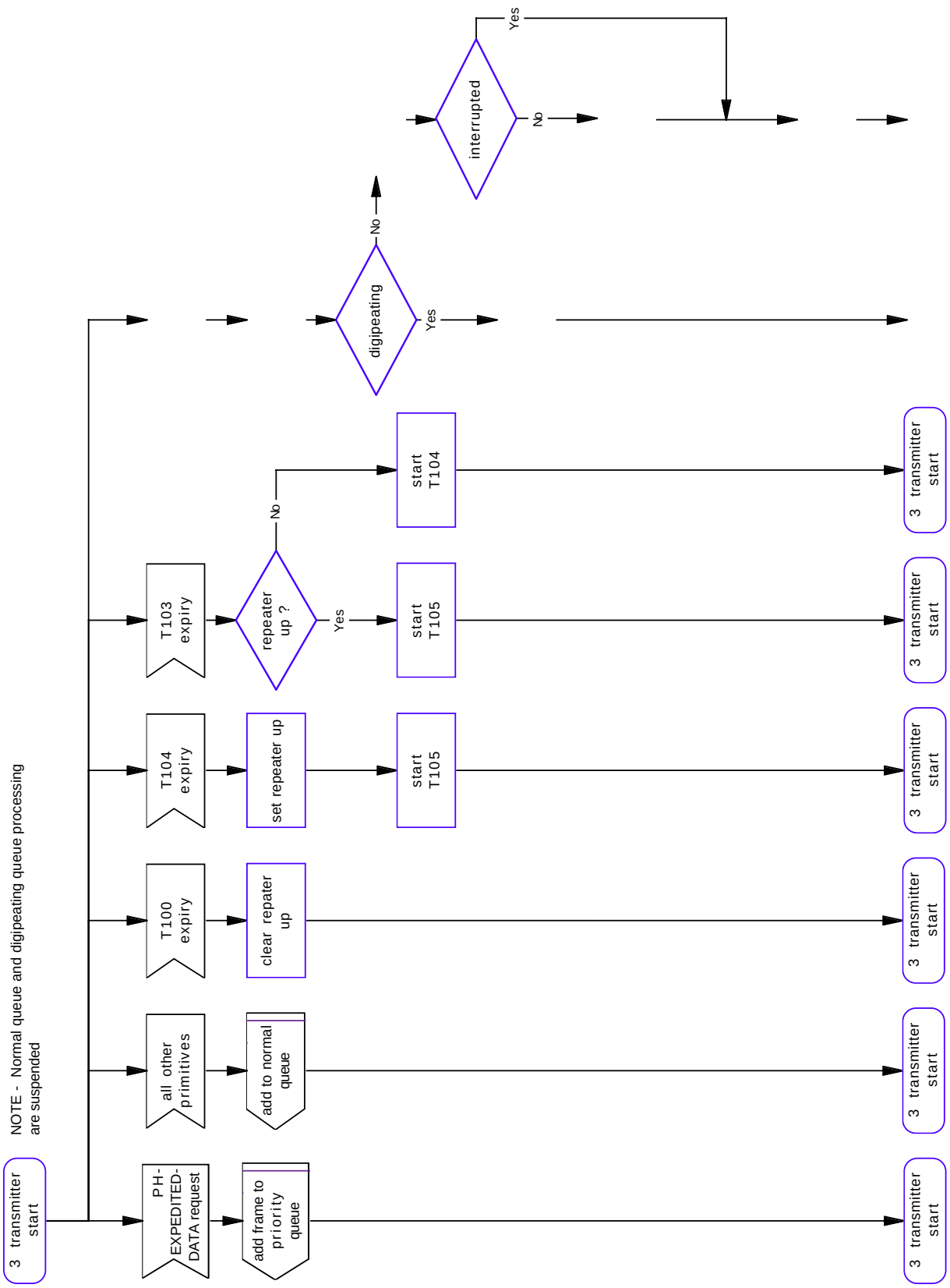


Figure C2a.5. Simplex physical transmitter start state.

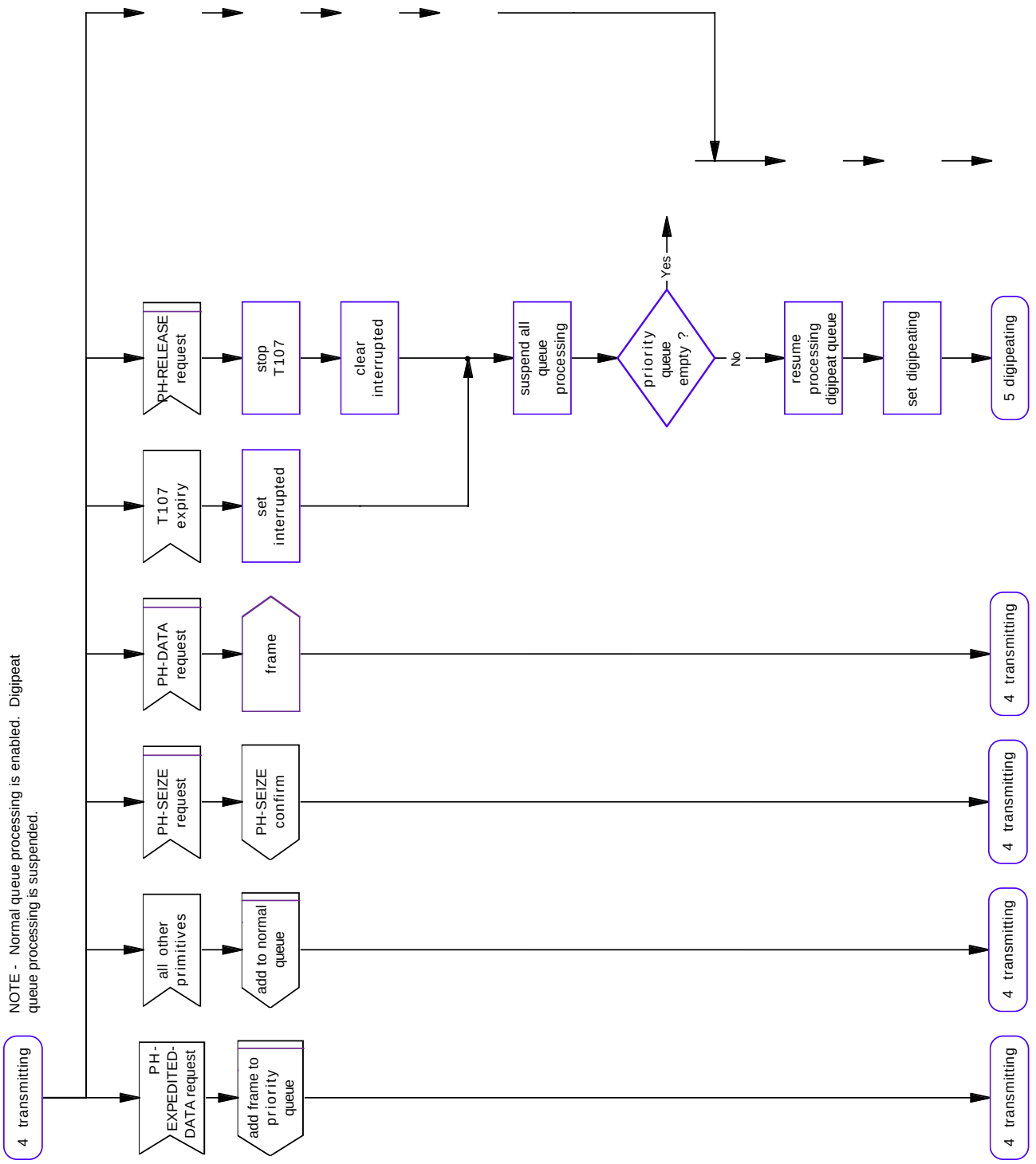


Figure C2a.6. Simplex physical transmitting state.

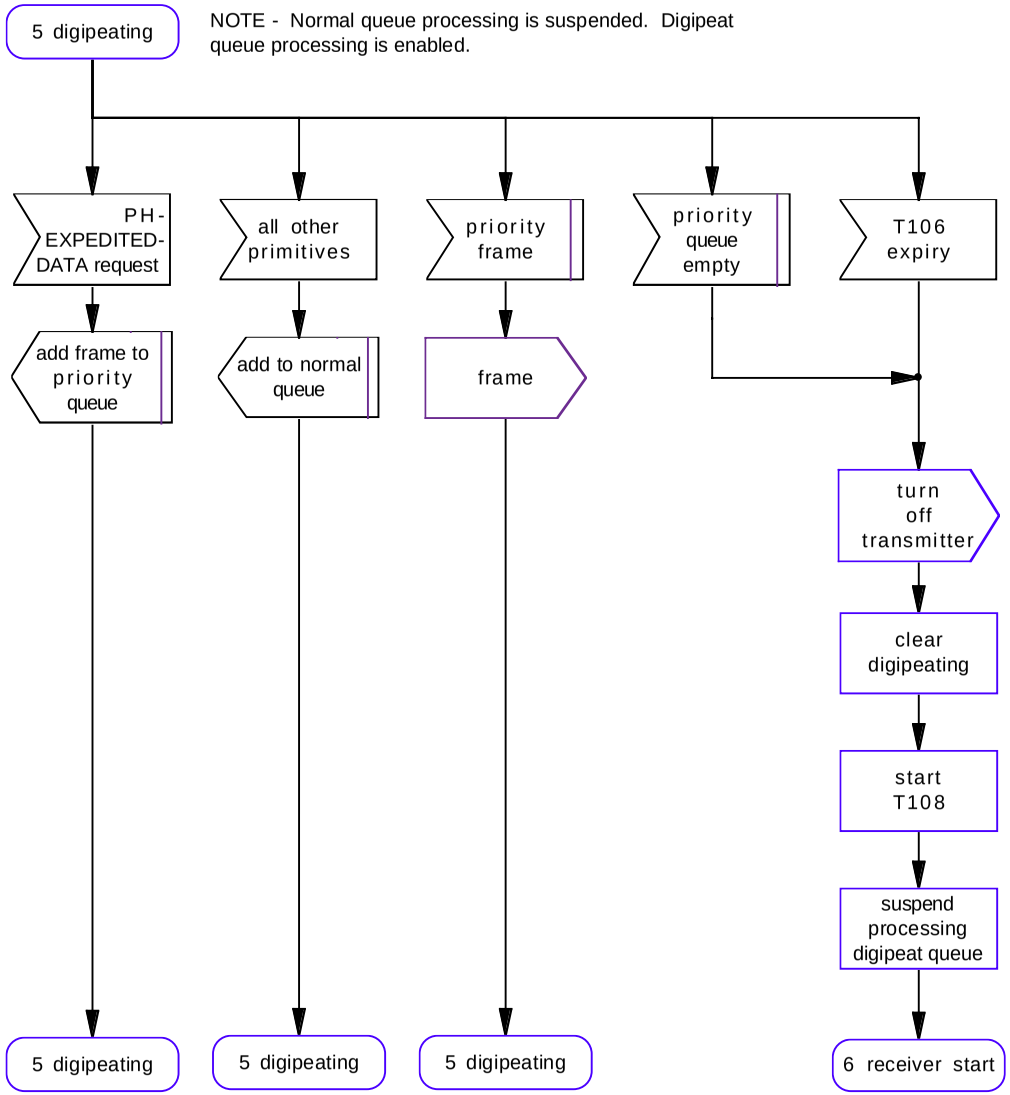


Figure C2a.7. Simplex physical digipeating state.

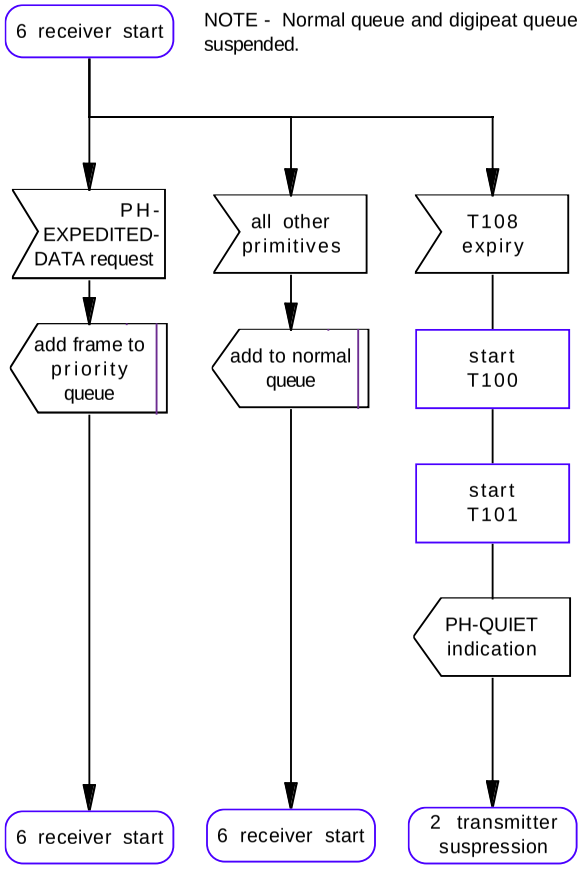


Figure C2a.8. Simplex physical receiver start state.

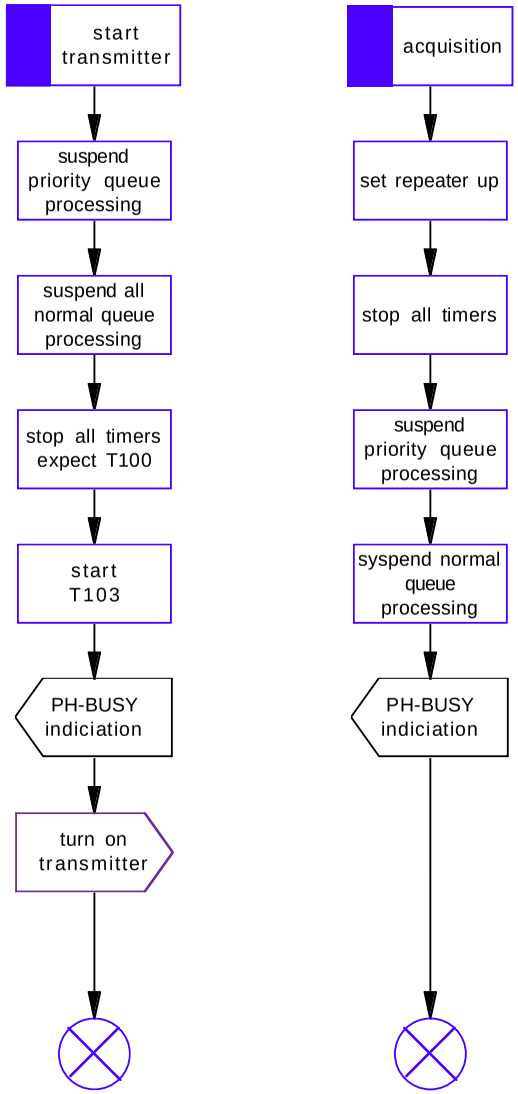


Figure C2a.9. Simplex physical subroutines C-2-A-12.

Appendix C2b

Duplex Physical Layer State Machines

C2b.1. Interaction with the Link Multiplexer

The Link Multiplexer State Machine directs the operation of the duplex Physical State Machine through the physical (PH) primitives described below.

- **PH-SEIZE Request**. This primitive requests the duplex state machine to begin transmitting. When that opportunity has been identified, the transmitter is started and a parameterized time is allowed for the synchronization of the remote station's receiver (known as TXDELAY in most implementations), then a PH-SEIZE Confirm primitive is returned to the link multiplexer.

- **PH-DATA Request**. This primitive from the Link Multiplexer State Machine provides a AX.25 frame of any type (UI, SABM, I, etc.) that is to be transmitted. An unlimited number of frames may be provided. If the transmission exceeds the 10-minute limit or the anti-hogging time limit, the duplex Physical State Machine shuts down the transmitter.

- **PH-RELEASE Request**. The Link Multiplexer State Machine provides this primitive when the submission of a sequence of frames to be transmitted on behalf of a particular AX.25 connection has been completed.

- **PH-EXPEDITED-DATA Request**. This primitive from the Link Multiplexer State Machine provides the AX.25 frame that is to be transmitted immediately. The duplex Physical State Machine gives preference to priority frames over normal frames. Priority frames can be provided by the link multiplexer at any time.

- **PH-DATA Indication**. During reception, the duplex Physical State Machine provides each AX.25 frame to the link multiplexer in a Frame primitive. No analysis is done on the frame by the duplex Physical State Machine; it does not examine lengths, the frame check sequence, or any other content of the frame; these responsibilities are carried out by the higher level state machines.

- **PH-BUSY Indication**. The duplex Physical State Machine provides this primitive whenever the channel becomes busy. "Busy" here means the detection of a valid modem synchronization sequence, HDLC Flags or HDLC frames — not the detection of an FM carrier on a two-meter radio. An indication of busy is provided to the higher layer state machines so that various timers which are supervise the AX.25 connection can be suspended. This avoids the undesirable situation on a busy channel where AX.25, having sent data and expecting an acknowledgment, times out and attempts retransmissions — and the only reason an acknowledgment was not received was because the remote station did not yet have a chance to make a transmission.

- **PH-QUIET Indication**. The duplex Physical State Machine provides this primitive whenever the channel becomes quiet.

C2b.2. Interface to the Hardware

As the lowest layer state machine in the standard, this machine manipulates a typical radio transceiver. The following primitives are used:

• **Turn On Transmitter**. This primitive keys the transceiver's PTT line.

• **Turn Off Transmitter**. This primitive unkeys the transceiver's PTT line.

• **Frame**. This primitive passes data for actual transmission of a frame. Although SDL representation of bit-by-bit transmission of the contents of a frame are possible, they are not used here because the additional complexity was not required. The Frame primitive, however, differs from all other primitives used in the state machines in one respect: it is not atomic. Under this model, the Frame primitive occupies time; this allows the duplex Physical State Machine to consume time associated with transmission, and to trigger the ten minute transmitter protection and anti-hogging timers.

  This primitive is also used to pass data from an actual reception of a frame.

• **Acquisition of Signal**. This primitive indicates the presence of modem synchronization, flag fill or frame structure.

• **Loss of Signal**. This primitive indicates the loss of modem synchronization, flag fill or frame structure.

C2b.3. Internal Operation of the Machine

The internal states, queues, flags and timers are summarized in Figure C2b.1. All queues are first-in, first-out. These items are used in a straightforward manner; no further explanation is deemed necessary here.

PH Primitives (Received from the Link Multiplexer):

PH-SEIZE Request
PH-RELEASE Request
PH-EXPEDITED-DATA Request
PH-DATA Request

PH Primitives (Sent to the Link Multiplexer):

PH-SEIZE Confirm
PH-BUSY Indication
PH-QUIET Indication
PH-DATA Indication

PH Primitives (Received from the Radio):

Acquisition of Signal
Loss of Signal
Frame

PH Primitives (Sent to the Radio):

Turn on Transmitter
Turn Off Transmitter
Frame

States:

0 — Receiver Ready
1 — Receiving
2 — Transmitter Ready
3 — Transmitter Start
4 — Transmitting

Error Codes:

No Error Codes Used.

Queues:

Normal Queue — holds all normal frames, plus Seize and Release Requests, in the order in that they arrived from the higher level.

Flags and Parameters:

Interrupted — set when anti-hogging or 10-minute transmitter limits have interrupted the transmission of normal frames.

Timers:

T105 — remote receiver sync
T106 — 10-minute transmission limit
T107 — anti-hogging limit

Figure C2b.1. Summary of primitives, states, queues, flags, errors and timers.

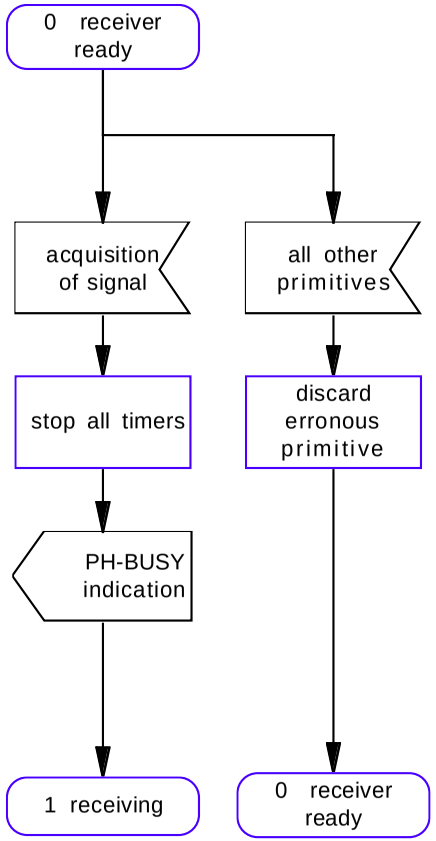


Figure C2b.2. Duplex physical receiver ready state.

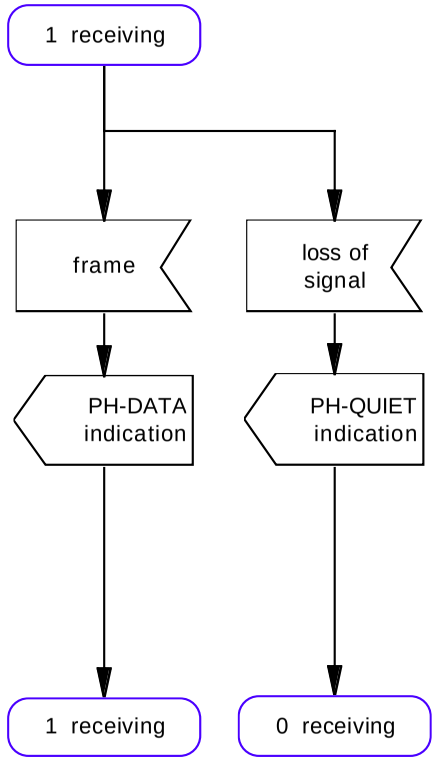


Figure C2b.3. Duplex physical receiving state.

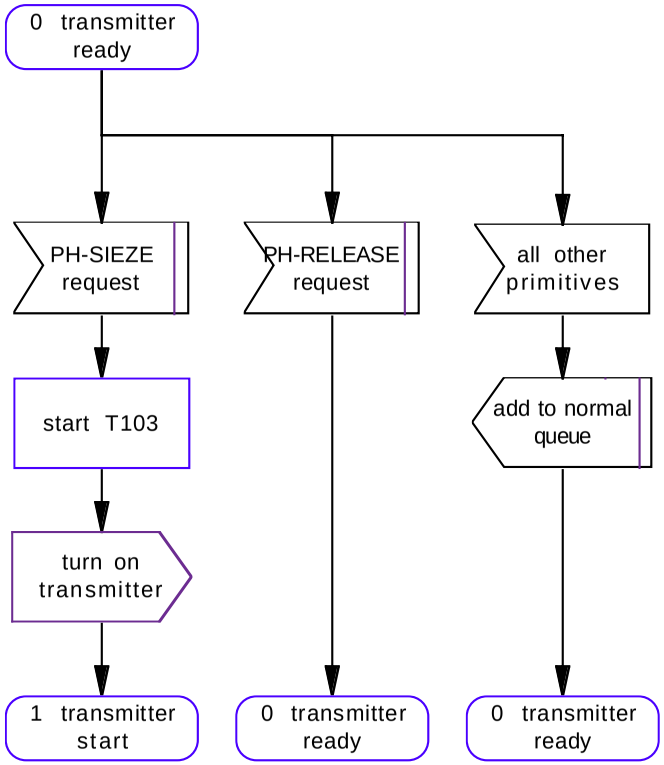


Figure C2b.4. Duplex physical transmitter ready state.

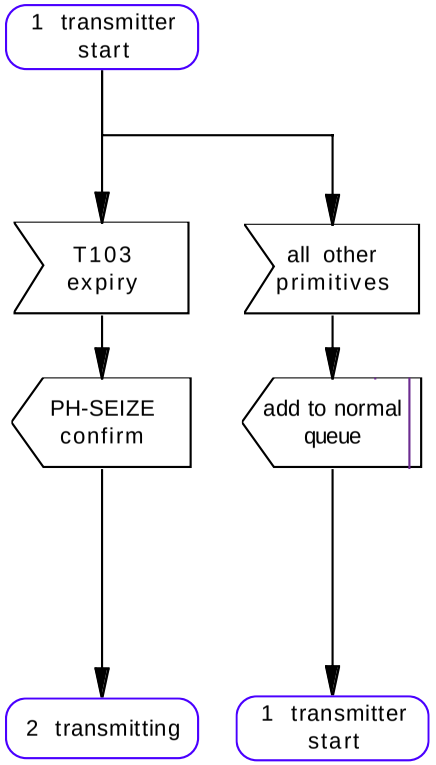


Figure C2b.5. Duplex physical transmitter start state.

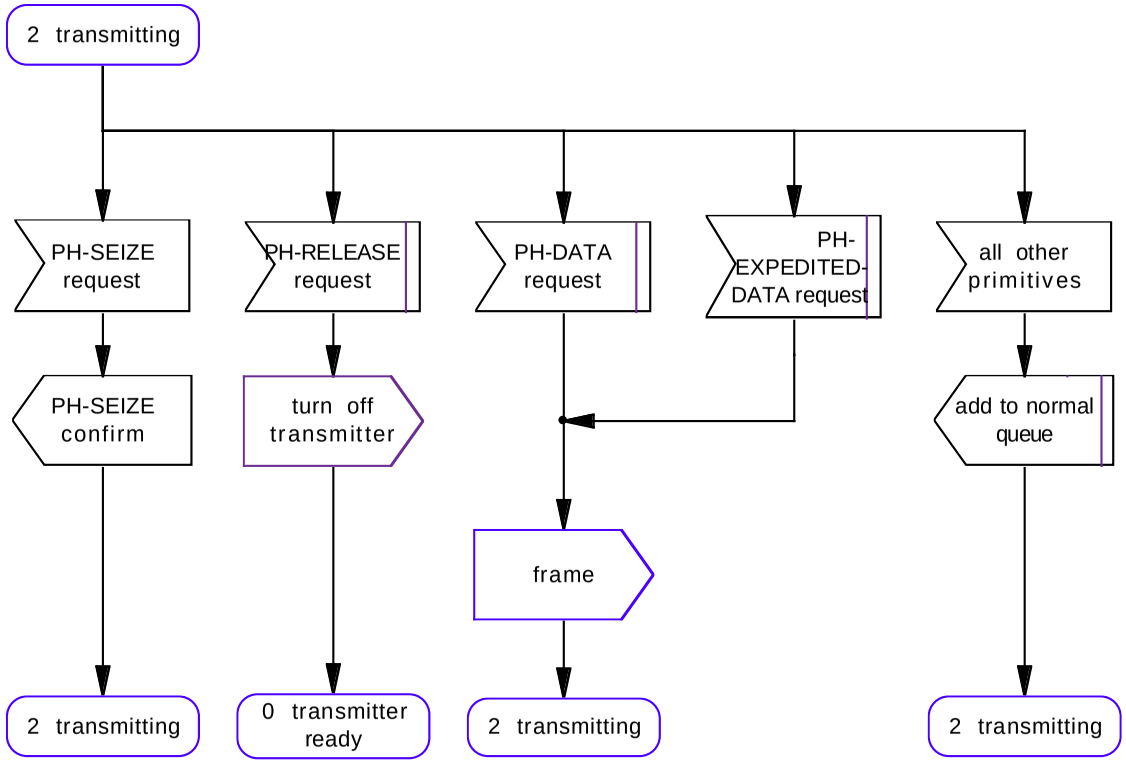


Figure C2b.6. Duplex physical transmitting state.

Appendix C3

Link Multiplexer State Machine

C3.1. Interaction with the Data-Link State Machine

The Data-link State Machine directs the operation of the Link Multiplexer State Machine through the link multiplexer (LM) primitives described below.

- **LM-SEIZE Request**. This primitive requests the Link Multiplexer State Machine to arrange for transmission at the next available opportunity. The Data-link State Machine uses this primitive when an acknowledgment must be made, but the exact frame in which the acknowledgment will be sent will be chosen when the actual time for transmission arrives. The Link Multiplexer State Machine uses the LM-SEIZE Confirm primitive to indicate that the transmission opportunity has arrived. After the Data-link State Machine has provided the acknowledgment, the Data-link State Machine gives permission to stop transmission with the LM Release Request primitive.
- **LM-DATA Request**. This primitive from the Data-link State Machine provides a AX.25 frame of any type (UI, SABM, I, etc.) that is to be transmitted. An unlimited number of frames may be provided. The Link Multiplexer State Machine accumulates the frames in a first-in, first-out queue until it is time to transmit them.
- **LM-DATA Indication**. This primitive from the Link Multiplexer State Machine provides a AX.25 frame of any type (UI, SABM, I, etc.) that has been received. An unlimited number of frames may be provided.

C3.2. Interaction with the Physical Layer State Machine

The Link Multiplexer State Machine works with the physical layer state machine. It is important to note that variations in the operating characteristics of radio channels are kept hidden from the Link Multiplexer State Machine. The Link Multiplexer State Machine uses the same primitives to communicate with the physical layer state machine, regardless of the latter's type.

- **PH-SEIZE Reques**t. This primitive is used by the Link Multiplexer State Machine before each transmission to request access to the radio channel.
- **PH-SEIZE Confirm**. This primitive notifies the Link Multiplexer State Machine when access has been obtained (i.e., the transmitter is operating, any intervening repeater has had an opportunity to be activated, the remote station's receiver has had an opportunity to become synchronized, and the channel is considered ready to send traffic).
- **PH-DATA Request**. This primitive is used by the Link Multiplexer State Machine to deliver each frame to the physical layer state machine.
- **PH-RELEASE Request**. This primitive is used when all frames that have been awaiting transmission for a given link have been submitted for transmission. The intention here is that a single transmission will contain frames for only one remote station. The PH-RELEASE Request primitive permits the physical layer state machine to release the channel for use by others, for digipeating, and for receipt of acknowledgments in a contention environment (such as shared simplex channels).

- **PH-DATA Indication**. This primitive is used by the physical layer state machine to provide incoming frames to the Link Multiplexer State Machine. The Link Multiplexer State Machine checks each incoming frame for FCS errors. Correctly-received frames are checked to see if digipeating by the station has been requested and if the digipeat function is enabled (a user specified parameter); if so, the frame is resubmitted to the physical layer state machine in a Digipeat Frame primitive. Correctly-received frames addressed to this station are delivered to the indicated higher-layer Data-link State Machine (see Section 5.3.1).

- **PH-EXPEDITE-DATA Request**. This primitive is used by the Link Multiplexer State Machine to submit a frame to the physical layer state machine to be transmitted immediately. (PH-SEIZE Request and PH-RELEASE Request are not used for digipeat operation.)

- **PH-BUSY Indication**. This primitive is used by the Link Multiplexer State Machine to suspend all AX.25 data-link timers.

- **PH-QUIET Indication**. This primitive is used by the Link Multiplexer State Machine to resume ticking on all AX.25 data- link timers.

The suspension of timers overcomes a problem noted in some implementations on busy channels. This problem occurs when frames are transmitted and a response is expected. If the channel is busy, it is possible for the retry timers (AX.25 timer T1) to expire before the remote station has had an opportunity to send any acknowledgment. This premature expiration causes needless retries and polling, which further clutters an already busy frequency.

C3.3. Internal Operation of the Machine

The internal states, queues and flags are summarized in Figure C3.1.

All queues are first-in, first-out. Three queues are used in conjunction with two flags to implement round-robin rotation among the various Data-link State Machines.

The Awaiting Queue contains all primitives received from Data-link State Machines that have not yet had an opportunity to transmit.

When a primitive pops off the Awaiting Queue, it and all other primitives from that same Data-link State Machine are placed in order on the Current Queue. The identity of this Data-link State Machine is maintained in the Current DL Flag. The Link Multiplexer State Machine then proceeds to obtain a transmission opportunity for that Data-link State Machine. Any further primitives received from that particular Data-link State Machine are added to the Current Queue. When the transmission opportunity arrives, everything in the Current Queue is conveyed to the physical layer state machine for transmission. (In the event of an overly large amount of information to be sent, the physical layer state machine makes whatever breaks in transmission are appropriate for reasonable channel sharing. This is done within the physical layer state machine and hidden from the higher layer.)

Once everything has been sent for the current Data-link State Machine, its identity is moved to the Served List. Any subsequent primitives from this Data-link State Machine are added to the Served Queue.

The Link Multiplexer State Machine then goes back to the Awaiting Queue to pop off the next primitive, and thereby identify which Data-link State Machine has the next transmission opportunity. If the Awaiting Queue is empty, then the Link Multiplexer State Machine concludes that all Data-link State Machines that had frames to be sent have now been served. The queue system is reset by converting the Served Queue into the new Awaiting Queue, and by purging all identifiers from the Served List.

LM Primitives (Received from LM):
LM-SEIZE Request
LM-RELEASE Request
LM-DATA Request

LM Primitives (Sent to LM):
LM-SEIZE Confirm
LM-DATA Indicate

PH Primitives (Received from PH):
PH-SEIZE Confirm
PH-QUIET Indication
PH-BUSY Indication
PH-DATA Indication

PH Primitives (Sent to PH):
PH-SEIZE Request
PH-RELEASE Request
PH-EXPEDITED-DATA Request
PH-DATA Request

States:
0 — Idle
1 — Seize Pending
2 — Seized

Error Codes:
No error codes used.

Queues:
Awaiting Queue — queue of primitives received from Data-link State Machines that are not presently using the transmitter.

Current Queue — queue of primitives received from the Data-link State Machine that is presently using the transmitter.

Served Queue — queue of primitives received from Data-link State Machines that already have used the transmitter.

Note: After all Data-link State Machines have had an opportunity to be served, then the Served Queue is converted to the Awaiting Queue.

Flags:

Current DL — Identifies the Data-link State Machine currently using the transmitter.

Served List — Identifies the Data-link State Machines that have already used the transmitter. This list is cleared when all Data-link State Machines with frames to send have been served.

Timers:

No timers used.

Figure C3.1. Summary of primitives, states, flags, errors and timers.

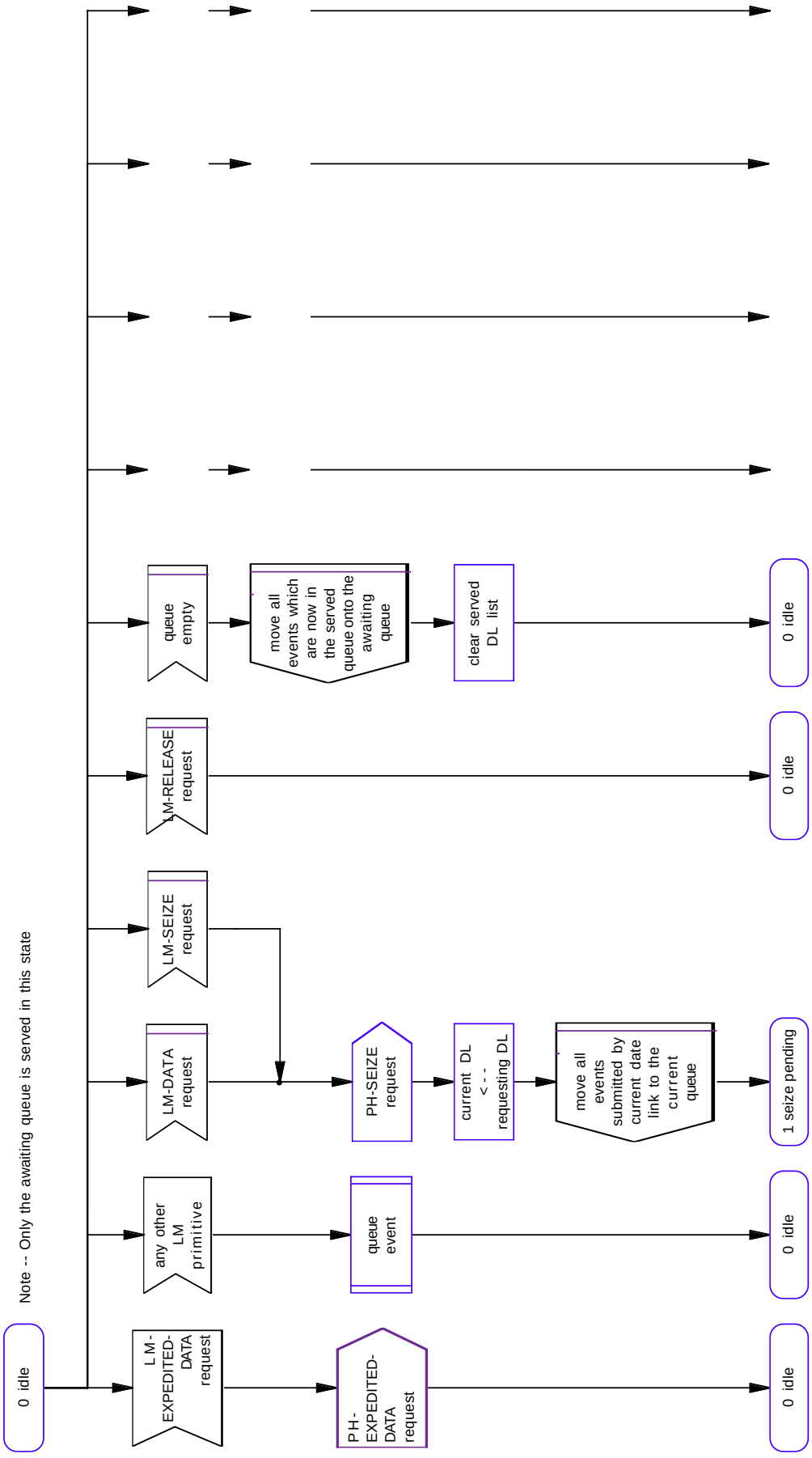


Figure C3.1. Link Multiplexer idle state.

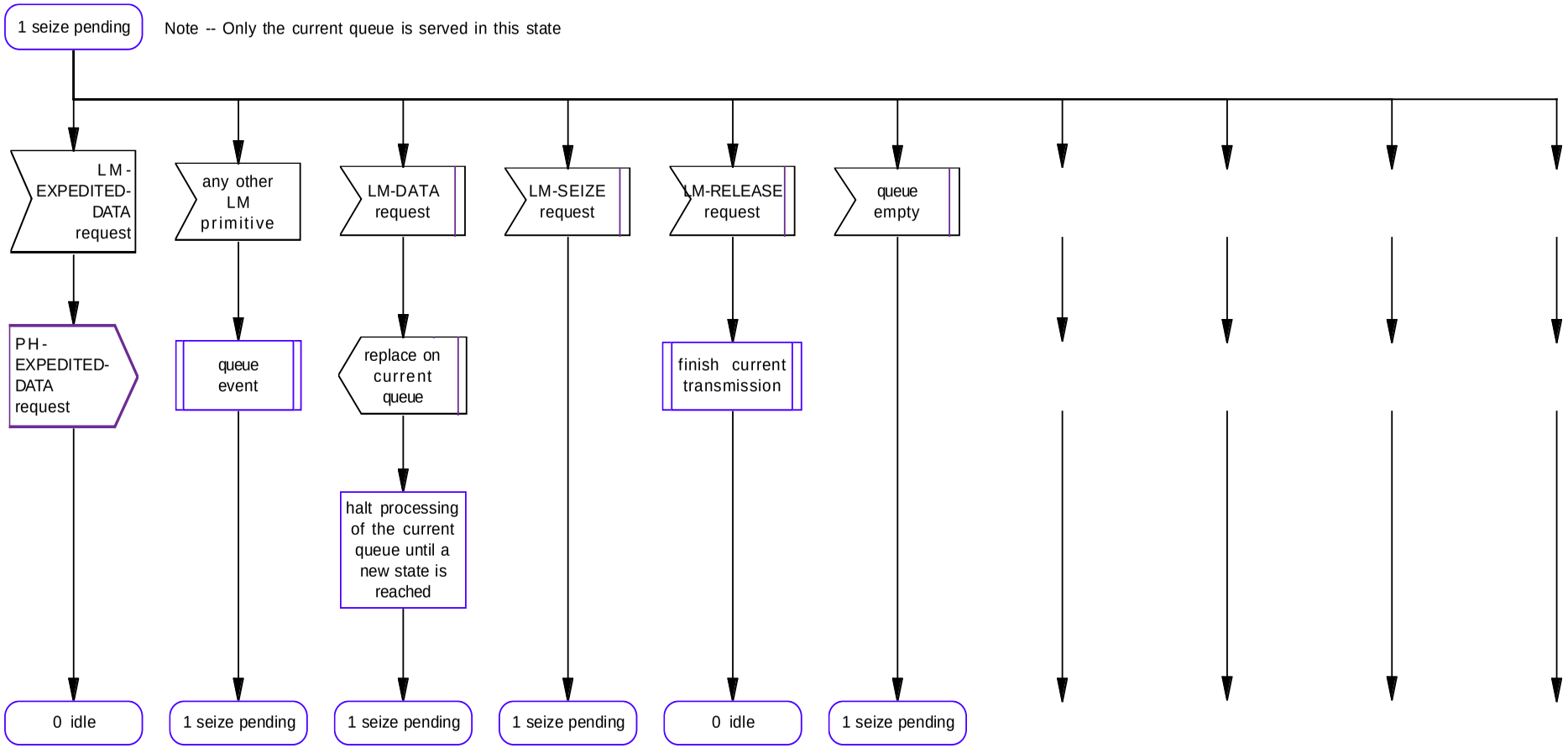


Figure C3.2. Link Multiplexer seize pending state.

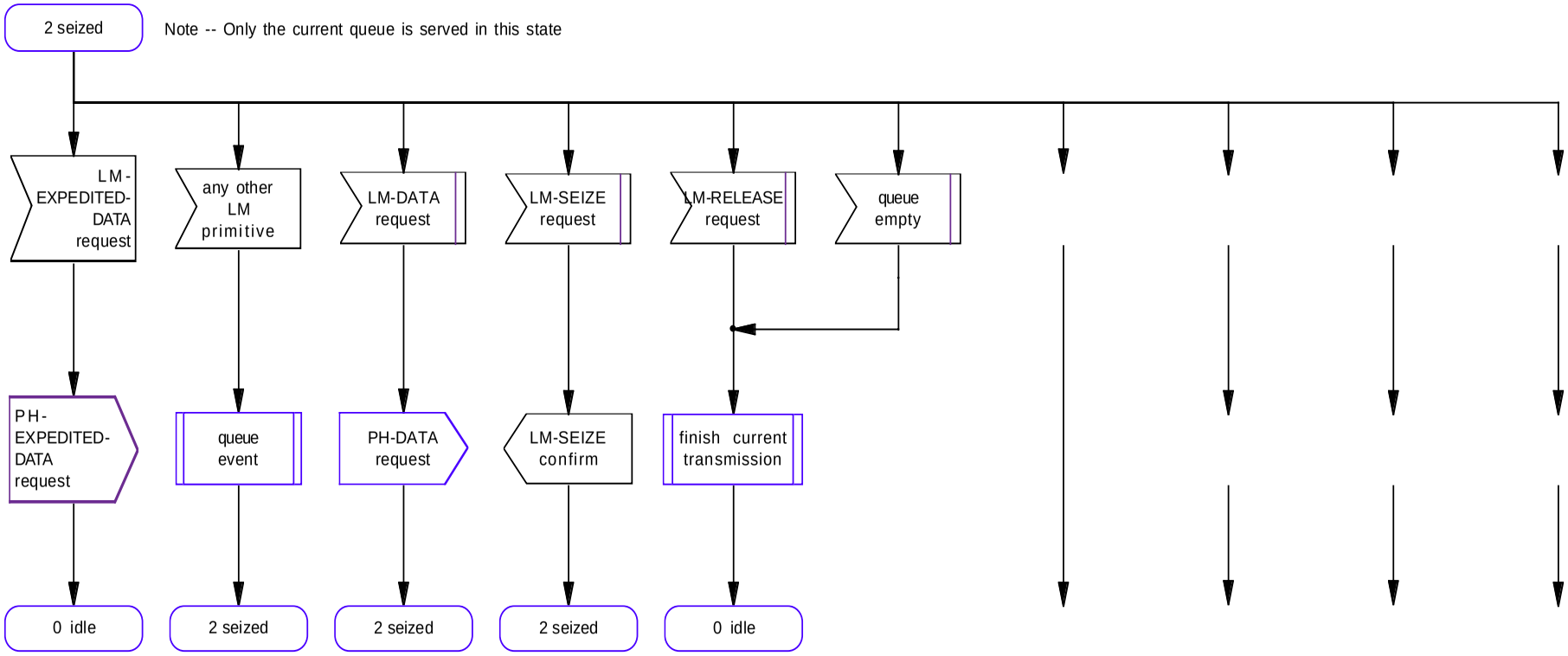
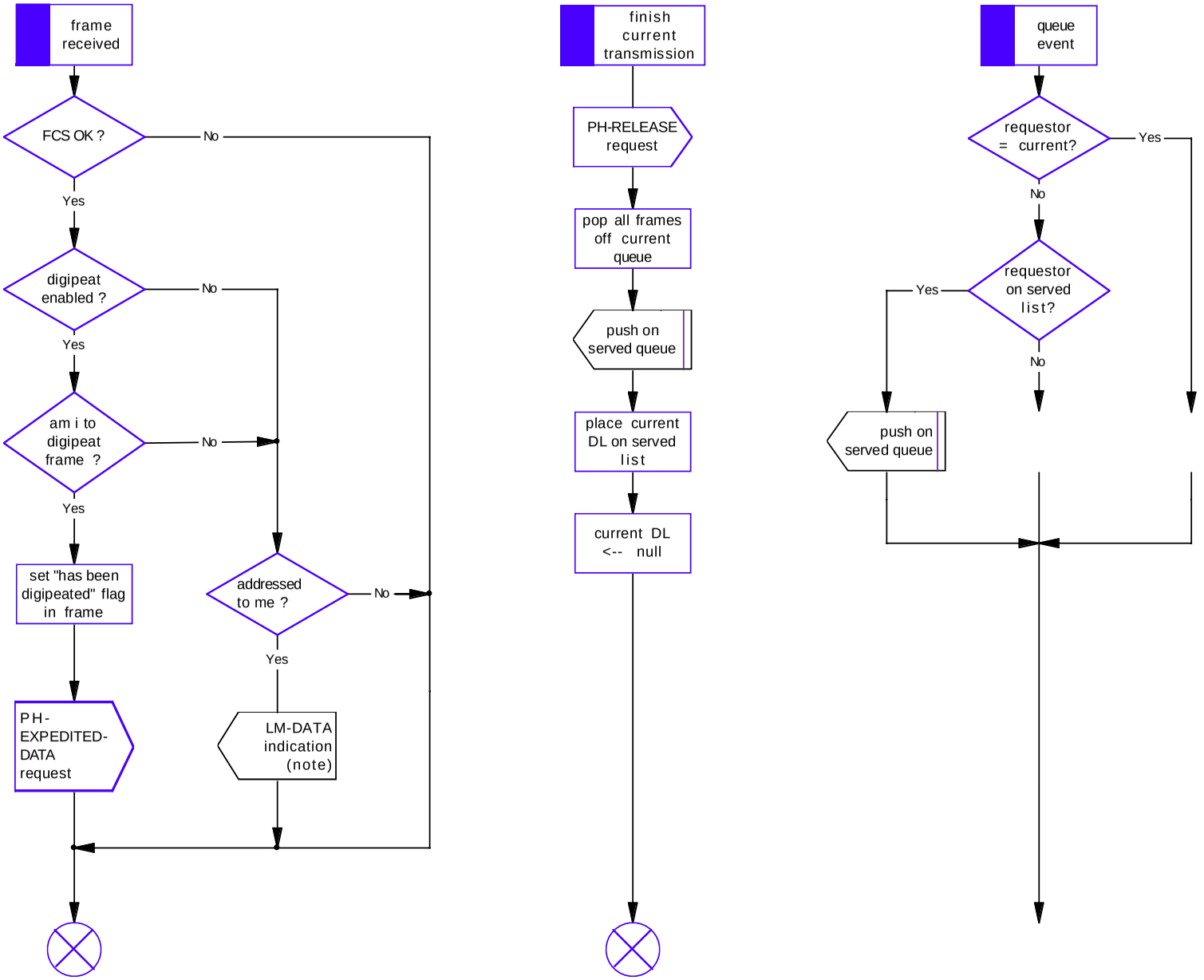


Figure C3.3. Link Multiplexer seized state.



Note - The LM-DATA indication primitive is sent to the data link machine which is responsible for communications with the indicated remote (source) station.

Figure C3.4. Link Multiplexer subroutines.

Appendix C4

Data-Link State Machine

C4.1. Interaction with the Data-Link Service Access Point

The data-link service access point directs the operation of the Data-link State Machine through the data link (DL) primitives described below. The segmenter state machine is between the Data-link State Machine and the data-link service access point. It passes all DL primitives except DL-DATA and DL-UNIT-DATA through transparently.

- **DL-CONNECT Request**. This primitive is used by the Layer 3 entity to request the establishment of a AX.25 connection.
- **DL-CONNECT Indication**. This primitive is used by the Data-link State Machine to indicate an AX.25 connection has been requested.
- **DL-CONNECT Confirm**. This primitive is used by the Data-link State Machine to indicate a AX.25 connection has been made.
- **DL-DISCONNECT Request**. This primitive is used by the Layer 3 entity to request the release of a AX.25 connection.
- **DL-DISCONNECT Indication**. This primitive is used by the Data-link State Machine to indicate an AX.25 connection has been released.
- **DL-DISCONNECT Confirm**. This primitive is used by the Data-link State Machine to indicate an AX.25 connection has been released and confirmed.
- **DL-DATA Request**. This primitive is used by the Layer 3 entity to request the transmission of data using connection oriented protocol. This frame is examined and acted upon by the segmenter, if necessary.
- **DL-DATA Indication**. This primitive is used by the reassembler to indicate reception of Layer 3 data using connection oriented protocol.
- **DL-UNIT-DATA Request**. This primitive is used by the Layer 3 entity to request the transmission of data using connectionless protocol. This frame is examined and acted upon by the segmenter, if necessary.
- **DL-UNIT-DATA Indication**. This primitive is used by the reassembler to indicate reception of Layer 3 data using connectionless protocol.
- **DL-ERROR Indication**. This primitive is used by the Data-link State Machine to indicate when frames have been received that are inconsistent with this protocol definition. This includes short frames, frames with inconsistent parameter values, etc.
- **DL-FLOW-OFF request**. This primitive is used by the Layer 3 entity to temporarily suspend the flow of incoming information.
- **DL-FLOW-ON Request**. This primitive is used by the Layer 3 entity to resume the flow of incoming information.

C4.2. Interaction with the Link Multiplexer State Machine

The Data-link State Machine directs the operation of the Link Multiplexer State Machine through the data link (LM) primitives described below.

- **LM-SEIZE Request**. This primitive is used by the Data-link State Machine to request the Link Multiplexer State Machine to arrange for transmission at the next available opportunity. The Data-link State Machine uses this primitive when an acknowledgment must be made, but the exact frame in which the acknowledgment is sent will be chosen when the actual time for transmission arrives.

- **LM-SEIZE Confirm**. This primitive indicates to the Data-link State Machine that the transmission opportunity has arrived.

- **LM-RELEASE Request**. This primitive is used by the Link Multiplexer State Machine to stop transmission.

- **LM-EXPEDITED-DATA Request**. This primitive is used by the Data-link State Machine to pass expedited data to the link multiplexer.

- **LM-DATA Request**. This primitive is used by the Data-link State Machines to pass frames of any type (SABM, RR, UI, etc.) to the Link Multiplexer State Machine.

- . This primitive is used by the Link Multiplexer State Machines to pass frames of any type (SABM, RR, UI, etc.) to the Data-link State Machine.

C4.3. Internal Operation of the Machine

The internal states, queues and flags are summarized in Figure C4.1.

All queues are first-in, first-out.

DL Primitives (Received from DL):

DL-CONNECT Confirm
DL-CONNECT Indicate
DL-DISCONNECT Confirm
DL-DISCONNECT Indicate
DL-DATA Indicate
DL-UNIT-DATA Indicate
DL-ERROR Indicate

DL Primitives (Sent to DL):

DL-CONNECT Request
DL-DISCONNECT Request
DL-DATA Request
DL-UNIT-DATA Request
DL-FLOW-OFF Request
DL-FLOW-ON Request

LM Primitives (Sent to LM):

LM-SEIZE Request
LM-RELEASE Request
LM-DATA Request
LM-EXPEDITED-DATA Request

LM Primitives (Received from LM):

LM-SEIZE Confirm
LM-DATA Indicate

States:

0 — Disconnected
1 — Awaiting Connection
2 — Awaiting Release
3 — Connected
4 — Timer Recovery

Error Codes:

A — F=1 received but P=1 not outstanding.
B — Unexpected DM with F=1 in states 3, 4 or 5.
C — Unexpected UA in states 3, 4 or 5.
D — UA received without F=1 when SABM or DISC was sent P=1.
E — DM received in states 3, 4 or 5.
F — Data link reset; i.e., SABM received in state 3, 4 or 5.
I — N2 timeouts: unacknowledged data.
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.
T — N2 timeouts: no response to enquiry.
U — N2 timeouts: extended peer busy condition.
V — No DL machines available to establish connection.

Queues:

I Frame Queue — queue of information to be transmitted in I frames.

Flags:

Layer 3 Initiated — SABM was sent by request of Layer 3; i.e., DL-CONNECT Request primitive.
Peer Receiver Busy — Remote station is busy and cannot receive I frames.
Own Receiver Busy — Layer 3 is busy and cannot receive I frames.
Reject Exception — A REJ frame has been sent to the remote station.
Selective Reject Exception — A SREJ frame has been sent to the remote station.
Acknowledge Pending — I frames have been successfully received but not yet acknowledged to the remote station.
SRT — Smoothed round trip time.
T1V — Next value for T1; default initial value is initial value of SRT.
N1 — Maximum number of octets in the information field of a frame, excluding inserted 0-bits.
N2 — Maximum number of retries permitted.

Timers:

T1 — Outstanding I frame or P-bit.
T3 — Idle supervision (keep alive).

Figure C4.1. Summary of primitives, states, flags, errors and timers.

Figure C4.2. Data-link disconnected state. (Pages 84-85)

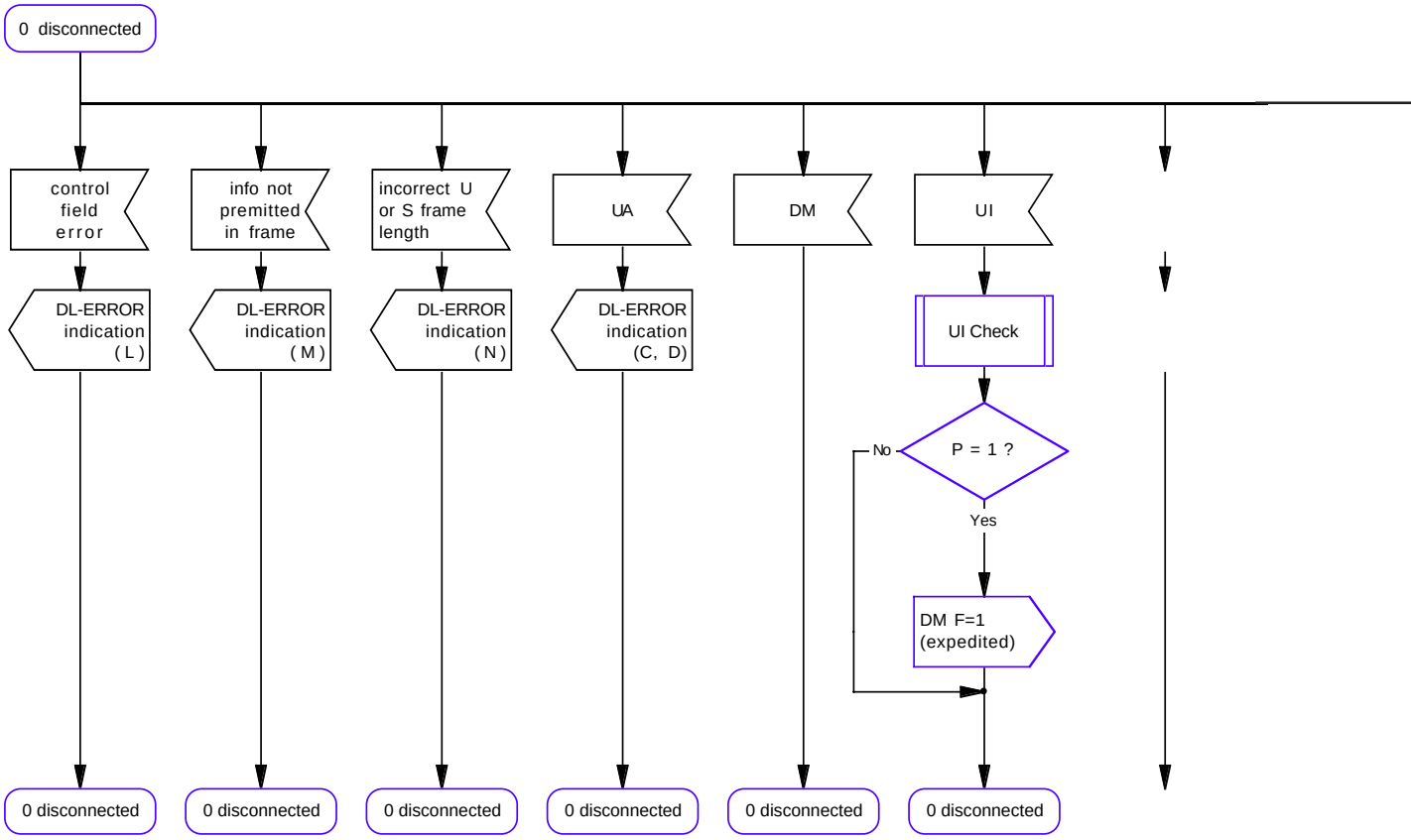


Figure C4.2. Data-link disconnected state. (continued)

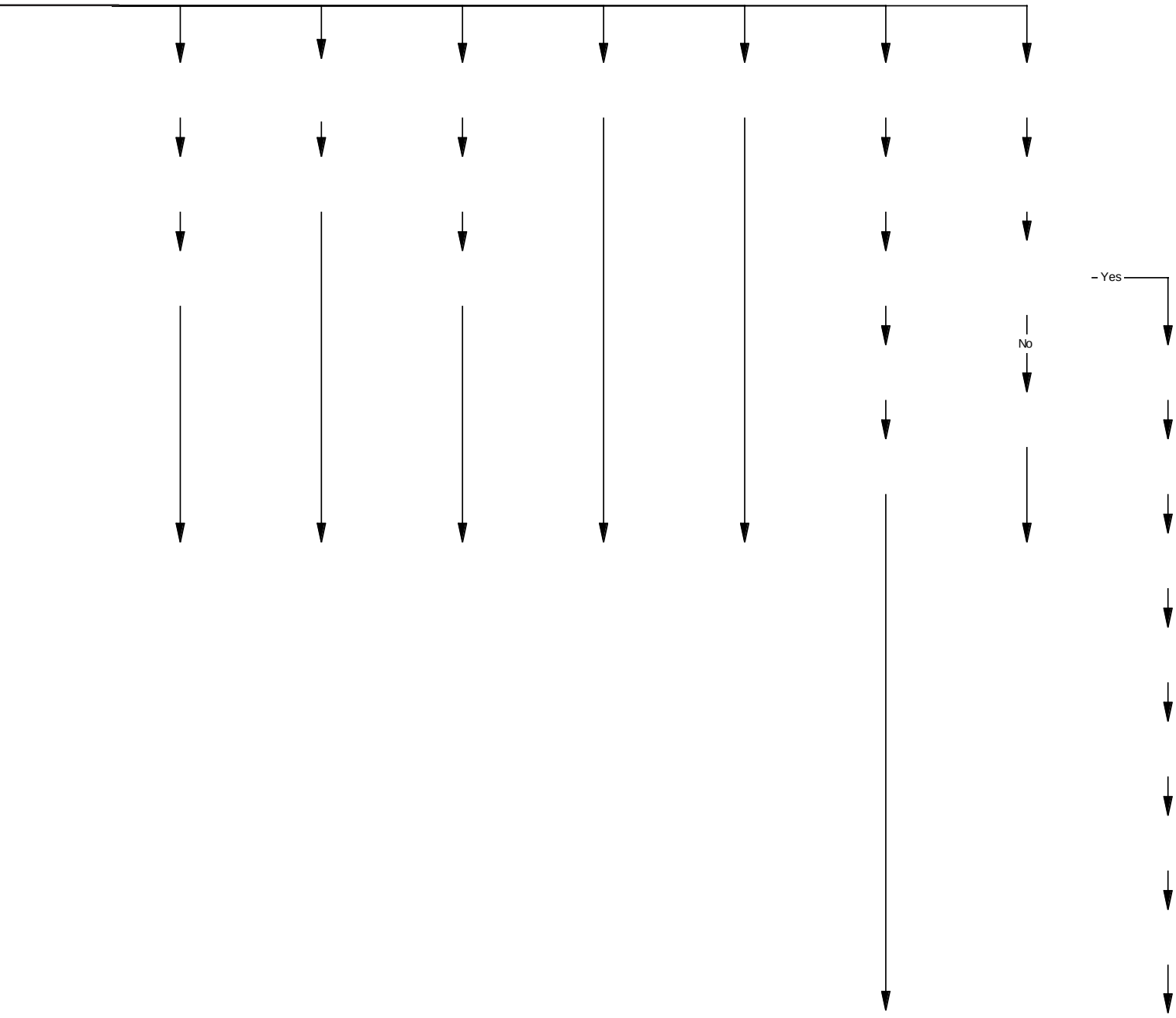


Figure C4.3. Data-link awaiting connection state. (Pages 86-88)

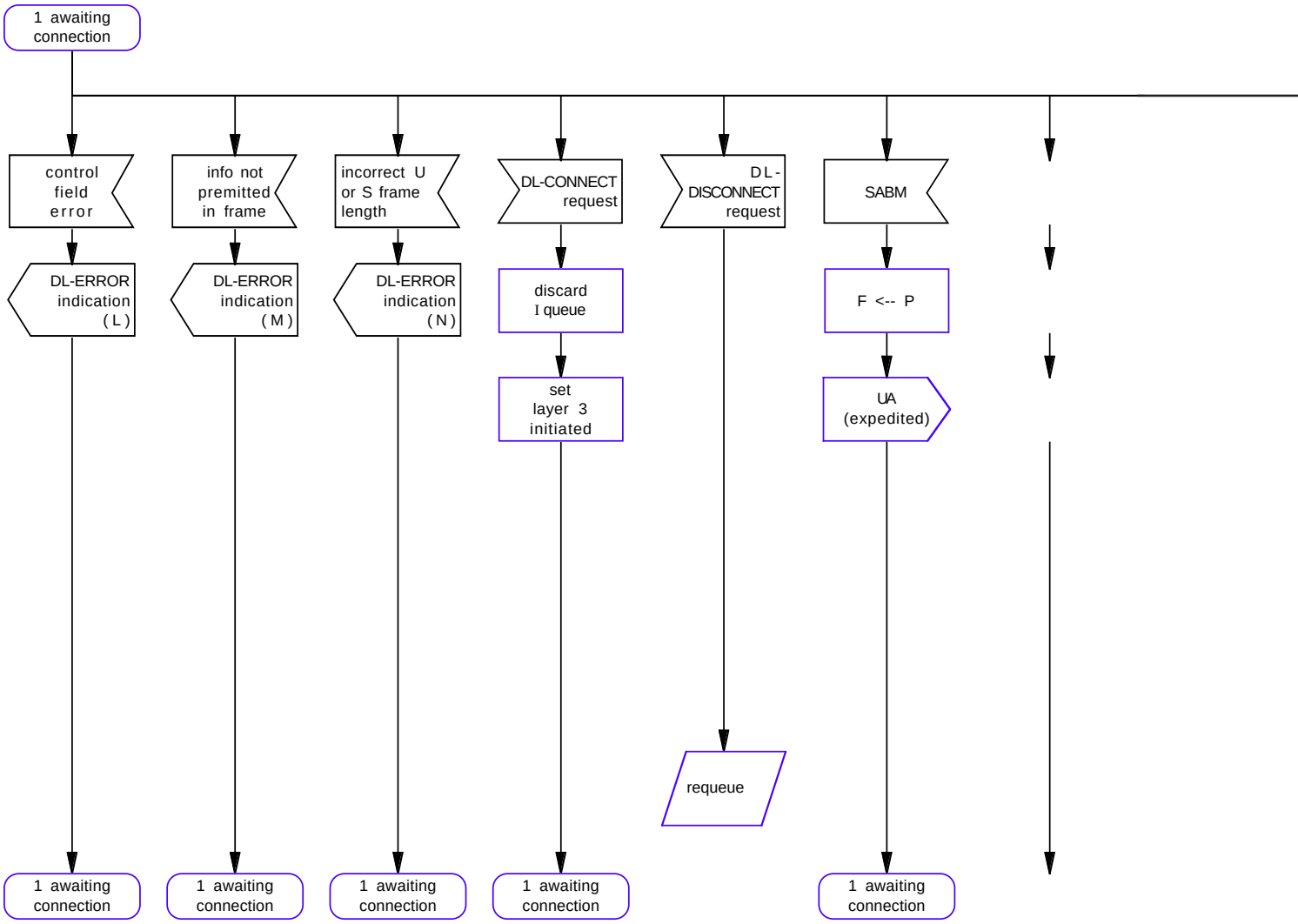


Figure C4.3. Data-link awaiting connection state. (continued)

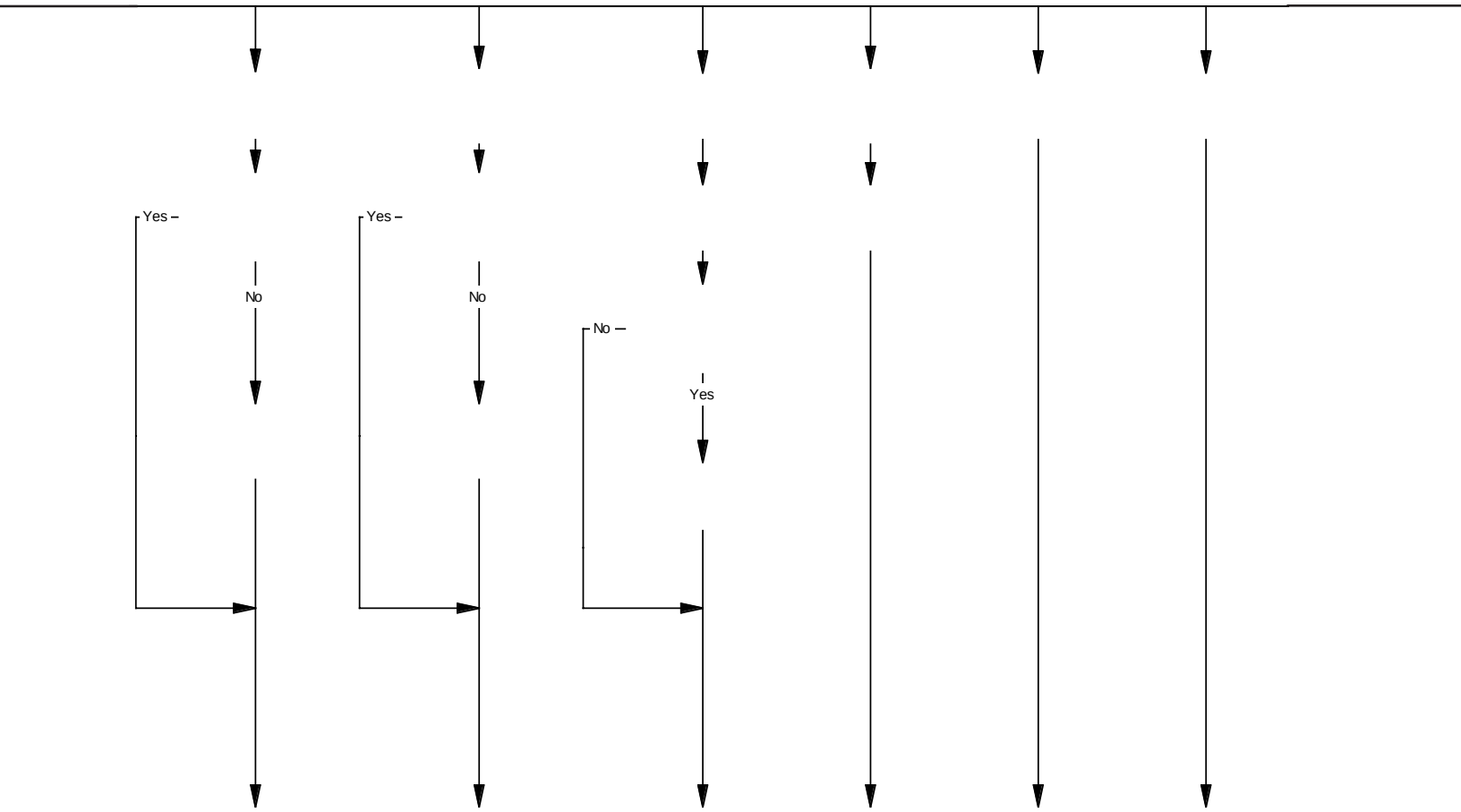


Figure C4.3. Data-link awaiting connection state. (continued)

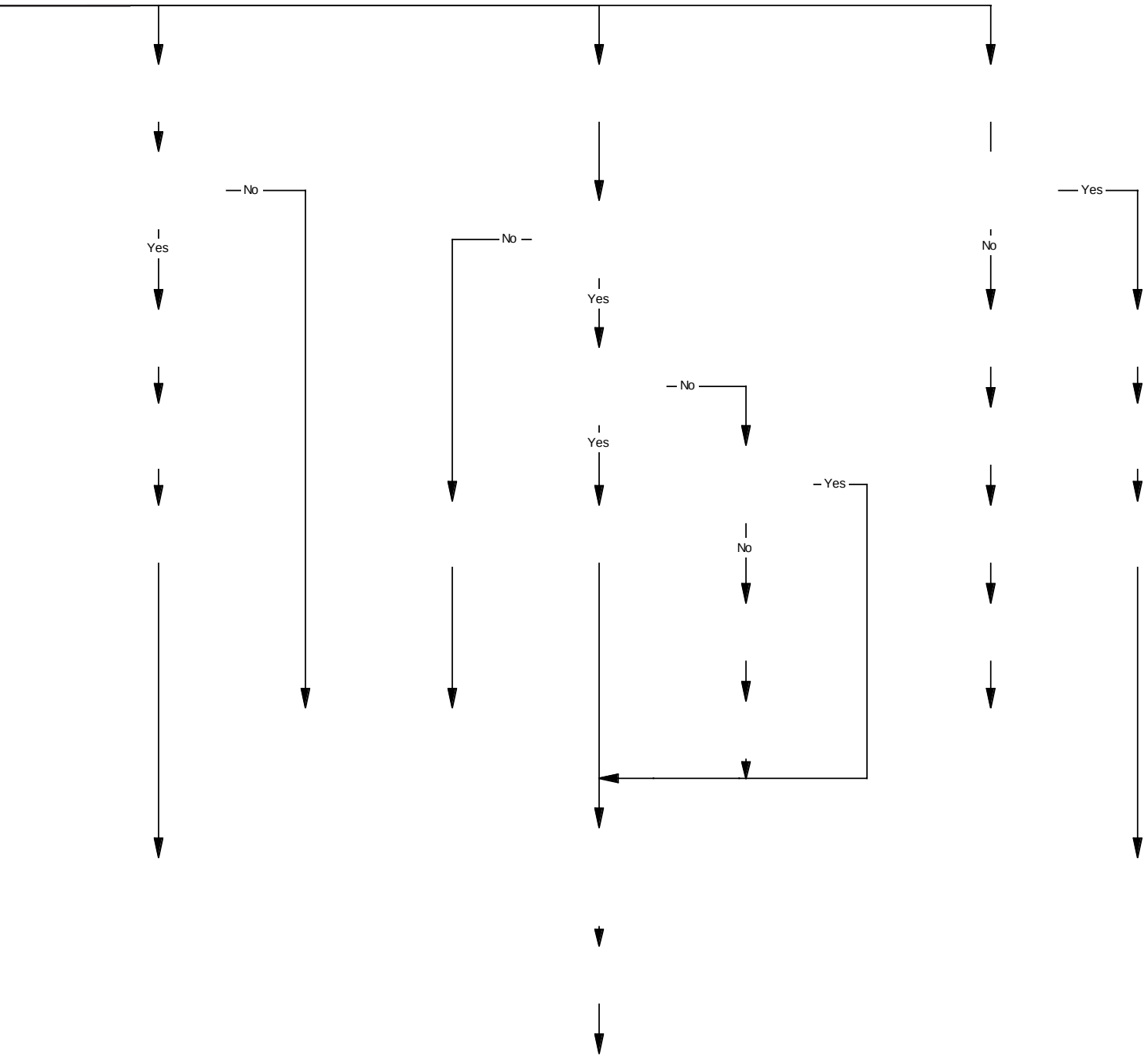


Figure C4.4. Data-link awaiting release state. (Pages 89-91)

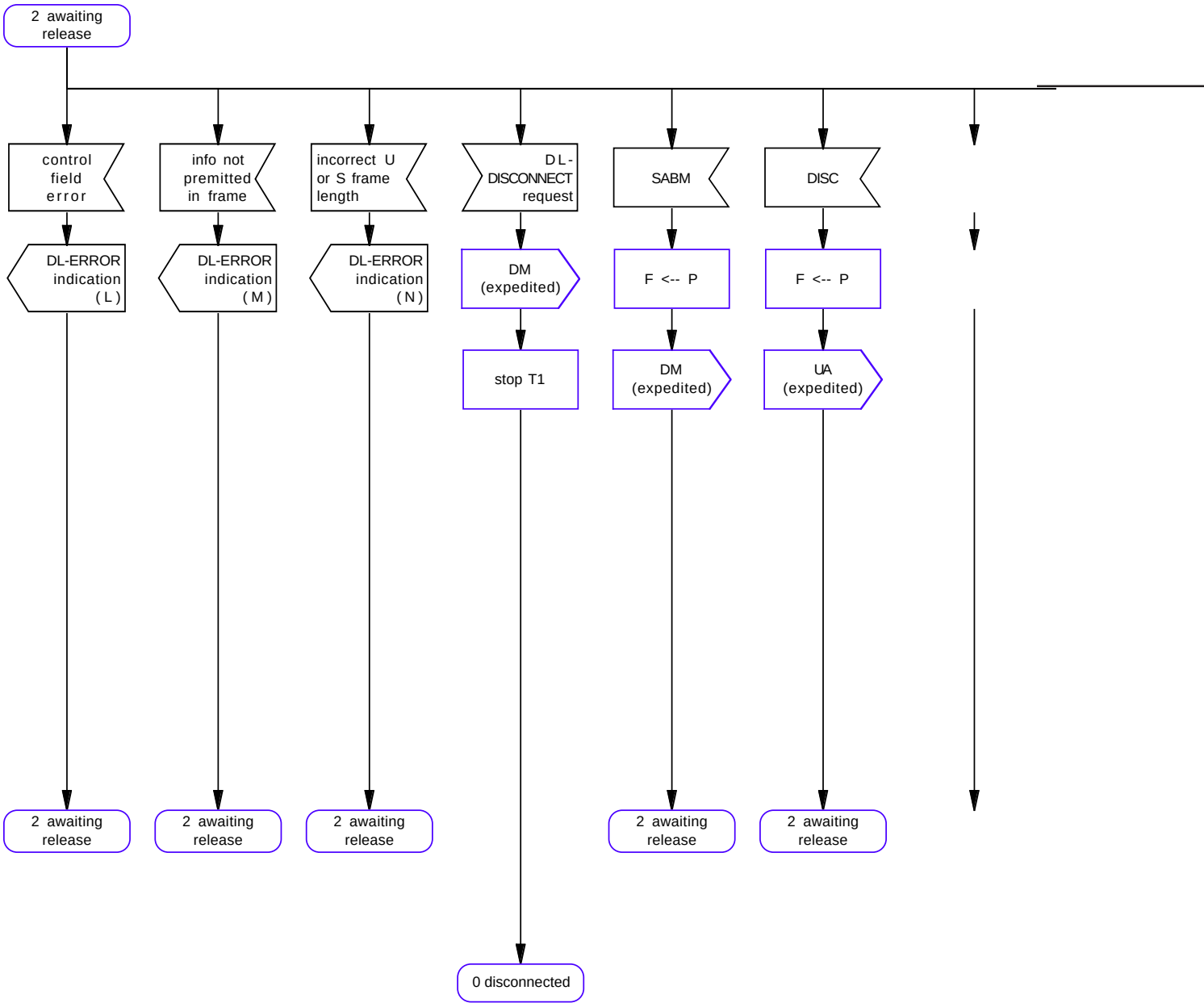


Figure C4.4. Data-link awaiting release state. (Continued)

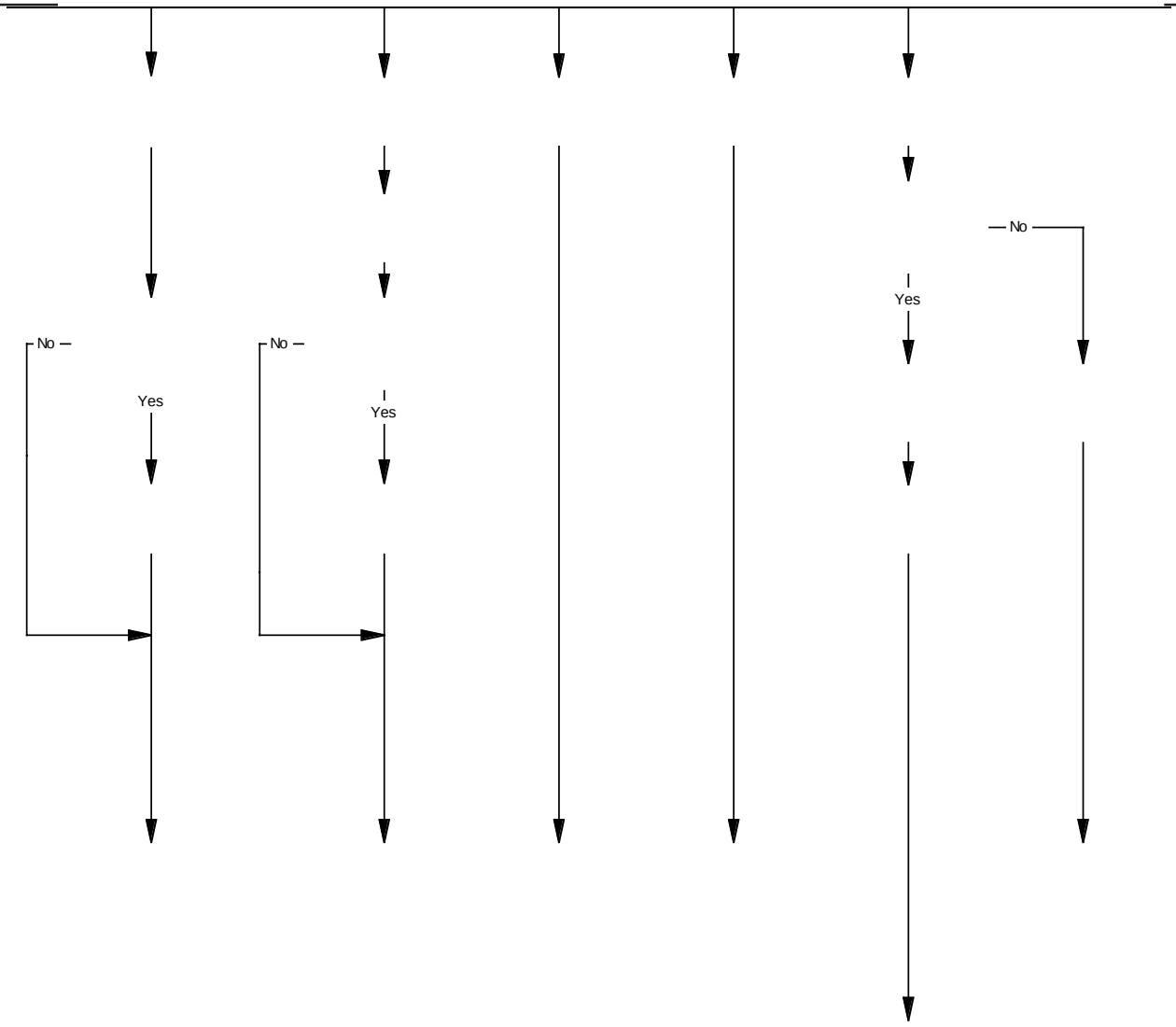


Figure C4.4. Data-link awaiting release state. (Continued)

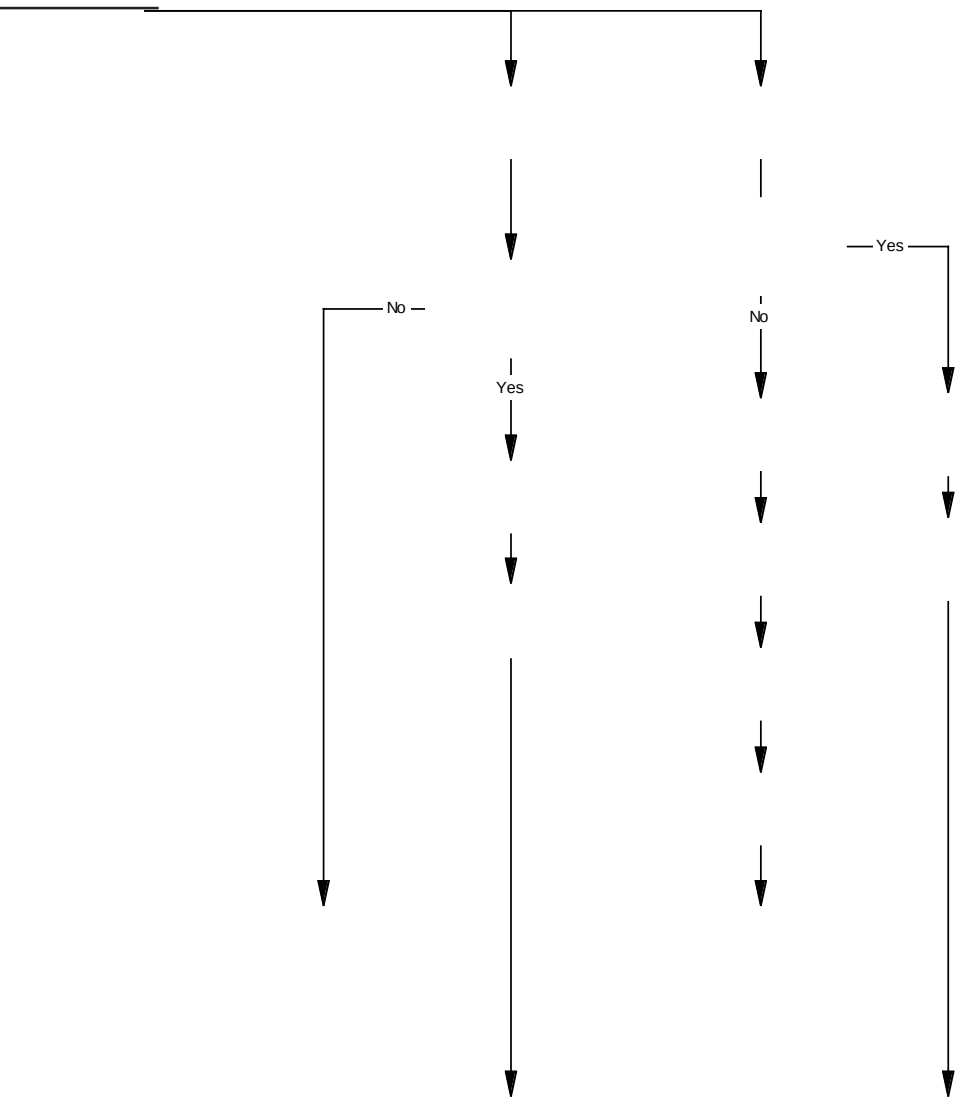


Figure C4.5. Data-link connected state. (Pages 92-97)

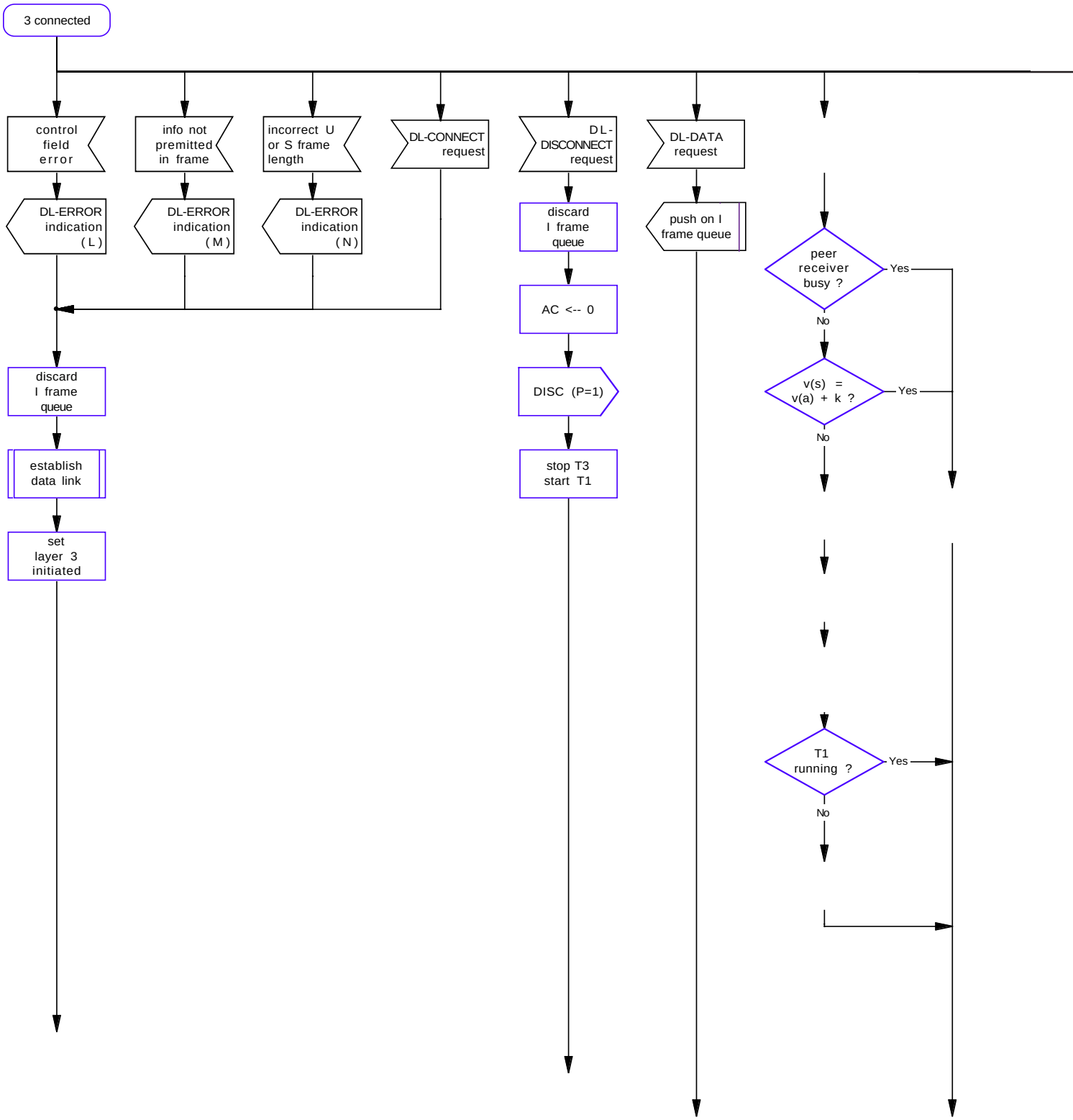


Figure C4.5. Data-link connected state. (Continued)

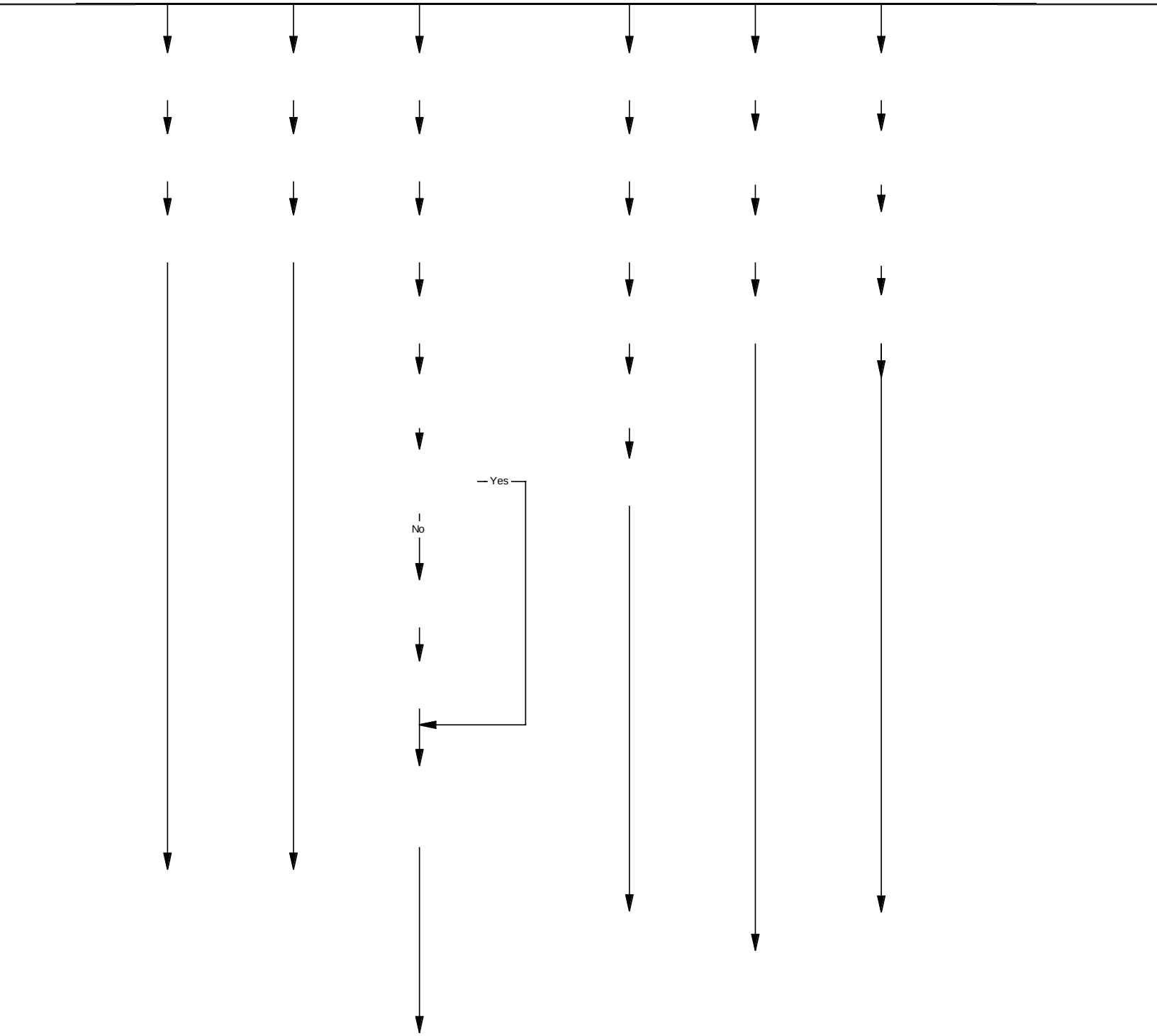


Figure C4.5. Data-link connected state. (Continued)

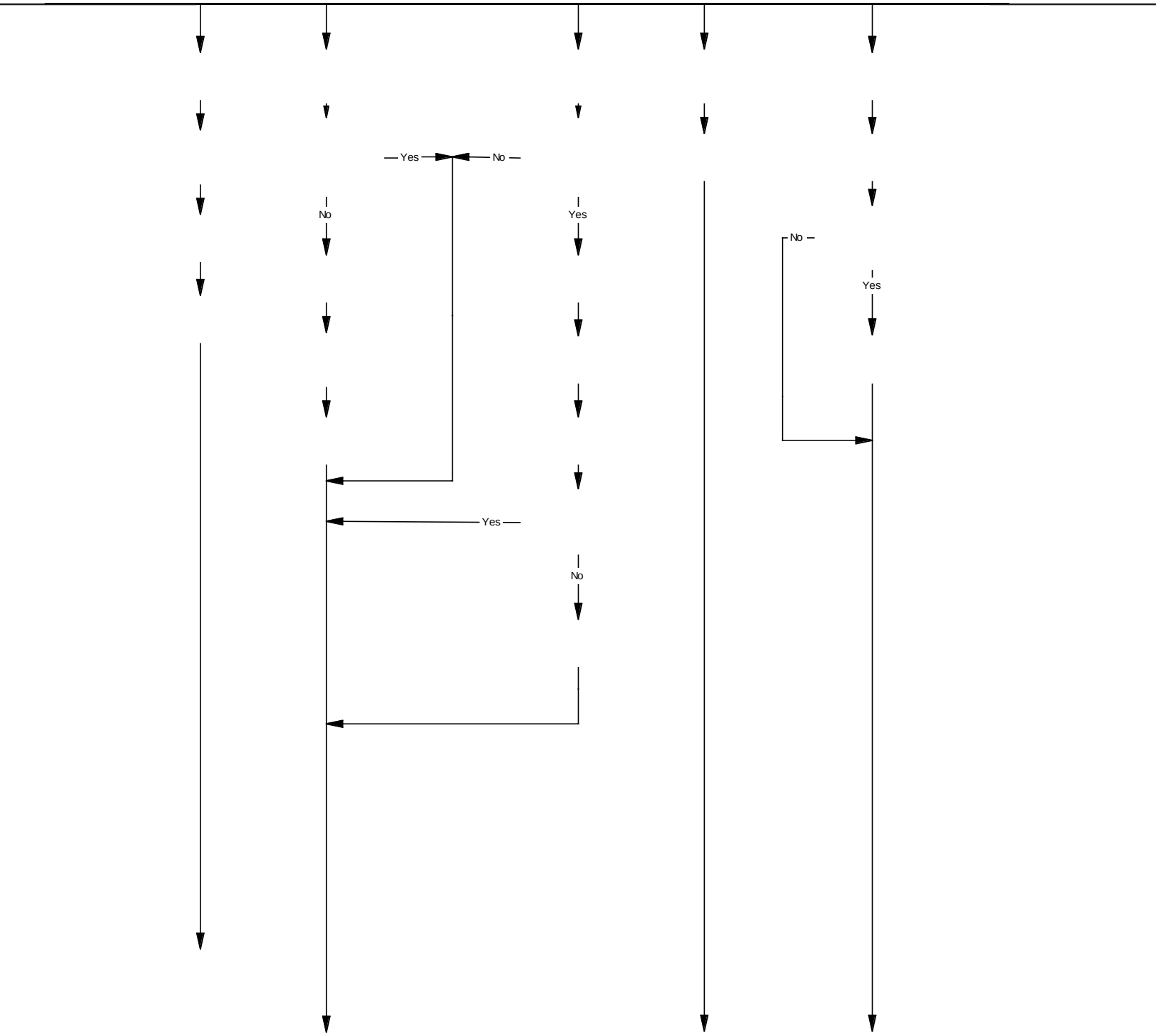


Figure C4.5. Data-link connected state. (Continued)

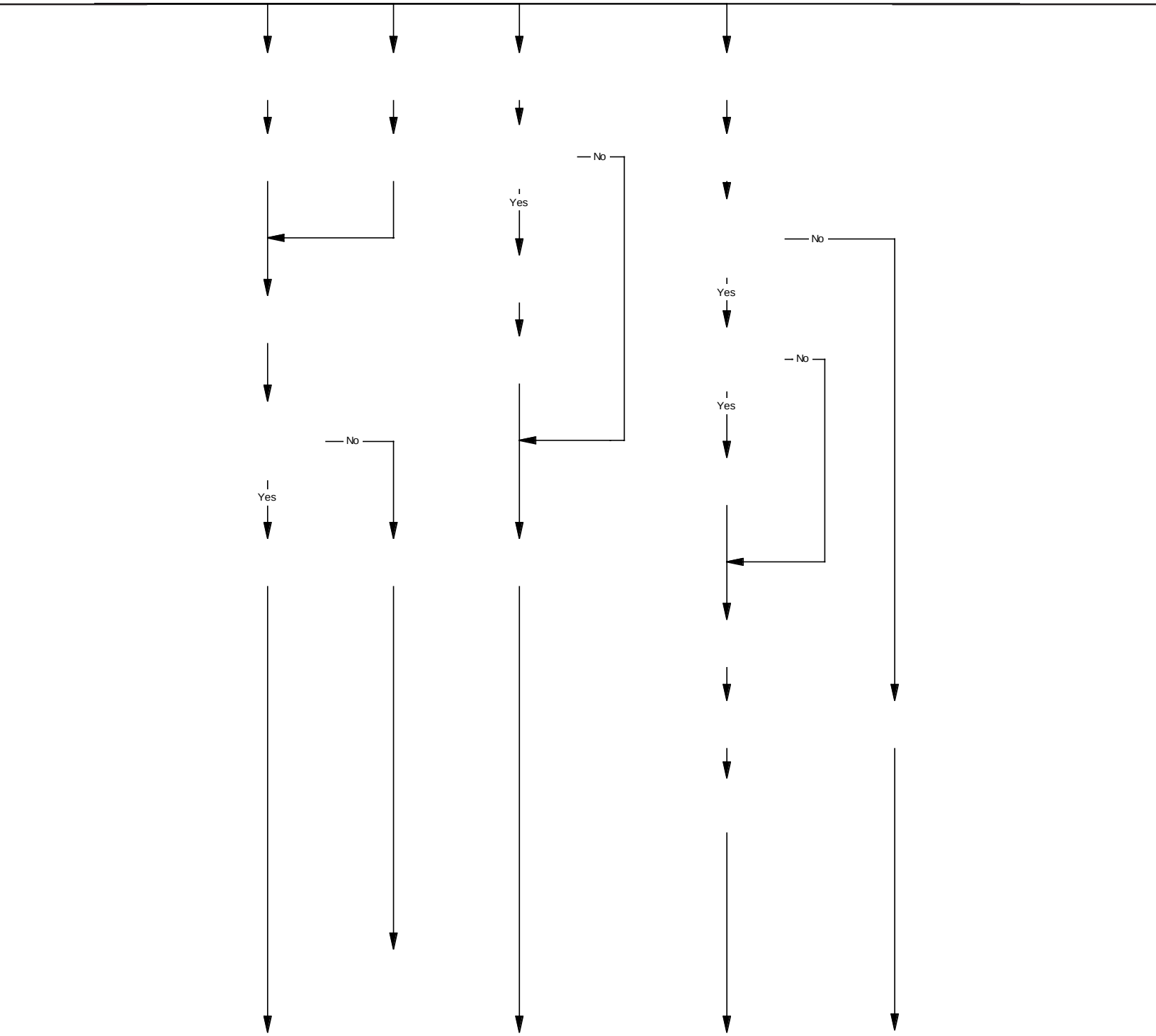


Figure C4.5. Data-link connected state. (Continued)

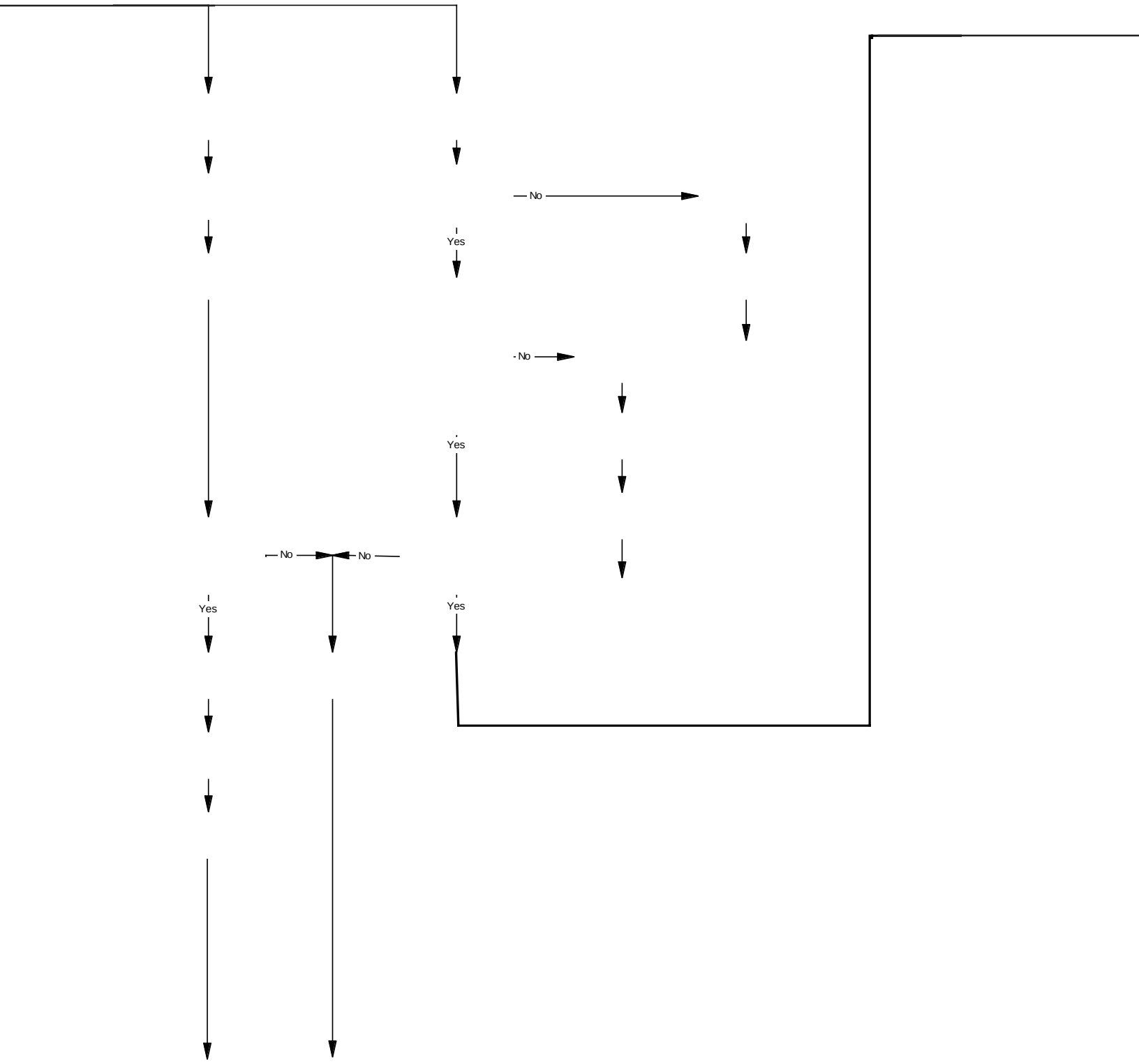


Figure C4.5. Data-link connected state. (Continued)

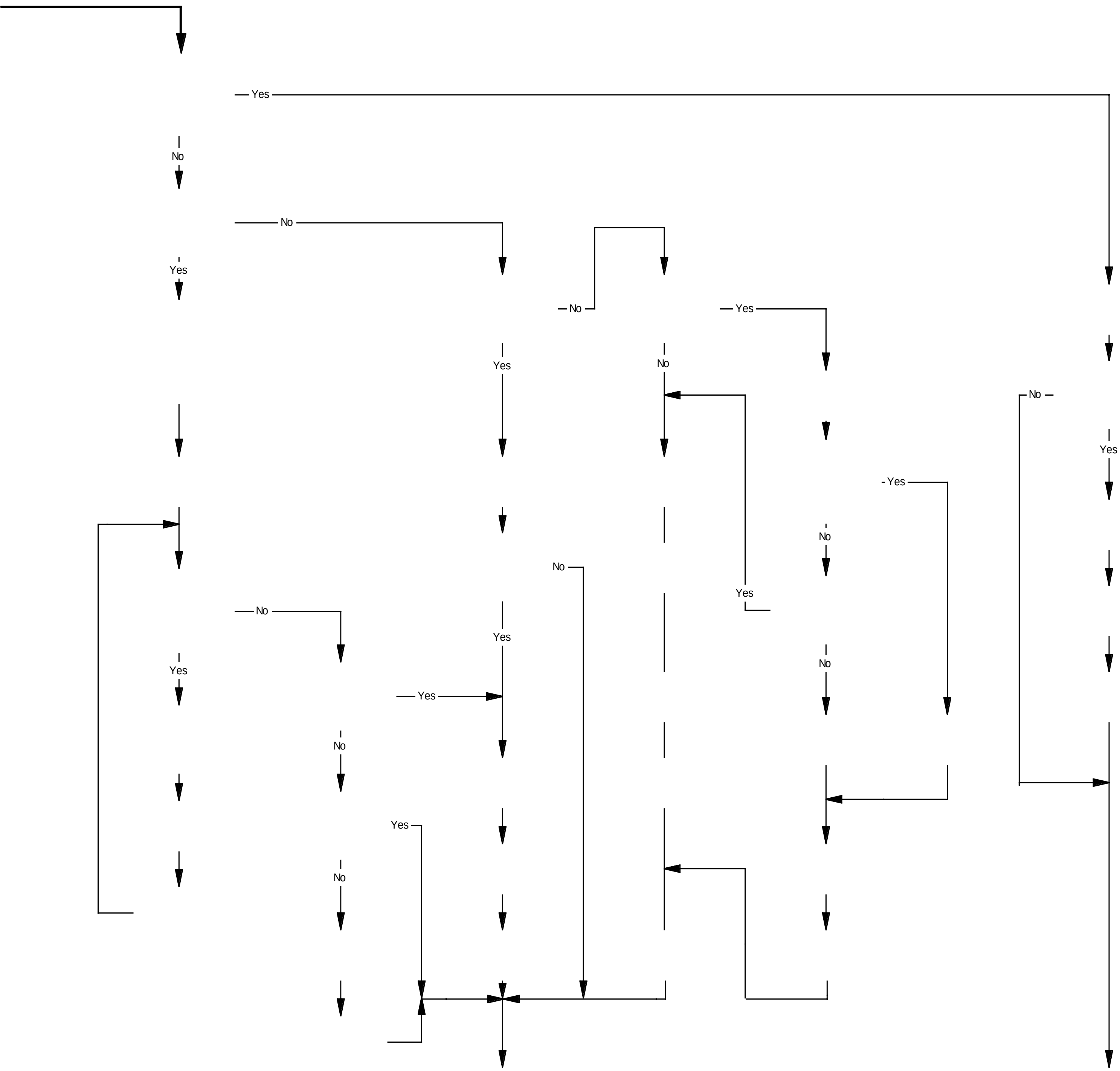


Figure C4.6. Data-link timer recovery state. (Pages 98-102)

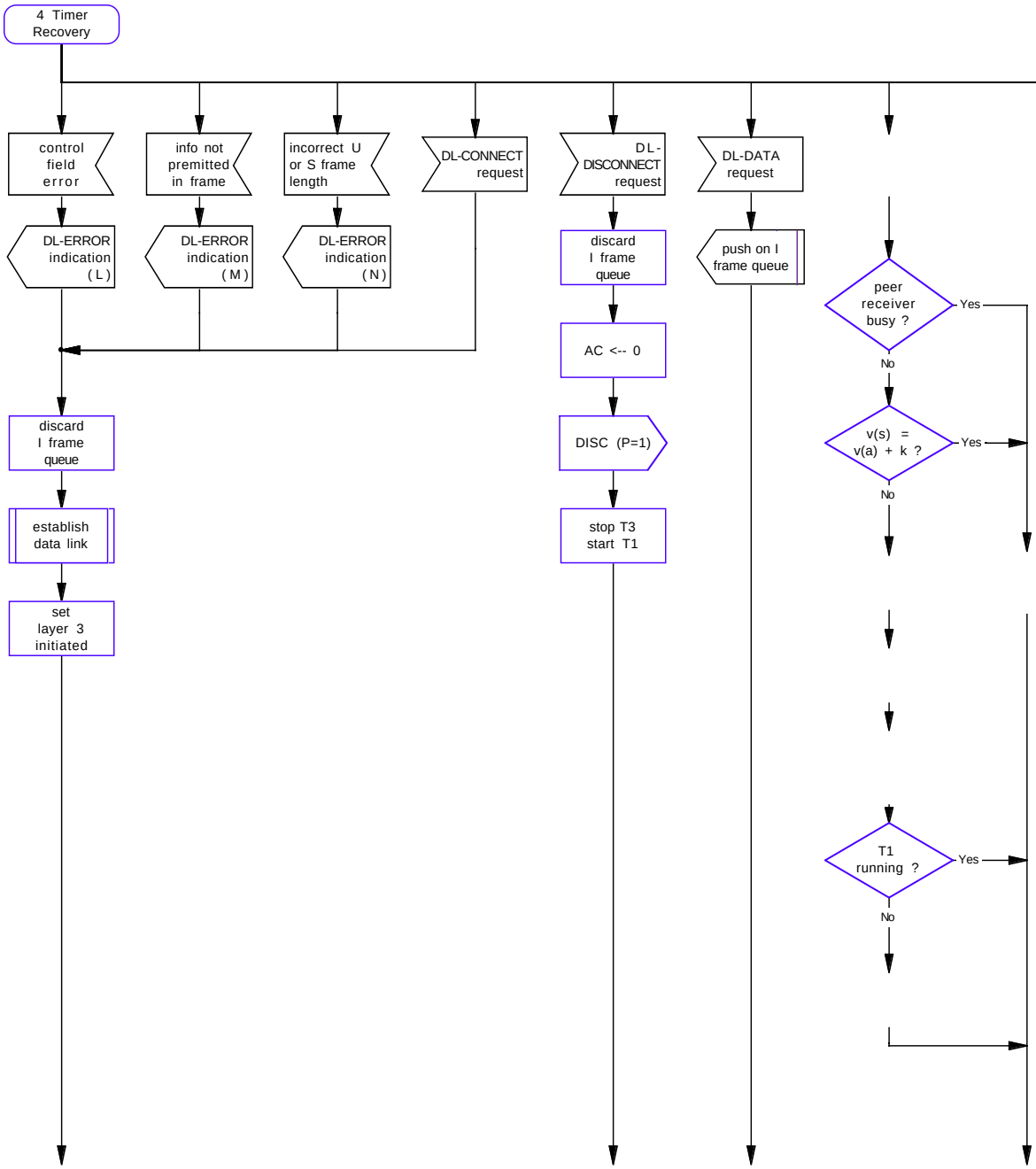


Figure C4.6. Data-link timer recovery state. (Continued)

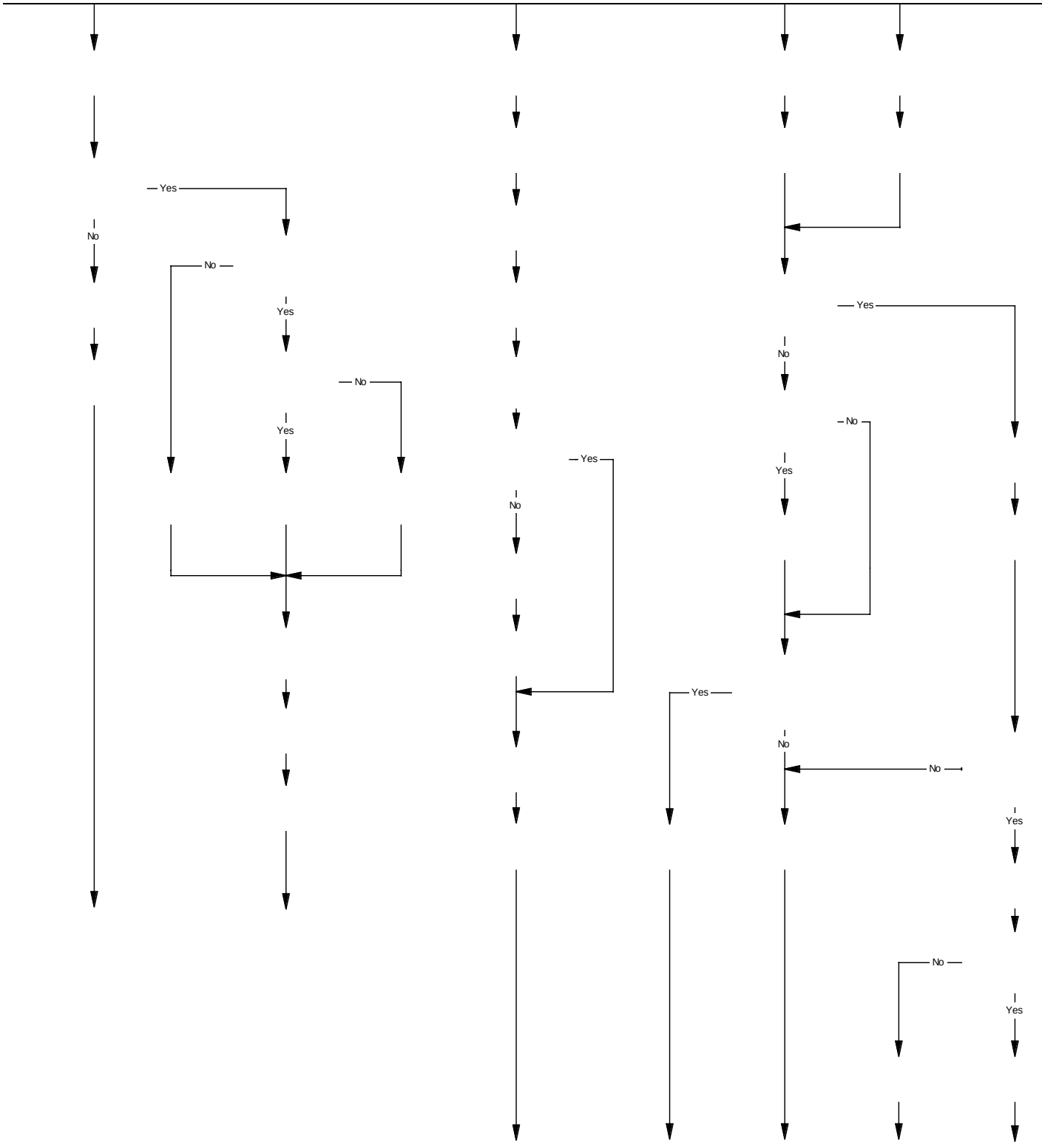


Figure C4.6. Data-link timer recovery state. (Continued)

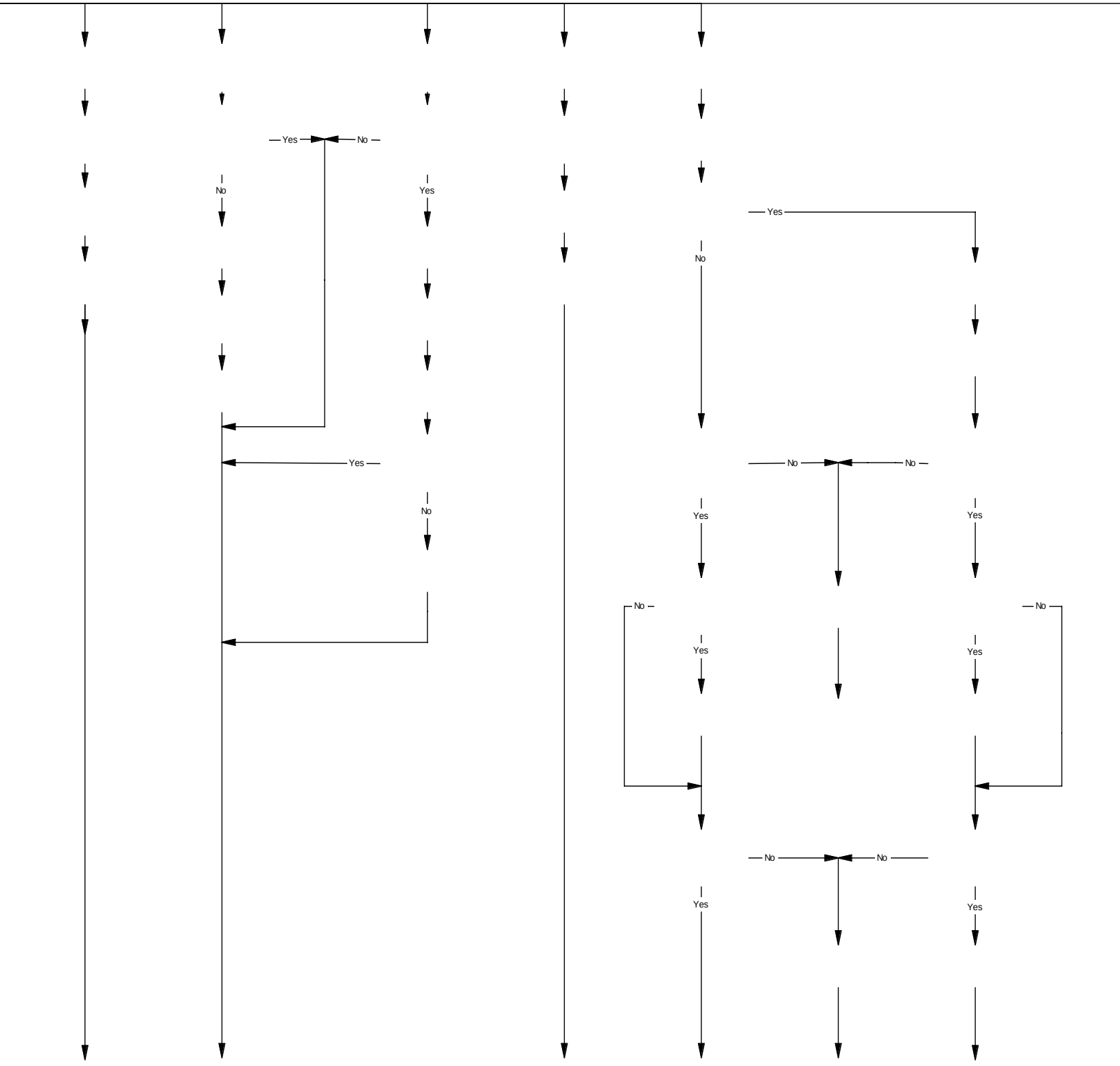


Figure C4.6. Data-link timer recovery state. (Continued)

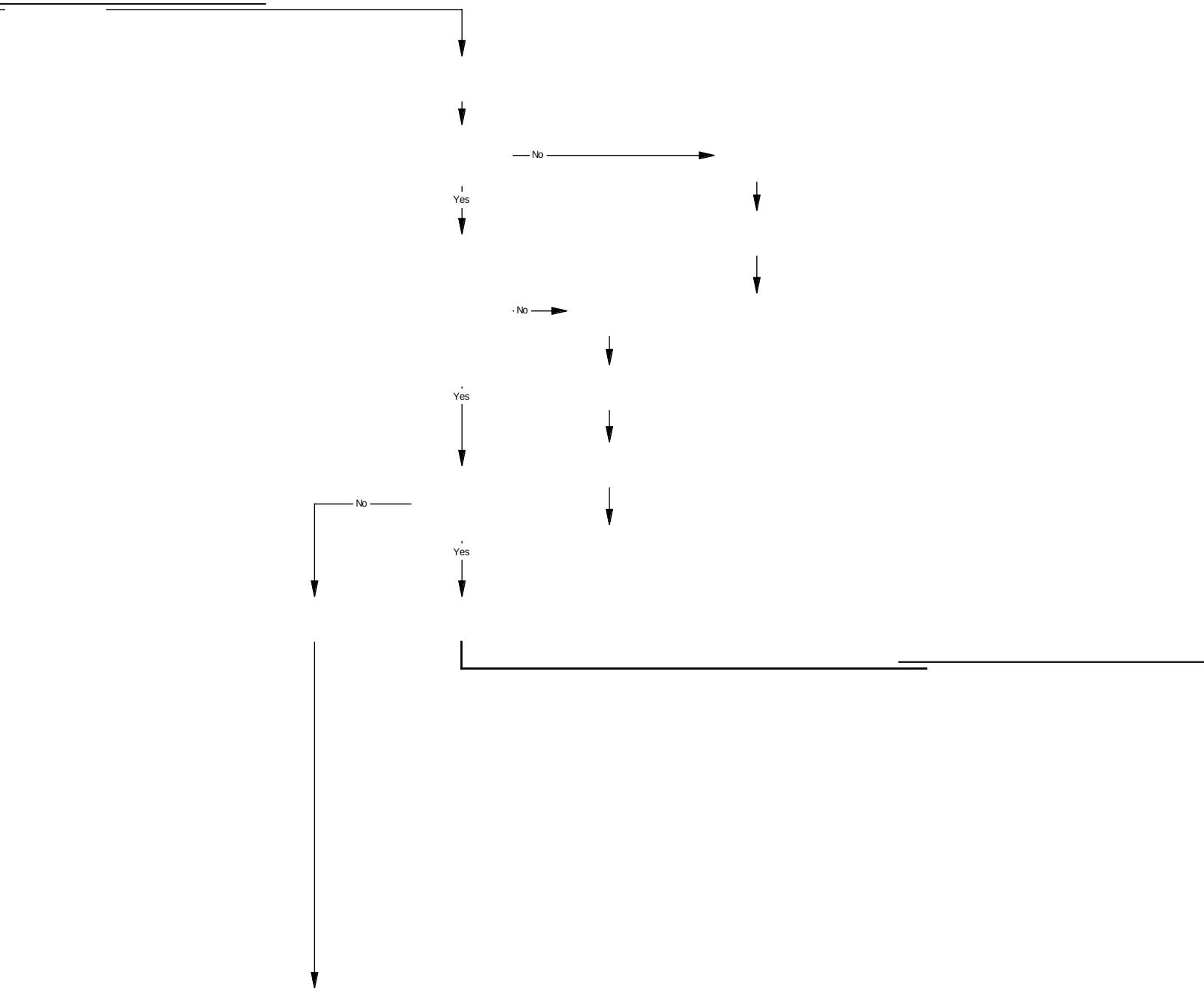


Figure C4.6. Data-link timer recovery state. (Continued)

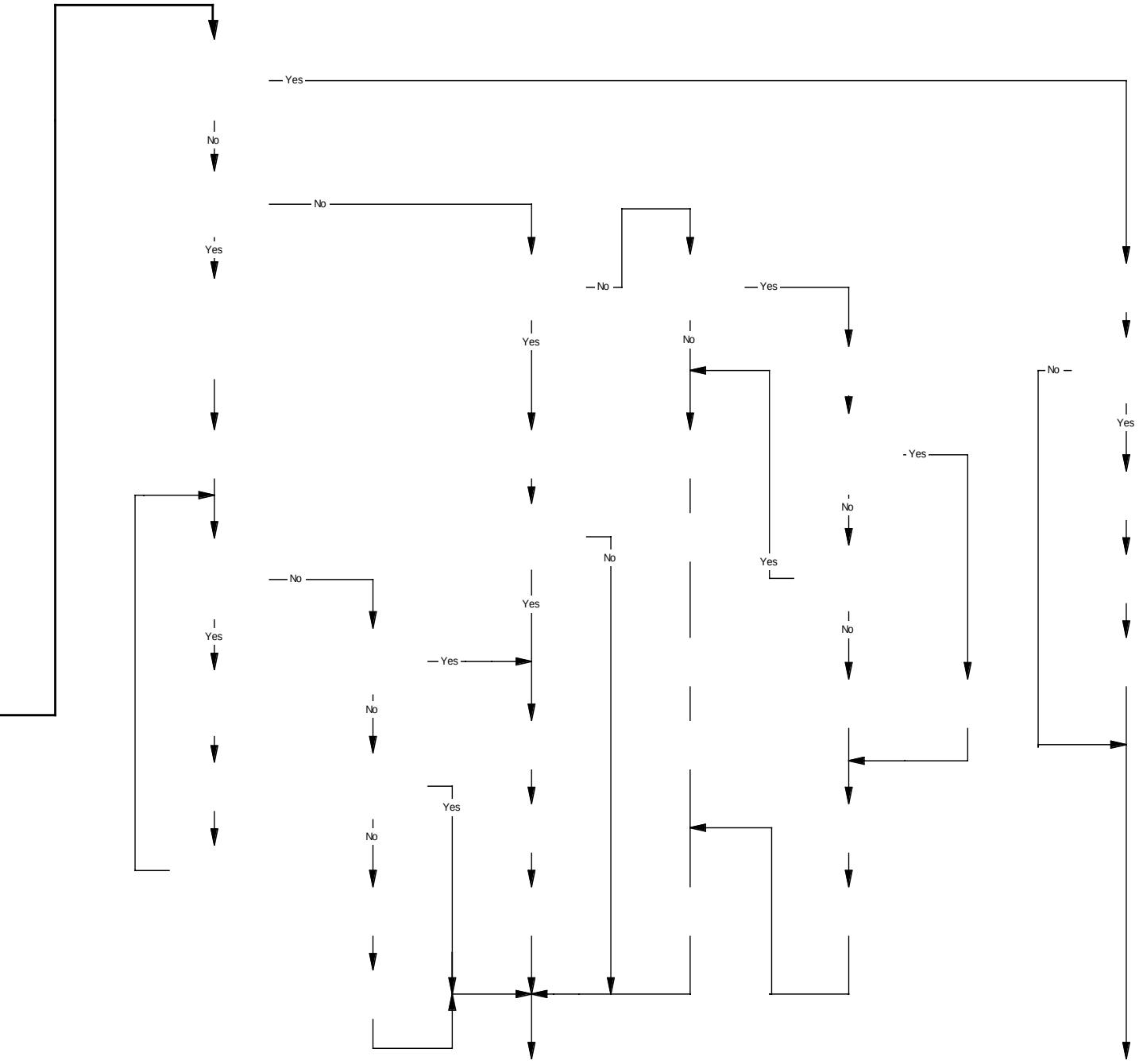


Figure C4.7. Data-link subroutines state. (Pages 103-106)

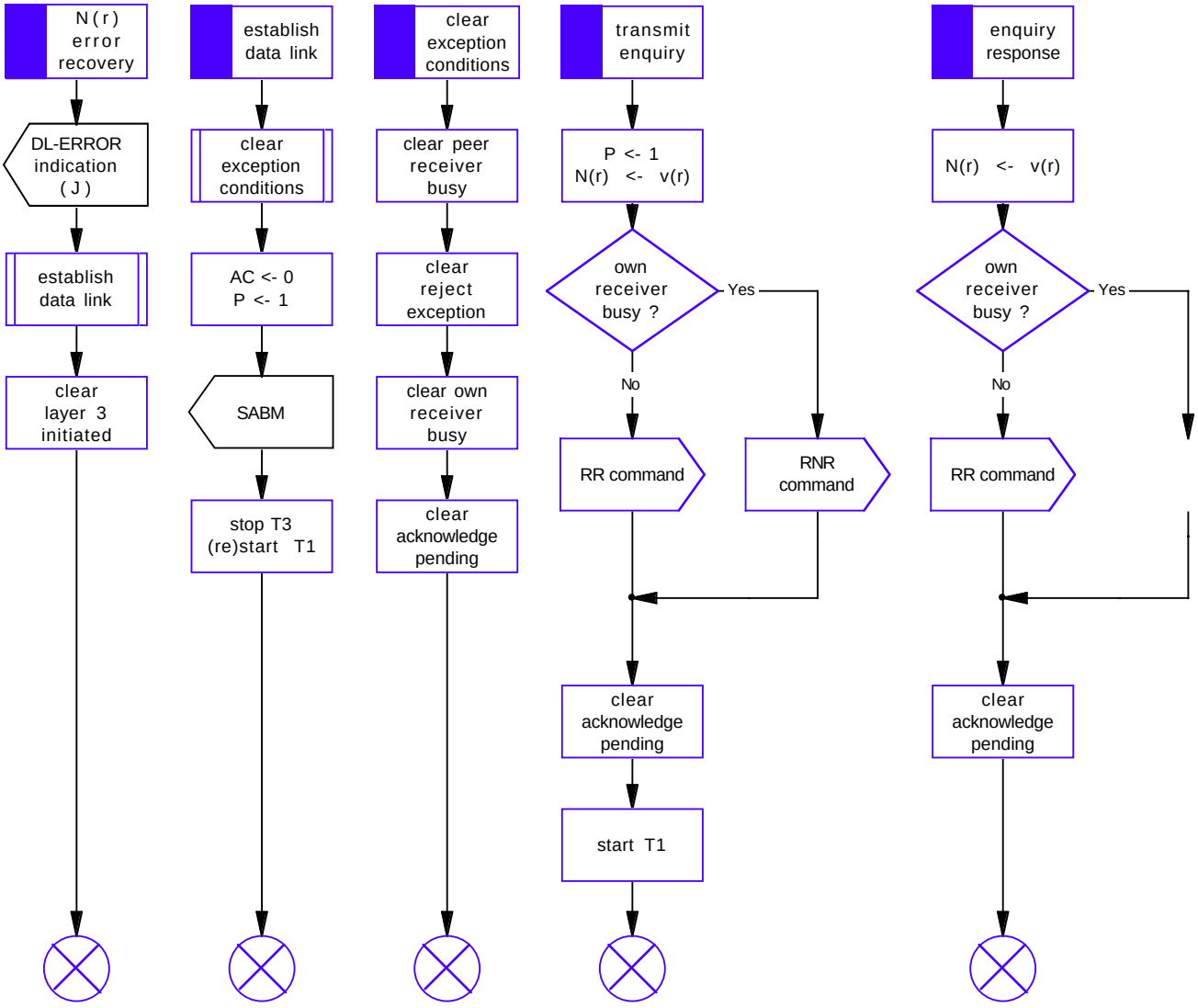


Figure C4.7. Data-link subroutines state. (Continued)

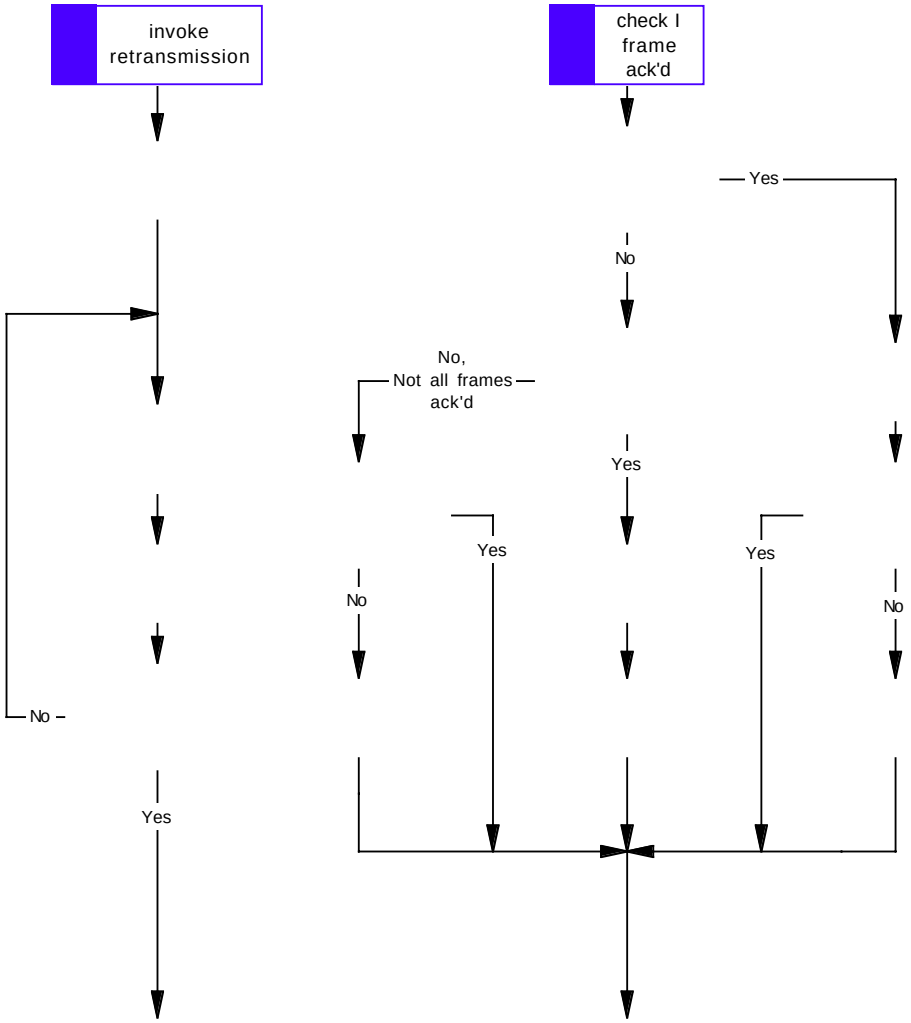


Figure C4.7. Data-link subroutines state. (Continued)

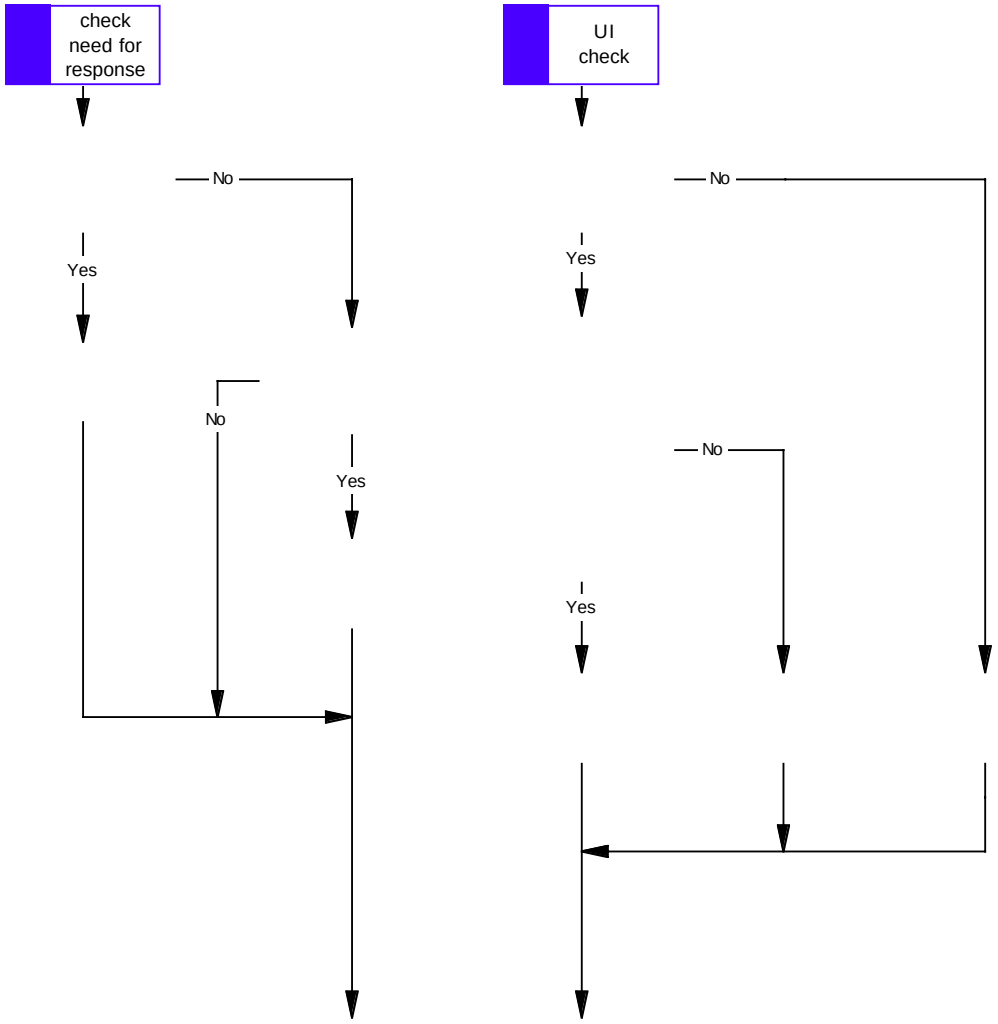
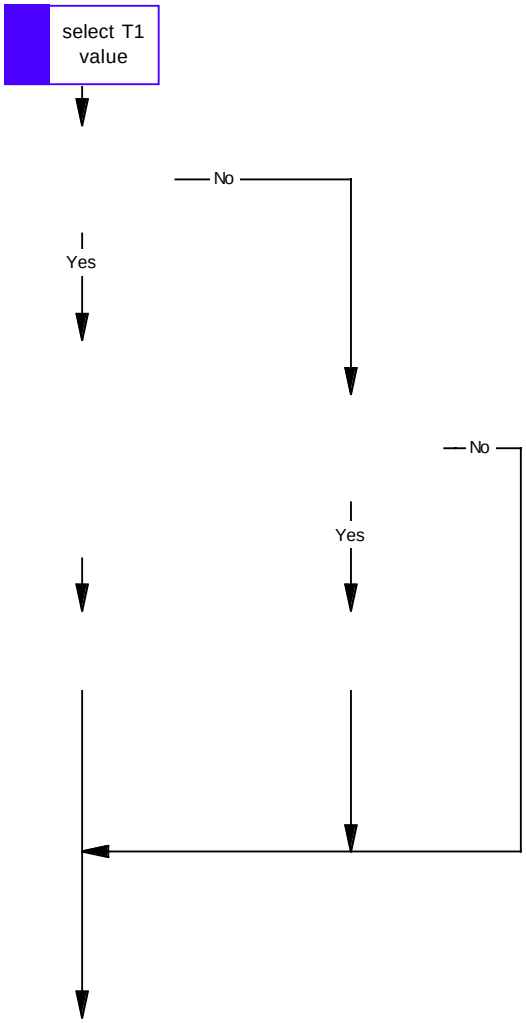


Figure C4.7. Data-link subroutines state. (Continued)



Appendix C5

Management Data-Link State Machine

C5.1. Interaction with the Data-Link Service Access Point

The Data-Link Service Access Point directs the operation of the Management Data-link State Machine through the management data link (MDL) primitives described below.

- **MDL-NEGOTIATE Request**. This primitive is used by the Layer 3 entity to request the Data-link State Machine to notify/negotiate.

- **MDL-NEGOTIATE Confirm**. This primitive is used by the Management Data-link State Machine to notify the Layer 3 entity notification/negotiation is complete.

- **MDL-ERROR Indicate**. This primitive is used by the Management Data-link State Machine to notify the Layer 3 entity notification/negotiation has failed.

C5.2. Interaction with the Link Multiplexer State Machine

The Management Data-link State Machine directs the operation of the Link Multiplexer State Machine through the link multiplexer (LM) primitives described below.

- **LM-DATA Request**. This primitive is used by the Data-link State Machines to pass frames of any type (SABM, RR, UI, etc.) to the Link Multiplexer State Machine.

- **LM-DATA Indication**. This primitive is used by the Link Multiplexer State Machines to pass frames of any type (SABM, RR, UI, etc.) to the Data-link State Machine.

C5.3. Internal Operation of the Machine

The internal states, queues and flags are summarized in Figure C5.1.

The Management Data-link State Machine handles the negotiation/notification of operational parameters. It uses a single command/response exchange to negotiate the final values of negotiable parameters.

The station initiating the AX.25 connection will send an XID command after it receives the UA frame. If the other station is using a version of AX.25 earlier than 2.2, it will respond with an FRMR of the XID command and the default version 2.0 parameters will be used. If the other station is using version 2.2 or better, it will respond with an XID response.

MDL Primitives (Received from MDL):
MDL-NEGOTIATE Confirm
MDL-ERROR Indicate

MDL Primitives (Sent to MDL):
MDL-NEGOTIATE Request

LM Primitives (Sent to LM):
LM-DATA Request

LM Primitives (Received from LM):
LM-DATA Indicate

States:
0 — Ready
1 — Negotiating

Error Codes:
A — XID command without P=1.
B — Unexpected XID response.
C — Management retry limit exceeded.
D — XID response without F=1.

Queues:
None used.

Flags:
RC — Retry count.
NM201 — Maximum number of retries of the XID command.

Timers:
TM201 — Retry timer for management functions.

Figure C5.1. Summary of primitives, states, flags, errors and timers.

Figure C5.1. Management Data-link ready state.

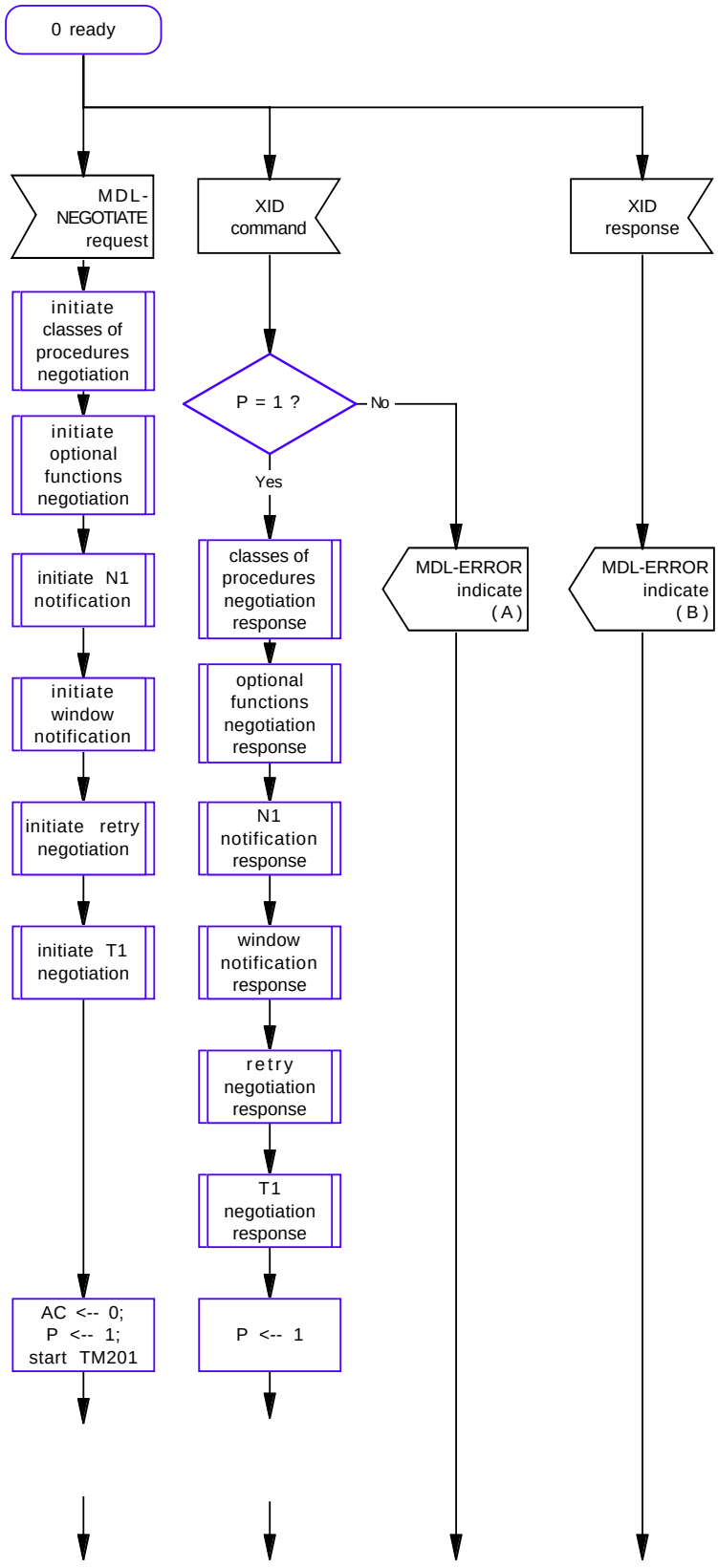


Figure C5.2. Management Data-link negotiating state.

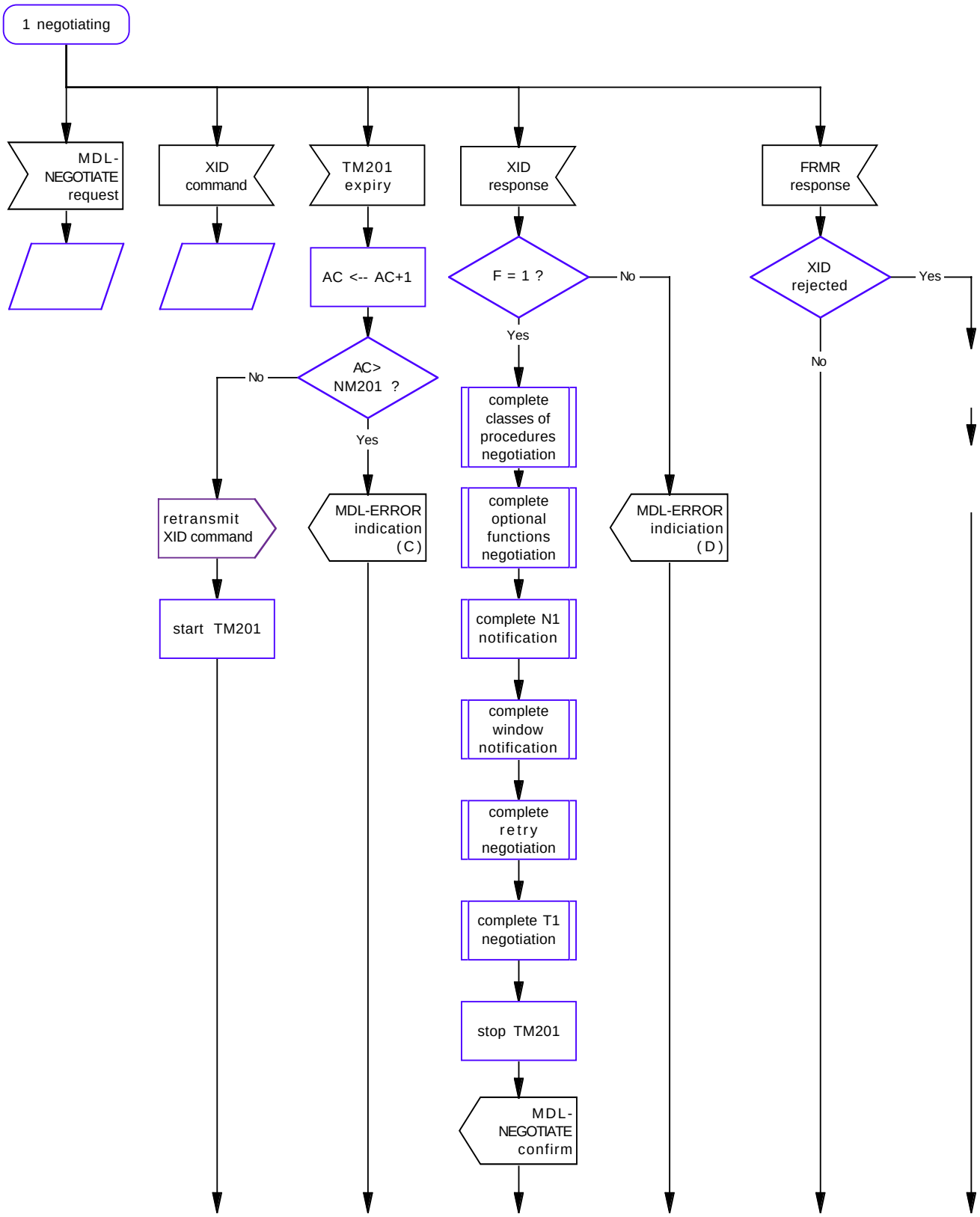


Figure C5.3. Management Data-link N1 notification subroutine.

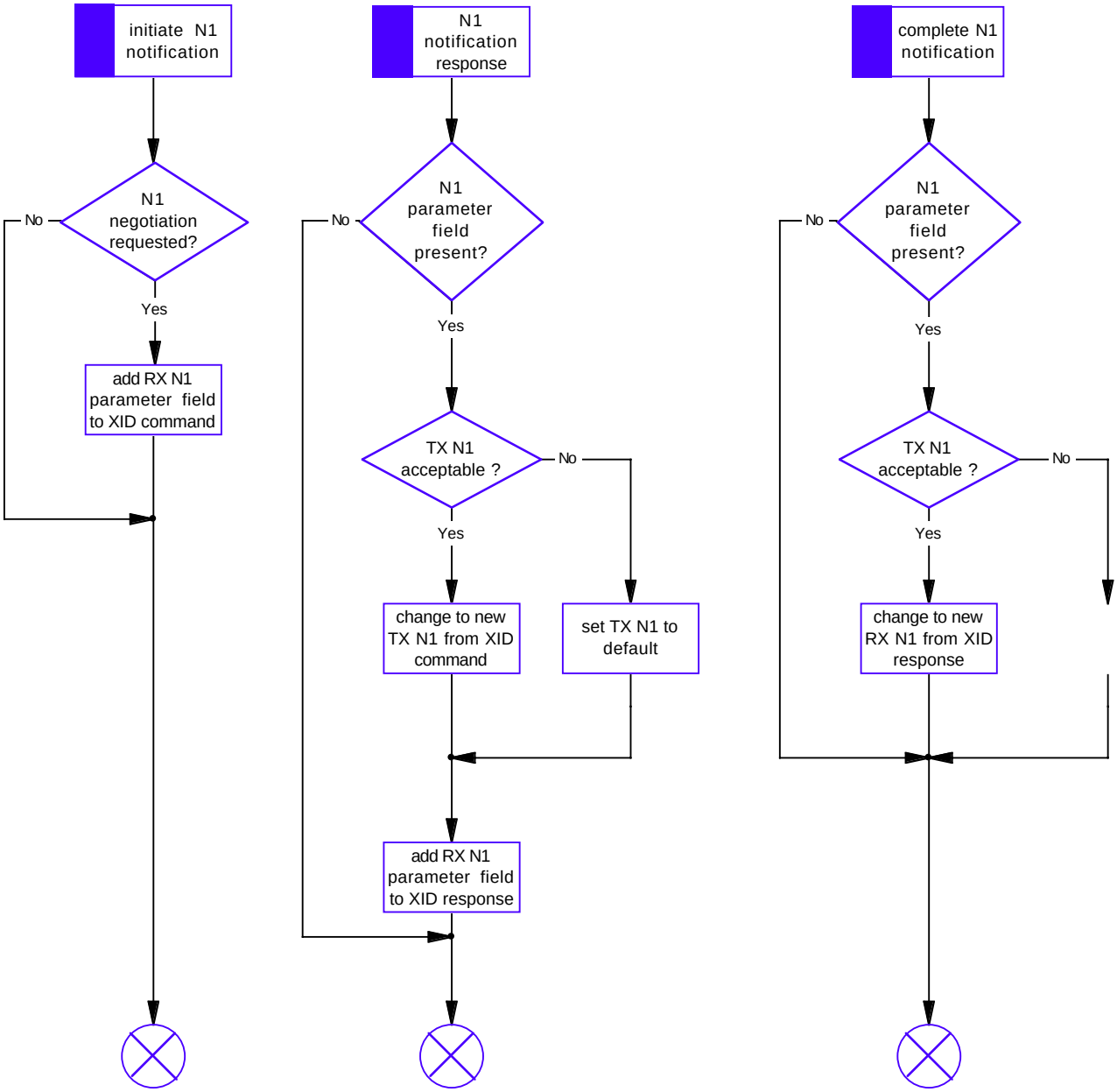


Figure C5.4. Management Data-link window notification subroutine.

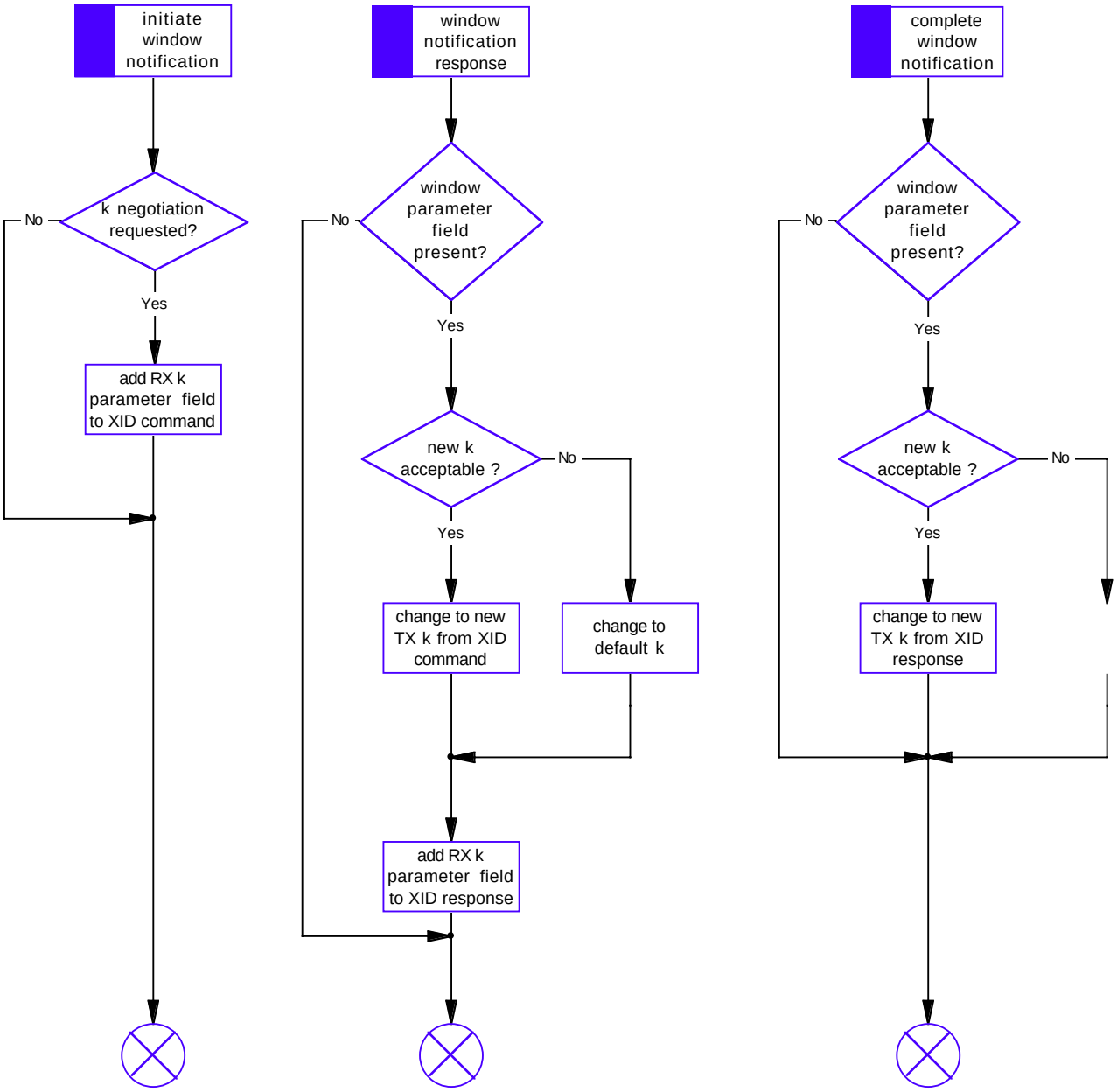


Figure C5.5. Management Data-link T1 negotiation subroutine.

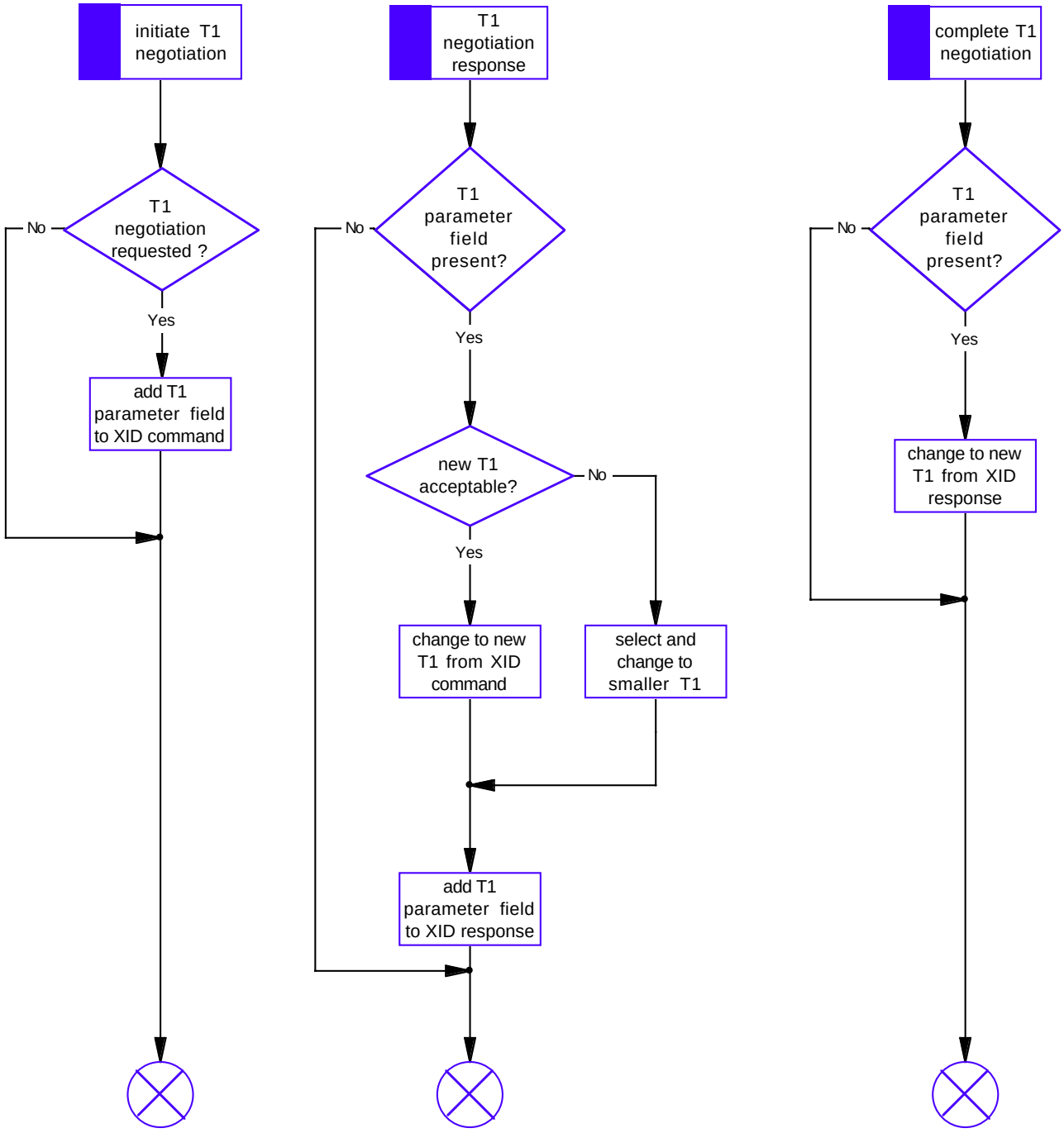


Figure C5.6. Management Data-link retry notification subroutine.

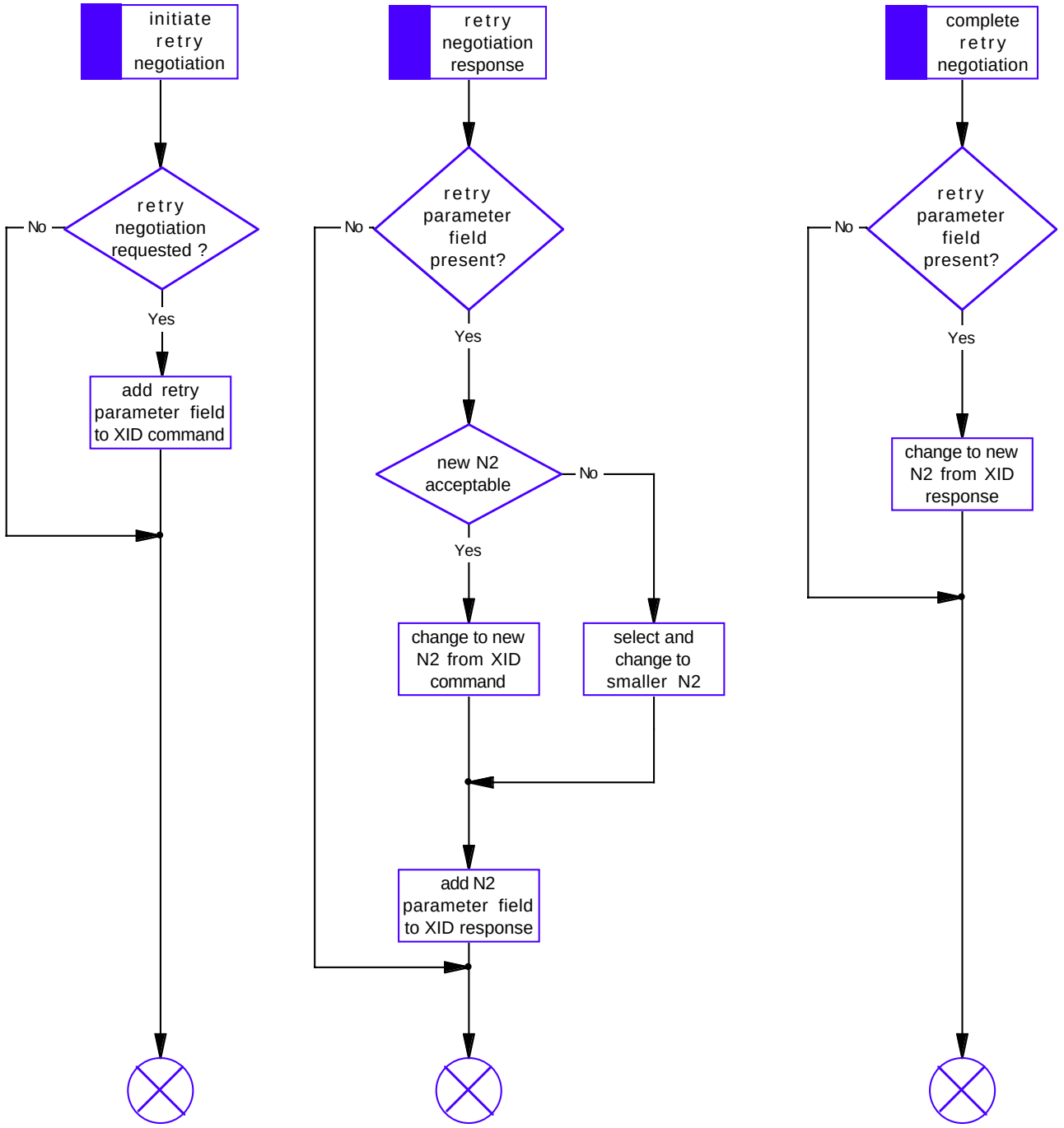


Figure C5.7. Management Data-link optional functions negotiation subroutine.

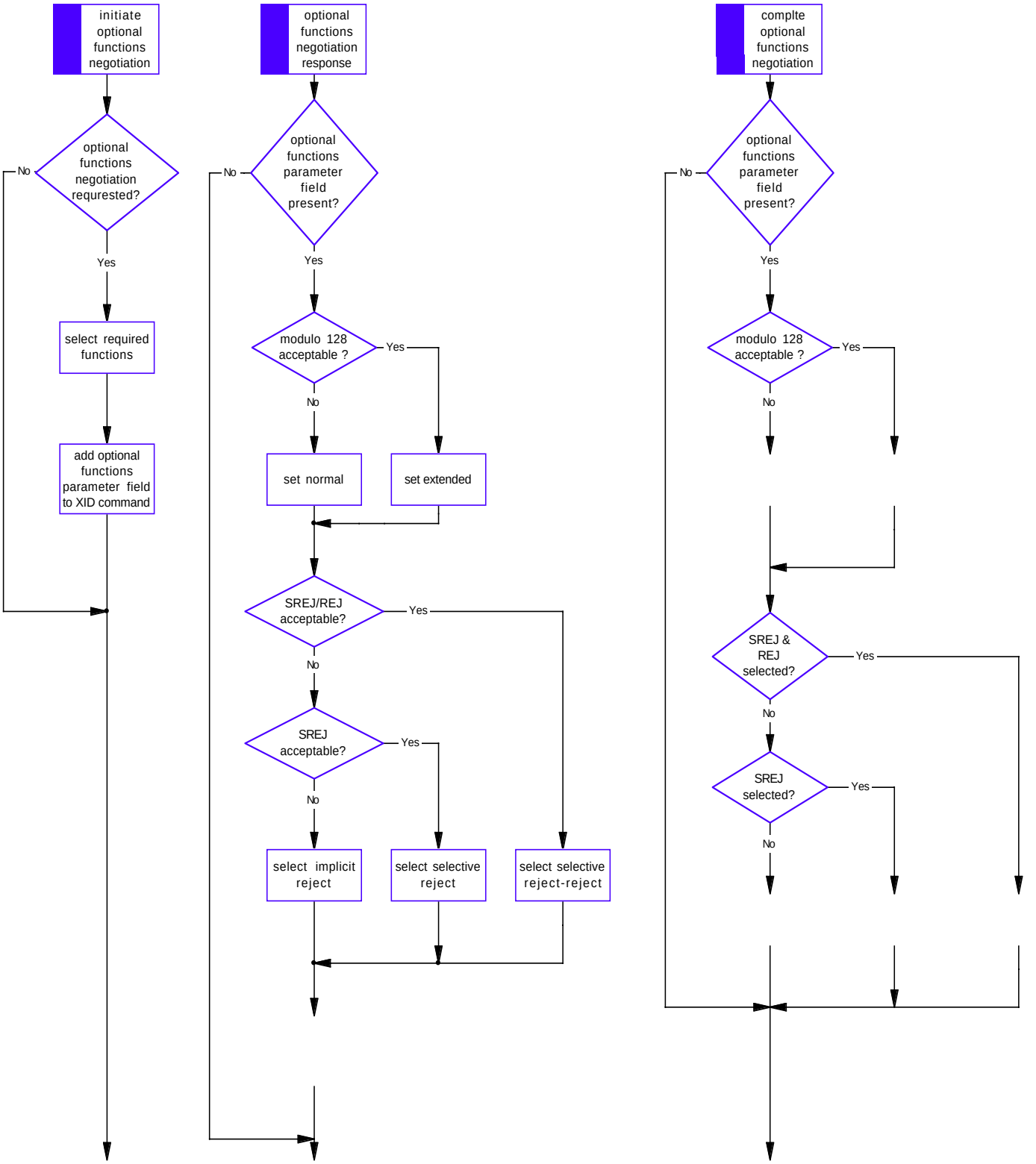
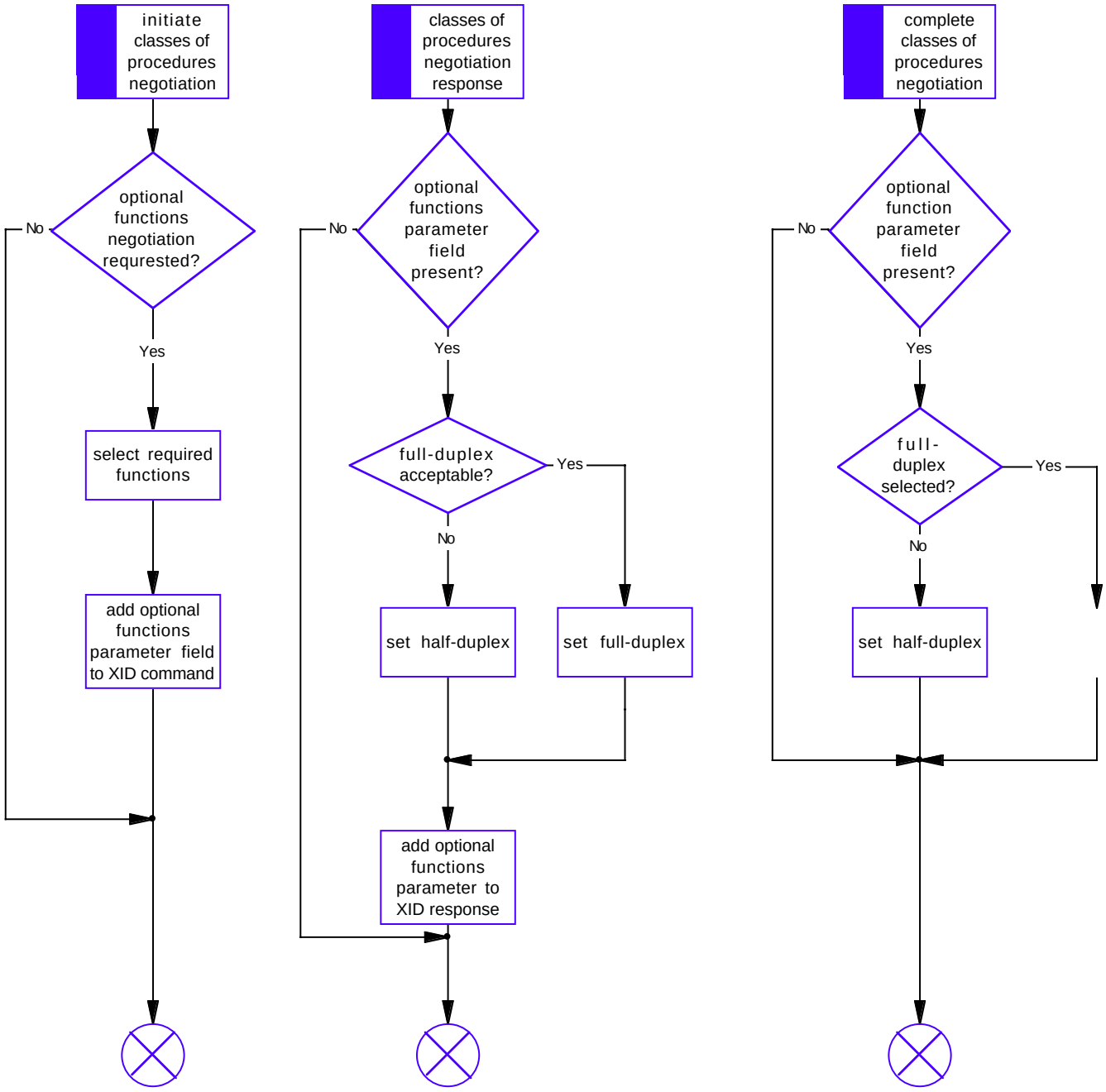


Figure C5.8. Management Data-link classes of procedure negotiation subroutines C-5-1.



Appendix C6

Segmenter/Reassembler

C6.1. Segmenter State Machine

Only the following DL primitives will be candidates for modification by the segmented state machine:

- **DL-DATA Request**. The user employs this primitive to provide information to be transmitted using connection-oriented procedures; i.e., using I frames. The segmenter state machine examines the quantity of data to be transmitted. If the quantity of data to be transmitted is less than or equal to the data link parameter N1, the segmenter state machine passes the primitive through transparently. If the quantity of data to be transmitted exceeds the data link parameter N1, the segmenter chops up the data into segments of length N1-2 octets. Each segment is prepended with a two octet header. (See Figures 3.1 and 3.2.) The segments are then turned over to the Data-link State Machine for transmission, using multiple DL Data Request primitives. All segments are turned over immediately; therefore the Data-link State Machine will transmit them consecutively on the data link.

- **DL-UNIT-DATA Request**. The user employs this primitive to provide information to be transmitted using connectionless procedures; i.e. using UI frames. The segmenter state machine examines the quantity of data to be transmitted. If the quantity of data to be transmitted is less than or equal to the data link parameter N1, the segmenter state machine passes the primitive through transparently. If the quantity of data to be transmitted exceeds the data link parameter N1, the segmenter chops up the data into segments of length N1-2 octets. Each segment is prepended with a two octet header. (See Figures 3.1 and 3.2.) The segments are then turned over to the Data-link State Machine for transmission, using multiple DL Data Request primitives. All segments are turned over immediately; therefore the Data-link State Machine will transmit them consecutively on the data link.

- All Other DL Primitives. All other DL primitives are passed through the segmenter state machine unchanged.

C6.2. Reassembler State Machine

All primitives from the Data-link State Machine are delivered transparently, except the following:

- **DL-DATA Indication**. This primitive is examined by the reassembler state machine. If the accompanying received data begins with an octet other than 0x08, it is assumed it has not been segmented, and is passed up transparently. If the data begins with 0x08, the reassembler state machine allocates buffers and switches to state 1. After various checks for errors, this segment and all remaining segments received in subsequent DL Data Indication primitives are assembled together to recreate the original larger data unit. If a segment is received without the proper PID or out of sequence, the accumulated packets are discarded, buffers are freed, a DL Error Indication is delivered and the state machine returns to state 0. The larger data unit is delivered with one DL Data Indication primitive.

• **DL-UNIT-DATA Indication**. This primitive is examined by the reassembler state machine. If the accompanying received data begins with an octet other than 0x08, it is assumed it has not been segmented, and is passed up transparently. If the data begins with 0x08, the reassembler state machine allocates buffers and switches to state 2. After various checks for errors, this segment and all remaining segments received in subsequent DL Unit Data Indication primitives are assembled together to recreate the original larger data unit. If a segment is received without the proper PID or out of sequence, the accumulated packets are discarded, buffers are freed, a DL Error Indication is delivered and the state machine returns to state 0. The larger data unit is delivered with one DL Unit Data Indication primitive.

• **Timer TR210 Expiry**. This primitive occurs when a segment is not received before timer TR210 times out. When this primitive is received, the accumulated packets are discarded, buffers are freed and the state machine returns to state 0. An DL Error Indication is passed to the higher level.

• **All Other DL Primitives**. All other DL primitives are passed through the reassembler state machine unchanged. If the state machine is in states 1 or 2 when another DL primitive is received, the accumulated packets are discarded, buffers are freed and the state machine returns to state 0. An DL Error Indication is passed to the higher level.

C6.3. Internal Operation of the Machine

The internal states, error codes, and timers are summarized in Figure C2.1.

C6.3.1. Internal Operation of the Segmenter State Machine

The segmenter state machine operation is quite straightforward. Only one state exists for this machine.

C6.3.2. Internal Operation of the Reassembler State Machine

The reassembler state machine resides in the Null state until the start of a segmented data stream is detected. At this point, a check is made to ensure that the first segment received is, in fact, the first segment of the message. This check is performed by examining octet 2, bit 8 of the segment header (see Figure 6.X). If this is not the first segment, then the reassembler state machine assumes that the actual first segment was lost somewhere, and signals an error. All segments will be discarded as they are received.

Assume now that the first segment was received correctly. The reassembler state machine then allocates sufficient storage to receive all the remaining segments; this prevents deadly embrace (resource deadlock) conditions. The reassembler state machine enters either the reassembling data state (if segments are arriving in I frames) or the reassembling unit data state (if segments are arriving in UI frames). A lengthy timer supervises both of these states; its purpose is to protect the reassembly process from hanging if a very long delay happens to occur (e.g., the remote station breaks down and never completes transmission). This timer is TR210: “R” for reassembler; “2” for level 2, the data link level of the OSI interconnect model; and “10” simply to avoid confusion with any other timers in this family of state machines.

Each incoming segment is examined to ensure that it is indeed the next expected segment. If the loss of a segment is detected, the entire accumulation of data is discarded and an error notification is provide to the AX.25 user. No attempt is made by the segmenter and reassembler state machines to recover segmented data units; this is left to the higher level AX.25 user. Rather, the reassembler state machine works to ensure that large data units are completely received and correctly reassembled over the data link. In other words, segmentation error detection is provided, but no segmentation error correction is provided.

The reassembler state machine also insists that, once the transmission of a segmented large data unit is begun, all segments will be transmitted until the complete large data unit has been transferred. No other event is permitted to occur over the data link. This constraint is imposed for two reasons:

a) to ensure that stations with multiple data links minimize the amount of buffer capacity tied up in partially received or transmitted large data units (which in turn reduces connectionless links); and

b) to minimize the delay in transmission of large data units, once the large data unit has reached the top of the queue.

C6.4. Final Observations

As mentioned above, the use of connection-oriented data-link procedures is recommended when segmentation is anticipated on data links with even moderately low collision levels. If connectionless data-link procedures (UI frames) are used to carry segments, the loss of a single UI frame will result in the loss of the entire segmented large data unit; higher level attempts at recovery will significantly increase congestion on the physical channel.

DL Primitives (Received from DLSAP
DL-DATA Request
DL-UNIT-DATA Request NOTE: all other primitives are passed transparently.

DL Primitives (Sent to DLSAP
DL-DATA Indication
DL-UNIT-DATA Indication
DL-ERROR Indication NOTE: all other primitives are passed transparently.

DL Primitives (Received from the Data-Link State Machine)
DL-DATA Indication
DL-UNIT-DATA Indication NOTE: all other primitives are passed transparently.

DL Primitives (Sent to the Data-Link State Machine)
DL-DATA Request
DL-UNIT-DATA Request NOTE: all other primitives are passed transparently.

Segmenter States:
0 — Ready

Reassembler States:

0 — Null
1 — Reassembling Data
2 — Reassembling Unit Data

Queues:

None.

Error Codes:

Y — data too long to segment
Z — reassembly error.

Flags and Parameters:

N — number of segments remaining to be reassembled.

Timers:

TR210 — time limit for receipt of next segment.

Figure C-6.1. Primitives, States, Queues, Flags, Parameters, Errors and Timers.

Figure C-6.2. Segmenter Ready State.

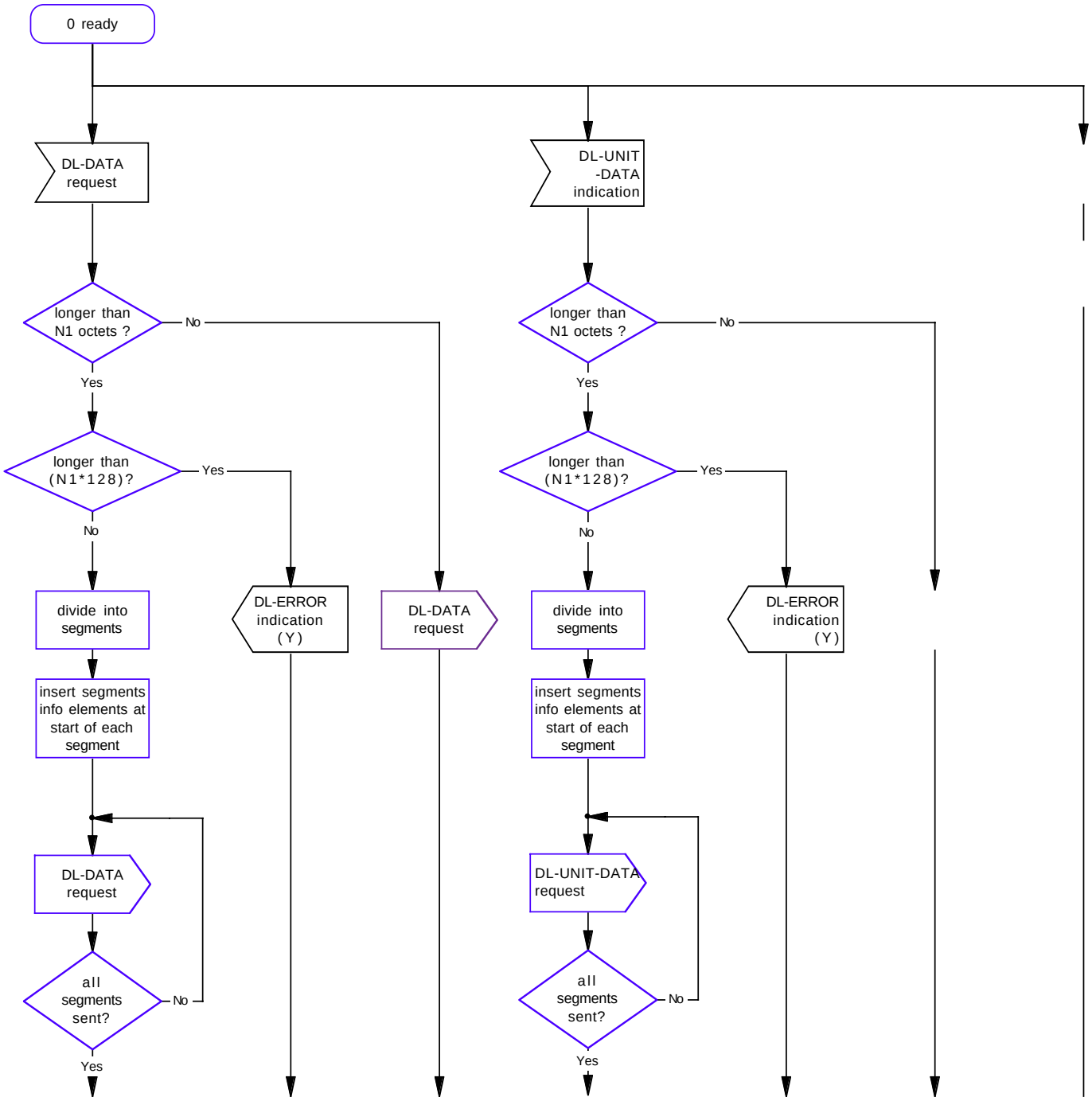


Figure C-6.3. Reassembler Ready State.

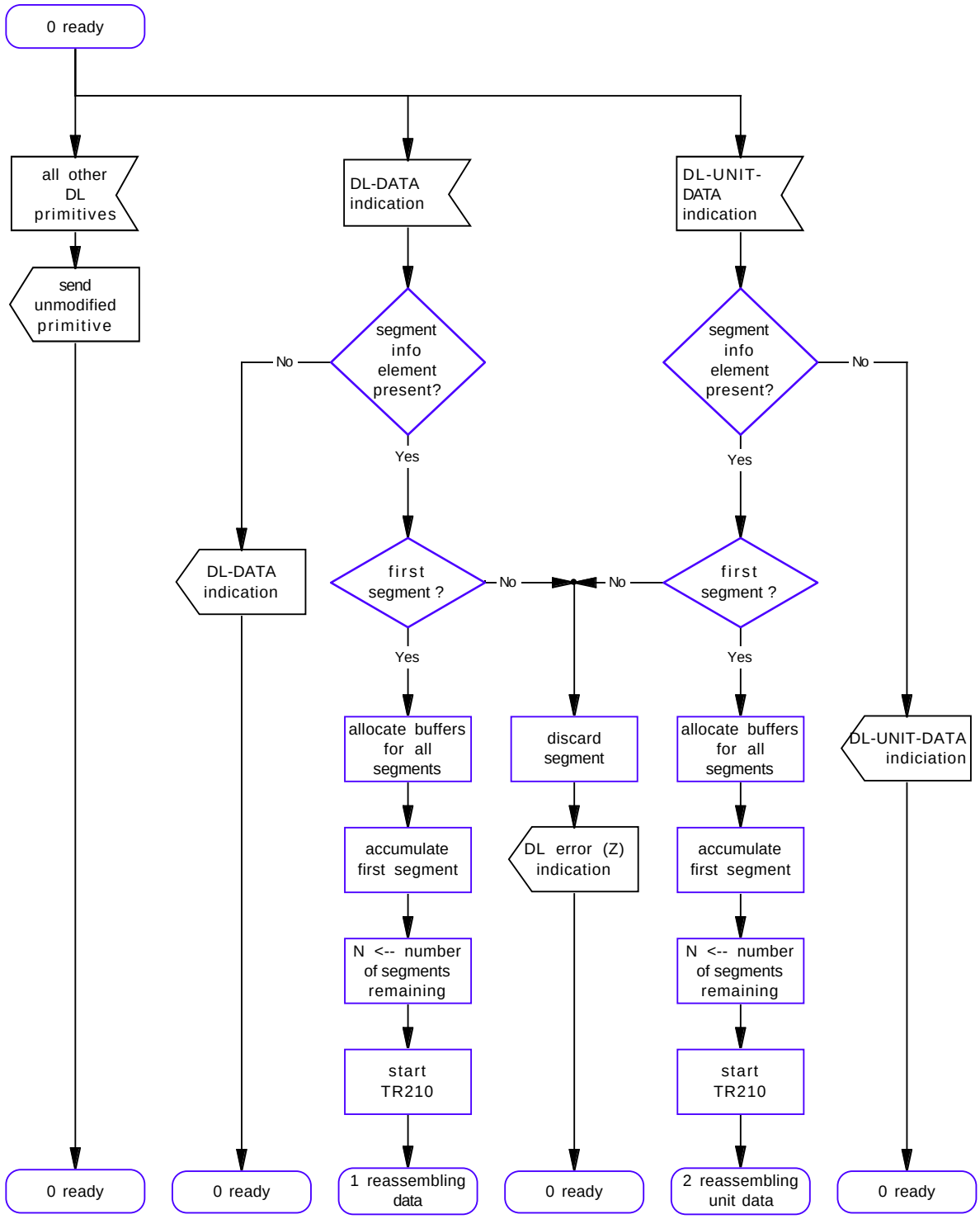


Figure C-6.4. Reassembler Assembling Data State.

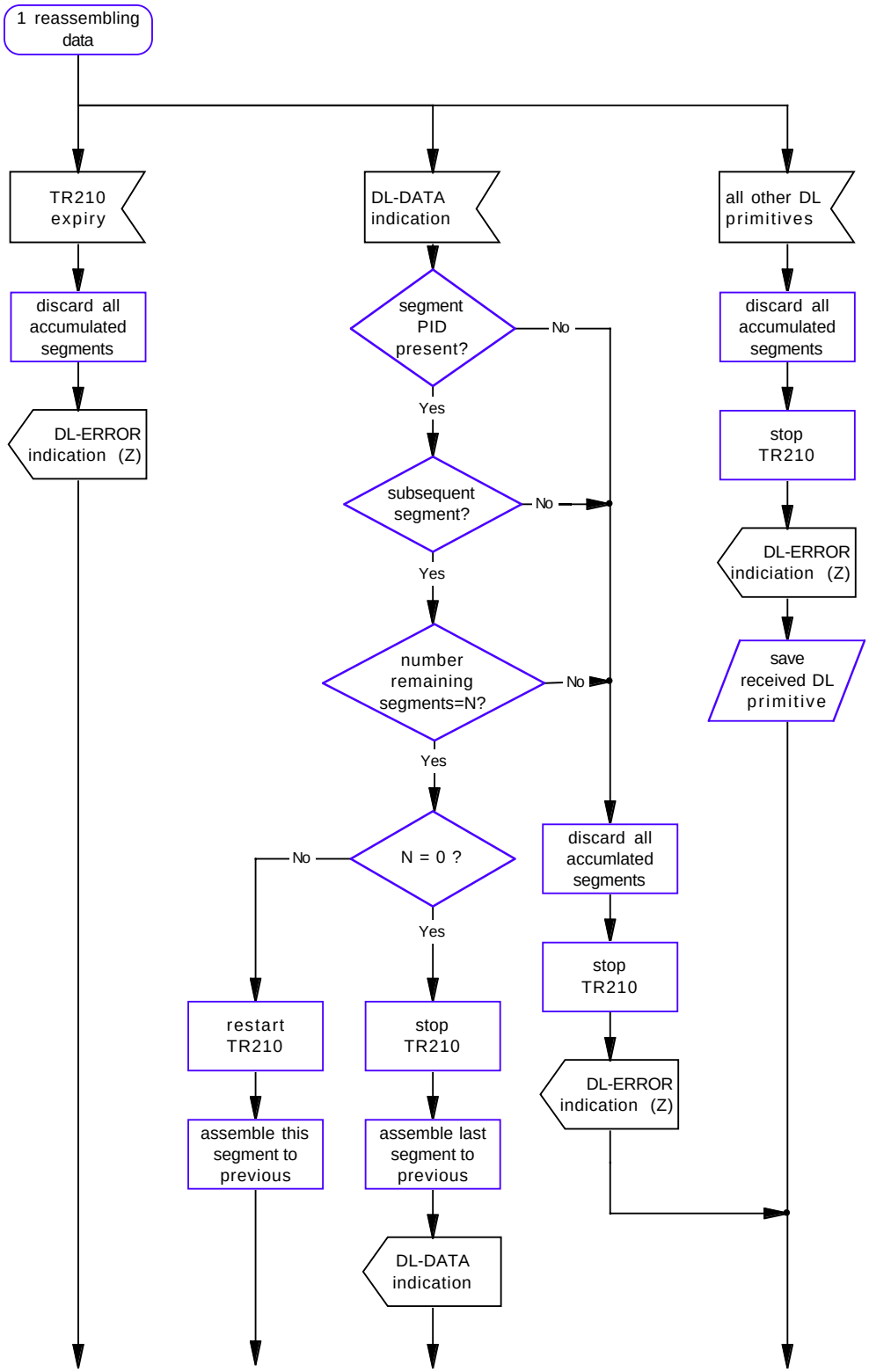
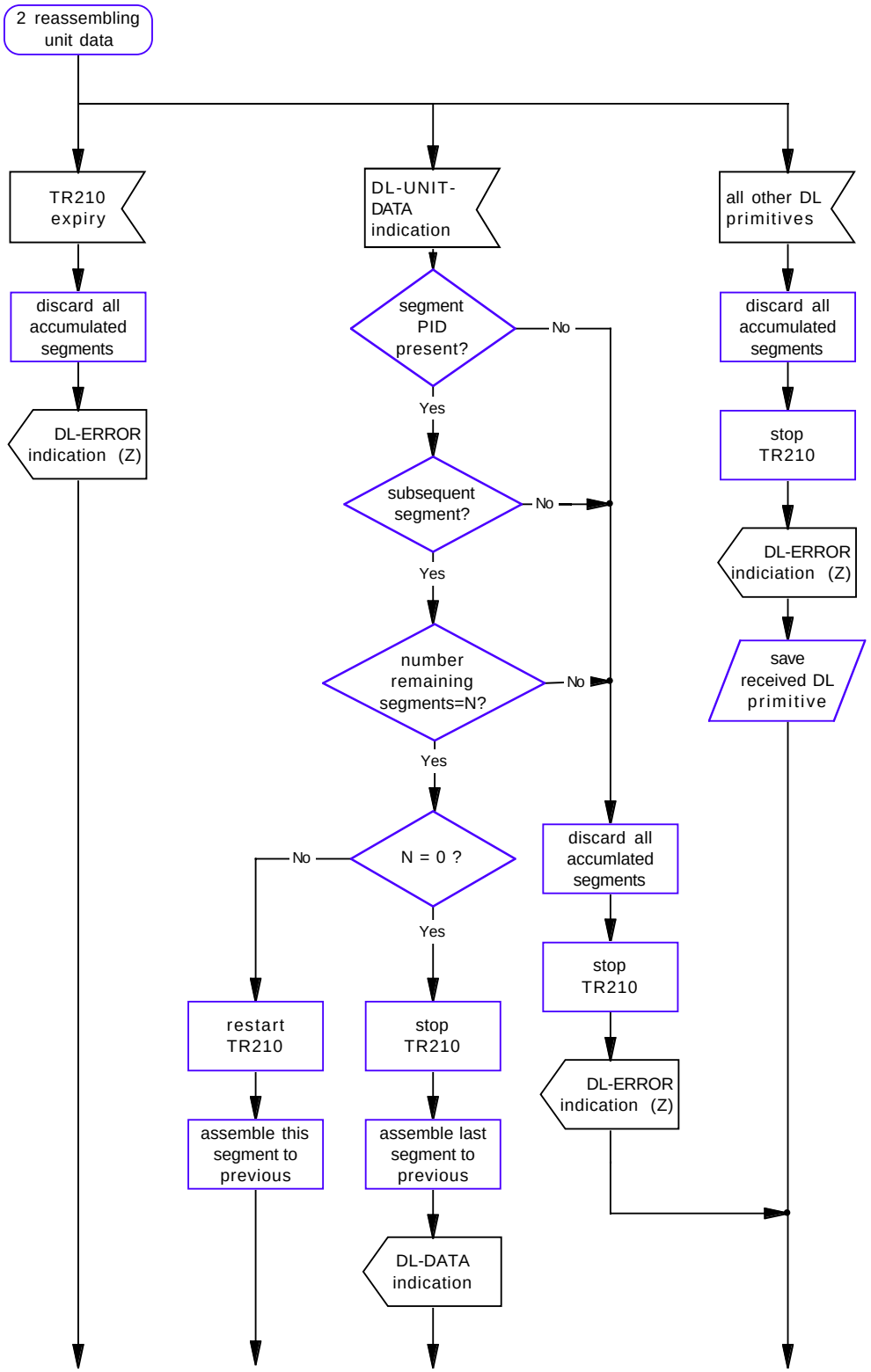


Figure C-6.5. Reassembler Assembling Unit Data State.



Appendix D

Data Link Service Access Point and Primitives

D.1. Model of a Data-Link Connection

A Data Link Service Access Point (DLSAP) is the point at which the data-link layer provides services to Layer 3. It provides a uniform programming interface to access the data-link services. This Appendix specifies this interface but does not specify the upper layer entities. In a basic user TNC, the upper layer entity may consist only of a Layer 7 user application with Layers 3-6 being null.

The following primitives are used to pass commands and receive responses from the DLSAP:

- **(Called Callsign)**. This primitive is used by the Layer 3 entity to request the establishment of an AX.25 connection.

- **DL-CONNECT Indication (Calling Callsign)**. This primitive is used by the Data-link State Machine to indicate an AX.25 connection has been requested.

- **DL-CONNECT Confirm (VOID)**. This primitive is used by the Data-link State Machine to indicate an AX.25 connection has been made.

- **DL-DISCONNECT Request**. This primitive is used by the Layer 3 entity to request the release of an AX.25 connection.

- **DL-DISCONNECT Indication**. This primitive is used by the Data-link State Machine to indicate an AX.25 connection has been released.

- **DL-DISCONNECT Confirm**. This primitive is used by the Data-link State Machine to indicate an AX.25 connection has been released and confirmed.

- **DL-DATA Request**. This primitive is used by the Layer 3 entity to request the transmission of data using connection-oriented protocol. This frame is examined and acted upon by the segmenter, if necessary.

- **DL-DATA Indication**. This primitive is used by the reassembler to indicate reception of Layer 3 data using connection-oriented protocol.

- **DL-UNIT-DATA Request**. This primitive is used by the Layer 3 entity to request the transmission of data using connectionless protocol. This frame is examined and acted upon by the segmenter, if necessary.

- **DL-UNIT-DATA Indication**. This primitive is used by the reassembler to indicate reception of Layer 3 data using connectionless protocol.

• **DL-ERROR Indication**. This primitive is used by the Data-link State Machine to indicate when frames have been received that are inconsistent with this protocol definition. This includes short frames, frames with inconsistent parameter values, etc. The error indications are discussed in the SDL appendices.

• **DL-FLOW-OFF Request**. This primitive is used by the Layer 3 entity to temporarily suspend the flow of incoming information.

• **DL-FLOW-ON Request**. This primitive is used by the Layer 3 entity to resume the flow of incoming information.

• **MDL-NEGOTIATE Request**. This primitive is used by the Layer 3 entity to request the Data-link State Machine to notify/negotiate.

• **MDL-NEGOTIATE Confirm**. This primitive is used by the Management Data-link State Machine to notify the Layer 3 entity that notification/negotiation is complete.

• **MDL-ERROR Indicate**. This primitive is used by the Management Data-link State Machine to notify the Layer 3 entity that notification/negotiation has failed.

D.2. Queue Model Concepts

The queue model represents the operation of the Data-Link Connection (DLC) in the abstract by a pair of queues linking the two DLSAPs. There is one queue for each direction of information flow (see Figure D.1).

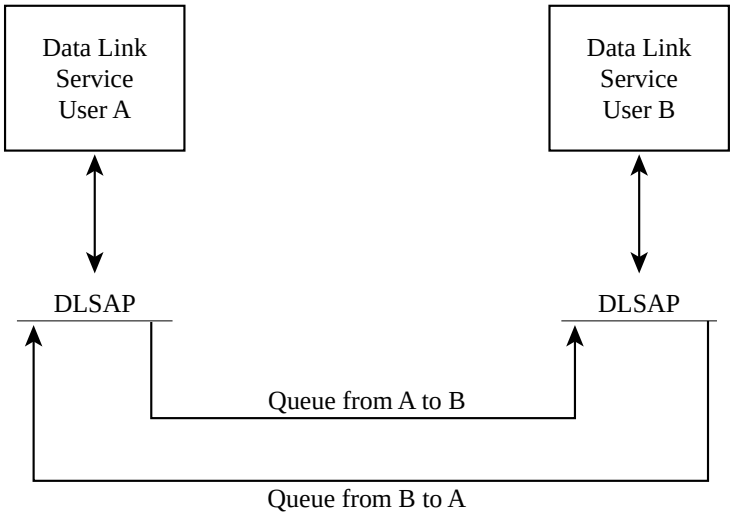


Figure D.1. Queue model of a data-link connection.

The pair of queues is considered to be available for each Data-Link Service (DLS) user. Each queue represents a DLS flow control function in one direction of transfer. The ability of a user to add objects to a queue will be determined by the behavior of the other DLS user in removing objects from the queue and by the state of the queue. Objects are entered or removed from the queue as a result of interactions at the two DLSAPs.

The following objects may be placed in a queue by a DLS user:

- A connect object, representing a DL-CONNECT primitive and its parameters.
- A data object, representing a DL-DATA primitive and its parameters.
- A disconnect object, representing a DL-DISCONNECT primitive and its parameters.

The following object may be placed in a queue by the DLS provider:

- A disconnect object, representing a DL-DISCONNECT primitive and its parameters.

The queues are defined to have the following general properties:

- A queue is empty before a connect object has been entered and can be returned to this state, with loss of its contents, by the DLS provider.
- Objects are entered into a queue by the sending DLS user, subject to control by the DLS provider. Objects may also be entered by the DLS provider.
- Objects are removed from the queue, under the control of the receiving DLS user.
- Objects are normally removed in the same order that they were entered.
- A queue has a limited capacity, but this capacity is not necessarily either fixed or determinable.

D.3. DLC Establishment

A pair of queues is associated with a DLC between two DLSAPs when the DLS provider receives a DL-CONNECT request primitive at one of the DLSAPs, and a connect object is entered into one of the queues. From the standpoint of the DLS users of the DLC, the queues remain associated with the DLC until a disconnect object representing a DL-DISCONNECT primitive is either entered or removed from the queue.

DLS user A, who initiates a DLC establishment by entering a connect object representing a DL-CONNECT request primitive into the queue from DLS user A to DLS user B, is not allowed to enter any other object, other than a disconnect object, into the queue until after the connect object representing the DL-CONNECT confirm primitive has been removed from the DLS user B to DLS user A queue. In the queue from DLS user B to DLS user A, objects can be entered only after DLS user B has entered a connect object representing a DL-CONNECT response primitive.

The properties exhibited by the queues while the DLC exists represent the agreements reached among the DLS users and the DLS provider during this connection establishment procedure.

D.4. Data Transfer

Flow control on the DLC is represented in this queue model by the management of the queue capacity, allowing objects to be added to the queues. The addition of an object may prevent the addition of a further object.

Once objects are in the queue, the DLS provider may manipulate pairs of adjacent objects, resulting in deletion. An object may be deleted if, and only if, the object that follows it is defined to be destructive with respect to the object. If necessary, the last object on the queue will be deleted to allow a destructive object to be entered; thus, a destructive object may always be added to the queue. Disconnect objects are defined to be destructive with respect to all objects.

The relationship between objects that may be manipulated in the above fashion are summarized in Figure D.2.

Preceding Object \ Following Object	Connect	Data	Sync	Disconnect
Connect	N/A	—	N/A	DES
Data	N/A	—	N/A	DES
Sync	N/A	—	N/A	DES
Disconnect	N/A	N/A	N/A	DES

Where:
N/A Not a valid state of the queue.
— Not to be destructive nor to be able to advance ahead.
DES To be destructive to the preceding object.

Figure D.2. Relationships between queue model objects.

Whether the DLS provider performs actions resulting in deletion or not will depend upon the behavior of the DLC users. In general, if a DLS user does not remove objects from a queue, the DLS provider shall, after some unspecified period of time, perform all the permitted deletions.

D.5. DLC Release

The insertion into a queue of a disconnect object, which may occur at any time, represents the initiation of the DLC release procedure. The release procedure may be destructive with respect to other objects in the two queues and eventually results in the emptying of the queues and the dissociation of the queues with the DLC.

The insertion of a disconnect object may also represent the rejection of a DLC establishment attempt or the failure to complete DLC establishment. In such cases, if a connect object representing a DL-CONNECT request primitive is deleted by a disconnect object, then the disconnect object is also deleted. The disconnect object representing the DL-CONNECT response.

D.6. Relationship of Primitives at the Two DLC Endpoints

A primitive issued at one DLC endpoint will, in general, have consequences at the other DLC endpoint. The relationship of primitives of each type at one DLC endpoint to primitives at the other DLC endpoint are defined in the appropriate subclauses discussed in Section 5.

A simple connection oriented transmission of data would be handled by the following primitives at the DLSAPs as shown in Figure D.3. Notice that the MDL primitives do not generate an Indicate primitive nor require a Response primitive from the Layer 3 entity in the station B. The MDL entities in the stations A and B work on a peer-to-peer relationship. The other primitives work in groups of four with the Request from the station A causing an Indicate in the station B, and a Response in station B causing a Confirm in the station A.

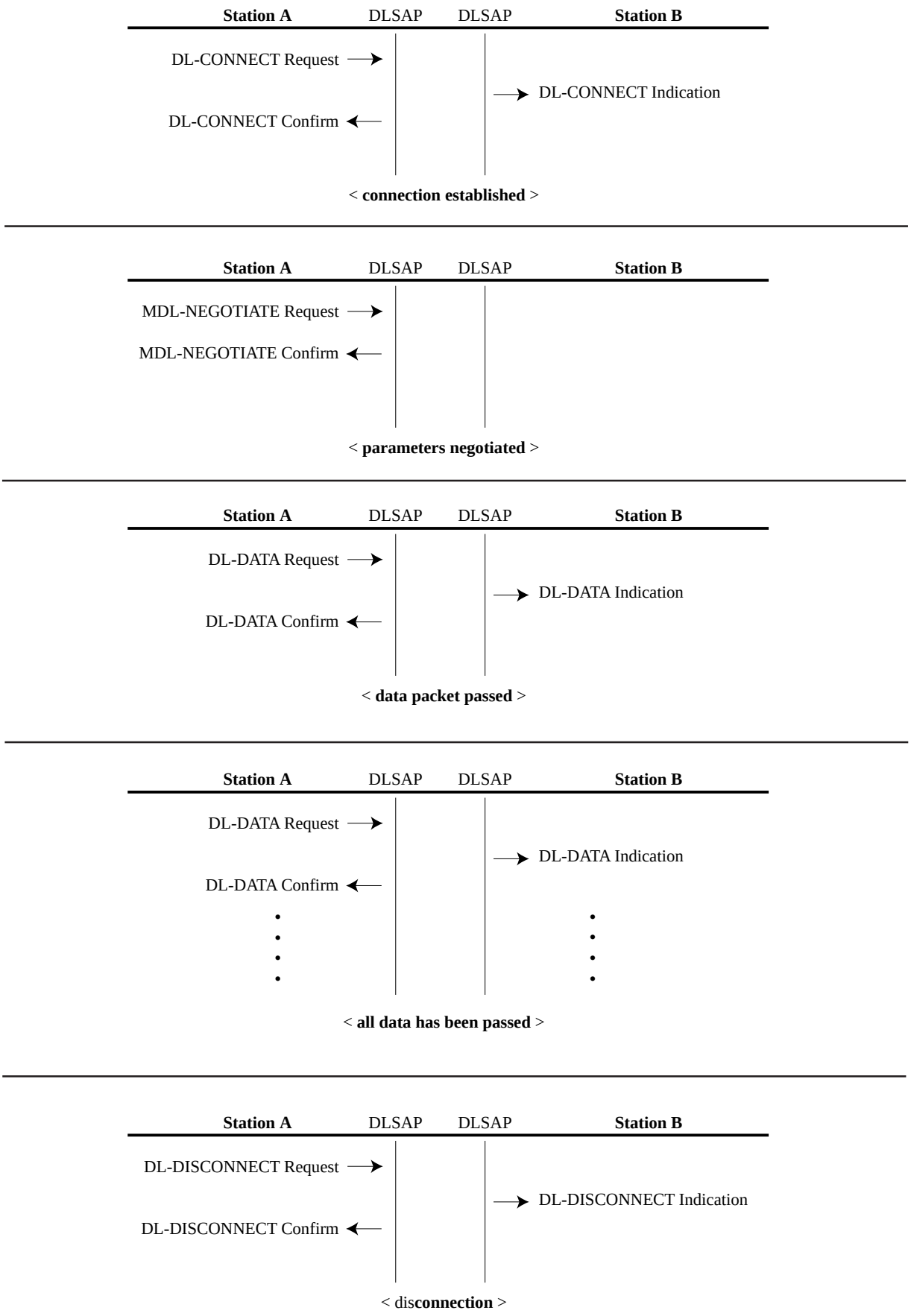


Figure D.3. Example of a connection-oriented data exchange.