ETSI/TC SMG
Released by : ETSI/PT 12
Release date: February 1992

RELEASE NOTE

Recommendation GSM 04.22

Radio Link Protocol for Data and Telematic Services
on the MS-BSS Interface

Previously distributed version : 3.7.0 (Updated Release 1/90)
New Released version February 92 : 3.7.0 (Release 92, Phase 1)

1. Reason for changes

No changes since the previously distributed version.

INTERIM
EUROPEAN
TELECOMMUNICATION
STANDARD

2nd FINAL DRAFT
pr I-ETS 300 026
February 1992

Source: ETSI TC-SMG

Reference: GSM 04.22

UDC: 621.396.21

Key words: European Digital Cellular Telecommunications System, Global System for Mobile Communications (GSM)

European digital cellular telecommunications system (phase 1); Radio Link Protocol (RLP) for data and telematic services on the Mobile Station - Base Station System (MS-BSS) interface and the Base Station System - Mobile Services Switching Centre (BSS-MSC) interface

ETSI

European Telecommunications Standards Institute
ETSI Secretariat: B.P.152 . F - 06561 Valbonne Cedex . France
TP. + 33 92 94 42 00 TF. + 33 93 65 47 16 Tx. 47 00 40 F

Whilst every care has been taken in the preparation and publication of this document, errors in content, typographical or otherwise, may occur. If you have comments concerning its accuracy, please write to "ETSI Standards Management Dept." at the address shown on the title page.

Foreword

Standard (I-ETS) has been produced by the Special Mobile Group (SMG), a Technical Committee of the European Telecommunications Standards Institute (ETSI).

The final drafts dealing with the GSM system were adopted by vote in May 1991 but were not published. This was because amendments, agreed by ETSI TC-SMG at subsequent meetings, were made to some of the drafts. However, other drafts have not been amended since the first vote.

This updated draft is now considered to be stable enough for submission to second vote.

This I-ETS describes the Radio Link Protocol (RLP) used for data transmission and telematic services within the European digital cellular telecommunications system (phase 1).

Reference is made within this I-ETS to the following technical specifications (NOTE 1):

GSM 04.21	Rate adaptation on the Mobile Station - Base Station System (MS-BSS) interface.
GSM 07.02	Terminal adaptation functions for services using asynchronous bearer capabilities.
GSM 07.03	Terminal adaptation functions for services using synchronous bearer capabilities.
GSM 09.04	Interworking between the Public Land Mobile Network (PLMN) and the Circuit Switched Public Data Networks (CSPDN).
GSM 09.05	Interworking between a Public Land Mobile Network (PLMN) and the Circuit Switched Public Data Networks (CSPDN) for PAD access.
GSM 09.06	Interworking between a Public Land Mobile Network (PLMN) and a PSPDN/ISDN for support of packet switched data transmission services.
GSM 09.07	General requirements on interworking between the PLMN and the ISDN or PSTN.

The above specifications are normative.

NOTE 1: ETSI has constituted stable and consistent documents which give technical specifications for the implementation of the European digital cellular telecommunications system. Historically, these documents have been identified as "GSM recommendations".

Some of these recommendations may subsequently become Interim European Telecommunication Standards (I-ETSs) or European Telecommunication Standards (ETSs), whilst the others will be renamed ETSI-GSM Technical Specifications. These ETSI-GSM Technical Specifications are, for editorial reasons, still referred to as GSM recommendations in some current GSM documents.

The numbering and version control system used for ETSI-GSM Technical Specifications is the same as that used for GSM recommendations.

NOTE 2: Items in this draft indicated as not complete, or requiring further study or work, are not required for the Phase 1 implementation of the European digital cellular telecommunications system.

ETSI/GSM

Released by: ETSI PT12

Date: February 1992

Recommendation: GSM 04.22

Title: RADIO LINK PROTOCOL (RLP) FOR DATA AND TELEMATIC SERVICES ON THE MS/BSS INTERFACE AND THE BSS/MSC INTERFACE

List of contents:

Original Language: English

Number of pages: 59

1. SCOPE

This recommendation specifies the Radio Link Protocol (RLP) for data transmission over the GSM PLMN. RLP covers the Layer 2 functionality of the ISO OSI Reference Model (IS 7498). It is based on ideas contained in IS 3309, IS 4335 and IS 7809 (HDLC of ISO) as well as X.25 and Q.92x (LAP-B and LAP-D of CCITT, resp.) RLP has been tailored to the special needs of digital radio transmission. RLP provides to its users the OSI Data Link Service (IS 8886).

RLP is intended for use with non-transparent data-transfer. Protocol conversion may be provided for a variety of protocol configurations. Those foreseen immediately are:

- Character-mode protocols using start-stop transmission (IA5)

- X.25 LAP-B.

For reasons of better presentation, material about protocol conversion has been placed within those recommendations concerned with the relevant Terminal Adaptors, i.e. Rec.07.02 for the asynchronous case and Rec.07.03 for the synchronous case. Care must be taken that that material also applies to Interworking Functions; see Rec. 09.04 - 09.07.

2. DEFINITIONS

command: An instruction represented in the RLP header, causing the receiving RLP entity to execute a specific function.

frame check sequence: A field of redundant information based on a cyclic code, used for error detection.

I + S frame: An RLP frame that is used for user information transfer, carrying supervisory information piggyback.

improper frame: An RLP frame having an FCS error or having a header the contents of which is inconsistent with this recommendation.

non-transparent: In PLMN data transmission, a configuration where at layer 2, protocol information of the fixed network is mapped on RLP elements, and vice versa.

piggybacking: Means by which one and the same frame can carry both user information and RLP related supervisory information.

response: A reply represented in the RLP-header, by which the sending RLP entity reports back about its status.

RLP frame: A sequence of contiguous bits, representing an RLP procedural element.

RLP header: That part of an RLP frame that encodes either a command or a response, located at the beginning of the RLP frame.

S frame: An RLP frame that contains supervisory information in the absence of user information.

transparent: In PLMN data transmission, a configuration where at layer 2 (and also at the layers above) no protocol conversion takes place.

U frame: An RLP frame that contains unnumbered protocol control information.

3. INTRODUCTION

RLP makes use of an underlying FEC (Forward Error Correction) mechanism. For RLP to perform adequatly it is assumed that the basic radio channel together with FEC provides for a block error rate of less than 10 %, where a block consists of 240 bits.

RLP frames are sent in strict alignment with the radio transmission. (For details, see Rec. 04.21). RLP frames are of fixed size of 240 bits. Whenever such a 240 bit frame is to be sent, the RLP entity has to provide the necessary protocol information to be contained in it. Provision is made for discontinuous transmission (DTX).

RLP spans from the mobile station (MS) to the interworking function (IWF), located at the nearest mobile switching centre (MSC), or beyond. Depending on the exact location of the IWF, handover of the MS may result in link-reset or even total loss of the connection.

In the terminology of HDLC, RLP is used in a balanced configuration, employing asynchronous operation, i.e. either station has the right to set-up, reset, or disconnect a link at any time. Procedural means are provided for to deal with contentious situations, should they ever occur.

RLP is full-duplex in the sense that it allows for information to be transferred in both directions simultaneously.

4. FRAME STRUCTURE

4.1. Basic frame structure

An RLP-frame has a fixed length of 240 bits consisting of a header (16 bits), an information field (200 bits), and an FCS (frame check sequence) field (24 bits). As a benefit of using strict alignment with underlying radio transmission there is no need for frame delimiters (like flags etc.) in RLP. In consequence, there is no "bit-stuffing" necessary in order to achieve code transparency. Frames cannot be aborted while being transmitted.

Header	Information	FCS
16 bit	200 bit	24 bit

Figure 1: Frame structure

4.2. RLP header

An RLP-header carries one of three types of control information, the first being unnumbered protocol control information (U frames), the second being supervisory information (S frames), the third being user information carrying supervisory information piggybacked (I + S frames).

4.3. Order of transmission

The header, as defined in section 5.2, shall be transmitted from left to right. The FCS shall be transmitted commencing with the highest order term. The order of bit transmission for the information field is from left to right.

4.4. Frame check sequence

The FCS shall be the ones complement of the modulo 2 sum of

a) the remainder of

$x^{216} (x^{23} + x^{22} + x^{21} + x^{20} + x^{19} + x^{18} + x^{17} + x^{16} + x^{15} + x^{14} + x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)$

divided modulo 2 by the generator polynomial

$x^{24} + x^{23} + x^{21} + x^{20} + x^{19} + x^{17} + x^{16} + x^{15} + x^{13} + x^8 + x^7 + x^5 + x^4 + x^2 + 1$

and

b) the remainder of the division modulo 2 by the generator polynomial

$x^{24} + x^{23} + x^{21} + x^{20} + x^{19} + x^{17} + x^{16} + x^{15} + x^{13} + x^8 + x^7 + x^5 + x^4 + x^2 + 1$

of the product of x^{24} by the content of the frame, excluding the FCS field. (The first bit transmitted corresponds to the highest order term.)

Implementation note: As a typical implementation, at the transmitter, the initial content of the register of the device computing the remainder of the division is preset to all ones and is then modified by division by the generator polynomial (as described above) of the header and information field; the ones complement of the resulting remainder is transmitted as the 24 bit FCS sequence.

At the receiver, the initial content of the register of the device computing the remainder is preset to all ones. The final remainder after multiplication by x^{24} and then division (modulo 2) by the generator polynomial

$x^{24} + x^{23} + x^{21} + x^{20} + x^{19} + x^{17} + x^{16} + x^{15} + x^{13} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + 1$

of the serial incoming protected bits and the FCS will be

0 1 1 0 1 1 0 1 1 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 (x^{23} to x^{0}, resp.)

in the absence of transmission errors.

5. ELEMENTS AND PROCEDURE

5.1. Modes

An RLP entity can be in one of two modes:

- Asynchronous Balanced Mode (ABM)
- Asynchronous Disconnected Mode (ADM)

5.1.1. Asynchronous Balanced Mode (ABM)

In ABM, which is the data link operational mode, either RLP entity may send commands at any time and may initiate response frame transmission without receiving explicit permission to do so from the other RLP-station. In ABM, frames shall be used for information field transfer and/or to indicate status changes in the RLP-station.

5.1.2. Asynchronous Disconnected Mode (ADM)

In ADM, which is the data-link non-operational mode, the RLP entity shall be logically disconnected from the data link and shall, therefore, neither transmit nor accept numbered information frames.

The RLP entity shall, however, be permitted to transmit and accept UI, TEST and XID frames. Either RLP entity can issue an SABM command at any time, in order to terminate the ADM state. In that case, entrance of the ABM state will be indicated by a UA response from the opposite station. If the opposite station is not able to enter ABM, it will indicate this by a DM response. All commands other than those mentioned above and any unsolicited response will be ignored in ADM under all circumstances.

5.2. Header and parameters

The formats defined for the header are listed in Figure 2 below.

5.2.1. Generally used bits

NOTES:
C/R = COMMAND/RESPONSE BIT
X = DON'T CARES
N(S) : Bit 4 low order bit
N(R) : Bit 11 low order bit

U	C/R	X	X	1	1	1	1	1	1	P/F	M1	M2	M3	M4	M5	X
S	C/R	S1	S2	0	1	1	1	1	1	P/F	N(R)					
I+S	C/R	S1	S2	N(S)						P/F	N(R)					
bit	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

S_1	S_2	
0	0	RR
0	1	REJ
1	0	RNR
1	1	SREJ

$M_1M_2M_3M_4M_5$	
1 1 1 0 0	SABM
0 0 1 1 0	UA
0 0 0 1 0	DISC
1 1 0 0 0	DM
1 1 1 1 0	NULL
0 0 0 0 0	UI
1 1 1 0 1	XID
0 0 1 1 1	TEST

Figure 2: Header format

5.2.1.1. Command/response bit, C/R

The C/R-bit is used to indicate whether the frame is a command or response frame and whether the P/F-bit is to be interpreted as a poll or final bit, resp. For commands, the C/R bit shall be set to "1", for responses it shall be set to "0".

5.2.1.2. Poll/Final bit, P/F

The P/F-bit is used to mark a special instance of command/response exchange. With a command, it is called the P-bit, with a response, it is called the F-bit. In any one direction, only one P/F-bit exchange may be outstanding at any time. A response with the F-bit set to "1" shall always reflect the latest receive status of the RLP entity.

A P/F-bit exchange always starts with a command frame with the P-bit set to "1", which shall be answered by a response frame with the F-bit set to "1" at the earliest response opportunity.

No unsolicited F-bit = "1" is allowed. Such a frame shall be considered "improper" (see 5.3.1). In ABM, the use of the P/F-bit is only allowed for checkpoint-recovery (see 5.3.3).

5.2.2. Unnumbered frames, U

5.2.2.1. Set asynchronous balanced mode SABM (11100)

The SABM encoding is used as a command only. It is always used with the P-bit set to 1.

The SABM command is used either to initiate a link for numbered information transfer, i.e. to go from ADM to ABM, or to reset a link already established for numbered information transfer. With an SABM command, no information transfer is allowed.

When issuing an SABM, the RLP entity has set to zero its internal variables for sending and receiving numbered information. The other RLP entity, on receiving an SABM command, will either confirm it by setting to zero its internal variables for sending and receiving numbered information and then issuing an UA (unnumbered acknowledgement) response or reject it by sending a DM (disconnected mode) response. In the former case, both entities have entered ABM and numbered information transfer may commence. In the latter case, both entities are in ADM.

When an SABM command is issued, a loss of information may occur. Appropriate action is in the responsibility of the layers above.

5.2.2.2. Unnumbered Acknowledge, UA (00110)

The UA encoding is used as a response only. It is used to positively acknowledge an SABM or DISC command. With the UA response, no information transfer is allowed.

5.2.2.3. Disconnect, DISC (00010)

The DISC encoding is used as a command only. It is used to disestablish a link, previously established for numbered information transfer, i.e. to terminate ABM and go into ADM. With the DISC command, no information transfer is allowed.

The other RLP-entity shall answer with a UA response before actioning the DISC command. When a DISC command is actioned, loss of information may occur. It is the responsiblity of the layers above, to provide for a "graceful" disconnect.

5.2.2.4. Disconnected Mode, DM (11000)

The DM encoding is used as a response only. It is used by RLP entity to report that it is in ADM and, as an answer to SABM, that it is unable to action a mode setting command. With the DM response, no information transfer is allowed.

5.2.2.5. Unnumbered Information, UI (00000)

The information field is to be interpreted as unnumbered information. Unnumbered information (UI) frames can be sent in both ADM and ABM. There is no acknowledgement of receipt of UI-frames within RLP.

5.2.2.6. Exchange Identification, XID (11101)

The information field is to be interpreted as exchange identification. This frame is used to negotiate and renegotiate parameters of RLP and layer 2 Relay function. XID frames can be sent in both ADM and ABM.

The negotiation procedure is one step i.e. one side will start the process by sending an XID command, offering a certain set of parameters from the applicable parameter repetoire (see table 1) the sending entity wants to negotiate proposing values within the allowed range. In return, the other side will send an XID response, either confirming these parameter values by returning the requested values, or offering higher or lower ones in their place (see Table 1 for sense of negotiation), except when the indicated RLP version is a lower one where a limited set of those parameters presented in the XID command may be answered according to the negotiated version. This normally will end the negotiation process. XID frames are always used with the P/F-bit set to "1".

Without any prior XID exchange, default values will apply (see section 5.4).

In the case of a collision of XID commands, all XID commands shall be ignored. The Mobile Station shall restart the parameter negotiation on expiry of T1, while the Interworking Function shall do so on expiry of twice the value of T1. An unsuccesful XID exchange shall be repeated on expiry of T1. After N2 times of unsuccessful repetition, the link shall be disconnected.

In the following, a list of parameters is given which constitute the parameter repetoire for this RLP version. In addition, the format of the XID information field is given.

Parameter Name	Type	Length (octets)	Range	Units	Sense of Negotiation
RLP version N° *	1	1	0..255	./.	down
IWF to MS window size	2	1	0..61	./.	down
MS to IWF window size	3	1	0..61	./.	down
Acknowledgement timer (T1)	4	1	0..255	10 ms	up
Retransmission attempts (N2)	5	1	0..255	./.	up
Reply delay (T2)**	6	1	0..255	10 ms	up

Table 1: XID parameters

* Note 1: The current version of RLP is "0". This is also the default value for the version N°. It is assumed that future versions of RLP will be backwards-compatible with former ones.

** Note 2: In case of negotiation of this parameter it may be necessary to negotiate also the "Acknowledgement timer" (T1).

The type and length are encoded within one octet, the type field occupying bits 8 to 5 and the length field occupying bits 4 to 1; 1 resp. 5 being the least significant bit. The least significant bit shall always be transmitted first.

A parameter item consists of the type/length-octet followed by the value of that parameter, where the length-indicator gives the number of octets the value actually occupies. Such parameter items may be arranged in arbitrary order, but must begin in the first octet of the XID-information field and follow on contiguously. The parameter list is delimited by parameter type zero.

5.2.2.7. Test, TEST (00111)

The information field of that frame is to be interpreted as test information. Test frames can be sent in both ADM and ABM. A test sequence is always initiated by sending a TEST command in one direction and completed by sending a TEST response in the other direction.

5.2.2.8. "Null" information, NULL (11110)

In ADM, null-frames shall be sent each time there is a send opportunity but no UI, TEST or XID frame is awaiting transmission.

The information field is to be interpreted as null information i.e. the information field is not used and its contents may be arbitrary.

5.2.3. Supervisory frames, S, and numbered information transfer and supervisory frames combined, I+S

In ABM, there are cases where there is no user information pending transmission. In consequence, supervisory (S) frames alone must be conveyed. In such cases, the information field is to be interpreted as null information, i.e. the information field is not used and may be of arbitrary contents.

For reasons of optimization in the special situation of digital radio transmission, numbered information transfer frames carry also supervisory type information ("piggy-backing"). Numbered information can be exchanged only in ABM.

Note: The extent to which piggy-backing is used by the sending RLP entity is optional. An RLP entity receiving any of allowed piggy-backed formats, however, shall take the appropriate actions. Implementors should be aware that not using the full capability of piggy-backing could, in certain circumstances, result in a less than optimal performance.

5.2.3.1. Send Sequence number, N(S)

The sequence number contains the number of the I frame. As far as N(S) is concerned, modulus 62 arithmetic is applied for frame numbering, thus allowing for a maximum window size of 61. On mutual agreement between the communicating parties, a smaller window size may be established. With the exception of SREJ conditions, information frames are transmitted in numerical order of their N(S). Normal information transfer is halted, when the number of outstanding, unacknowledged frames is equal to the currently established window size.

5.2.3.2. Receive sequence number, N(R)

The N(R) field is used in ABM to designate the next information frame to be sent by the other RLP entity and to confirm that all frames up to and including N(R) - 1 have been received properly. As an exception to this, in the case of SREJ (selective reject), N(R) designates the information frame that is selectively rejected and thus requested for retransmission. In this case, no previously received frames are confirmed, unless a SREJ response is sent with the F-bit set to "1".

N(R) provides for a modulus of 62, thereby allowing for a maximum window size of 61, i.e. a maximum of 61 information frames may be outstanding at any time.

5.2.3.3. Receive ready, RR (00)

The RR encoding can be used either as command or response. In ABM, it is used by an RLP entity to confirm all information frames up to and including N(R)-1. In doing so, the RLP-station allows the other station to transmit up to k additional information frames, counting from N(R) onwards. The issue of an RR command/response clears any previous busy condition in that direction.

5.2.3.4. Reject, REJ (01)

The REJ encoding can be used either as command or response. It is used by an RLP entity to indicate that in numbered information transfer one or more out-of sequence frames have been received. Frames up to and including N(R)-1 have been received correctly, frames N(R) and following are requested to be retransmitted. Following retransmission of those frames, further frames awaiting initial transmission may be sent. With respect to each direction of transmission, only one REJ condition may exist at any given time. It is further not allowed to establish a REJ condition, while there still are uncleared SREJ conditions.

A REJ condition is cleared on

- receipt of the frame numbered N(R)
- on time-out
- or on reset (SABM)

An REJ shall be sent at the earliest opportunity. On time-out, REJ frames shall not be repeated. An RLP-entity receiving an REJ frame with the same N(R), which has already been the starting frame of a retransmission sequence due to P/F-bit checkpointing, shall inhibit the retransmission due to that particular REJ frame.

5.2.3.5. Receive not ready, RNR (10)

The RNR encoding can be used either as command or response. It is used by an RLP entity to indicate that it is temporarily not ready to receive numbered information frames. In that case, the RLP entity is said to be in the busy condition. All frames up to and including N(R)-1 shall be considered acknowledged. Subsequent frames, if any, shall not be considered confirmed. The acceptance status of those is a matter of further status exchange.

5.2.3.6. Selective reject, SREJ (11)

The SREJ encoding can be used either as command or response. The SREJ command/response is used to request retransmission of a single frame, thus, under certain circumstances, providing for more efficient error recovery than by REJ. No acknowledgement of received I frames is indicated by an SREJ frame, thus allowing an RLP entity to transmit one or more SREJ frames with a different N(R) before earlier SREJ conditions have been cleared.

An SREJ condition shall be cleared

- on receipt of an information frame with N(S) equal N(R) of the SREJ
- on time out
- on reset (SABM)

No SREJ shall be issued during a pending REJ condition. For each frame, only one SREJ condition may exist at any time.

SREJ frames shall be sent at the earliest possibility. On time-out, SREJ frames shall not be repeated.

Note: Sending SREJ commands/responses is not mandatory.

5.3. Error Recovery

5.3.1. Improper frames

Frames containing an FCS error or having a control field the contents of which is not implemented or inconsistent with those defined in this recommendation are called improper frames. Improper frames shall be ignored, i.e. the receiving RLP station shall not make any use of their contents.

5.3.2. N(S) sequence error

In numbered information transfer, any information frame with an N(S) out of the normal sequence shall lead to an N(S) sequence error condition, unless that frame is requested for retransmission by an SREJ, sent at an earlier time. There are three mechanisms to deal with N(S) sequence errors.

- REJ recovery
- SREJ recovery
- P/F-bit recovery (checkpointing),

the first two being the responsibility of the receiving station, the last being the responsibility of the sending station. There are no strict rules as to whether REJ or SREJ recovery shall be applied, however, if a station decides to initiate REJ or SREJ recovery, it shall do so at the earliest opportunity. The information part of out-of sequence frames shall be discarded, unless the receiving station intends to initiate SREJ recovery.

5.3.3. Time-out and checkpointing

All frames requiring a response or acknowledgement shall be guarded by time-out. (Timer value for further study.) In detail, those frames are those containing

- SABM
- DISC
- REJ
- SREJ

- numbered information
- any frame with the P-bit set to "one" in ABM, i.e. checkpointing

5.3.3.1. Treatment of errors during link establishment, link reset and link disconnect

An SABM, which is not answered by either UA or DM within the timer period, shall be repeated up to N2 times (Action on finally unsuccessful SABM is for further study, pending decisions on station management.)

A DISC, which is not answered by UA within the timer period, shall be repeated up to N2 times. If the DISC is finally unanswered, the RLP station will go into ADM in any case. For this reason, it is the responsibility of the management of any RLP entity to put the RLP entity into ADM, should there be an indication of a permanent outage, i.e. a loss of connectivity longer than N2 times the timer value.

5.3.3.2. Treatment of errors during numbered information transfer

The last frame of a sequence of numbered information frames shall also be guarded by time-out. If neither a positive acknowledgement nor a REJ is received, the RLP entity will start checkpoint recovery, i.e. the station will send a frame with the P-bit set to "1", requesting the latest status information from the other entity, indicated by the F-bit set to "1". In that case, status information is carried either by RR or RNR responses. A P-bit set to "1" shall only be sent with a Supervisory Frame.

If such status information is received, retransmission from N(R) onwards will be performed if appropriate. However, no frame sequence starting with a given N(R) shall be retransmitted more than N2 times. If there is a frame sequence that cannot be transmitted successfully after N2 repetitions, the RLP link shall be reset or disconnected.

If no status information is received during the time-out period, this request will be repeated up to N2 times. If still there is no valid status reported back, the RLP link shall be reset or disconnected.

5.3.4. Contentious situations

Due to the asynchronous procedure, various contentious situations may arise. A contention of SABMs shall result into both entities be set into ABM or be reset. A contention of DISC's shall result into both entities be disconnected. A contention of SABM and DISC shall result into both entities be disconnected.

5.4. List of system parameters

The system parameters are as follows:

5.4.1 . Timer T1

The period of Timer T1 is regarded to start at the beginning of the transmission of the relevant frame.

The negotiation (or default) value is defined to be the earliest instant to enter recovery.

The period of Timer T1 at the end of which retransmission of a frame may be initiated according to the procedures described in 5.3 above, is a system parameter agreed for a period of time.

The proper operation of the procedure requires that Timer T1 be greater than the maximum time between transmission of frames (SABM, DM, DISC, I or supervisory commands) and the reception of the corresponding frame returned as a response to this frame (UA, DM or acknowledging frame). Therefore, the RLP entity should not delay the response or acknowledging frame returned to the above frame by more than a value T2. T2 is a system parameter, which is less than T1.

5.4.2. Maximum number of retransmissions N2

The value of the maximum number of retransmissions N2 of a frame following the running out of Timer T1 is a system parameter agreed for a period of time.

5.4.3. Maximum number of outstanding I frames k

The maximum number (k) of sequentially numbered I frames that may be outstanding (i.e. unacknowledged) at any given time is a system parameter which can never exceed 61. It shall be agreed for a period of time.

Name	Range of values	Default and recommended value
k MS → IWF	0 - 61	61
k IWF → MS	0 - 61	61
T1	> 380 ms > 600 ms	480 ms (fullrate) 780 ms (halfrate)
T2		< 80 ms (fullrate) < 80 ms (halfrate)
N2	> 0	6

T2 < T1 - (2 * transmission delay)

Table 2: RLP parameter values

5.5. Support for discontinuous transmission (DTX)

In both ADM and ABM, whenever the RLP entity has no numbered or unnumbered supervisory commands/responses and no information transfer frames pending transmission, the RLP entity shall indicate to the lower layer that the DTX function may be invoked.

6. SERVICE DEFINITIONS

6.1. Introduction

This chapter defines the service provided by the RLP-sublayer to the L2R-sublayer at the boundary between the RLP-sublayer and the L2R-sublayer.

The relationships between RLP-sublayer, L2R-sublayer and RLP-protocol are shown in figure 3.

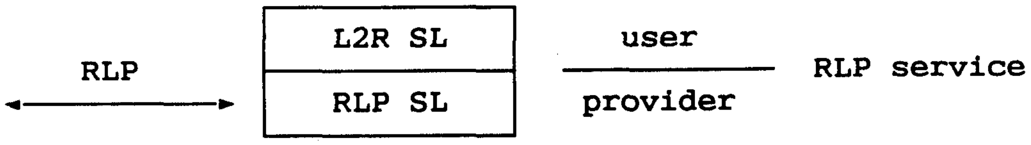


Figure 3: Basic relationship between RLP and L2R

The RLP service is defined in terms of

- the primitive actions and events of the service.
- the parameters associated with each primitive action and event.
- the inter-relationship between, and the valid sequence of, these actions and events.

6.2. Conventions

For the description of the Data Link Service, the following conventions are used with time-sequence diagrams:

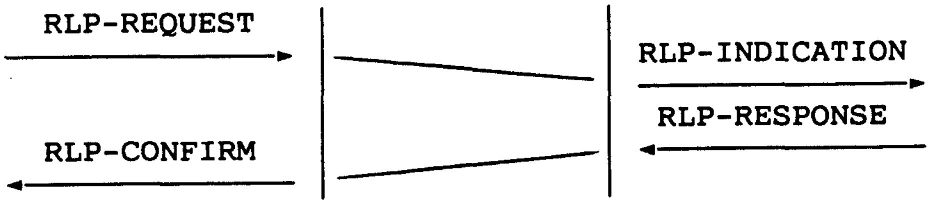


Figure 4: Confirmed service with acknowledgement

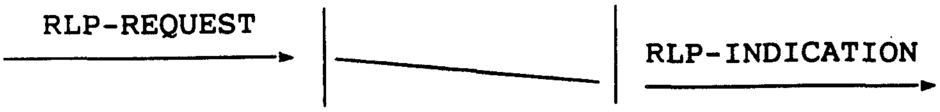


Figure 5: Unconfirmed service

In time-sequence diagrams, time moves from top to bottom. Arrows indicate the flow of information. Such flow of information may be subject to implicit flow-control. Skewed lines indicate a logical relationship between arrows. For clarity, the absence of such a relation may be marked by the symbol "~" (tilde).

6.3. Queue Model

Between the two endpoints of an RLP-connection, there exists a flow control function. As a means of specifying this flow control feature and its relationship with other capabilities of the RLP, the following queue model is provided.

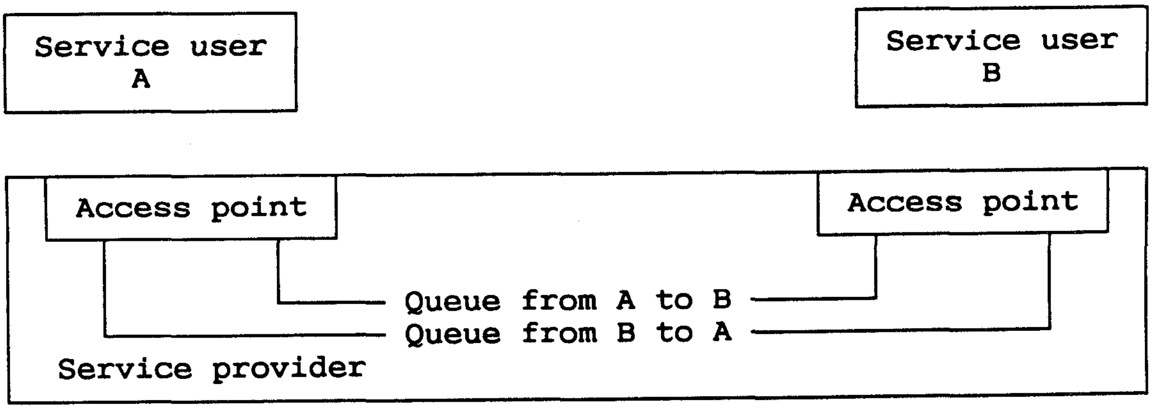


Figure 6: Queue Model

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

a) connect
b) connection-mode data (numbered information)
c) reset
d) disconnect

The following objects may be placed in a queue by a service provider:

a) reset
b) synchronization mark
c) disconnect

Note: Other possible objects (i.e. unnumbered information, identification, test) are irrelevant (-) to the queue model and for reasons of simplicity are not shown.

The relation between these objects is shown in the following table:

Following Preceding	Connect	Data	Reset	Sync Mark	Disconnect
Connect	NA	——	——	NA	DES
Data	NA	——	DES	NA	DES
Reset	NA	——	DES	——	DES
Synchronization Mark	NA	——	DES	NA	DES
Disconnect	NA	NA	NA	NA	DES

Legend:

NA : Not applicable
-- : not destructive, not able to advance ahead of the preceding object
DES : Destructive to the preceding object

6.4. List of Primitives

Link establishment

RLP-CONNECT-REQUEST
RLP-CONNECT-INDICATION
RLP-CONNECT-RESPONSE (-NEG)
RLP-CONNECT-CONFIRM (-NEG)

Normal Data Transfer

RLP-DATA-REQUEST (INF)
RLP-DATA-INDICATION (INF)

Reset

RLP-RESET-REQUEST
RLP-RESET-INDICATION
RLP-RESET-RESPONSE
RLP-RESET-CONFIRM

Release

RLP-DISCONNECT-REQUEST
RLP-DISCONNECT-INDICATION

Miscellaneous

unnumbered information

RLP-UNITDATA-REQUEST (INF)
RLP-UNITDATA-INDICATION (INF)

Exchange Identification

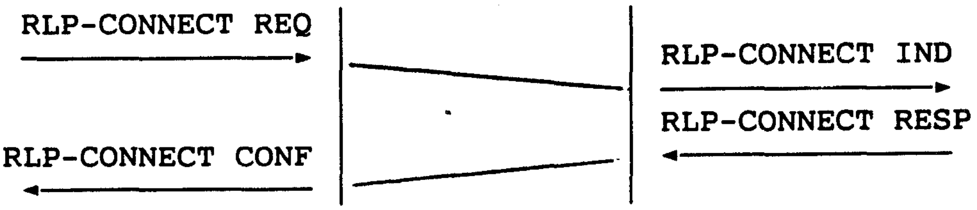
RLP-XIDDATA-REQUEST (INF)
RLP-XIDDATA-INDICATION (INF)

Test

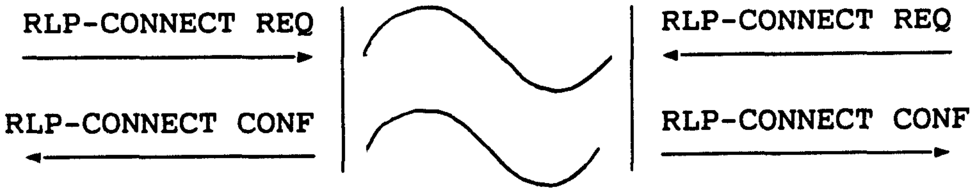
RLP-TESTDATA-REQUEST (INF)
RLP-TESTDATA-INDICATION (INF)

6.5. Possible RLP time sequence diagrams

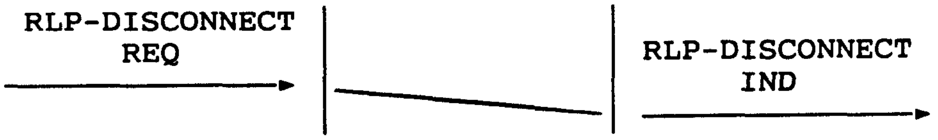
a) Connection establishment (without collision)



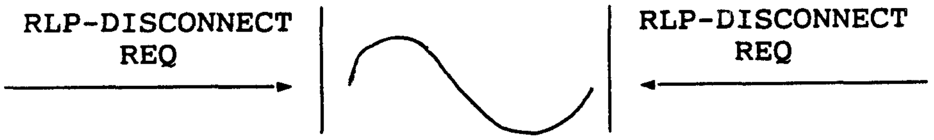
b) Connection establishment (with collision)



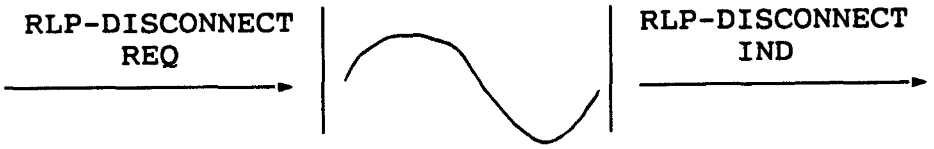
c) User invoked release (without collision)



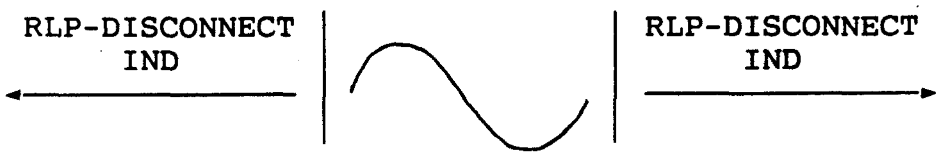
d) Collision of user invoked releases



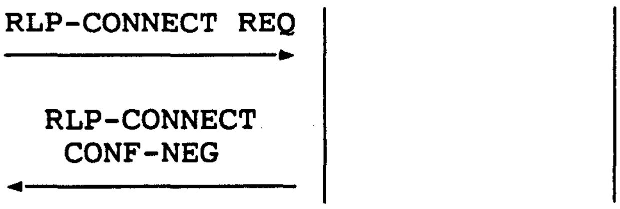
e) Simultaneous user and provider invoked release



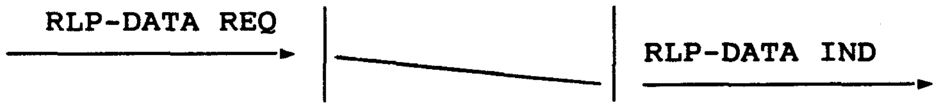
f) Provider invoked release



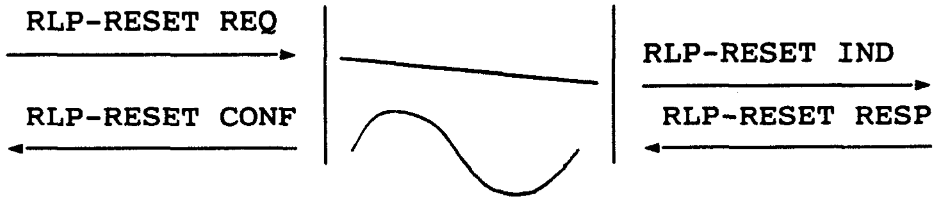
g) Provider rejection of establishment



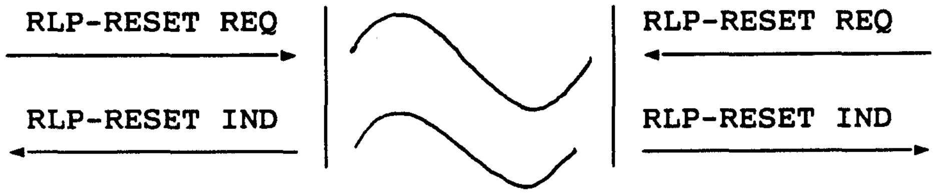
h) Normal data transfer



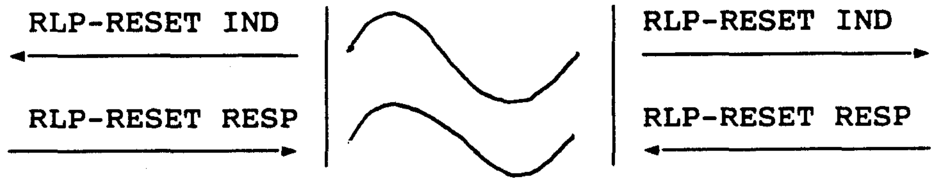
i) User invoked reset



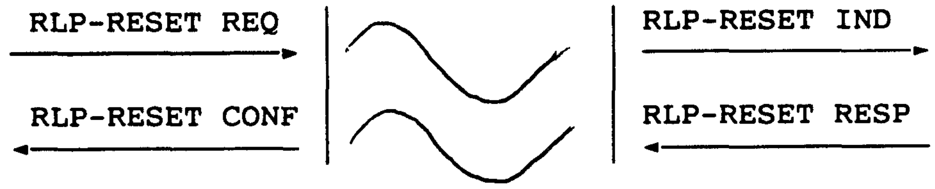
j) Collision of user invoked resets



k) provider invoked reset



l) simultaneous user and provider inkoved reset



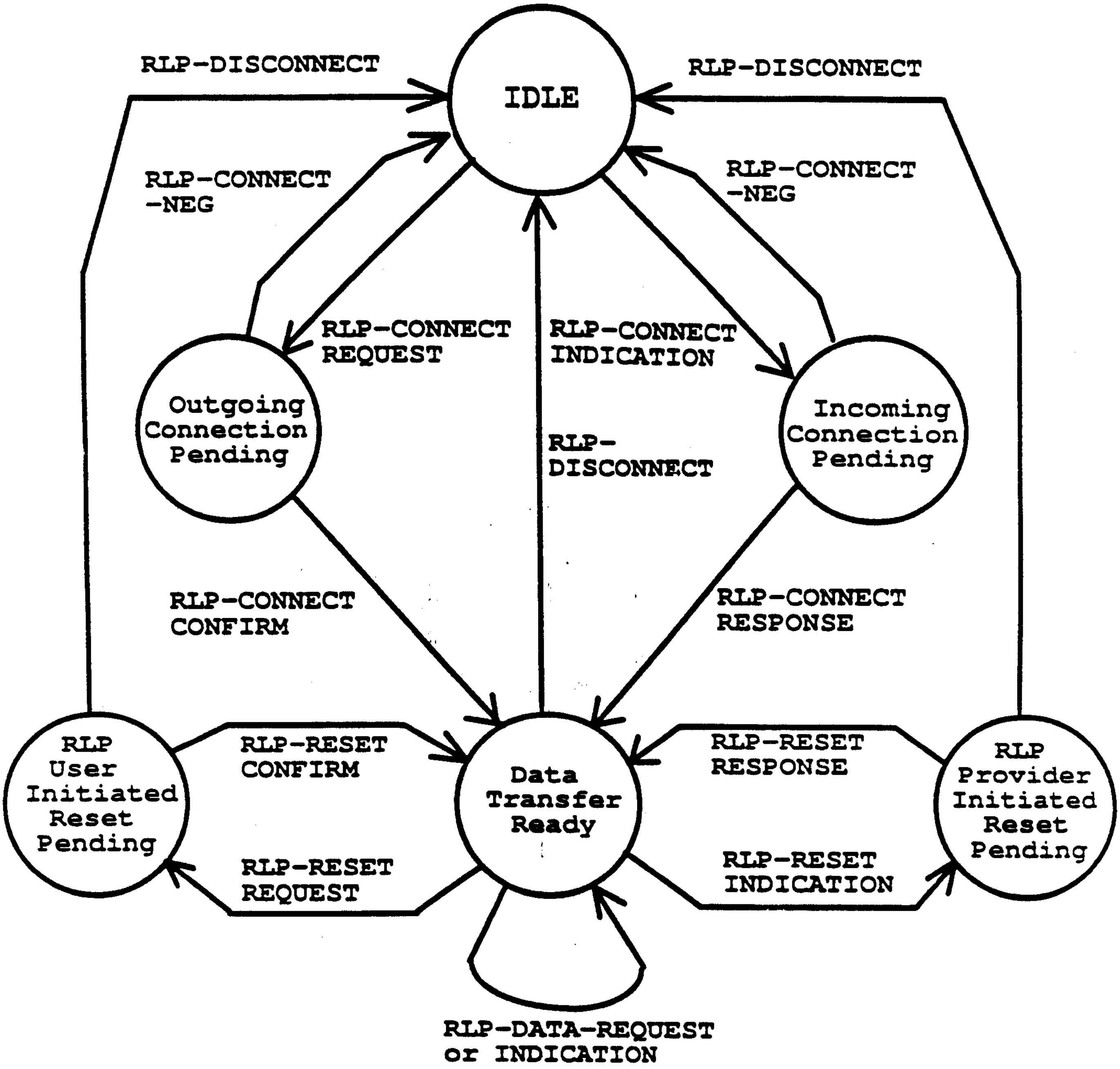


Figure 7: State transition diagram for sequence of RLP connection-mode service primitives

7. REFERENCES

ISO 3309:1984 Information processing systems-Data communications- High-level data link control procedures-Frame structure

ISO 4335:1987 Information processing systems-Data communication- High-level data link control procedures-Consolidation of elements of procedures

ISO 7498:1984 Information processing systems- Open Systems Interconnection-Basic Reference Model

ISO 7498:1984/ Add.1:1987 Information processing systems- Open Systems Interconnection-Basic Reference Model-Addendum 1: Connectionless-mode transmission

ISO 7776:1986 Information processing systems- Data communication- High-level data link control procedures- Description of the X.25 LAPB compatible DTE data link procedures

ISO 7809:1984 Information processing systems- Data communication- High-level data link control procedures- Consolidation of classes of procedures

ISO 7809:1984/ Add.1:1987 Information processing systems- Data communication- High-level data link control procedures- Consolidation of classes of procedures; Addendum 1

ISO 7809:1984/ Add.2:1987 Information processing systems- Data communication- High-level data link control procedures- Consolidation of classes of procedures- Addendum 2: Description of optional functions

ISO/TR 8509:1987 Information processing systems- Open Systems Interconnection- Service conventions

ISO 8885:1987 1987-08-15 Information processing- Data communication- High-level data link control procedures- General purpose XID frame information field content and format

ISO 8885/DAD1 Information processing systems- Data communication- High-level data link control procedures- General purpose XID frame information field content and format- Addendum 1: Additional operational parameters for the parameter negotiation Data Link Layer subfield and definition of a Multilink parameter negotiation Data Link Layer subfield

Final Text ISO/DIS 8886.2 Information processing systems- Data communication- Data link service definition for Open Systems Interconnection

I.440 (=Q.920) ISDN user-network interface data link layer-General aspectsRedbook III.5

I.441 (=Q.921) ISDN user-network interface data link layer-SpecificationRedbook III.5

X.25 Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuitRedbook VIII.3

A.2 List of RLP entity events

The interface is indicated by l:lower, u:upper and m:management. From the formal definition point of view this distinction of course is unneccessary.

event#	name	semantic	interface
1	Attach_Req	Switch to "ADM and Attached"	m
2	Conn_Req	Connect request	u
3	Conn_Resp	Connect response	u
4	Data_Req(Data)	Data transfer request (user data in Data)	u
5	Detach_Req	Switch to "ADM and Detached"	m
6	Disc_Req	Disconnect request	u
7	DISC(P)	PDU DISC received (P-bit in P)	l
8	DM(F)	PDU DM received (F-bit in F)	l
9	Ready_Ind	Indication that a new PDU may be sent	m
10	RR_I(C_R,P_F,NR,NS,Data)	I-frame RR received	l
11	RNR_I(C_R,P_F,NR,NS,Data)	I-frame RNR received	l
12	REJ_I(C_R,P_F,NR,NS,Data)	I-frame REJ received	l
13	SREJ_I(C_R,P_F,NR,NS,Data)	I-frame SREJ received	l
14	RR(C_R,P_F,NR)	S-frame RR received	l
15	RNR(C_R,P_F,NR)	S-frame RNR received	l
16	REJ(C_R,P_F,NR)	S-frame REJ received	l
17	SREJ(C_R,P_F,NR)	S-frame SREJ received	l
18	SABM(P)	PDU SABM received	l
19	UA(F)	PDU UA received (F-bit in F)	l
20	UI_Req(Data)	Unnumbered Information transfer request	u
21	UI(C_R,P_F,Data)	UI PDU received	l
22	T	Timeout (Timer of the sender expired)	m
23	T_RCVR	Timeout (Timer of the receiver exp.)	m
24	T_TEST	Timeout (Test timer expired)	m
25	T_XID	Tineout (Xid timer expired)	m
26	Test_Req(Data)	Test request (Test data in Data)	m
27	TEST(C_R,P_F,Data)	TEST command/response PDU received (C/R-bit in C_R, P/F-bit in P_F, Data in Data)	l
28	XID_Req(Data)	Exchange ID request	m
29	XID(C_R,P_F,Data)	XID command/response PDU received	l

variable name	type and range	semantic
S[M]	record array ...	Sender Slots (M slots, numbered 0 to M-1)
S[n].Data	char[25]	user information to be sent.
S[n].State	(idle, send, wait)	(S[n].State = send) means, data has to be sent (with sequence# n).
SF	(RR,RNR,REJ,SREJ)	to store the last superv. PDU type
T	Timer	used by the data sender if waiting for I-frame acknowledgements or F-bits.
T_RCVR	Timer	used by the receiver to timeout a REJ or a SREJ condition.
T_TEST	Timer	used by the sender of a TEST frame if waiting for a TEST response.
T_XID	Timer	used by the sender of a XID frame if waiting for the XID response.
UA_FBit	(0, 1)	value of the F-Bit used in the next UA response.
UA_State	(idle, send)	if (UA_State = send) an UA PDU has to be sent.
VA	(0, 1, ..., Nmax)	frame sequence number of oldest not yet acknowledged I-frame (if VA = VS then there are no unacknowledged frames).
VD	(0, 1, ..., Nmax)	slot number used in the next Data_Req.
VR	(0, 1, ..., Nmax)	receiver sequence number (the next received I-frame is expected to carry this sequence number).
VS	(0, 1, ..., Nmax)	sender sequence number (under normal operating conditions the next I-frame is assigned this number).
WS	(1, 2,..., M-1)	the current windowsize (the maximum number allowed for unacknowledged I-frames).

variable name	type and range	semantic
Poll_Count	(0, 1, ..., N2)	to count the tramsmissions of poll requests.
Poll_State	(idle, send, wait)	(Poll_State = send) means, a supervisory PDU with P-bit set to one has to be sent. (Poll_State = wait) means, the RLP entity waits for the response with F-bit set to one.
R[M]	record array ...	Receiver slots (M slots, numbered 0 to M-1)
R[n].Count	(0, 1, ..., N2)	to count the transmissions of SREJ(n).
R[n].Data	char[25]	to store user information
R[n].State	(idle, rcvd, ackn, srej, wait)	(R[n].State = rcvd) means, data has been received (with sequence number n). (R[n].State = ackn) means, data has been received and acknowledged. (R[n].State = srej) means, the retransmission of data has to be requested using srej(n). (R[n].State = wait) means, the entity waits for the requested retransmitted data.
REJ_Count	(0, 1, ..., N2)	to count the transmissions of REJ
REJ_State	(idle, send, wait)	The REJ_State is send if and only if a REJ PDU has to be sent.
returncode	Integer	used in procedures to report a result
RRReady	Boolean	Remote Receiver Ready
SABM_Count	(0, 1, ..., N2)	to count the transmissions of SABM
SABM_State	(idle, send, wait)	if (.._State = send) the SABM PDU has to be sent. if (.._State = wait) the RLP entity waits for the UA response.

variable name	variable type and range	semantic
Ackn_FBit	(0, 1)	Value of the F-Bit used in the next acknowledging PDU.
Ackn_State	(idle, send)	(Ackn_State = send) means, an acknowledging PDU (Supervisory or Data) has to be sent.
C_R	(0, 1)	to store the C/R-Bit value of a received S- or I-frames
Data	char[25]	to store temporarily the information part (user data) of a received I-frame.
DISC_Count	(0, 1, ..., N2)	to count the transmissions of DISC.
DISC_PBit	(0, 1)	The value of the P-bit in the next DISC command PDU.
DISC_State	(idle, send, wait)	if (DISC_State = send) the DISC command PDU has to be sent at the next possible opportunity. if (DISC_State = wait) the RLP entity waits for the corresponding response.
DM_FBit	(0, 1)	Value of the F-Bit used in the next DM response PDU.
DM_State	(idle, send)	if (DM_State = send) the PDU DM has to be sent.
DTX_SF	(N, RR, RNR)	to store the last Supervisory frame for DTX (only RR or RNR can be suppressed)
DTX_VR	(0, 1, ..., Nmax)	to store the last transmitted value of VR (used to decide the DTX condition)
F	(0, 1)	to store temporarily the F-bit of a received response PDU.
NR	(0, 1, ..., Nmax)	to store temporarily the receive sequence number of a received S- or I-frame.
NS	(0, 1, ..., Nmax)	to store temporarily the send sequence number of a received I-frame.
P	(0, 1)	to store temporarily the P-bit of a received command PDU
P_F	(0, 1)	to store temporarily the P- or F-bit of received command or response PDUs.

A.1 List of RLP entity states

A.1.1 (main) states

state number	state symbol	state name
0	S0	ADM and Detached
1	S1	ADM and Attached
2	S2	Pending Connect Request
3	S3	Pending Connect Indication
4	S4	ABM and Connection Established
5	S5	Disconnect Initiated
6	S6	Pending Reset Request
7	S7	Pending Reset Indication
8	S8	Permanent Error

A.1.2 state variables

The main states are further distinguished by the values of the state variables.

However, not every state variable is used (evaluated/ defined) in every state.

First some constants need to be defined:

M	= 32	number of different sequence numbers
Nmin	= 0	smallest sequence number
Nmax	= 31	largest sequence number (M - 1)
N2	= 6	maximum number of retransmissions

A RLP SDL Diagrams

This annex describes a model implementation of an RLP entity.

The description should help to clarify GSM recommendation 04.22, the RLP service and protocol definition.

However, it is not intended to restrict any implementation of an RLP entity in any way, on condition the implementation shows the correct behaviour at the RLP protocol level.

The model implementation consists of three processes. Process "SEND_PDU" adds the CRC to a given PDU and hands it to the lower layer entity for transmission. Process "RECEIVE_PDU" gets a received PDU block, checks the value of the CRC and the bits of the PDU header. If the CRC has the right value and if the header is syntactically correct, the receipt event is signalled to the "RLP_KERNEL" process, which is the protocol handling automaton.

Each process is described as an extended finite state machine (using SDL-Diagrams).

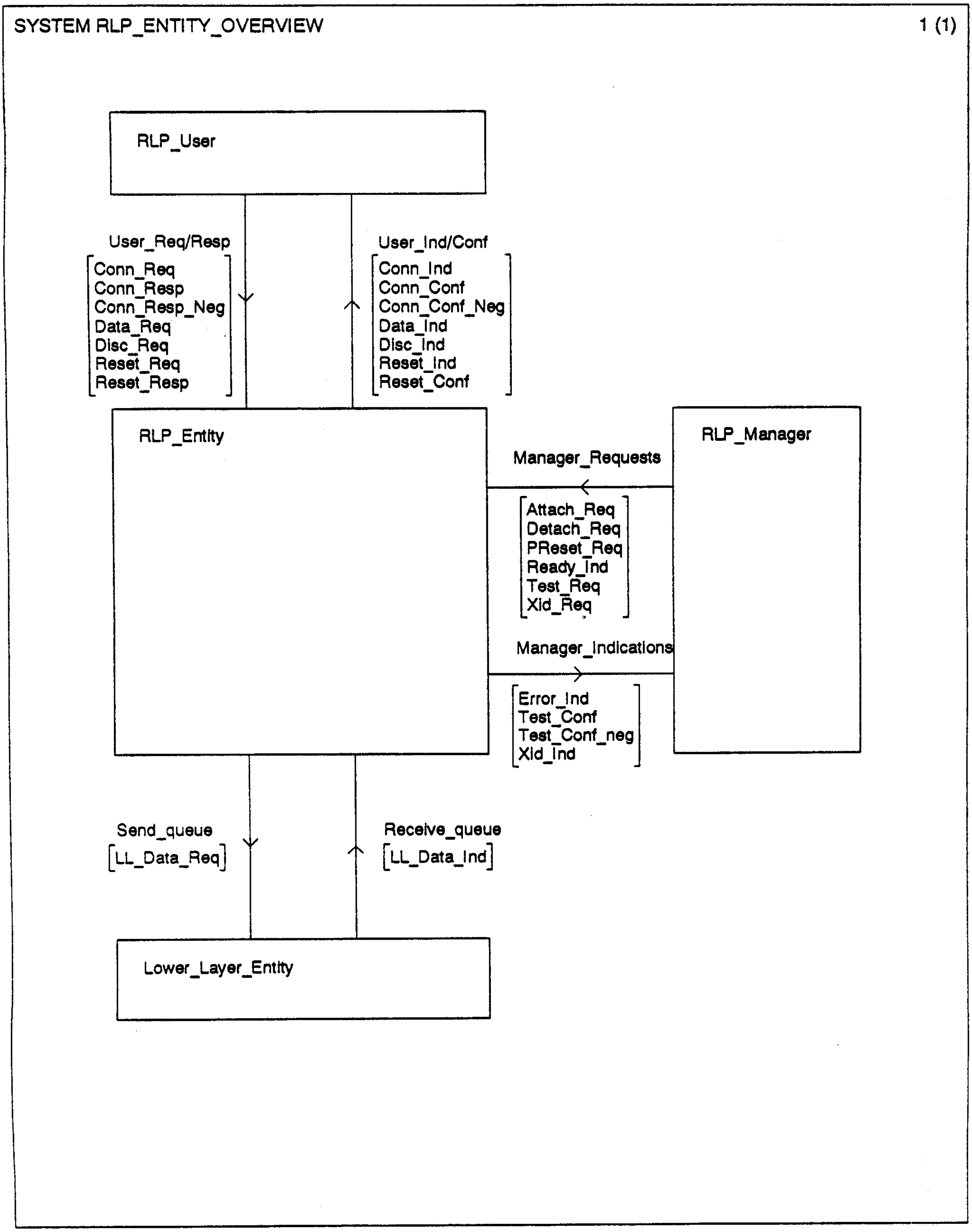
Each state of the automaton is described by a (main-)state number and a corresponding (main-)state name. The state may further be distinguished by the value of other state variables. This scheme is used because not every state variable needs to be defined in every state. The states are defined in chapter A.1.

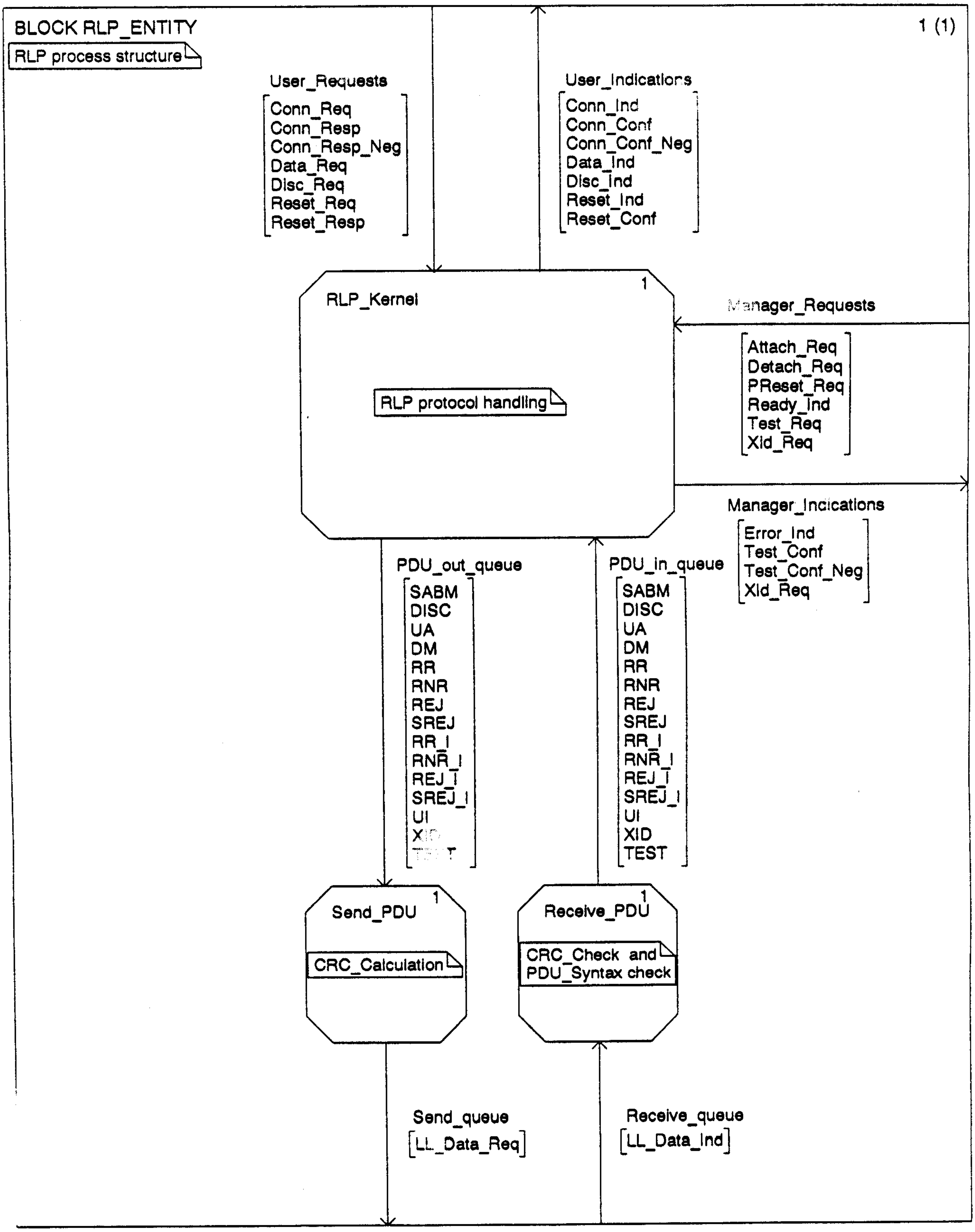
The RLP machine reacts on events, which may be classified as

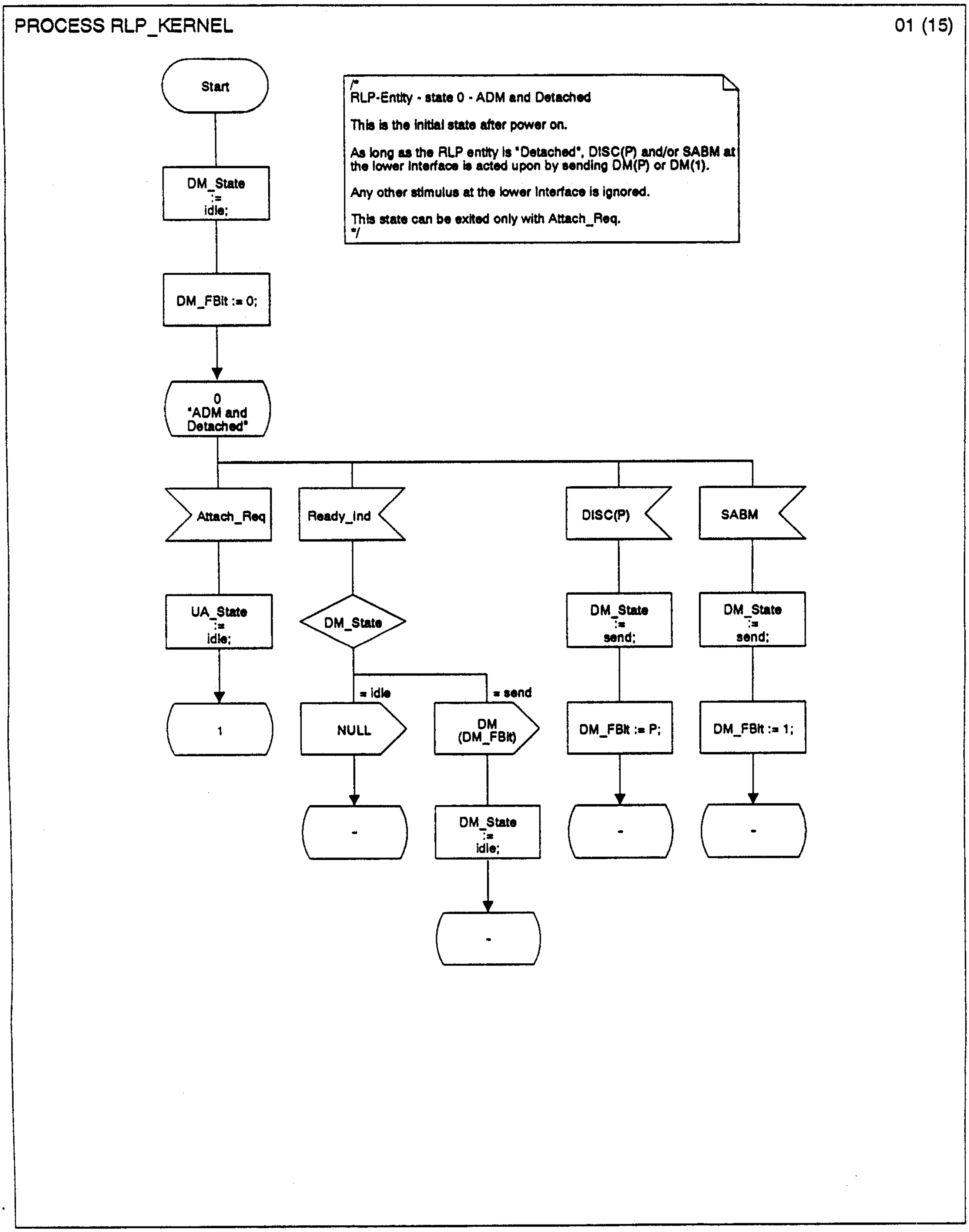
- lower layer interface events
- upper layer interface events

and - station management or internal events

The events of the RLP-Kernel are described in chapter A.2.


SYSTEM RLP_ENTITY_OVERVIEW
1 (1)
RLP_User
User_Req/Resp
Conn_Req
Conn_Resp
Conn_Resp_Neg
Data_Req
Disc_Req
Reset_Req
Reset_Resp
User_Ind/Conf
Conn_Ind
Conn_Conf
Conn_Conf_Neg
Data_Ind
Disc_Ind
Reset_Ind
Reset_Conf
RLP_Entity
Manager_Requests
Attach_Req
Detach_Req
PReset_Req
Ready_Ind
Test_Req
Xid_Req
RLP_Manager
Manager_Indications
Error_Ind
Test_Conf
Test_Conf_neg
Xid_Ind
Send_queue
LL_Data_Req
Receive_queue
LL_Data_Ind
Lower_Layer_Entity


BLOCK RLP_ENTITY
RLP process structure
1 (1)
User_Requests
Conn_Req
Conn_Resp
Conn_Resp_Neg
Data_Req
Disc_Req
Reset_Req
Reset_Resp
User_Indications
Conn_Ind
Conn_Conf
Conn_Conf_Neg
Data_Ind
Disc_Ind
Reset_Ind
Reset_Conf
RLP_Kernel
1
RLP protocol handling
Manager_Requests
Attach_Req
Detach_Req
PReset_Req
Ready_Ind
Test_Req
Xid_Req
Manager_Indications
Error_Ind
Test_Conf
Test_Conf_Neg
Xid_Req
PDU_out_queue
SABM
DISC
UA
DM
RR
RNR
REJ
SREJ
RR_I
RNR_I
REJ_I
SREJ_I
UI
XID
TEST
PDU_in_queue
SABM
DISC
UA
DM
RR
RNR
REJ
SREJ
RR_I
RNR_I
REJ_I
SREJ_I
UI
XID
TEST
Send_PDU
1
CRC_Calculation
Receive_PDU
1
CRC_Check and
PDU_Syntax check
Send_queue
LL_Data_Req
Receive_queue
LL_Data_Ind



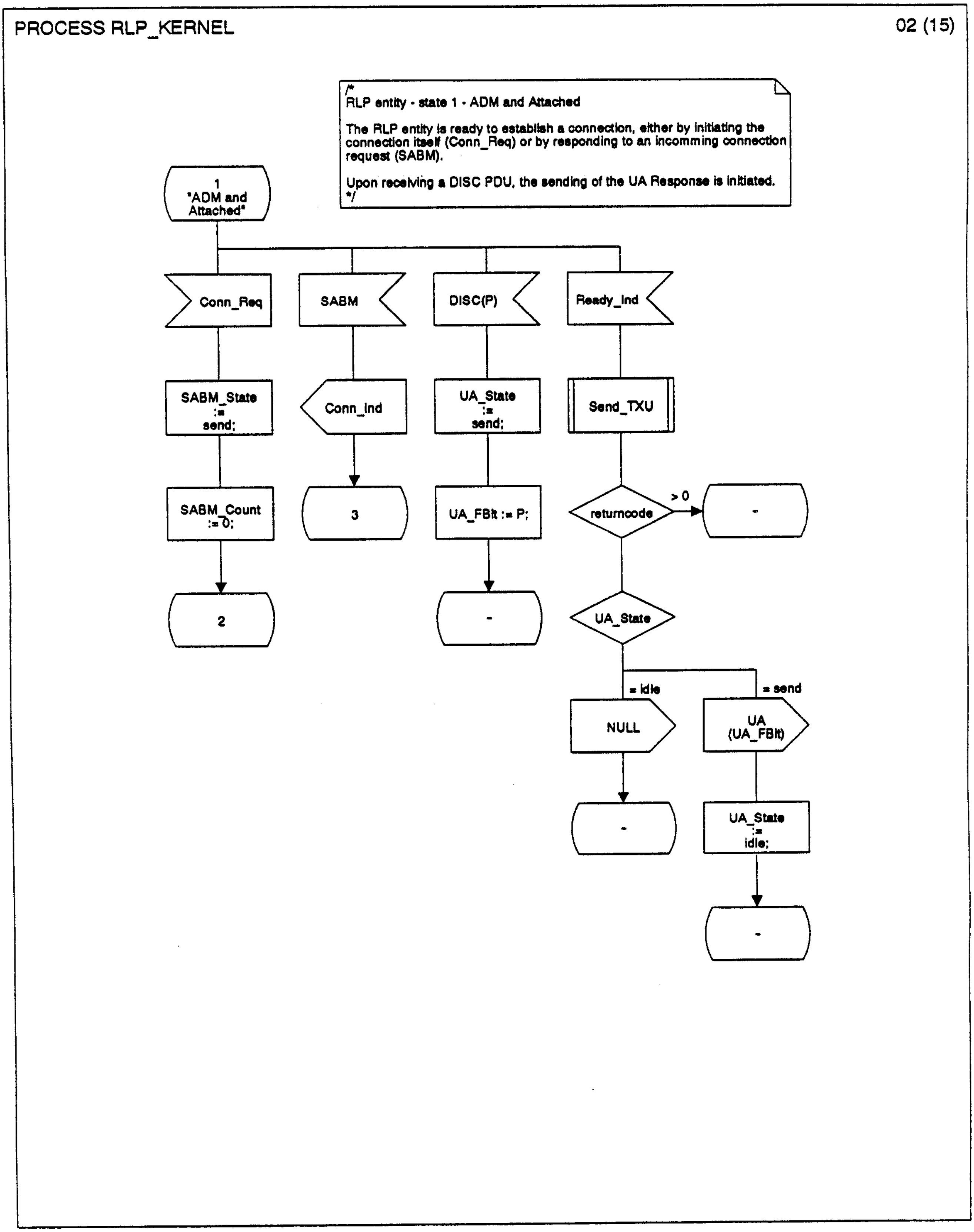
/*
RLP-Entity - state 0 - ADM and Detached

This is the initial state after power on.

As long as the RLP entity is "Detached", DISC(P) and/or SABM at the lower interface is acted upon by sending DM(P) or DM(1).

Any other stimulus at the lower interface is ignored.

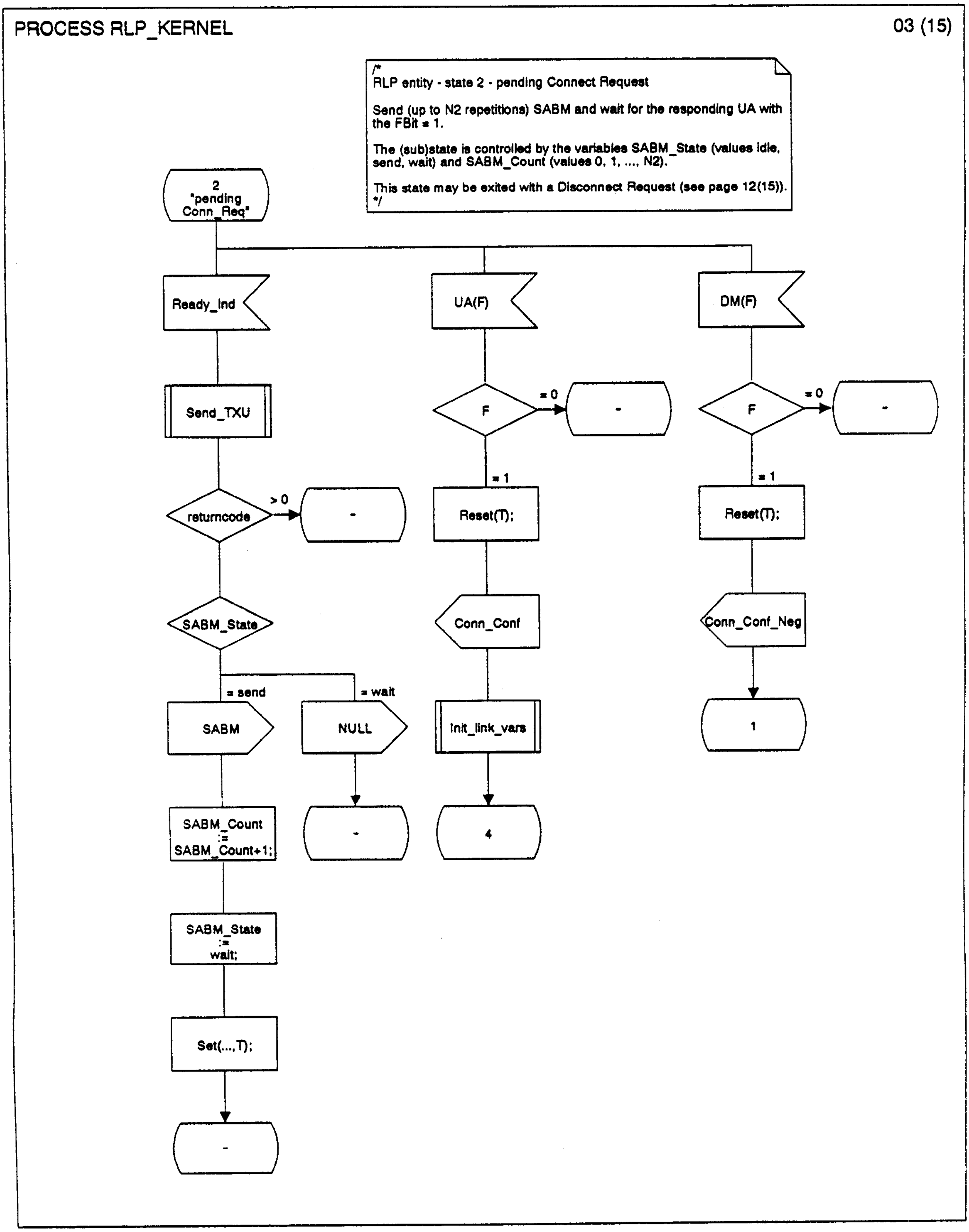
This state can be exited only with Attach_Req.
*/



/*
RLP entity - state 1 - ADM and Attached

The RLP entity is ready to establish a connection, either by initiating the connection itself (Conn_Req) or by responding to an incomming connection request (SABM).

Upon receiving a DISC PDU, the sending of the UA Response is initiated.
*/

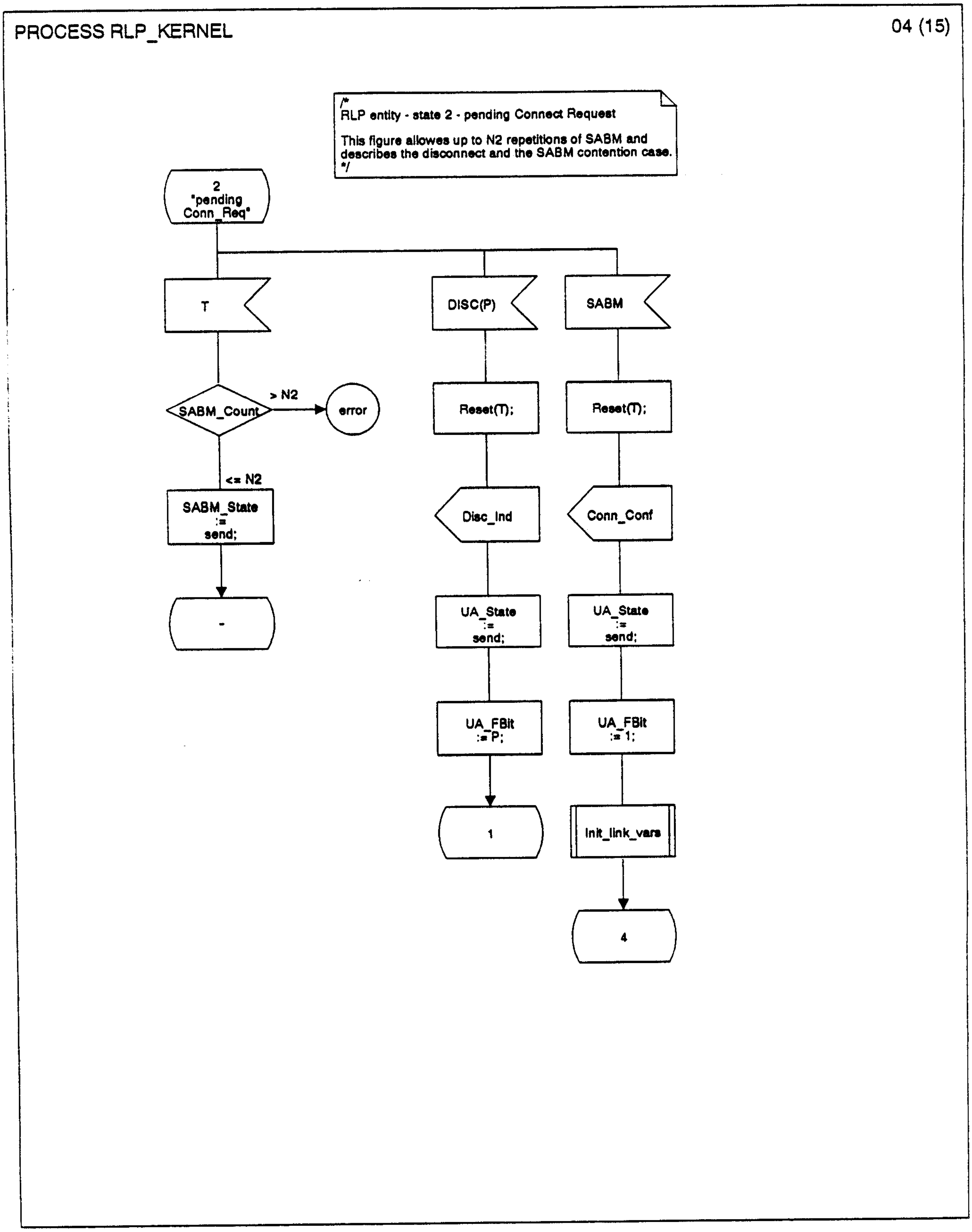


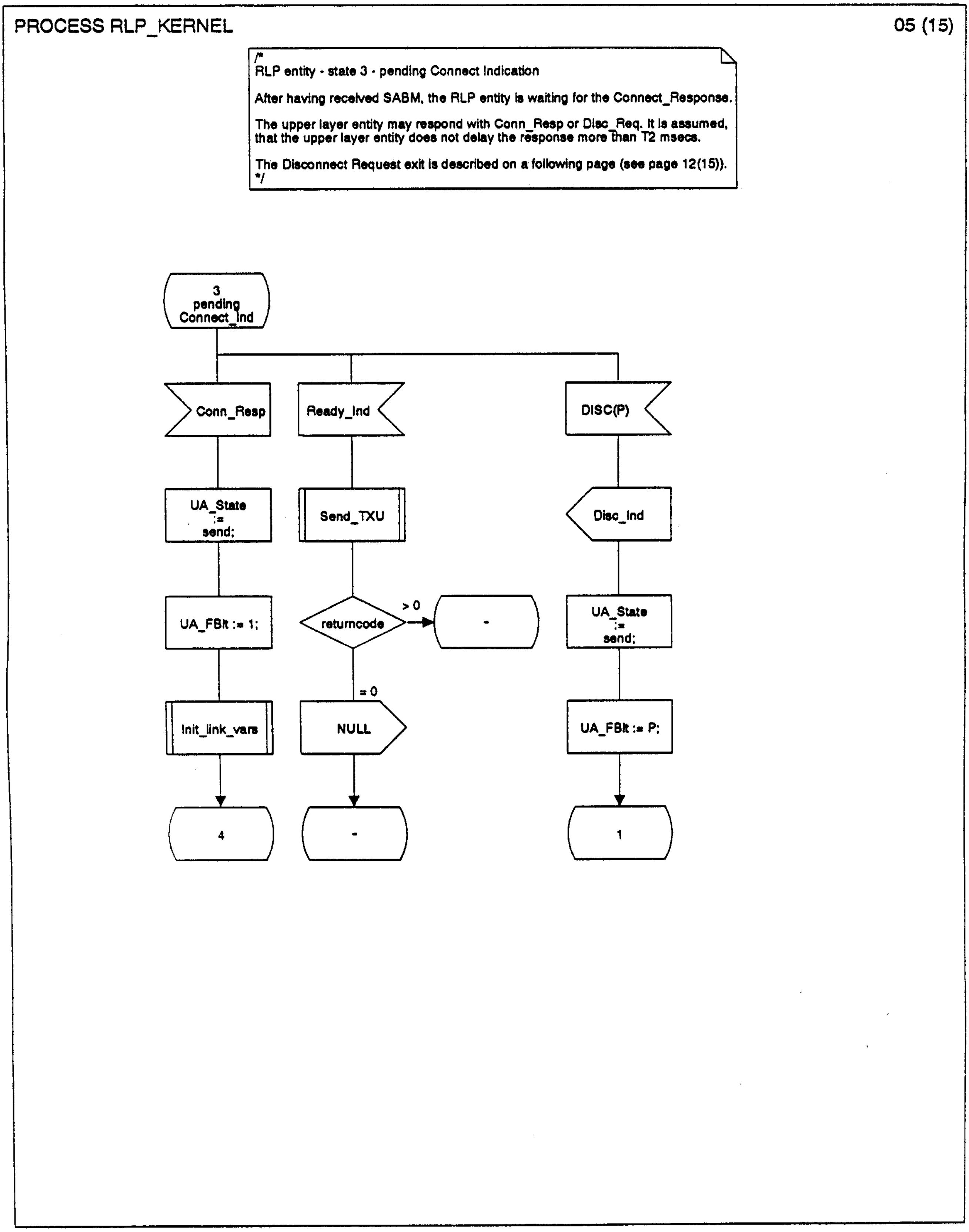
/*
RLP entity - state 2 - pending Connect Request

Send (up to N2 repetitions) SABM and wait for the responding UA with the FBit = 1.

The (sub)state is controlled by the variables SABM_State (values idle, send, wait) and SABM_Count (values 0, 1, ..., N2).

This state may be exited with a Disconnect Request (see page 12(15)).
*/


PROCESS RLP_KERNEL
04 (15)
/*
RLP entity - state 2 - pending Connect Request
This figure allowes up to N2 repetitions of SABM and describes the disconnect and the SABM contention case.
*/
2 "pending Conn_Req"
T
SABM_Count
> N2
error
<= N2
SABM_State := send;
-
DISC(P)
Reset(T);
Disc_Ind
UA_State := send;
UA_FBit := P;
1
SABM
Reset(T);
Conn_Conf
UA_State := send;
UA_FBit := 1;
Init_link_vars
4



PROCESS RLP_KERNEL

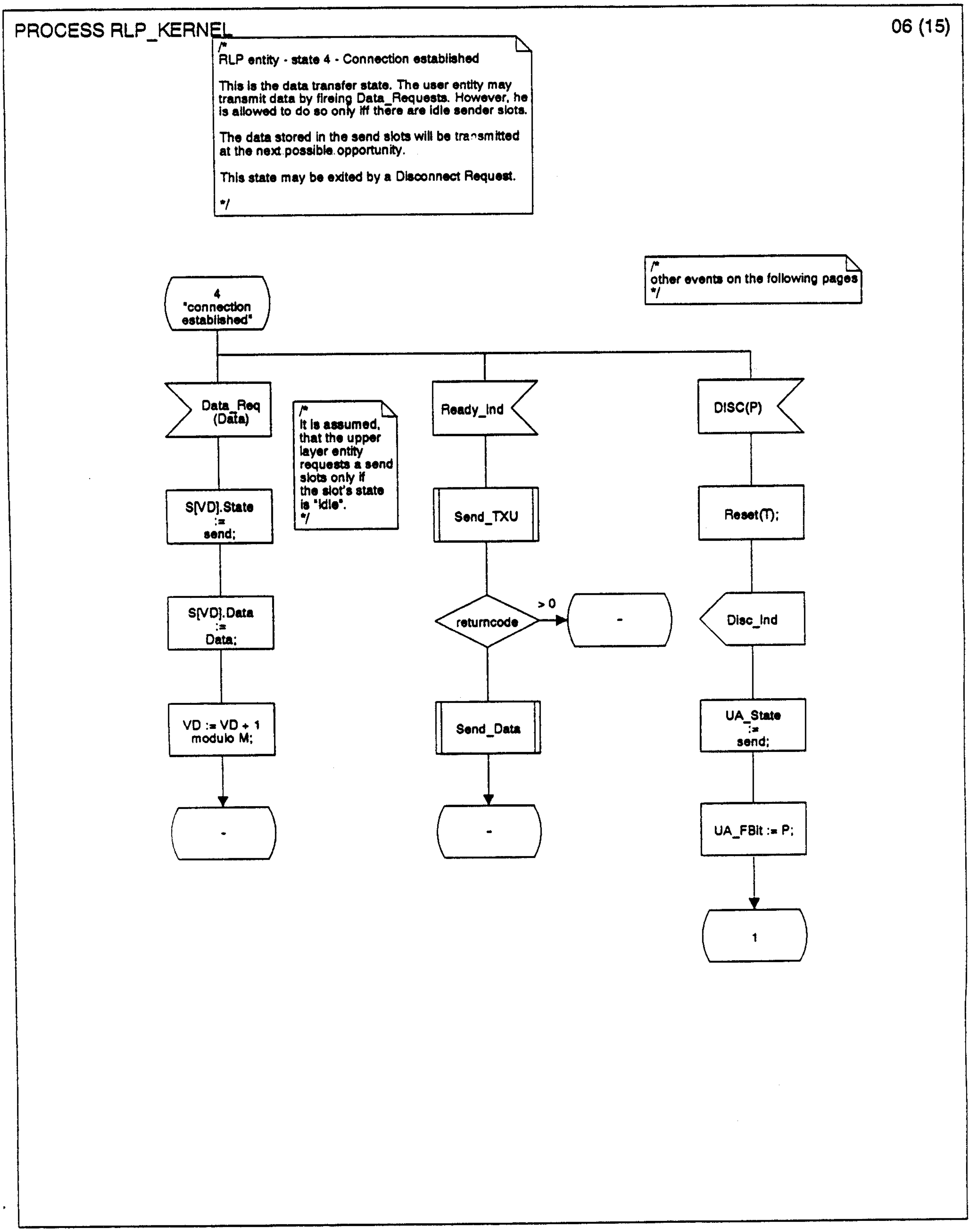
05 (15)

/*
RLP entity - state 3 - pending Connect Indication

After having received SABM, the RLP entity is waiting for the Connect_Response.

The upper layer entity may respond with Conn_Resp or Disc_Req. It is assumed, that the upper layer entity does not delay the response more than T2 msecs.

The Disconnect Request exit is described on a following page (see page 12(15)).
*/



/*
RLP entity - state 4 - Connection established

This is the data transfer state. The user entity may transmit data by fireing Data_Requests. However, he is allowed to do so only iff there are idle sender slots.

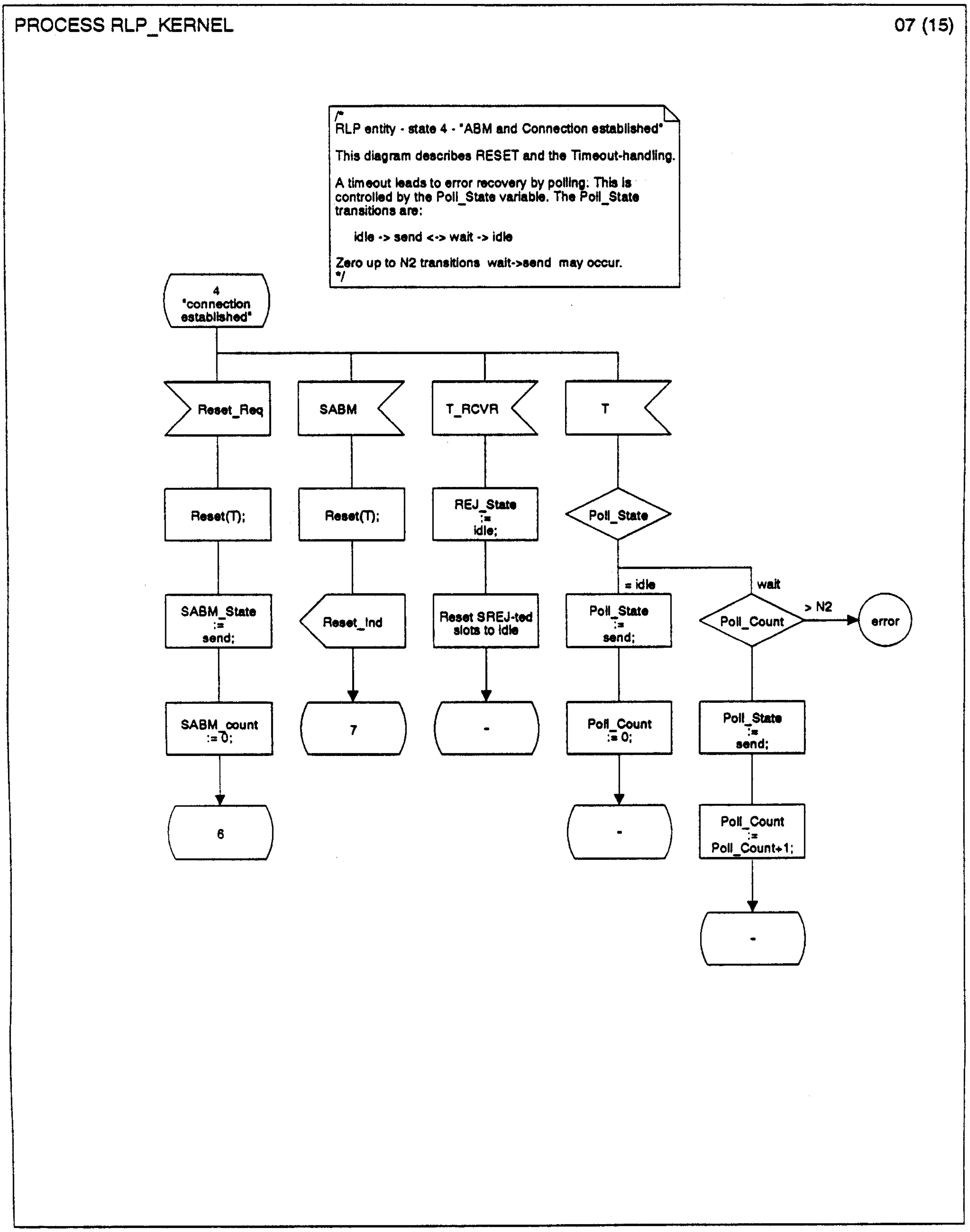
The data stored in the send slots will be transmitted at the next possible opportunity.

This state may be exited by a Disconnect Request.

*/

/*
other events on the following pages
*/

/*
It is assumed, that the upper layer entity requests a send slots only if the slot's state is "idle".
*/



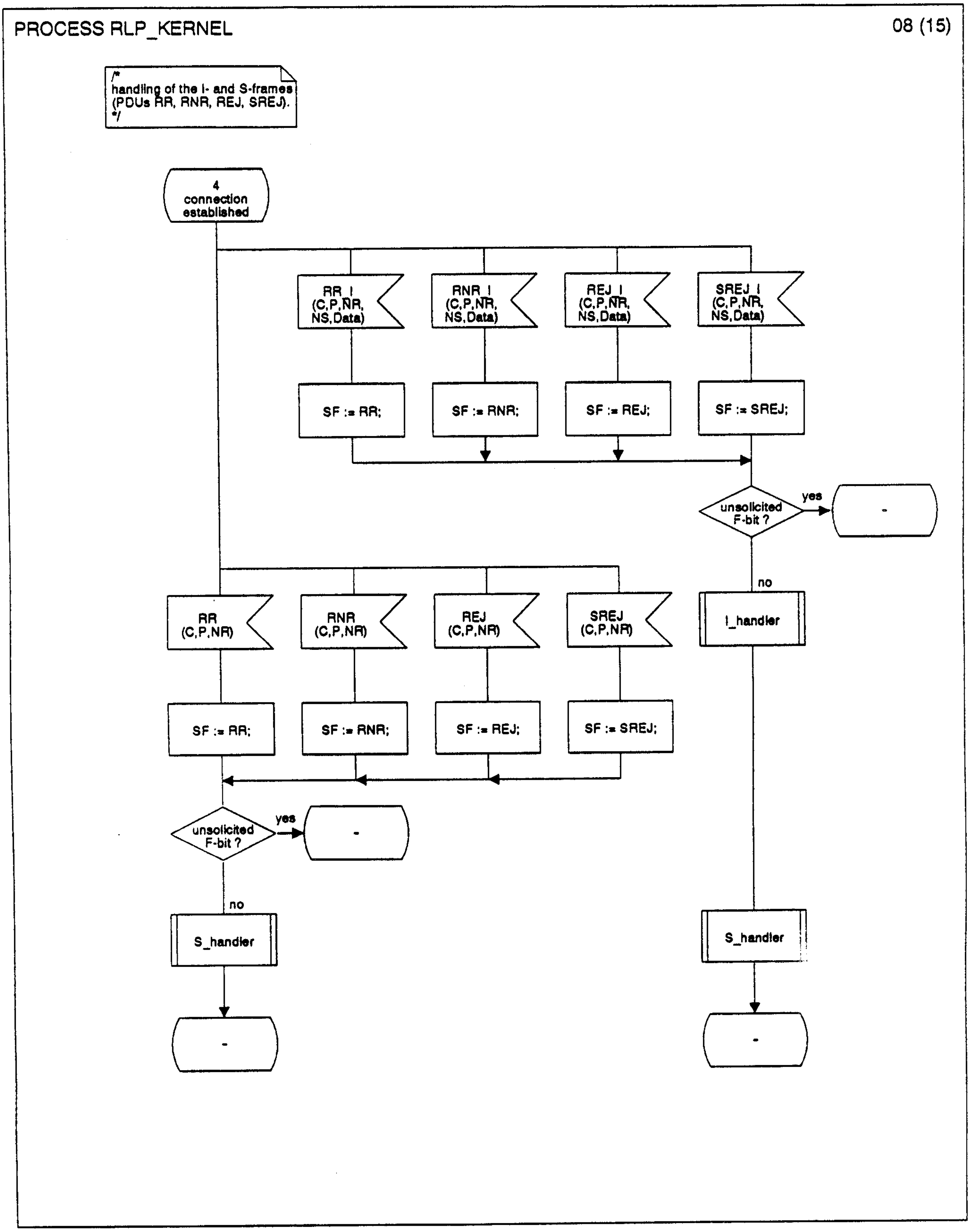
/*
RLP entity - state 4 - "ABM and Connection established"

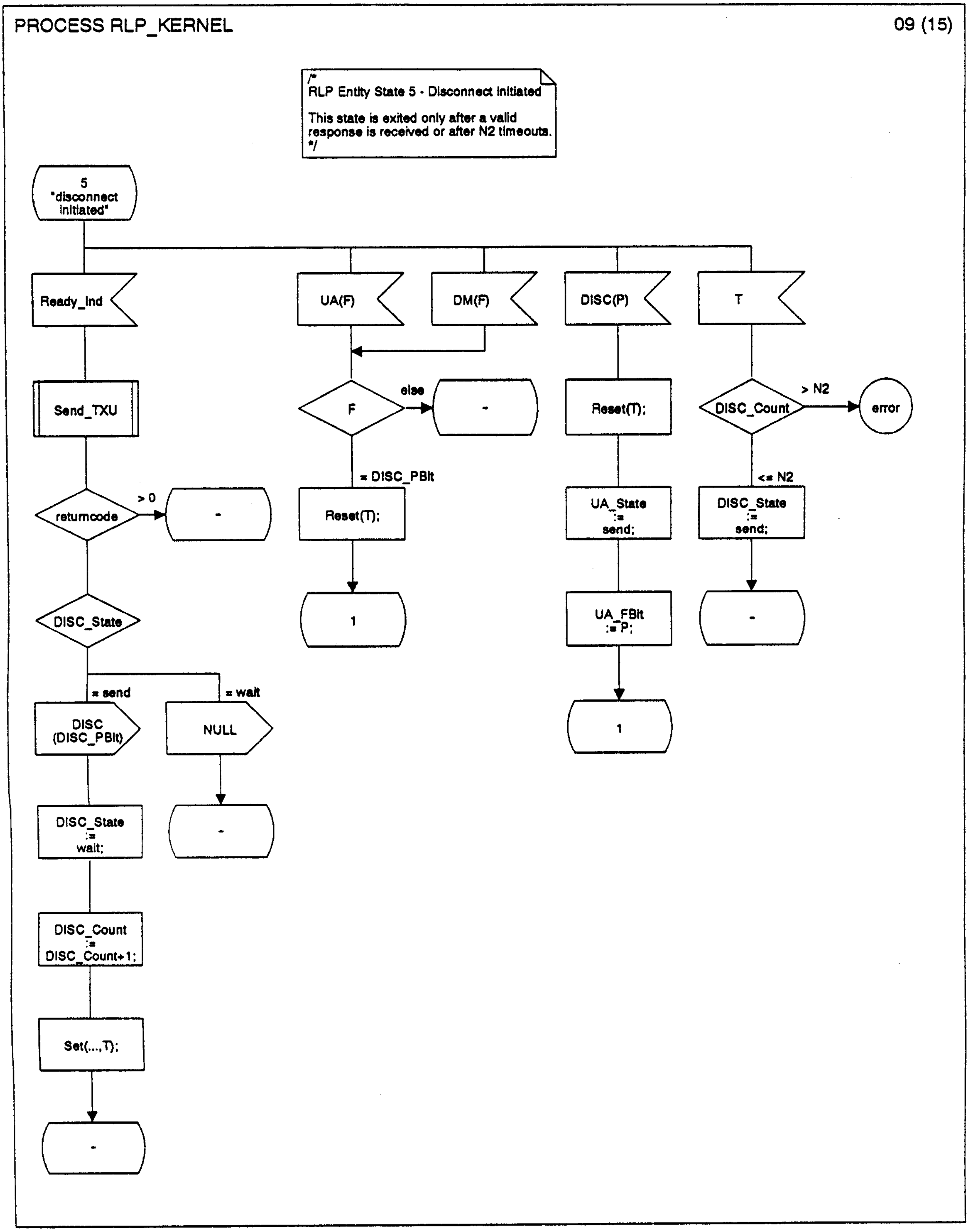
This diagram describes RESET and the Timeout-handling.

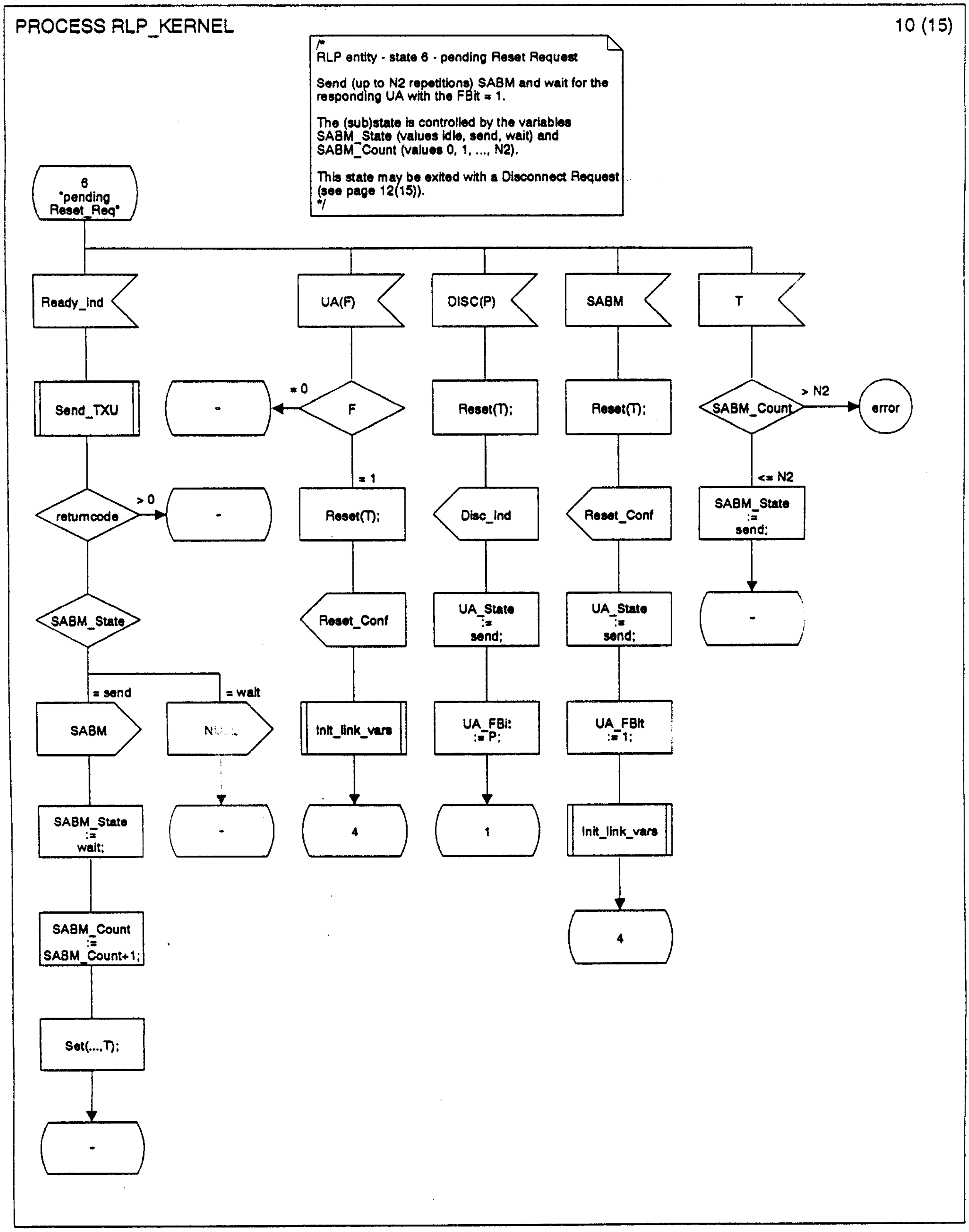
A timeout leads to error recovery by polling. This is controlled by the Poll_State variable. The Poll_State transitions are:

idle -> send <-> wait -> idle

Zero up to N2 transitions wait->send may occur.
*/


PROCESS RLP_KERNEL
08 (15)
/*
handling of the I- and S-frames
(PDUs RR, RNR, REJ, SREJ).
*/
4
connection
established
RR_I
(C,P,NR,
NS,Data)
RNR_I
(C,P,NR,
NS,Data)
REJ_I
(C,P,NR,
NS,Data)
SREJ_I
(C,P,NR,
NS,Data)
SF := RR;
SF := RNR;
SF := REJ;
SF := SREJ;
unsolicited
F-bit ?
yes
-
no
I_handler
RR
(C,P,NR)
RNR
(C,P,NR)
REJ
(C,P,NR)
SREJ
(C,P,NR)
SF := RR;
SF := RNR;
SF := REJ;
SF := SREJ;
unsolicited
F-bit ?
yes
-
no
S_handler
S_handler
-
-


PROCESS RLP_KERNEL
09 (15)
/*
RLP Entity State 5 - Disconnect Initiated
This state is exited only after a valid response is received or after N2 timeouts.
*/
5 "disconnect initiated"
Ready_Ind
Send_TXU
returncode
> 0
-
DISC_State
= send
= wait
DISC (DISC_PBit)
NULL
-
DISC_State := wait;
DISC_Count := DISC_Count+1;
Set(...,T);
-
UA(F)
DM(F)
F
else
-
= DISC_PBit
Reset(T);
1
DISC(P)
Reset(T);
UA_State := send;
UA_FBit := P;
1
T
DISC_Count
> N2
error
<= N2
DISC_State := send;
-



/*
RLP entity - state 6 - pending Reset Request

Send (up to N2 repetitions) SABM and wait for the responding UA with the FBit = 1.

The (sub)state is controlled by the variables SABM_State (values idle, send, wait) and SABM_Count (values 0, 1, ..., N2).

This state may be exited with a Disconnect Request (see page 12(15)).
*/

PROCESS RLP_KERNEL

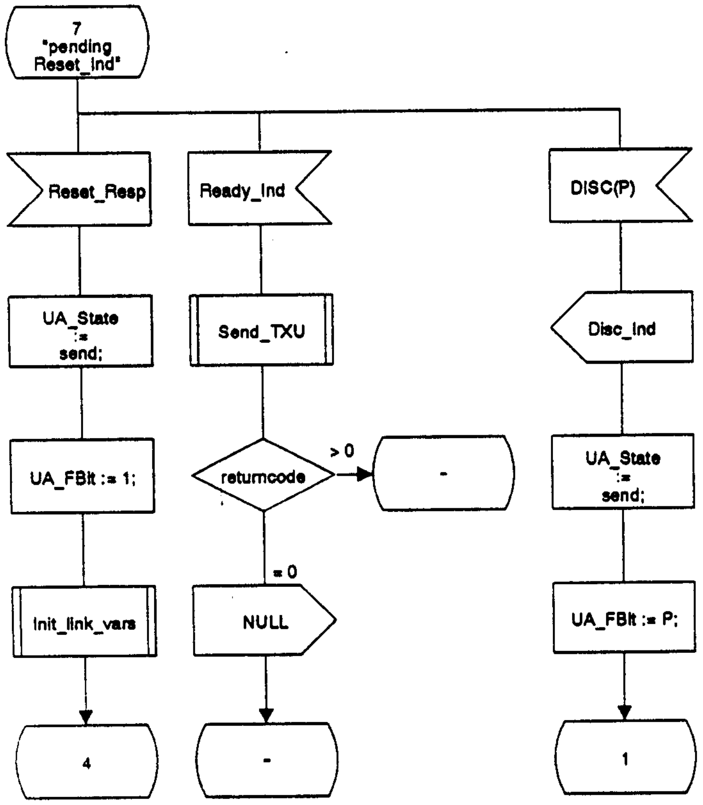
11 (15)

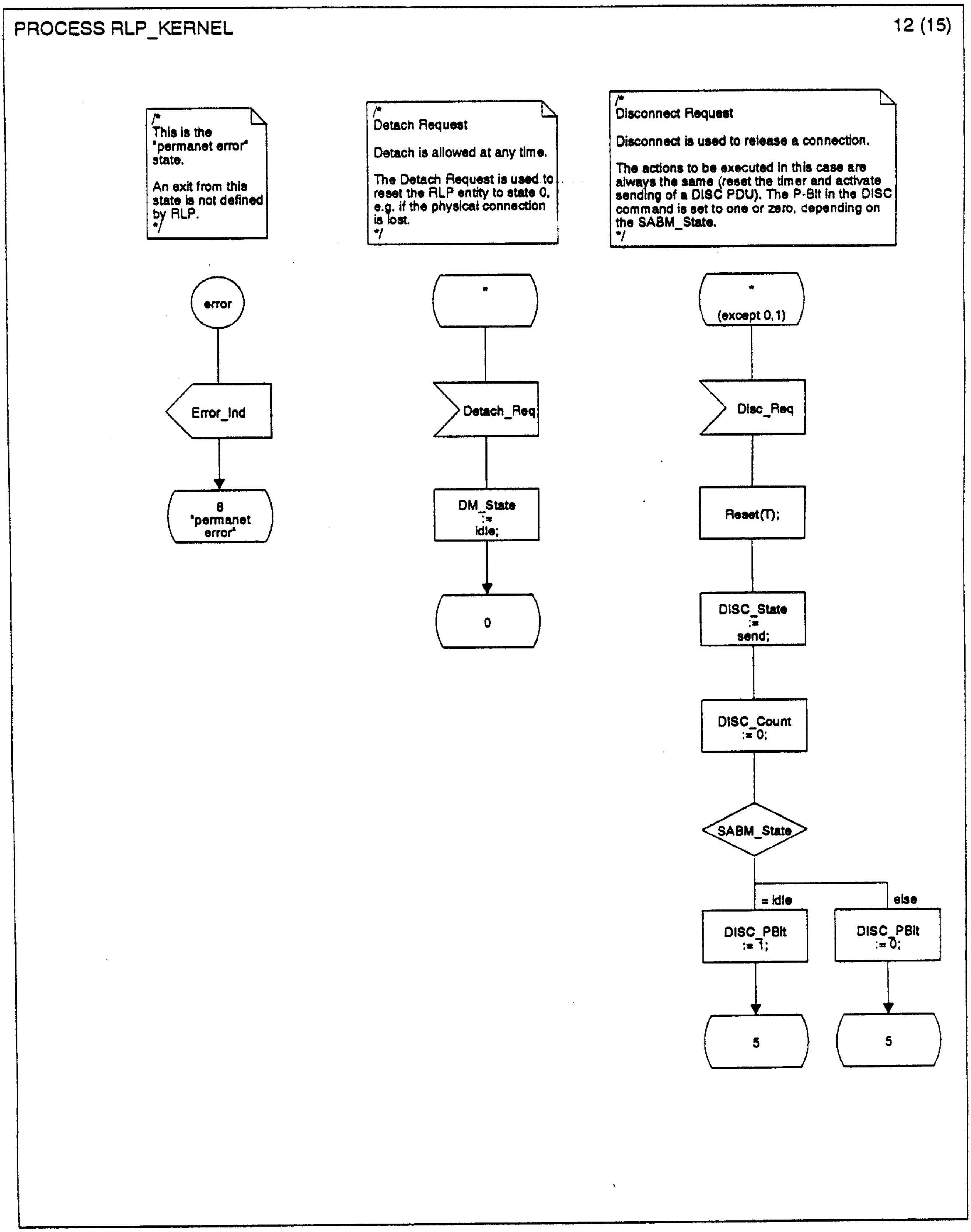
/*
RLP entity - state 7 - pending Reset Indication

After having received SABM, the RLP entity is waiting for the Reset_Response.

The upper layer entity may respond with Reset_Resp or Disc_Req. It is assumed, that the upper layer entity does not delay the response more than T2 msecs.

The Disconnect Request exit is described on a following page (see page 12(15)).
*/





PROCESS RLP_KERNEL

12 (15)

/*
This is the "permanet error" state.

An exit from this state is not defined by RLP.
*/

/*
Detach Request

Detach is allowed at any time.

The Detach Request is used to reset the RLP entity to state 0, e.g. if the physical connection is lost.
*/

/*
Disconnect Request

Disconnect is used to release a connection.

The actions to be executed in this case are always the same (reset the timer and activate sending of a DISC PDU). The P-Bit in the DISC command is set to one or zero, depending on the SABM_State.
*/

PROCESS RLP_KERNEL

13 (15)

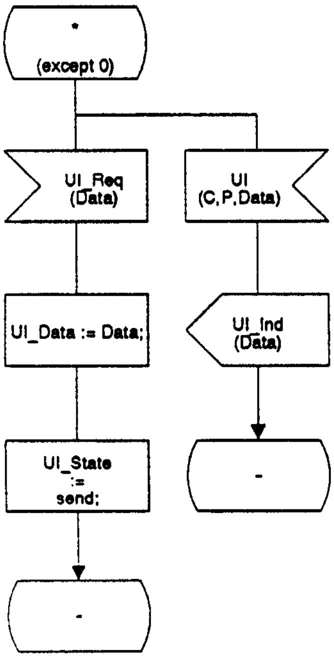
/*
UI handling (UI_Req, UI)

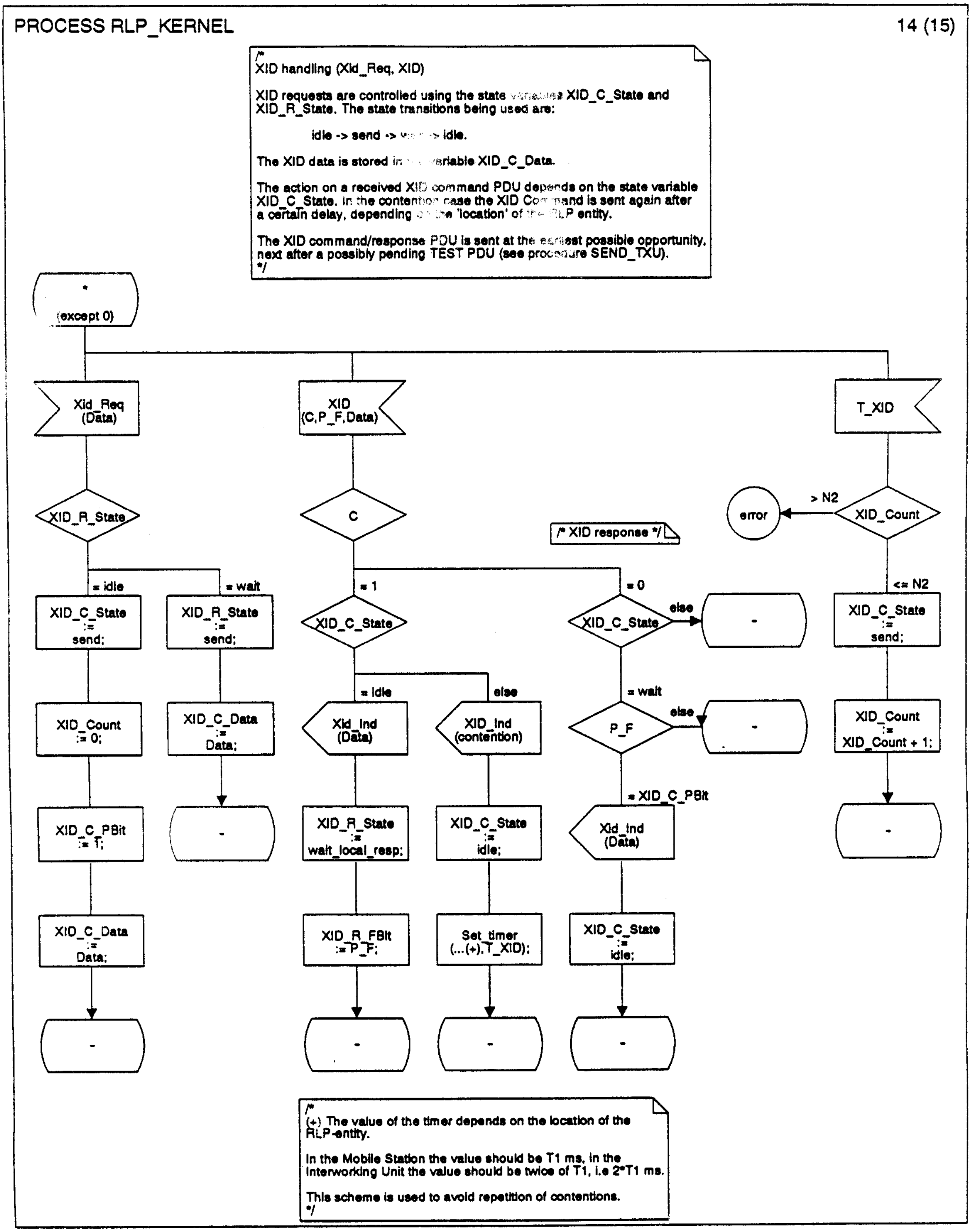
UI_Requests are controlled using the state variable UI_State. The values (state transitions) are: idle -> send -> idle.

It is assumed, that the upper layer entity issues an UI_Req only if the RLP entity's UI_State is idle.

The UI data is stored in the variable UI_Data.

The UI PDU is generated at the next possible opportunity, i.e. after the higher priviledged PDUs (TEST PDUs, iff any) have been transmitted.
*/





/*
XID handling (Xid_Req, XID)

XID requests are controlled using the state variables XID_C_State and XID_R_State. The state transitions being used are:

idle -> send -> wait -> idle.

The XID data is stored in the variable XID_C_Data.

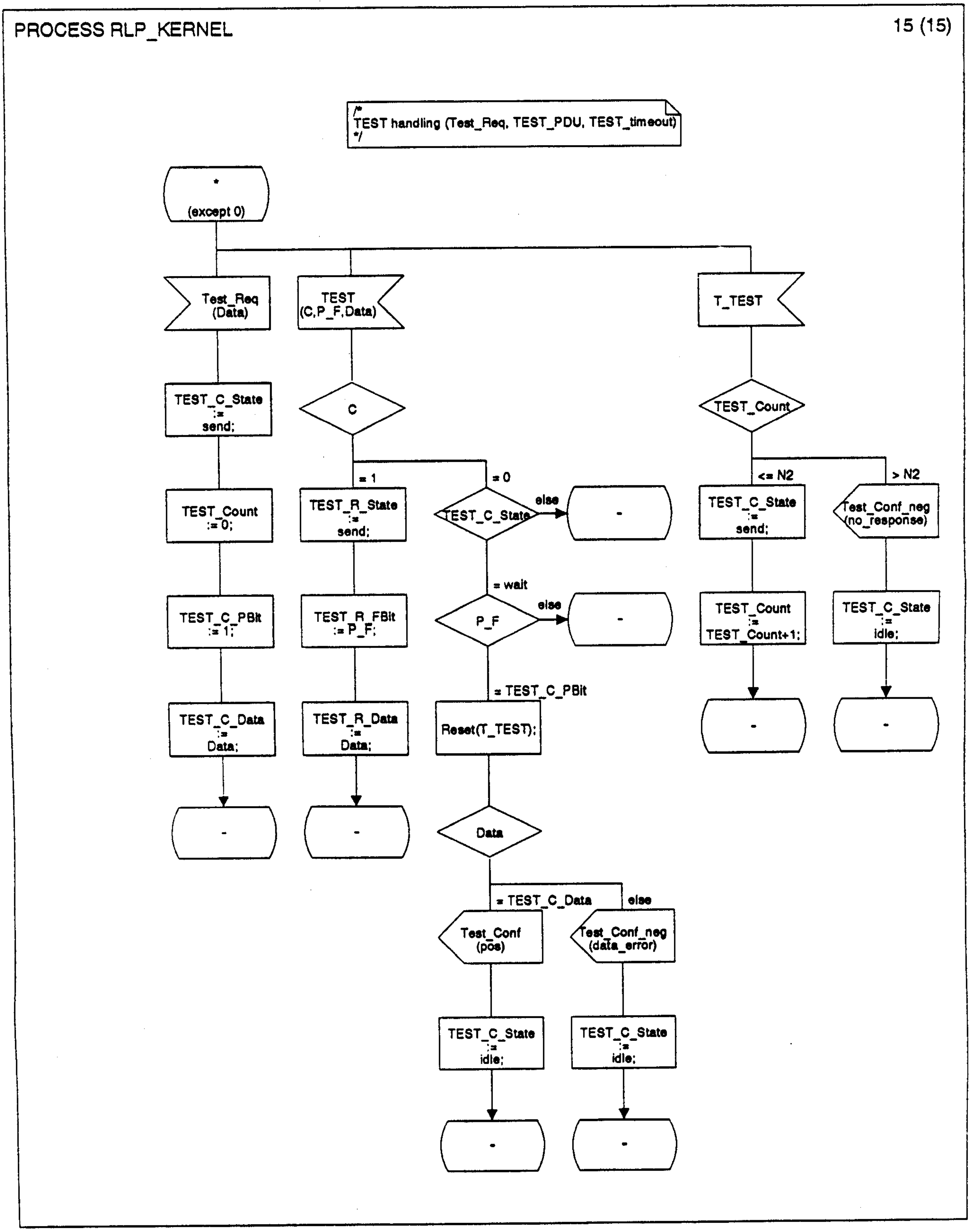
The action on a received XID command PDU depends on the state variable XID_C_State. In the contention case the XID Command is sent again after a certain delay, depending on the 'location' of the RLP entity.

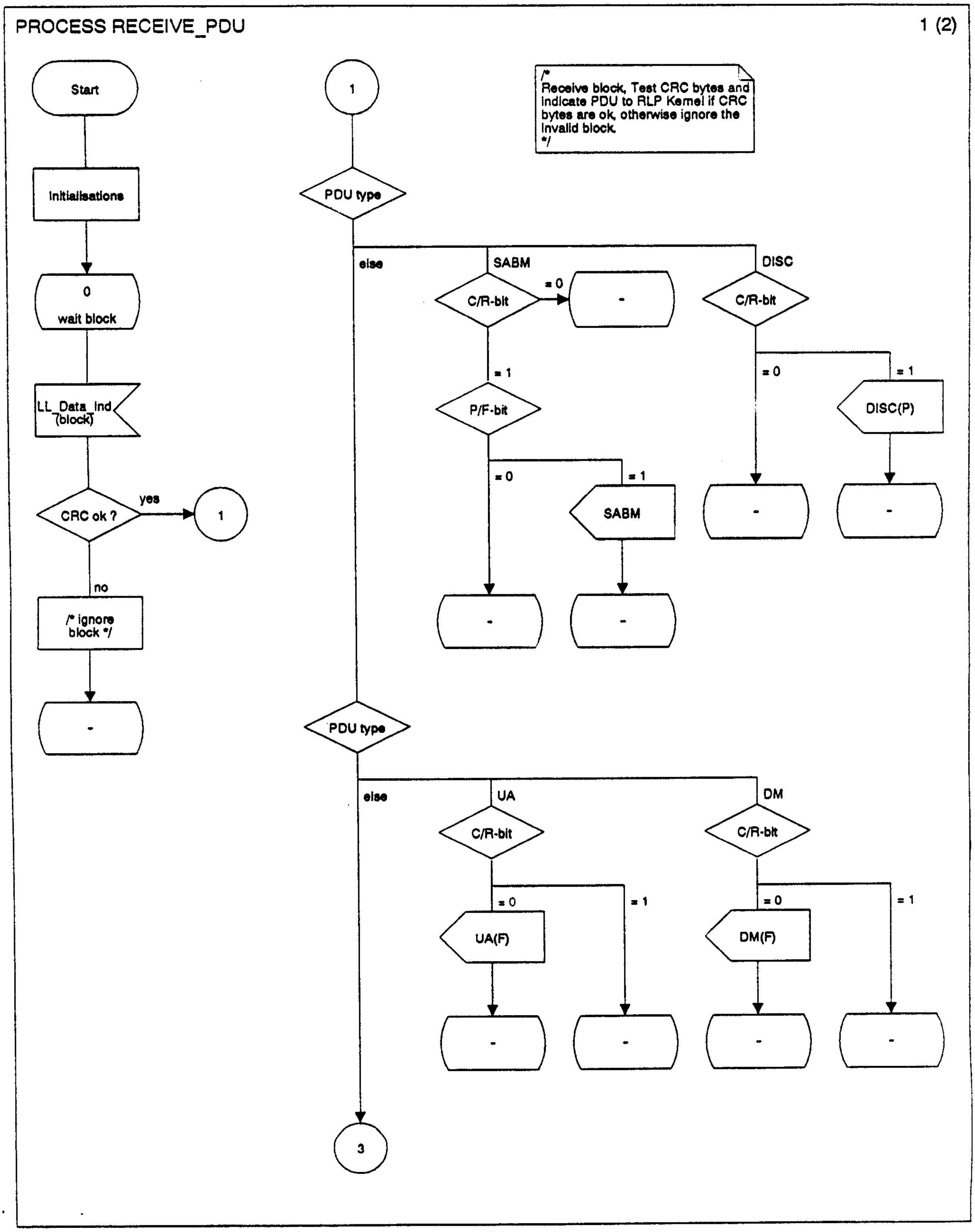
The XID command/response PDU is sent at the earliest possible opportunity, next after a possibly pending TEST PDU (see procedure SEND_TXU).
*/

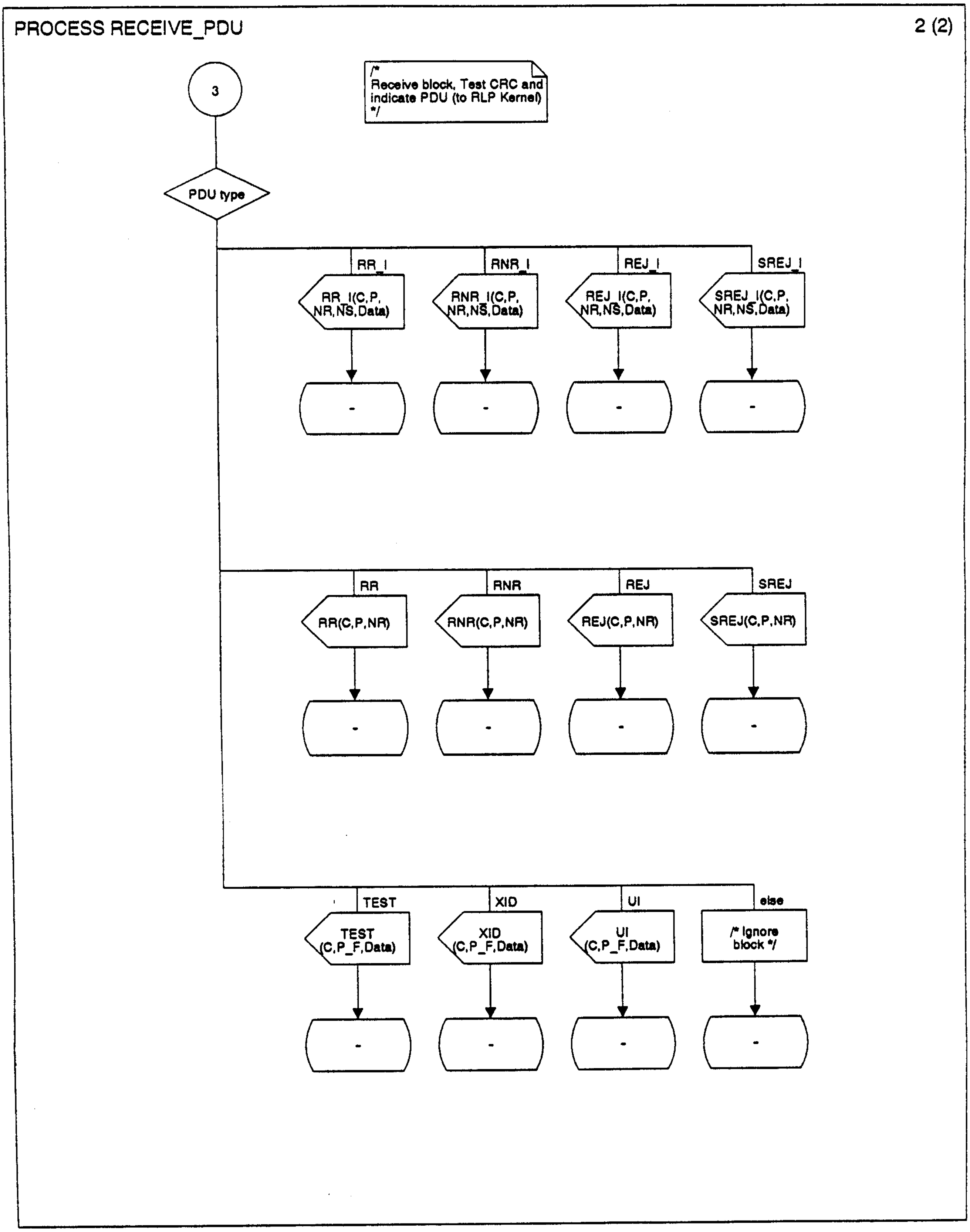
/*
(+) The value of the timer depends on the location of the RLP-entity.

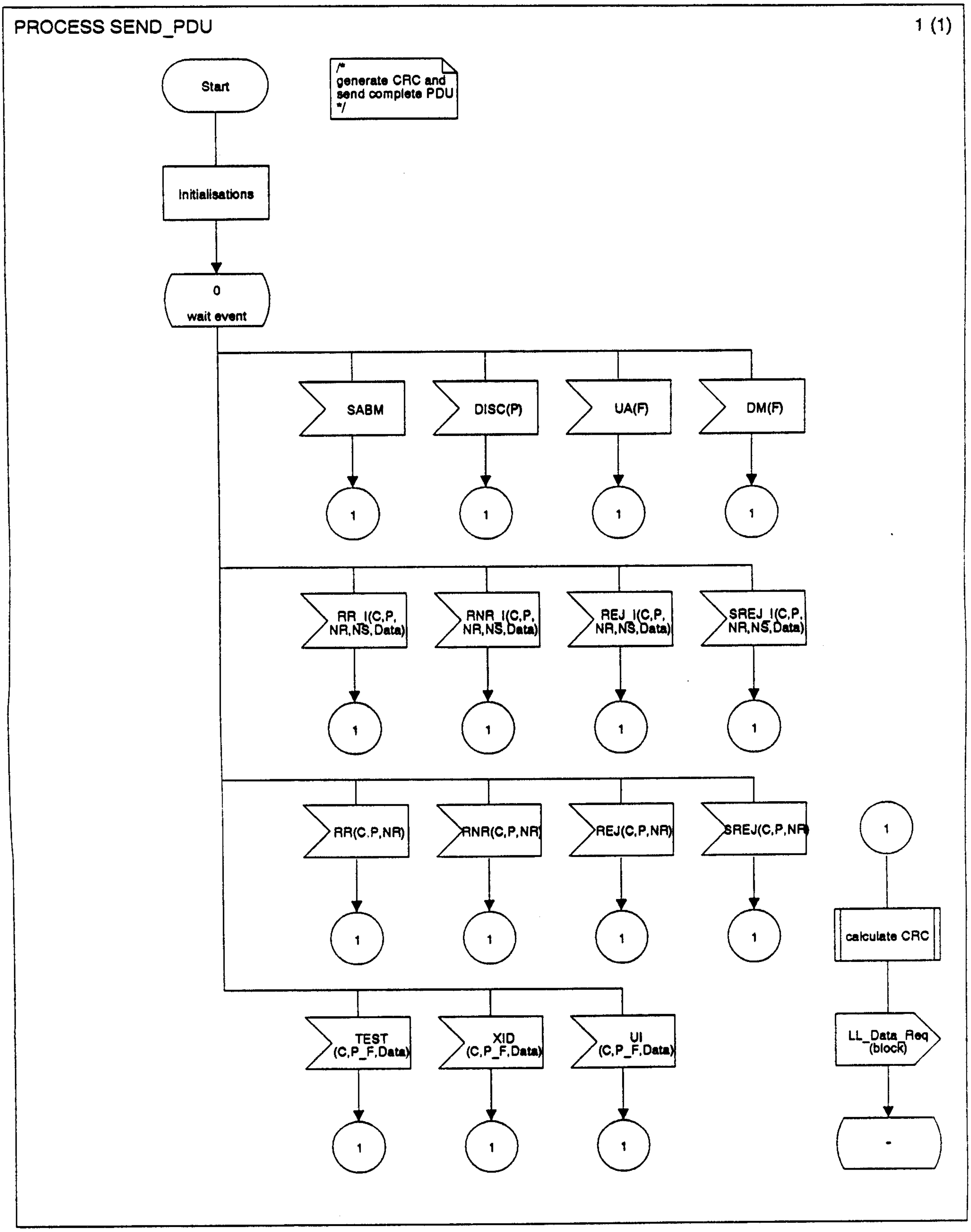
In the Mobile Station the value should be T1 ms, in the Interworking Unit the value should be twice of T1, i.e 2*T1 ms.

This scheme is used to avoid repetition of contentions.
*/

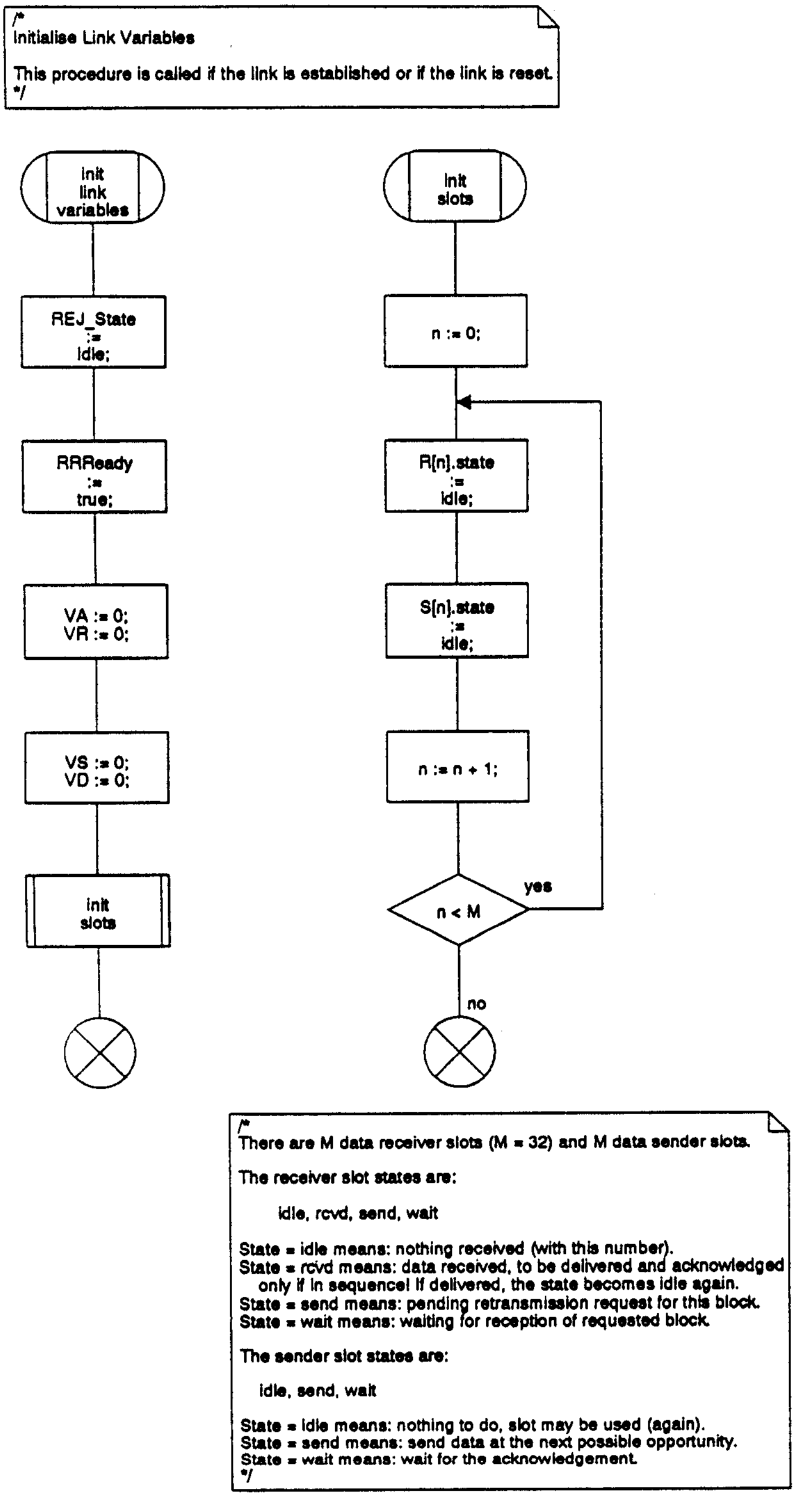

PROCESS RLP_KERNEL
15 (15)
/*
TEST handling (Test_Req, TEST_PDU, TEST_timeout)
*/
*
(except 0)
Test_Req
(Data)
TEST_C_State
:=
send;
TEST_Count
:= 0;
TEST_C_PBit
:= 1;
TEST_C_Data
:=
Data;
-
TEST
(C,P_F,Data)
C
= 1
TEST_R_State
:=
send;
TEST_R_FBit
:= P_F;
TEST_R_Data
:=
Data;
-
= 0
TEST_C_State
else
-
= wait
P_F
else
-
= TEST_C_PBit
Reset(T_TEST);
Data
= TEST_C_Data
Test_Conf
(pos)
TEST_C_State
:=
idle;
-
else
Test_Conf_neg
(data_error)
TEST_C_State
:=
idle;
-
T_TEST
TEST_Count
<= N2
TEST_C_State
:=
send;
TEST_Count
:=
TEST_Count+1;
-
> N2
Test_Conf_neg
(no_response)
TEST_C_State
:=
idle;
-


PROCESS RECEIVE_PDU
1 (2)
Start
Initialisations
0
wait block
LL_Data_ind
(block)
CRC ok ?
yes
1
no
/* ignore block */
-
1
PDU type
/*
Receive block, Test CRC bytes and indicate PDU to RLP Kernel if CRC bytes are ok, otherwise ignore the invalid block.
*/
else
SABM
C/R-bit
= 0
-
= 1
P/F-bit
= 0
= 1
SABM
-
-
DISC
C/R-bit
= 0
= 1
DISC(P)
-
-
PDU type
else
UA
C/R-bit
= 0
= 1
UA(F)
-
-
DM
C/R-bit
= 0
= 1
DM(F)
-
-
3


PROCESS RECEIVE_PDU
2 (2)
3
/* Receive block, Test CRC and indicate PDU (to RLP Kernel) */
PDU type
RR_I
RR_I(C,P, NR,NS,Data)
-
RNR_I
RNR_I(C,P, NR,NS,Data)
-
REJ_I
REJ_I(C,P, NR,NS,Data)
-
SREJ_I
SREJ_I(C,P, NR,NS,Data)
-
RR
RR(C,P,NR)
-
RNR
RNR(C,P,NR)
-
REJ
REJ(C,P,NR)
-
SREJ
SREJ(C,P,NR)
-
TEST
TEST (C,P_F,Data)
-
XID
XID (C,P_F,Data)
-
UI
UI (C,P_F,Data)
-
else
/* Ignore block */
-


PROCESS SEND_PDU
1 (1)
Start
/* generate CRC and send complete PDU */
Initialisations
0
wait event
SABM
DISC(P)
UA(F)
DM(F)
RR_I(C,P, NR,NS,Data)
RNR_I(C,P, NR,NS,Data)
REJ_I(C,P, NR,NS,Data)
SREJ_I(C,P, NR,NS,Data)
RR(C,P,NR)
RNR(C,P,NR)
REJ(C,P,NR)
SREJ(C,P,NR)
TEST (C,P_F,Data)
XID (C,P_F,Data)
UI (C,P_F,Data)
1
calculate CRC
LL_Data_Req (block)
-

PROCEDURE INIT_LINK_VARS 1 (1)



/*
Initialise Link Variables

This procedure is called if the link is established or if the link is reset.
*/

/*
There are M data receiver slots (M = 32) and M data sender slots.

The receiver slot states are:

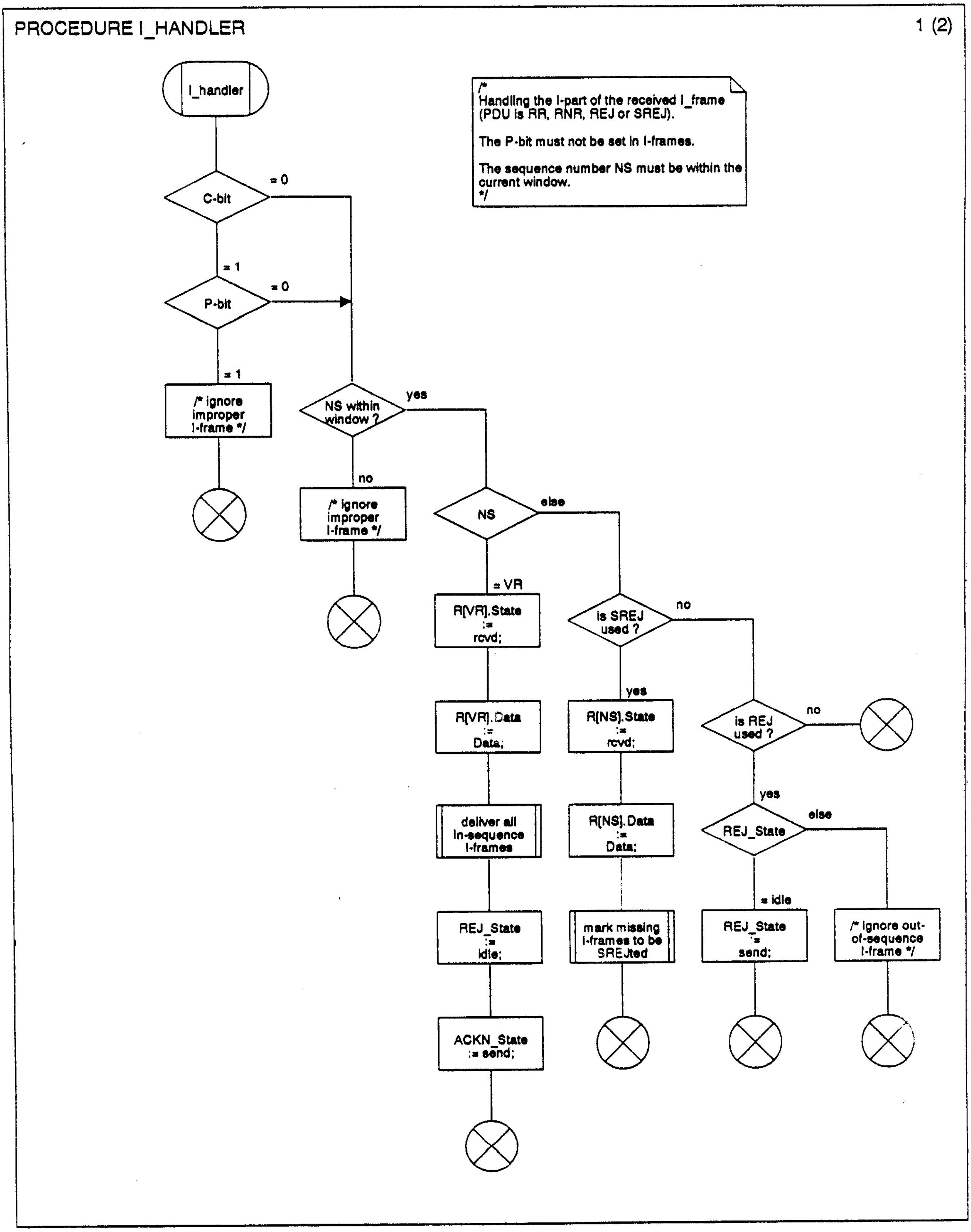
idle, rcvd, send, wait

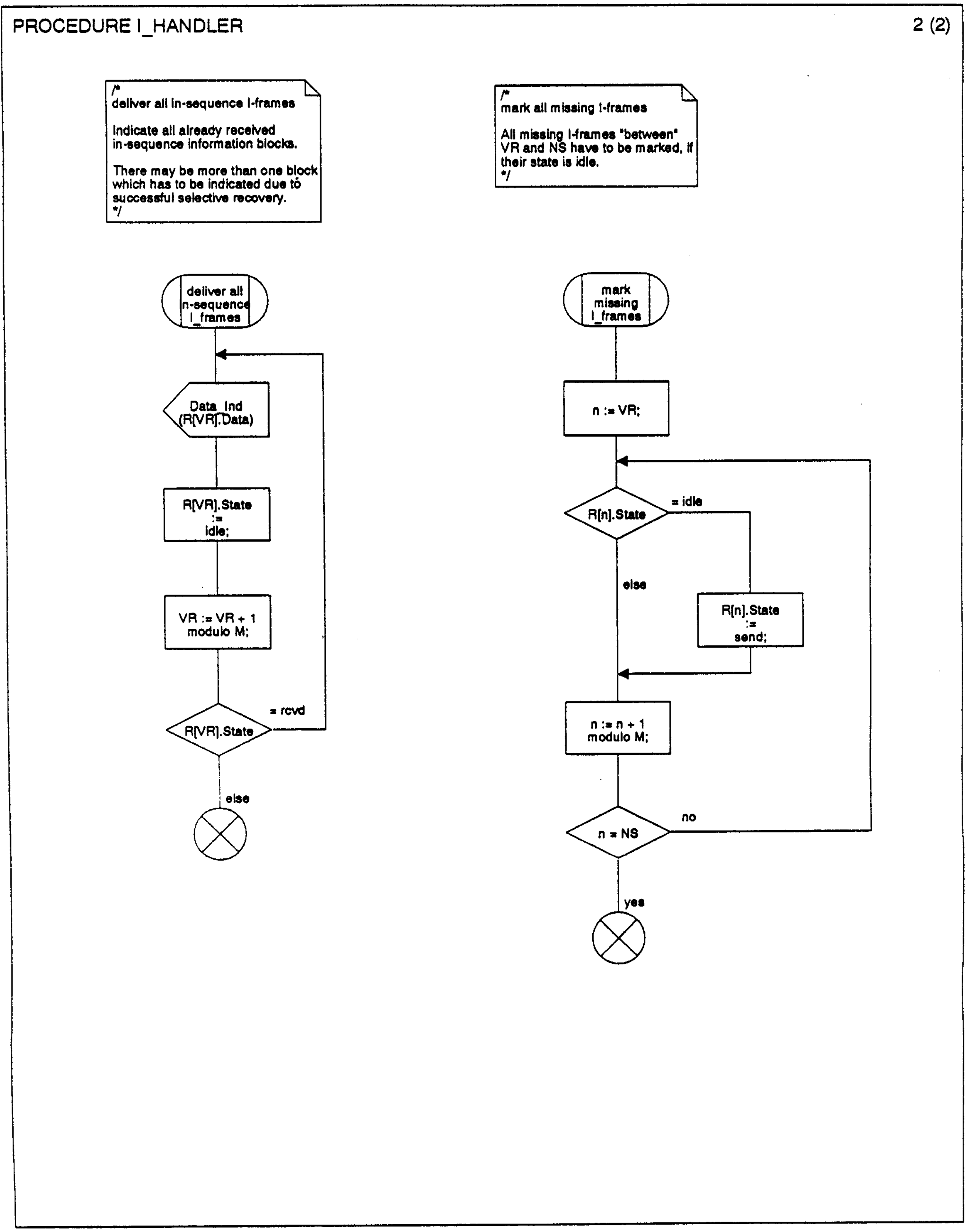
State = idle means: nothing received (with this number).
State = rcvd means: data received, to be delivered and acknowledged only if in sequence! If delivered, the state becomes idle again.
State = send means: pending retransmission request for this block.
State = wait means: waiting for reception of requested block.

The sender slot states are:

idle, send, wait

State = idle means: nothing to do, slot may be used (again).
State = send means: send data at the next possible opportunity.
State = wait means: wait for the acknowledgement.
*/

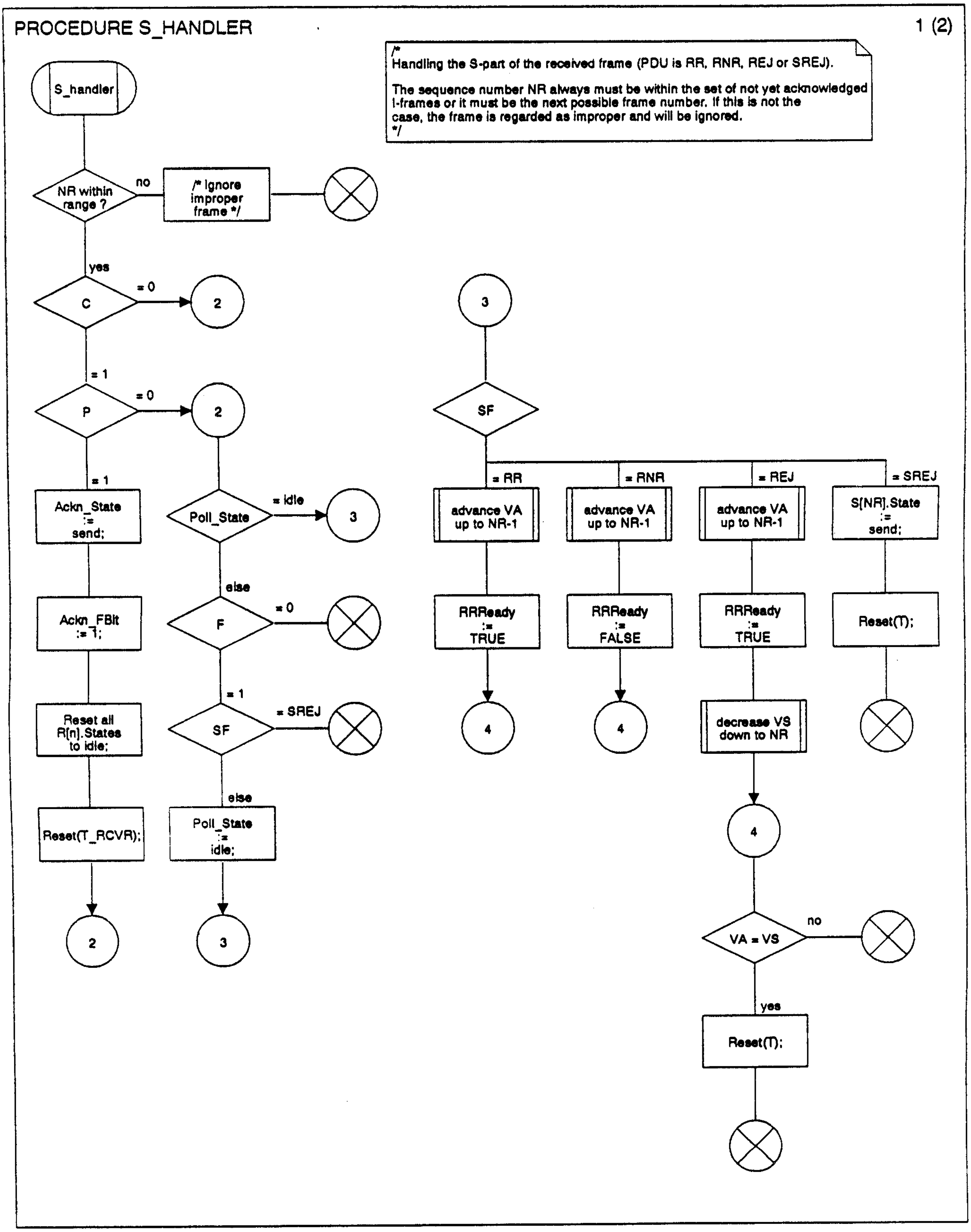

PROCEDURE I_HANDLER
1 (2)
I_handler
/*
Handling the I-part of the received I_frame
(PDU is RR, RNR, REJ or SREJ).
The P-bit must not be set in I-frames.
The sequence number NS must be within the current window.
*/
C-bit
= 0
= 1
P-bit
= 0
= 1
/* ignore improper I-frame */
NS within window ?
yes
no
/* ignore improper I-frame */
NS
else
= VR
R[VR].State := rcvd;
R[VR].Data := Data;
deliver all in-sequence I-frames
REJ_State := idle;
ACKN_State := send;
is SREJ used ?
no
yes
R[NS].State := rcvd;
R[NS].Data := Data;
mark missing I-frames to be SREJted
is REJ used ?
no
yes
REJ_State
else
= idle
REJ_State := send;
/* ignore out-of-sequence I-frame */


PROCEDURE I_HANDLER
2 (2)
/*
deliver all in-sequence I-frames

Indicate all already received in-sequence information blocks.

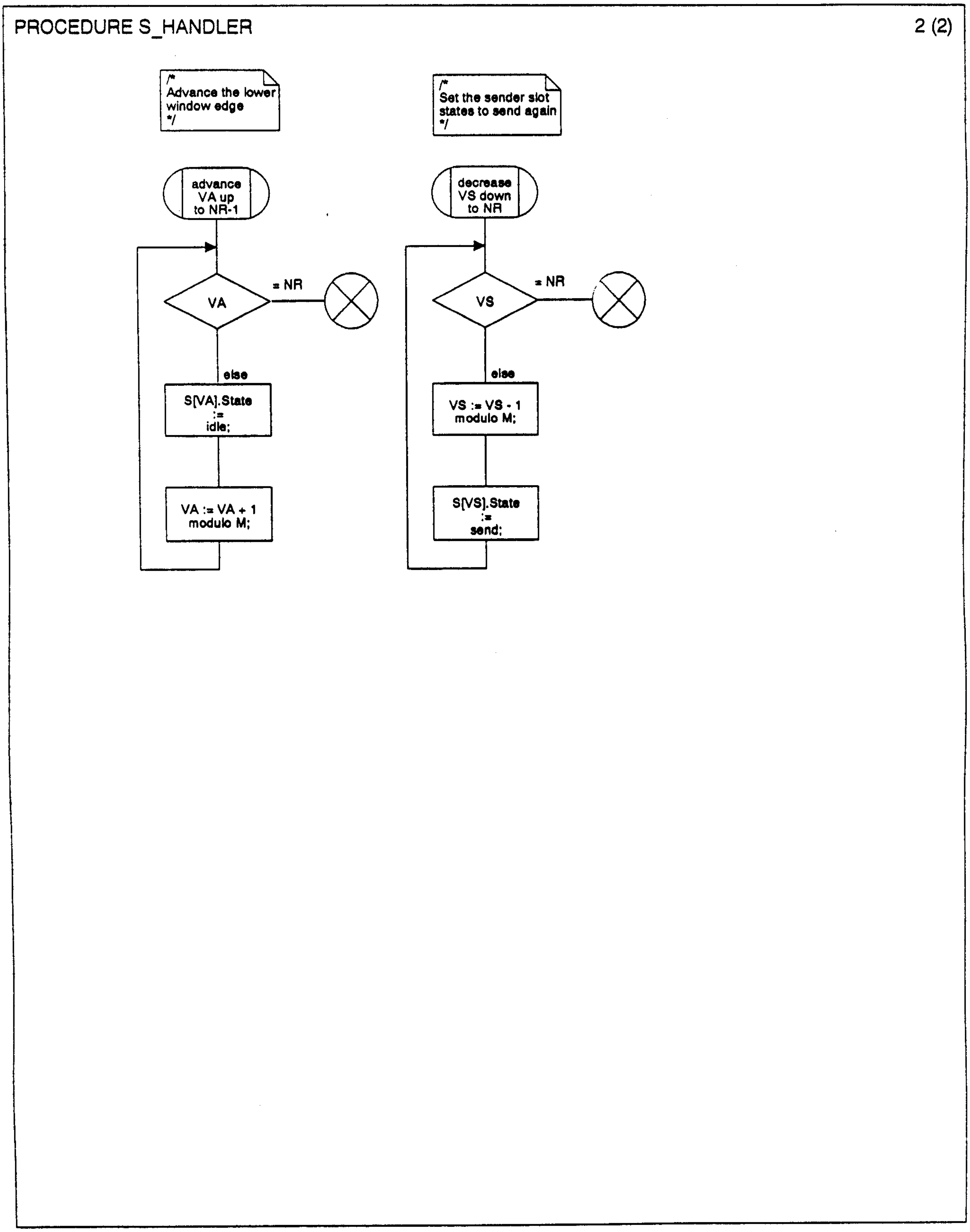
There may be more than one block which has to be indicated due to successful selective recovery.
*/
/*
mark all missing I-frames

All missing I-frames "between" VR and NS have to be marked, if their state is idle.
*/
deliver all in-sequence I_frames
Data_Ind (R[VR].Data)
R[VR].State := idle;
VR := VR + 1 modulo M;
R[VR].State
= rcvd
else
mark missing I_frames
n := VR;
R[n].State
= idle
else
R[n].State := send;
n := n + 1 modulo M;
n = NS
no
yes



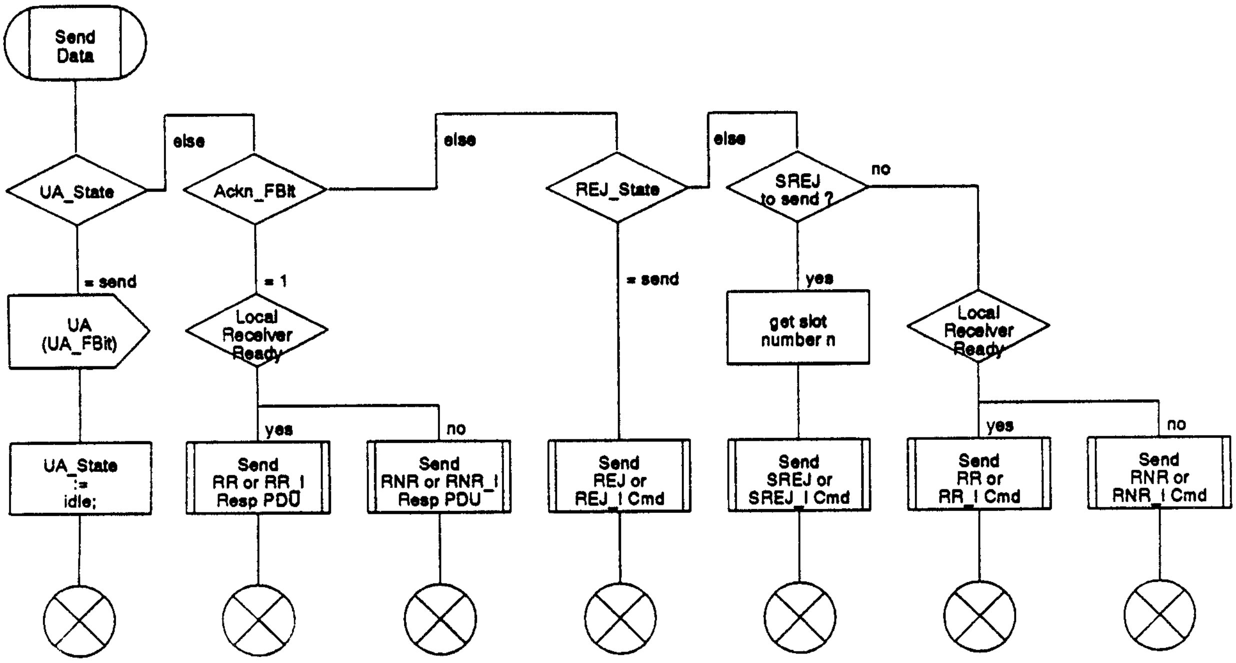
/*
Handling the S-part of the received frame (PDU is RR, RNR, REJ or SREJ).

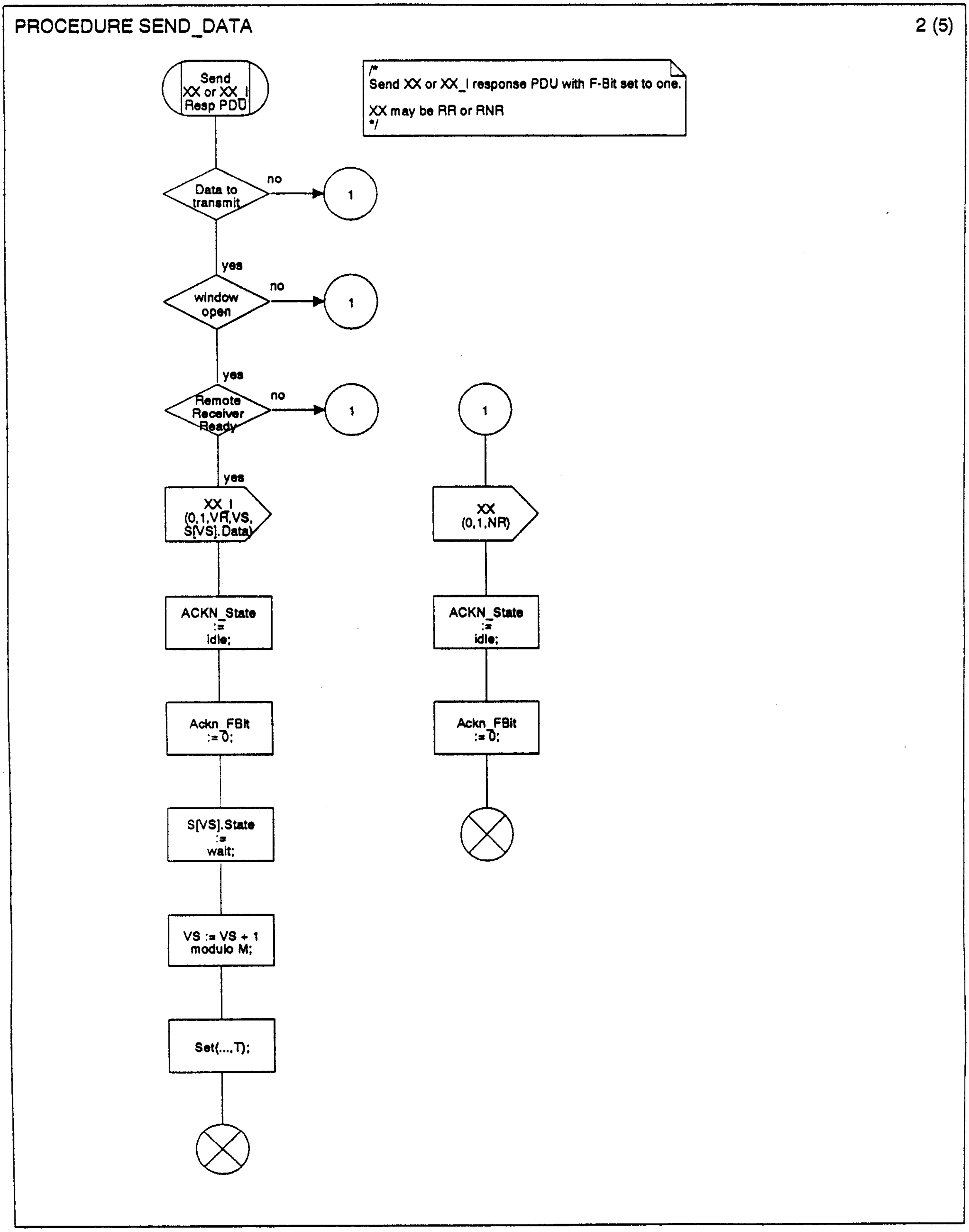
The sequence number NR always must be within the set of not yet acknowledged I-frames or it must be the next possible frame number. If this is not the case, the frame is regarded as improper and will be ignored.
*/

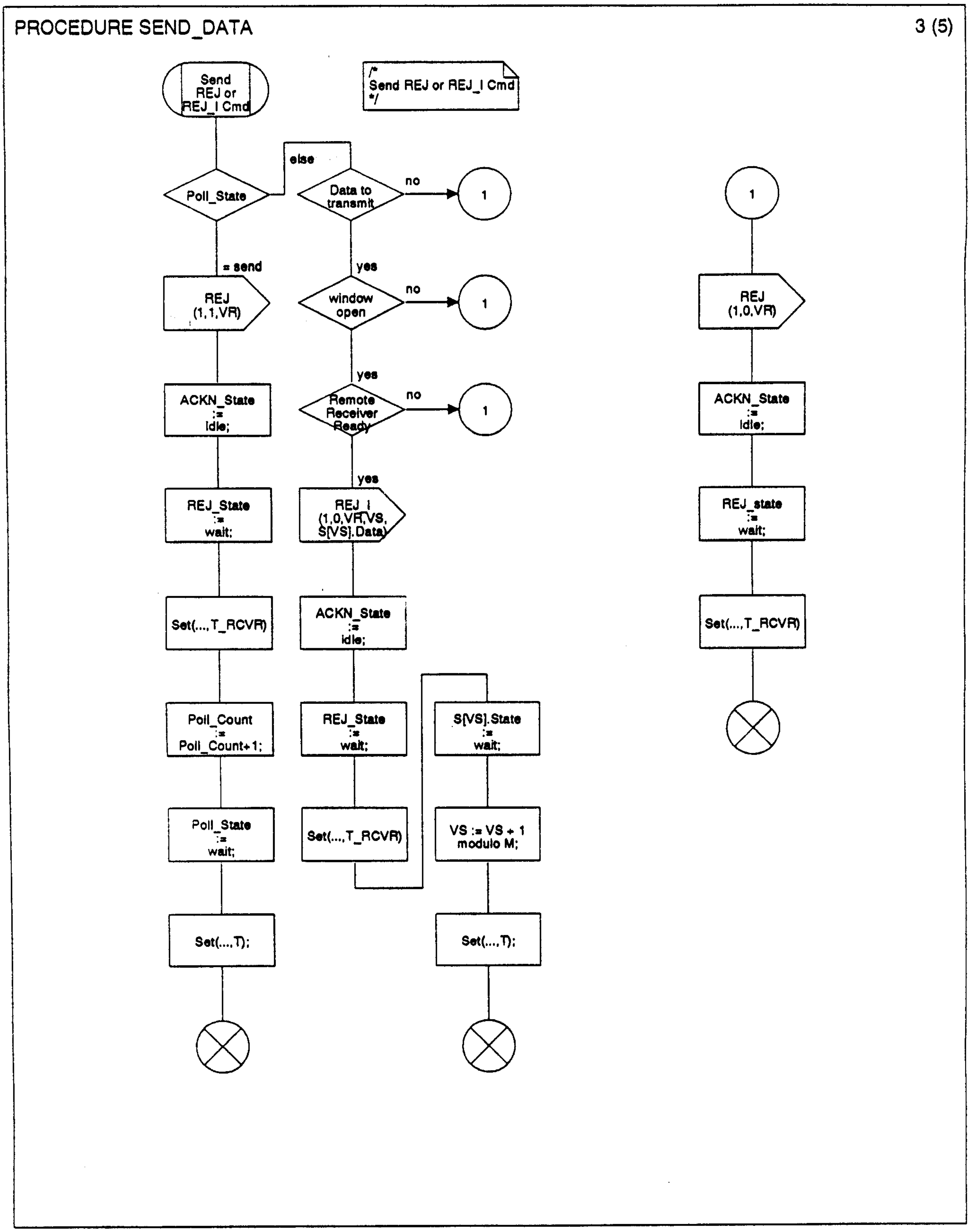

PROCEDURE S_HANDLER
2 (2)
/*
Advance the lower window edge
*/
advance VA up to NR-1
VA
= NR
else
S[VA].State := idle;
VA := VA + 1 modulo M;
/*
Set the sender slot states to send again
*/
decrease VS down to NR
VS
= NR
else
VS := VS - 1 modulo M;
S[VS].State := send;

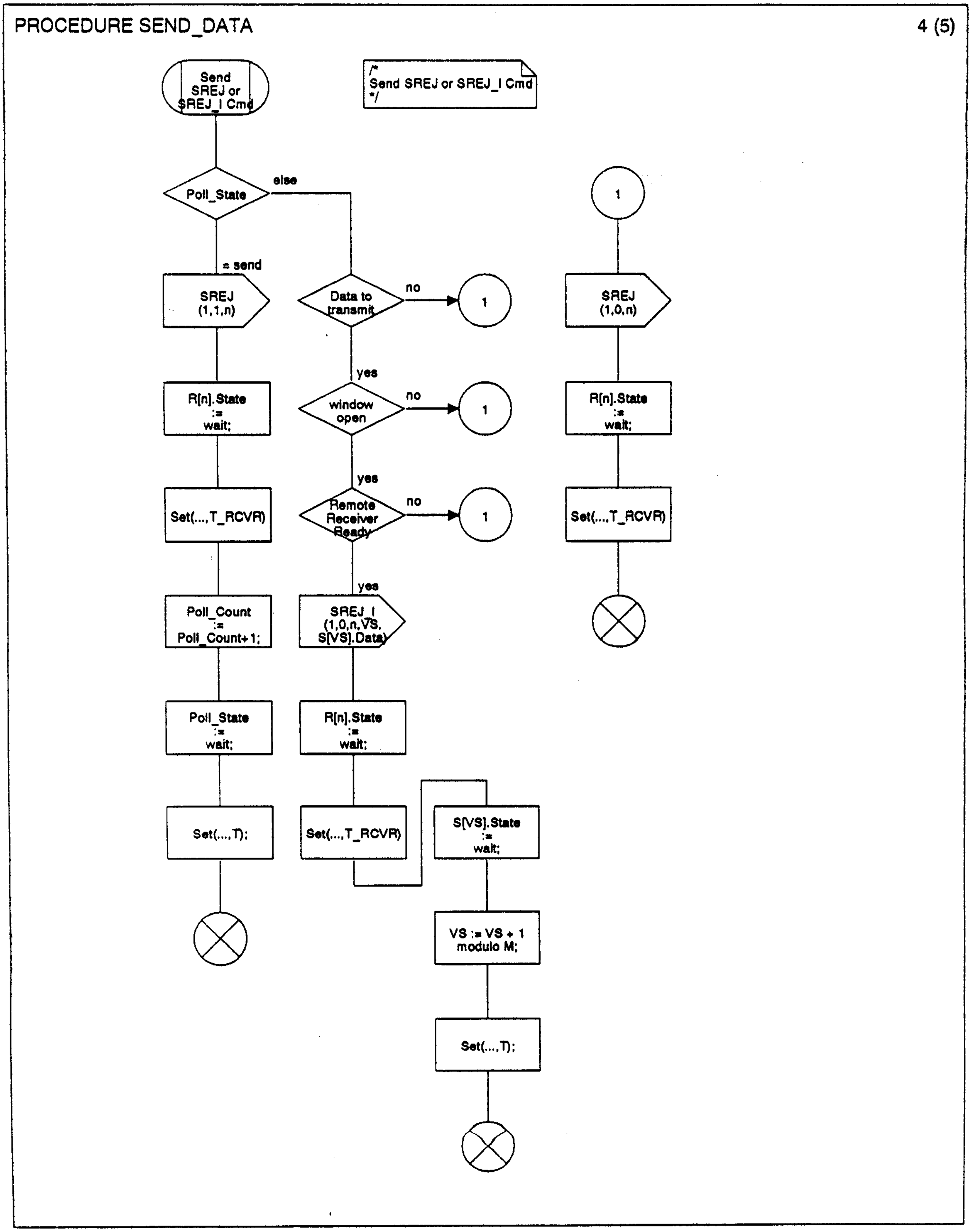
PROCEDURE SEND_DATA

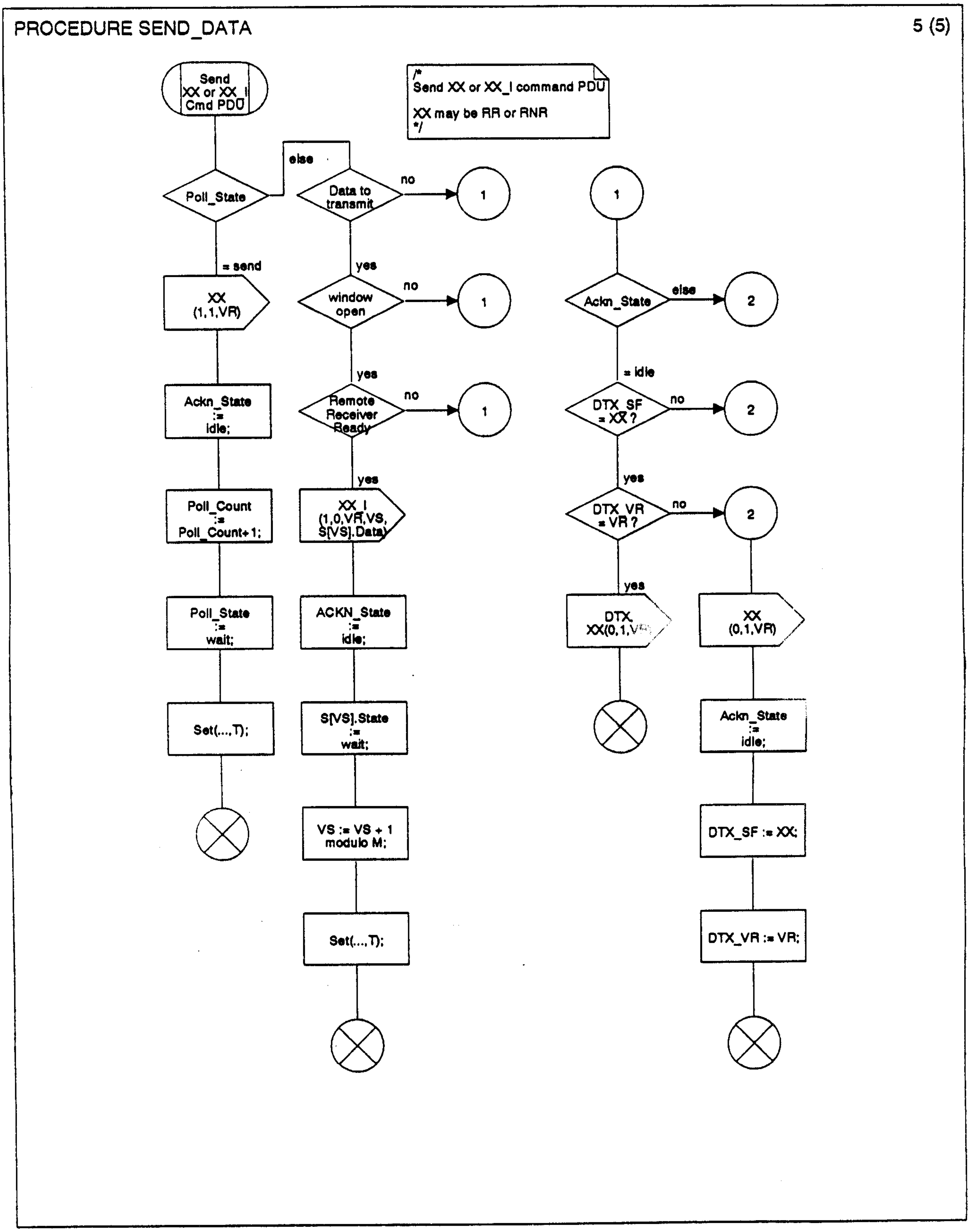
1 (5)

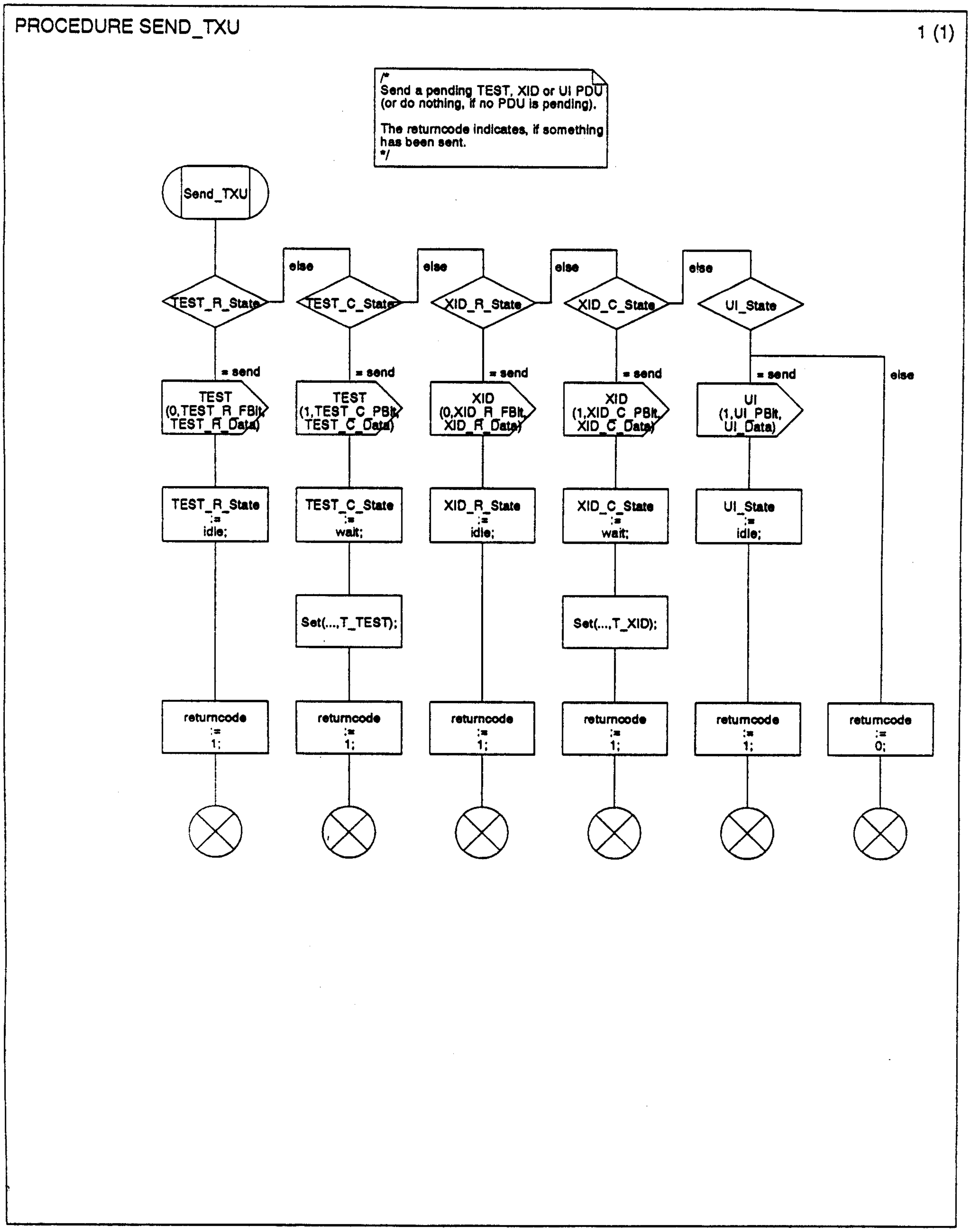



PROCEDURE SEND_DATA
2 (5)
Send XX or XX_I Resp PDU
/* Send XX or XX_I response PDU with F-Bit set to one.
XX may be RR or RNR */
Data to transmit
no
1
yes
window open
no
1
yes
Remote Receiver Ready
no
1
yes
1
XX_I (0,1,VR,VS, S[VS].Data)
XX (0,1,NR)
ACKN_State := idle;
ACKN_State := idle;
Ackn_FBit := 0;
Ackn_FBit := 0;
S[VS].State := wait;
VS := VS + 1 modulo M;
Set(...,T);


PROCEDURE SEND_DATA
3 (5)
Send REJ or REJ_I Cmd
/* Send REJ or REJ_I Cmd */
Poll_State
else
Data to transmit
no
1
= send
yes
REJ (1,1,VR)
window open
no
1
yes
ACKN_State := idle;
Remote Receiver Ready
no
1
yes
REJ_State := wait;
REJ_I (1,0,VR,VS, S[VS].Data)
Set(...,T_RCVR)
ACKN_State := idle;
Poll_Count := Poll_Count+1;
REJ_State := wait;
S[VS].State := wait;
Poll_State := wait;
Set(...,T_RCVR)
VS := VS + 1 modulo M;
Set(...,T);
Set(...,T);
1
REJ (1,0,VR)
ACKN_State := idle;
REJ_state := wait;
Set(...,T_RCVR)


PROCEDURE SEND_DATA
4 (5)
Send SREJ or SREJ_I Cmd
/* Send SREJ or SREJ_I Cmd */
Poll_State
else
= send
SREJ (1,1,n)
R[n].State := wait;
Set(...,T_RCVR)
Poll_Count := Poll_Count+1;
Poll_State := wait;
Set(...,T);
Data to transmit
no
1
yes
window open
no
1
yes
Remote Receiver Ready
no
1
yes
SREJ_I (1,0,n,VS, S[VS].Data)
R[n].State := wait;
Set(...,T_RCVR)
S[VS].State := wait;
VS := VS + 1 modulo M;
Set(...,T);
1
SREJ (1,0,n)
R[n].State := wait;
Set(...,T_RCVR)


PROCEDURE SEND_DATA
5 (5)
Send XX or XX_I Cmd PDU
/* Send XX or XX_I command PDU
XX may be RR or RNR */
Poll_State
else
= send
Data to transmit
no
1
yes
window open
no
1
yes
Remote Receiver Ready
no
1
yes
XX (1,1,VR)
Ackn_State := idle;
Poll_Count := Poll_Count+1;
Poll_State := wait;
Set(...,T);
XX_I (1,0,VR,VS, S[VS].Data)
ACKN_State := idle;
S[VS].State := wait;
VS := VS + 1 modulo M;
Set(...,T);
1
Ackn_State
else
2
= idle
DTX_SF = XX ?
no
2
yes
DTX_VR = VR ?
no
2
yes
DTX XX(0,1,VR)
2
XX (0,1,VR)
Ackn_State := idle;
DTX_SF := XX;
DTX_VR := VR;


PROCEDURE SEND_TXU
1 (1)
/*
Send a pending TEST, XID or UI PDU
(or do nothing, if no PDU is pending).
The returncode indicates, if something
has been sent.
*/
Send_TXU
TEST_R_State
else
TEST_C_State
else
XID_R_State
else
XID_C_State
else
UI_State
= send
TEST (0,TEST_R_FBit, TEST_R_Data)
TEST_R_State := idle;
returncode := 1;
= send
TEST (1,TEST_C_PBit, TEST_C_Data)
TEST_C_State := wait;
Set(...,T_TEST);
returncode := 1;
= send
XID (0,XID_R_FBit, XID_R_Data)
XID_R_State := idle;
returncode := 1;
= send
XID (1,XID_C_PBit, XID_C_Data)
XID_C_State := wait;
Set(...,T_XID);
returncode := 1;
= send
UI (1,UI_PBit, UI_Data)
UI_State := idle;
returncode := 1;
else
returncode := 0;

Document history		
July 1990	Public Enquiry	PE 12: 1990-07-09 to 1990-11-30
May 1991	Vote	V 6: 1991-03-11 to 1991-03-15
February 1992	Vote	V 18: 1992-02-17 to 1992-04-10