Supplement to 7th Computer Networking Conference Proceedings

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

Attached are three additional documents for the 7th Computer Networking Conference:

- AX.25 Data Link State Machine -- SDL Revisions & Corrections -- 1988 Sep 30
- Parameter Negotiation between Consenting AX.25 Stations
- Segmenter State Machine

AX.25 Data Link State Machine

SDL Revisions & Editorial Corrections 1988 Sep 30

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

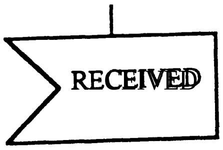
Attached are revisions and editorial corrections to the SDL diagrams which appeared in the 7th Networking Conference proceedings. You should find the following four changed pages:

- summary page
- Connected State, state 3 -- page 2 of 2
- Timer Recovery State, state 4 -- page 1 of 2
- Subroutines

DATA LINK

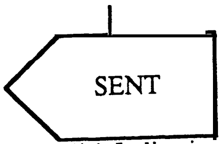
Summary of Primitives, States, Queues, Flags, Parameters, Errors, and Timers

DL Primitives



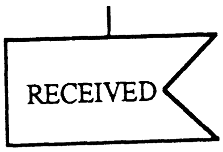
DL Establish Request

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



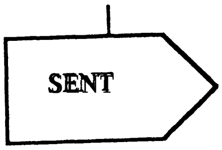
DL Establish Indication
DL Establish Confirm
DL Release Indication
DL Data Indication
DL Unit Data Indication
DL Error Indication

LM Primitives



LM Seize Confirm

DM
UA
SABM
DISC
RR
RNR
REJ
UI
FRMR
I



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

States

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

Queues

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

Timers

T1 -- outstanding I-frame or P-bit.
T3 -- idle supervision (keep alive).

Error Codes

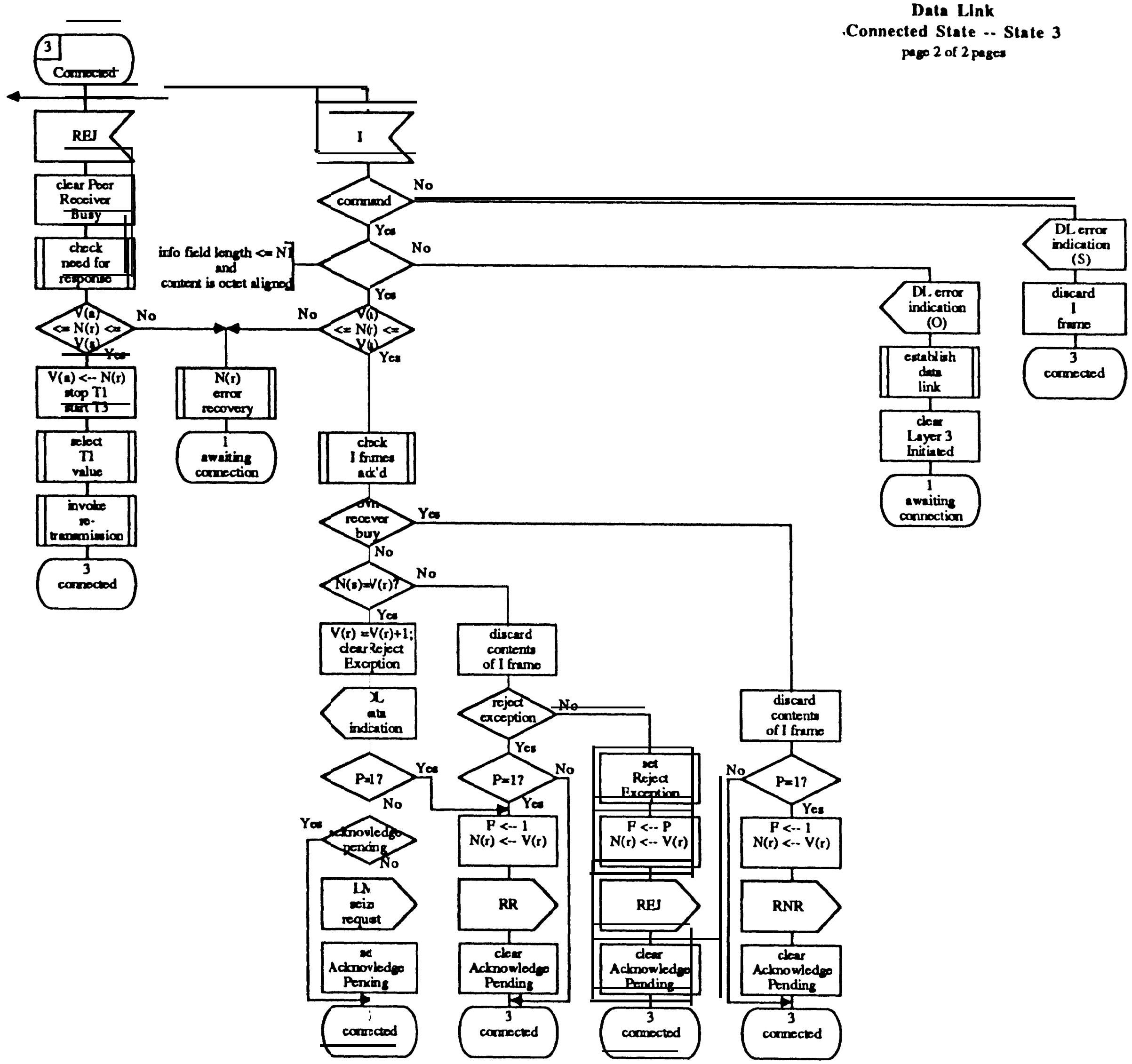
A -- F=1 received but P=1 not outstanding.
B -- Unexpected DM with F=1 in states 3, 4, and 5.
C -- Unexpected UA in states 3, 4, and 5.
D -- UA received without F=1 when SABM or DISC was sent with P=1.
E -- DM received in states 3, 4, or 5.
F -- Data link reset; i.e., SABM received in state 3 or 4.
I -- 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.

Flags & Parameters

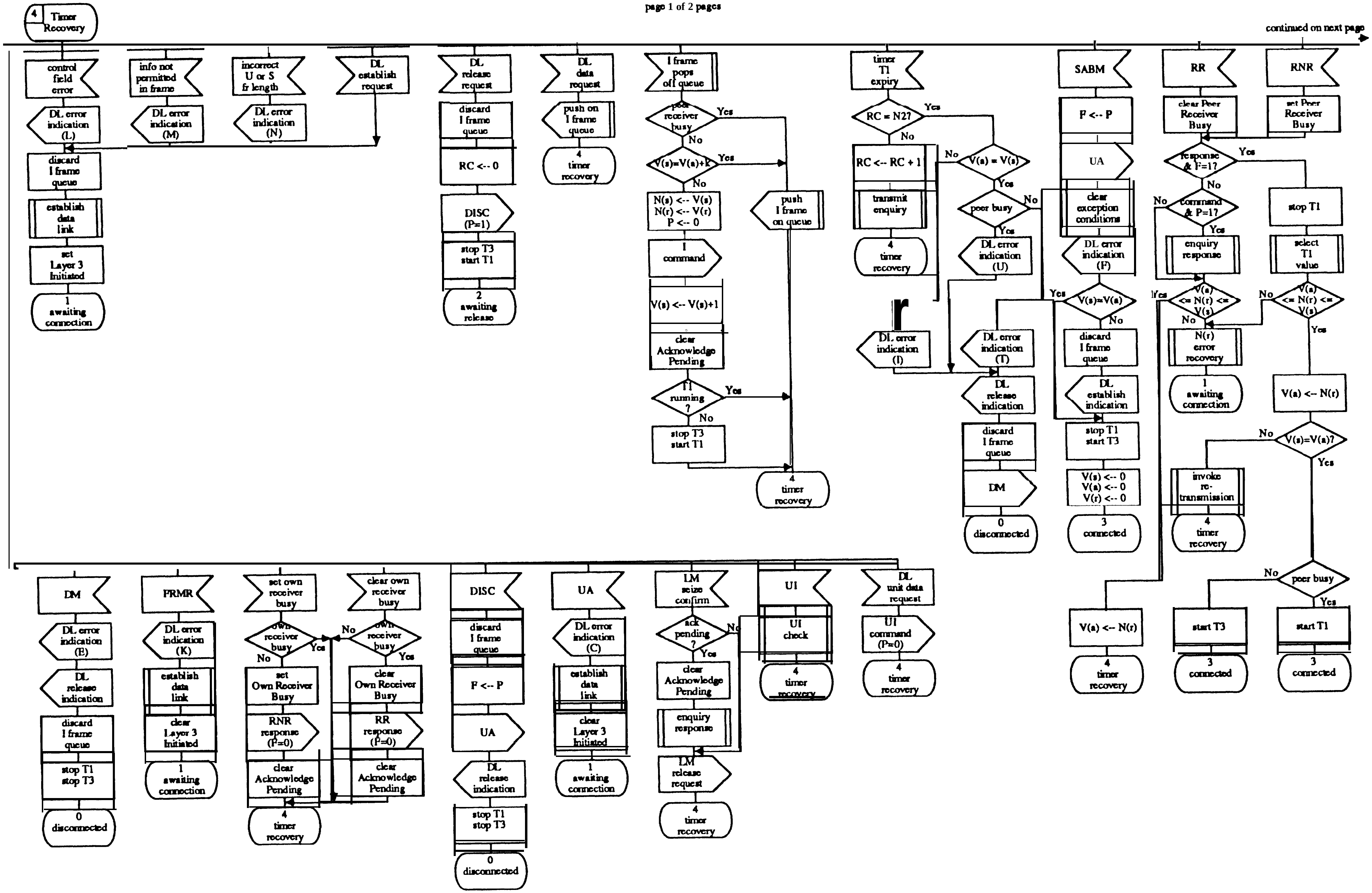
Layer 3 Initiated -- SABM was sent by request of Layer 3; i.e., DL-Establish-Request primitive.
Peer Receiver Busy -- remote station is busy and can not receive I-frames.
Own Receiver Busy -- Layer 3 is busy and can not receive I-frames.
Reject Exception -- a REJect frame has been sent to the remote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

rev 88 Sep 30 K3NA

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

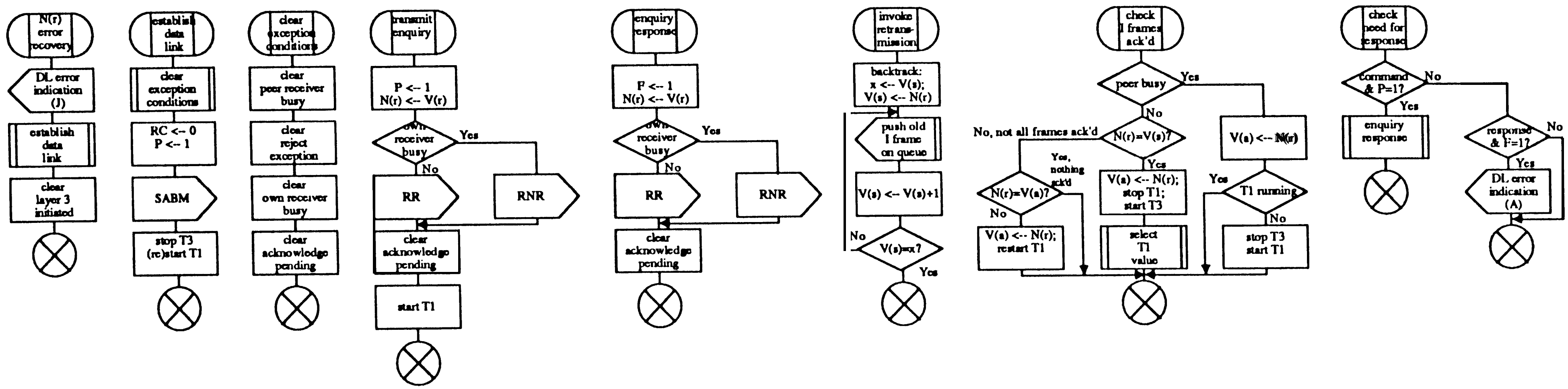


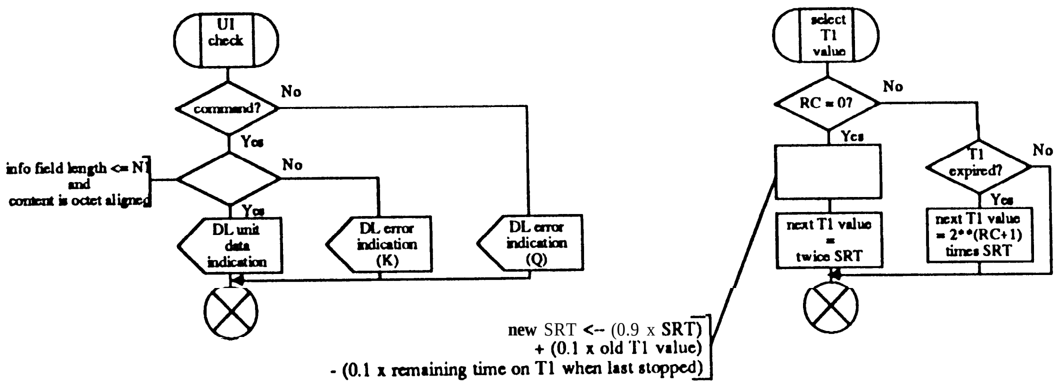
Data Link
Timer Recovery State -- State 4



rev 8% Sep 30 K3NA

Data Link Subroutines





Parameter Negotiation between Consenting AX.25 Stations

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

0. Summary

This paper is part of a series of papers which introduce enhancements and other improvements in AX.25. This particular paper provides a mechanism for the negotiation of connection parameters on an AX.25 link between two consenting stations and the affected digipeaters. The negotiation mechanism is based on existing provisions for link parameter notification and negotiation which are found in a number of international telecommunications standards (ISO and CCITT). These procedures are backwards compatible with existing AX.25 Revision 2.0 implementations; and may also be expanded to include the notification or negotiation of other aspects of link operation.

1. Status of Proposal

The ARRL Digital Committee specifically directed me to develop a proposal for parameter negotiation mechanisms to provide for automatic adjustment of the following aspects of an AX.25 link between two stations:

a) maximum frame size (N1) and window size.
b) round trip timers.
c) physical transmission speed.
d) segmentation procedures.

The following material is still in a draft state. You are invited to review and comment on this material. Comments are desired so that the final publication is as useful as possible to its readers.

2. Notification versus Negotiation

Notification and negotiation are related but distinct concepts.

Notification occurs when one station provides to another some information about its present operating parameters. The operating parameters of the sending station are *not* changed; the information provided is considered advisory in nature. The receiving station may, however, elect to adjust its internal operating parameters based on the contents of the notification.

For example, notification could be used to inform the remote station that, henceforth, all future UI and I frames will contain up to 4096 octets of data in the information field. The remote station may wish to adjust its internal buffer allocation procedures accordingly when such a notice is received.

Negotiation occurs when stations exchange information about operating parameters with the intention to reach a mutually-agreed value. The operating parameters of both stations may be changed as a result of the negotiation. Generally, negotiated parameters have a *default value* towards which negotiation is made.

For example, negotiation could begin with one station indicating that it wishes to send UI and I frames which contain up to 4096 octets of data in the information field. The default value in AX.25 today is 256 octets. The remote station (or an intervening digipeater) may indicate that only 1024 octets can be accomodated, with the final result that both stations settle on the 1024 octet value. Here, negotiation took place from the initial proposed value towards the default value.

The choice of permitting negotiation or notification should be made on a parameter by parameter basis; some parameters are better negotiated, whilst others are more satisfactorily handled by simple notification.

There are also some parameters which *are* best left unnegotiated; for example, it is not appropriate to allow one station to change the callsign stored in the TNC of another station.

3. Outline of Procedure

The proposed procedure utilizes standard XID (Exchange Identification) command and response frames. The XID frame is defined in national and international standards for high-level data link control (HDLC) protocols.

In the architecture of extended finite state machines described in other companion papers, this procedure is executed by a new state machine, the data link management entity. Figure 1 illustrates the relationship between this state machine and other state machines associated with AX.25. Note that, for a pure digipeater, a data link management state machine exists even though no instances of data link state machines or AX.25 users may exist within the pure digipeater.

Notification/negotiation may occur at any time during the life of the link to another station.

Notification/negotiation affect only the parameters used on a particular link between two particular stations. System-wide default parameters within a TNC should not be changed by a remote station through this procedure, nor should a remote station change parameters associated with links to other (third-party) stations.

3.1 Procedure Between Stations Equipped for XID

Notification/negotiation begins with the AX.25 user sending an **MDL Negotiate Request** primitive to the data link management state machine. (MDL stands for Management -- Data Link.) The data link management state machine transmits an XID command frame with the Poll bit set to '1'. Timer TM201 supervises the procedure. The transmitting data link state machine enters state 1, negotiating.

The remote station's data link management state machine processes the contents of the received XID command, and then replies with an XID response frame with the Final bit set to '1'. State 0, ready, is then re-entered. The local AX.25 users are not notified, since a connection to user may not exist at this point in time.

Upon receipt of the XID response the initiating state machine stops timer TM201, delivers an **MDL Negotiate Confirm** primitive to the AX.25 user, and returns to state 0, ready.

If timer TM201 expires, the XID command is retransmitted. NM201 attempts are made before the notification/negotiation procedure is abandoned and an MDL **Error** primitive is delivered to the AX.25 user.

3.2 Processing of XID by Digipeaters

Intervening digipeaters must examine the contents of the XID to determine if the parameters being adjusted are compatible with their capabilities and their responsibilities; for example, under this proposal a digipeater would refuse to permit a change in operating speed since, as a result of making the change, the digipeater becomes unavailable to other stations still using the original speed. This examination is carried out by the data link management state machine of the digipeater.

In order to positively indicate that the XID frame has been examined by each intervening digipeater, the digipeater inserts its callsign into the information field of the XID frame. The destination station which receives the XID *command* must check to see if all intervening digipeaters have performed their compatibility checks. If one or more digipeater callsigns are absent within the information field of the XID frame, then two possibilities exist:

a) the negotiation/notification is invalid, since intervening digipeaters did not consent to or recognize (i.e., not an AX.25 Revision 2.1 implementation) the procedure. Appropriate action to abandon the notification/negotiation procedure is taken by each station.

b) the negotiation/notification procedure is valid, because the parameter being negotiated does not affect digipeater operation.

Whether (a) or (b) should be followed is indicated below for each specific notification/negotiation procedure.

3.3 Backwards Compatibility

If an existing AX.25 Revision 2.0 station receives an XID command, it will either:

a) if in the disconnected state, ignore the frame; or,

b) if in the connected state, transmit an FRMR response.

The station initiating notification/negotiation will detect case (a) through T1 timeout, and can detect case (b) by inspection of the FRMR response. In both cases the MDL Error primitive is returned to the AX.25 user and default parameters are used.

An existing AX.25 Revision 2.0 digipeater will digipeat the XID frames blindly without checking the contents for compatibility. The mechanism for insuring that each digipeater has checked in §

4. Format of XID Frames

The XID frame is a U-type frame, and contains an information field.

4.1 XID Control Field

The XID frame control field contains:

8	7	6	5	4	3	2	1	<-- transmitted bit number
1	0	1	**p/f**	1	1	1	1	

4.2 Organization of the Information Field Within the XID Frame

The XID information field is organized in a modular fashion, following the structuring rules adopted by the CCITT in Recommendation Q.931.

The basic building block is the ***information element*** which contains related information about a particular parameter being notified or negotiated. Information elements may be included in any order within the information field. Only the information elements relevant to the parameters being notified or negotiated are included in a particular XID frame.

Each information element is comprised of three parts:

a) the information element identifier, a single octet whose value indicates which information element follows.

b) the length, a single octet containing a binary number. The binary number represents the length, in octets, of the remaining contents of the information element; i.e., excluding the information element identifier and length fields.

c) the contents, containing one or more fields in an integral number of octets. The exact organization depends on the specific information element, as diagrammed below.

All three parts are transmitted together as an indivisible unit. An information element may be repeated, if appropriate.

Important flexibilities are gained by the inclusion of the length field. The contents of the information element may be variable in length, and can even be expanded over time as a result of subsequent enhancements. Furthermore, a particular implementation need not implement every possible information element. In the event that an unimplemented information element is received (i.e., the value of the information element identifier is not recognized), the receiver can ignore that element and, by checking the length field, skip down to the next information element.

4.3 Digipeater Compatible

This information element is used to inform the XID receiver that the indicated digipeater has examined *the* contents of *the* XID *command* and has determined that it is compatible with the parameter(s) as presently indicated.

All AX.25 Revision 2.1 digipeaters will examine the contents of each information element in any XID command to be digipeated. Those information element which were understood (i.e., the information element identifier was recognized by the digipeater implementation) will be processed according to the specifications in § 5 below. The digipeater shall add a Digipeater Compatibility Information Element, containing the digipeater callsign, to the digipeated XID command. Within the Digipeater Compatibility Information Element, bit flags are set indicating which information elements are implemented by the digipeater. This permits "forward compatibility", whereupon, in the future, additional information elements can be included without:

- affecting previous digipeater implementations
- confusing the remote station on the link as to whether all information elements were understood by all intervening equipments.

Existing AX.25 Revision 2.0 digipeaters will digipeat ***the MD*** command ***without*** including a Digipeater Compatible Information Element containing its callsign, since they do not presently understand this frame.

The Digipeater Compatible Information Element is not included in XID *responses*.

The format of the Digipeater Compatible Information Element is shown in Figure 2.

4.4 N1

This information element is used to negotiate the value of data link parameter N1, maximum information field length. The format of the N1 Information Element is shown in Figure 3.

4.5 Window Size

This information element is used to negotiate the value of the data link parameter k, window size. The format of the Window Size Information Element is shown in Figure 4.

4.6 Initial Round Trip Time

This information element is used to notify the remote station of the initial value to be used in calculating round trip time. The format of the Initial Round Trip Time Information Element is shown in Figure 5.

4.7 Transmission Speed

This information element is used to negotiate the transmission speed to be used on the link. The format of the Transmission Speed Information Element is shown in Figure 6.

4.8 Average Information Transfer Rate

This information element is used to notify the remote station of the average information transfer rate which is expected to occur on the connection. The format of the Average Information Transfer Rate Information Element is shown in Figure 7.

4.9 Changeover

This information element is used to coordinate the synchronized change from one data link operating condition to another. The changeover procedures are general purpose in that they can be employed for a variety for data link changes in the future. However, at the present time only the transmission speed negotiation procedures employ a synchronized changeover. Note that, because of the inability of digipeaters to guarantee end-to-end delivery of XID frames, synchronized changeover is not permitted on links containing digipeaters. The format of the Changeover Information Element is shown in Figure 8.

5. Notification/Negotiation Procedures for Specific Parameters

This section describes how each station evaluates and responds to notification or negotiation of the various data link parameters.

5.1 Segmentation Availability Notification

The segmentation procedures (described in a companion paper) are presently proposed as an inherent part of the AX.25 Revision 2.1 protocols, as are these general procedures for notification/negotiation. Any station which sends an XID frame is, by definition, operating according to the proposed AX.25 Revision 2.1 and therefore can also employ the segmentation procedures.

Therefore, to notify a remote station that the segmentation procedures (and other aspects of AX.25 Revision 2.1) are available for use, the data link management state machine shall cause an XID frame to be transmitted to the remote station.

No information element is required in the XID frame to notify the remote station that segmentation and other AX.25 Revision 2.1 enhancements are available.

Intervening digipeaters are transparent to segmentation procedures. Thus, the absence of one or more intervening digipeater identifications (detected by comparing the Digipeater Compatible Information Elements within the XID frame with the digipeater callsigns in the XID frame's address field) does not preclude the use of segmentation procedures; i.e., § 3.2 case (b) applies.

5.2 N1 Negotiation

Parameter N1 is the maximum number of octets which may be contained in the information field of an I, UI or XID frame; the flags, address fields, frame check sequence, and 0-bits added for transparency are *not* included in calculating this limit. The N1 value applies to both directions of information transfer.

The default value of N1 is 256 octets. N1 negotiation procedures are used when one station wishes to arrange with a remote station to use a larger or smaller value of N1. Permissible values of N1 are powers of 2; e.g., 32, 64, 128, 256, 512, 1024, 2048, and 4096 are all examples of permissible values of N1.

N1 negotiation procedures begin with the transmission of an XID command with the N1 Information Element containing the proposed new value of N1. Note that, at the time of transmission of the XID command, the new value of N1 *is not yet authorized for use.* Any ongoing link operations shall continue to use the present (old) value of N1 for the time being.

Since the combination of N1 and window size affects the buffer requirements of intervening digipeaters, the data link management state machine within each digipeater on the path must evaluate the proposed N1. If the proposed N1 is not acceptable to the digipeater, the digipeater shall replace the proposed value with a new, *smaller,* proposed value. For example, if a value of 4096 octets was proposed, an intervening digipeater may trim down the value to 1024 octets... or even all the way back to 16 octets (which might occur if a two-port digipeater which relayed from an excellent radio path to a poor radio path was involved).

The remote station receiving the XID command first compares the digipeater address fields with the Digipeater

Compatibility Information Elements. If one or more digipeaters failed to include a Digipeater Compatibility Information Element, or if *one* or more digipeaters indicate that the N1 Information Element was *not* evaluated, the station concludes that there are some incompatible digipeaters in the path, and that N1 negotiation can not be completed. The XID response frame is transmitted with the N1 Information Element coded as "default N1 must be used".

If all Digipeater Compatibility Information Elements are present, the station receiving the XID command then evaluates the proposed value for N1 and either accepts it or selects a new smaller value. At this point the final value for N1 has been obtained; this final value is recorded in the N1 Information Element contained in the XID response.

As the XID response works its way back to across the link, intervening digipeaters and the station initiating N1 negotiation shall note the final value chosen for N1 and plan accordingly.

5.3 Window Size Negotiation

Parameter k is *the* maximum number of I frames which may be outstanding (i.e., transmitted but not yet acknowledged) on the data link. The k value is independent for each direction of data transfer on the link.

The default value of k is seven I frames. Window size negotiation procedures are used when one station wishes the remote station to prepare for larger or small quantities of outstanding frames. Any integer in the range from one to seven is permitted.

Note -- The use of window sizes larger than seven requires extended sequence numbering and the SABME command. The ARRL Digital Committee has not yet discussed whether extended sequence numbering should be a part of AX.25 Revision 2.1.

Window size negotiation procedures begin with the transmission of an XID command with the Window Size Information Element containing the proposed new value of k. Note that, at the time of transmission of the XID command, *the* new value of k *is not yet authorized for use. Any* ongoing link operations shall continue to use the present (old) value of k for the time being.

Since the combination of N1 and k affects the buffer requirements of intervening digipeaters, the data link management state machine within each digipeater on the path must evaluate the proposed k. If the proposed k is not acceptable to the digipeater, the digipeater shall replace the proposed value with a new, *smaller,* proposed value. For example, if a value of 6 was proposed, an intervening digipeater may trim down the value to 4 or even all the way to 1.

The remote station receiving the XID command first compares the digipeater address fields with the Digipeater Compatible Information Elements. If one or more digipeaters failed to include a Digipeater Compatible Information Element, or if one or more digipeaters indicated that the Window Size Information Element was *not* evaluated, the station concludes that *there* are some incompatible digipeaters in the path, and that window size negotiation can not be completed. The XID response frame is transmitted with the Window Size Information Element coded as "default k must be used".

If all Digipeater Compatibility Information Elements are present, the station receiving the XID command then evaluates the proposed value for k and either accepts it or selects a new smaller value. At this point the final value for k has been obtained; this final value is recorded in the Window Size Information Element contained in the XID response.

As the XID response works its way back to across the link, intervening digipeaters and the station initiating window size negotiation shall note the final value chosen for k and plan accordingly.

5.4 Initial Round Trip Time Notification

The initial round trip time notification procedures allow one station to suggest to another an appropriate initial value for the smoothed round trip time calculations. The suggestion could be based, for instance, on the present smoothed round trip times seen on other data links over the same radio path, or may be completely arbitrary. The value of smoothed round trip time is ultimately calculated independently by each station on the link, based on recent historical trends for acknowledgements.

Initial round trip time notification procedures begin with the transmission of an XID command with the Initial Round Trip Time Information Element containing the new value for smoothed round trip time to be used by the remote station.

Since round trip timing is not done by digipeaters, each digipeater takes *no* action on this information element.

The remote station shall substitute the round trip time value contained in the XID command in place of its present value in the data link state machine. An XID response must be transmitted (to stop timer TM201); if no other items are being negotiated, the XID response frame may wind up containing an empty information field.

The absence of one or more intervening digipeater identifications (detected by comparing the Digipeater Compatible

Information Elements within the XID frame with the digipeater callsigns in the XID frame's address field) does not preclude the use of initial round trip time notification procedures; i.e., § 3.2 case (b) applies.

5.5 Transmission Speed Negotiation

AX.25 has no default value for transmission speed, although 1200 bit/s and 300 bit/s are commonly used on VHF and HF, respectively. Transmission speed negotiation procedures are used when one station wishes to execute a synchronized change to a new (faster or slower) transmission speed.

In order to achieve a fully synchronized and reliable changeover from one speed to another, a two phase procedure is employed. In the first phase, negotiation of the new speed is performed. In the second phase, the stations execute the changeover to the new speed. The use of both phases is recommended but not mandatory; if ***apriori*** knowledge exists that all stations can operate at the new speed, the negotiation phase may be omitted.

Note -- This proposal does not support the alteration of the transmission speed of digipeaters. Changing the transmission speed of a digipeater would remove the digipeater from service for other users. Thus, transmission speed negotiation can be executed only if no digipeaters are part of the link.

5.5.1 Negotiation

Transmission speed negotiation procedures begin with the transmission of an XID command with the Transmission Speed Information Element containing a list of one or more transmission speeds, in order of preference. Note that, at the ***time*** of transmission of ***this*** first XID ***command,*** a new transmission ***speed is not yet authorized for use. Any ongoing*** link operations shall continue to use the present transmission speed for the time being.

Since any possible speed change will affect the ability to use intervening digipeaters, the data link management state machine within each digipeater on the path checks to determine if transmission speed negotiation is being attempted. If an attempt is being made to negotiate transmission speed, the digipeater alters the Transmission Speed Information Element to "present speed must be used".

The remote station receiving the XID command should also check to see if transmission speed negotiation is being attempted over a link with digipeaters. If so, the XID response frame is transmitted with the Transmission Speed Information Element coded as "present speed must be used".

If no digipeaters are included in the link, the station receiving the XID command then evaluates the proposed values for transmission speed and selects the first one in the list which it can support. The selected value is reported back across the link in the Transmission Speed Information Element contained in the XID response. No change in transmission speed occurs at this point, however.

When the XID response is received at the initiating station, the execution of the changeover begins.

5.5.2 Execution of Changeover

Upon completion of successful transmission speed selection, a new XID command is transmitted at the old speed by the initiating station, containing *only* two information elements:

- the Transmission Speed Information Element, coded with the selected new speed, and
- the Execute Changeover Information Element, coded to indicate "execute changeover".

The initiating station starts TM202 and enters state 2: changeover. Immediately after the XID command has been sent, the station shifts to the new speed to await the XID response. Note -- it is desirable, but not mandatory, that the station also continue to monitor at the old speed for possible indication of changeover failure, and to receive other frames from the remoton until changeover completes.

An intervening digipeater (if any) which detects an XID frame (command or response) containing the Changeover Information Element shall alter the coding of that information element to read "changeover failure".

The remote station, upon detecting the Changeover Information Element coded "execute changeover", shall understand that a changeover is in progress. The Transmission Speed Information Element shall be examined to determine if the new speed can be supported. If the new speed can be supported, the station shall immediately shift to the new speed and shall transmit an XID response with the Changeover Information Element coded "changeover confirmed". If the new speed can not be supported, the station shall transmit an XID response containing the Changeover Information Element coded "changeover failure".

On the other hand, if the remote station detects the Changeover Information Element coded "changeover failure", it shall understand that an attempt was made to made a non-negotiated changeover which could not be supported by one or more intermediate digipeaters. The remote station shall transmit an XID response frame containing the Changeover Information Element as it was coded in the XID command.

The initiating station, upon receipt of an XID response containing the Changeover Information Element coded "changeover confirmed", shall stop timer TM202; shall send an MDL Neogitate Confirm primitive to the AX.25 user; and shall enter state 0: ready. The changeover procedures shall be considered completed.

The initiating station, upon receipt of an XID response containing the Changeover Information Element coded "changeover failure", shall stop timer TM202; shall revert back to the previous speed; shall send a MDL Error primitive to the AX.25 user; and shall enter state 0: ready.

If timer TM202 expires the initiating station shall conclude that the changeover failed; shall revert back to the previous speed; shall send a MDL Error primitive to the AX.25 user; and shall enter state 0: ready.

5.6 Average Information Transfer Rate Notification

The average information transfer rate notification procedures allow one station to inform the other of the anticipated long-term average information transfer rate; such averages can sometimes help in determining whether sufficient processor respources remain available to offer the human user reasonable response time to requests. The average information transfer rate is independently established for each direction of data transfer.

Average information transfer rate notification procedures begin with the transmission of an XID command with the Average Information Transfer Rate Information Element containing the new value for estimated average information transfer rate to be used by the remote station for planning purposes.

Since this average does not affect digipeaters, each digipeater takes no action on this information element.

Upon receipt of the XID command, the remote station shall transmit an XID response; the XID response may also contain an Average Information Transfer Rate Information Element, indicating the anticipated average information transfer rate in the reverse direction on the data link.

The absence of one or more intervening digipeater identifications (detected by comparing the Digipeater Compatible Information Elements within the XID frame with the digipeater callsigns in the XID frame's address field) does not preclude the use of the average information transfer rate notification procedures; i.e., § 3.2 case (b) applies.

6. Combinations of Procedures

Simultaneous negotiation and notification of various parameters is permitted and would be conducted by including all of the relevant information elements in a single XID frame. Figure 9 contains an example of an XID frame.

Changeover procedures must be conducted independently of negotiation/notification procedures. Although the changeover procedures are only exercised by the transmission speed change described above, other synchronized changeovers could be envisioned (transmitter frequency, etc) and could be combined together into a single XID frame.

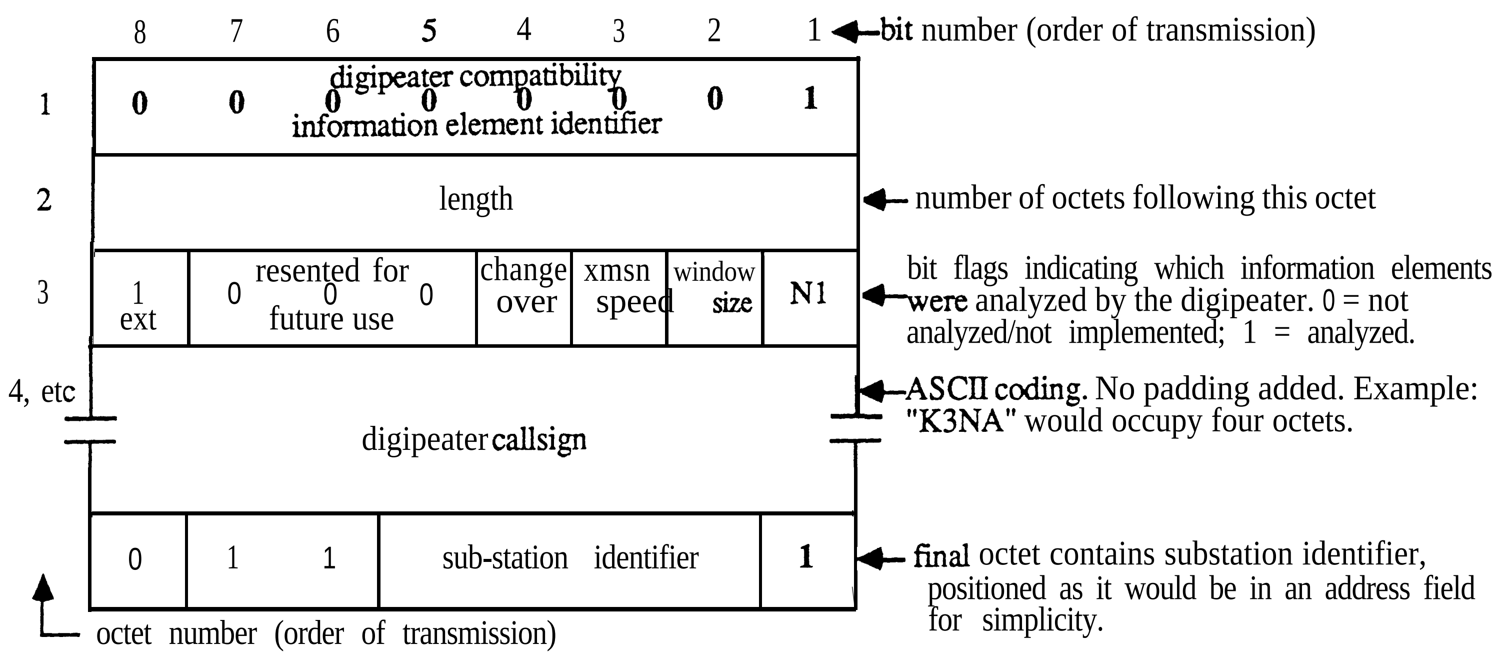
7. SDL Representation of Procedures

Attached at the end of this paper are two sets of SDL diagrams. One set defines the negotiation, notification, and changeover procedures for the stations at each end of an AX.25 data link. The second set defines the negotiation and changeover procedures for intervening digipeaters on the AX.25 data link.

8. Summary

A general purpose, flexible mechanism for automatically adjusting various link parameters, and for conducting a synchronized change of transmission speed, has been proposed. The structure and format of these procedures are both backwards compatible, and yet able to accomodate further expanded applications which might be developed in the future.

Your comments and suggestions are welcome.



Note -- Octer 3 bit 8 is an extension bit provided for future expansion. Therefore all equipment must interprete this bit, when set to 1, to indicate that this is the last octet of bit flags. When set to 0, this bit indicates that another octet of bit flags follows.

Figure 2 -- Digipeater Compatibility information element

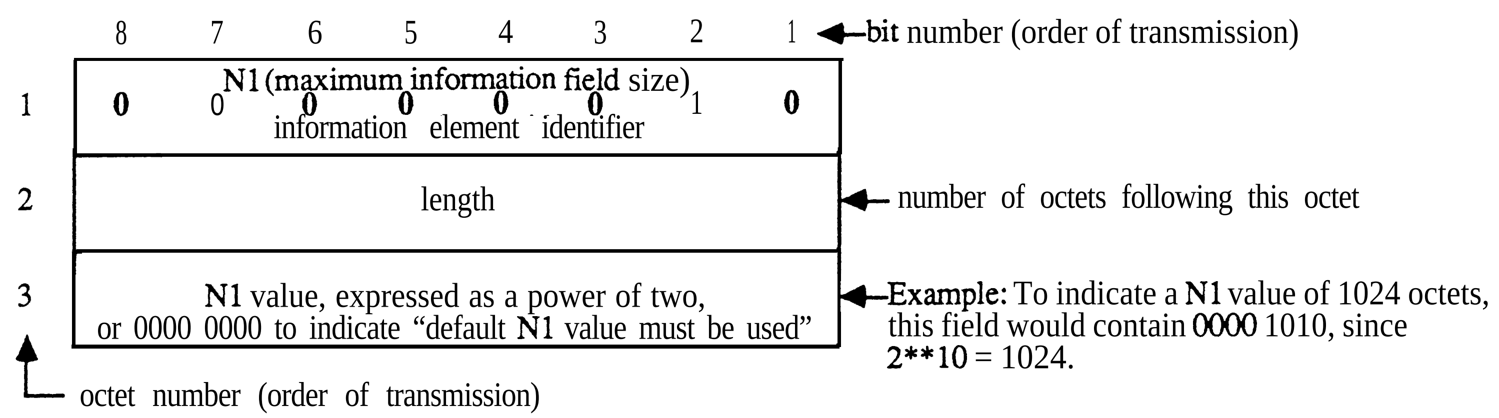


Figure 3 -- Nl information element

Octet	8	7	6	5	4	3	2	1
1	0	0	0	0	0	0	1	1
	window size information element identifier							
2	length							
3	window size (k) 0000 0000 indicates "default value must be used"							

bit number (order of transmission)

number of octets following this octet

k is coded in binary; e.g., 0000 0111 represents a window size of 7.

octet number (order of transmission)

Figure 4 -- Window Size Information Element

Octet	8	7	6	5	4	3	2	1
1	[illegible]	[illegible]	[illegible]	[illegible]	0	1	0	0
	round trip time information element identifier							
2	length							
3	initial round time time, in units of 0.1 seconds							

bit number (order of transmission)

number of octets following this octet

coded in binary; e.g., 0011 0000 represents 4.8 seconds

octet number (order of transmission)

Figure 5 -- Initial Round Trip Time Information Element

Octet	8	7	6	5	4	3	2	1 ← bit number (order of transmission)
1	0	0	0	0	0	1	0	1
	transmission speed information element identifier							
2	length								← number of octets following this octet
3	1 ext	0 reserved for future use	0	transmission speed coded according to the following table				

↑ octet number (order of transmission)

transmission speed encodings:
0 0000 -- present speed must be used.
0 0001 -- 0.6 kbit/s
0 0010 -- 1.2 kbit/s
0 0011 -- 2.4 kbit/s
0 0100 -- 3.6 kbit/s
0 0101 -- 4.8 kbit/s
0 0110 -- 7.2 kbit/s
0 0111 -- 8 kbit/s
0 1000 -- 9.6 kbit/s
0 1001 -- 14.4 kbit/s
0 1010 -- 16 kbit/s
0 1011 -- 19.2 kbit/s
0 1100 -- 32 kbit/s
0 1110 -- 38 kbit/s
0 1111 -- 56 kbit/s
1 0000 -- 64 kbit/s
1 0011 -- 384 kbit/s
1 0101 -- 1536 kbit/s
1 0111 -- 1920 kbit/s
1 1001 -- 0.050 kbit/s
1 1010 -- 0.074 kbit/s
1 1011 -- 0.110 kbit/s
1 1100 -- 0.150 kbit/s
1 1101 -- 0.2 kbit/s
1 1110 -- 0.3 kbit/s
1 1111 -- 12 kbit/s

Note 1 -- Octet 3, bit 8 is reserved for use as an extension bit. Since all transmission speed encodings presently fit into one octet, this bit (for the time being) will always be set to "1".

Note 2 -- When multiple transmission speeds are to be listed octet 3 shall be repeated. The first instance of octet 3 shall be considered to be the most preferred transmission speed; the second instance shall be considered the second-choice speed; etc.

Figure 6 -- Transmission Speed Information Element

octet	8	7	6	5	4	3	2	1
1	0	0	0	0	0	1	1	0
	average information transfer rate information element identifier							
2	length							
3	1 ext	average information transfer rate						

bit number (order of transmission)

octet number (order of transmission)

Octet 1: average information transfer rate information element identifier

Octet 2: number of octets following this octet

Octet 3: average information transfer rate field is coded using the same values as for transmission speed.

Note -- See the transmission speed information element for a description on the use of octet 3, bit 8.

Figure 7 -- Average Information Transfer Rate Information Element

octet	8	7	6	5	4	3	2	1
1	0	0	0	0	0	1	1	1
	changeover information element identifier							
2	length							
3	0	0	0	0	0	0	changeover status code	
	reserved for future use							
4 et;:	callsign of station where changeover failure occurred							

bit number (order of transmission)

octet number (order of transmission)

Octet 2: number of octets following this octet

Octet 3: status codes:
- 00 -- execute changeover.
- 01 -- changeover confirmed
- 10 -- changeover failure
- 00 -- reserved for future use.

Octet 4 etc.: This field contains the callsign of the station which set the changeover status code to “changeover fail&“. When the changeover status code has any other value, a callsign is NOT included in this information element.

Figure 8 -- Changeover Information Element

8	7	6	5	4	3	2	1	← bit order of transmission
(address fields)								
1	0	1	1	1	1	1	1	← **XID control field; P/F bit = 1.**
0	0	0	0	0	0	1	0	← **Nl information element.**
0	0	0	0	0	0	0	1	← one octet follows this octet.
0	0	0	0	1	0	1	0	← Nl value = 1024 octets.
0	0	0	0	0	0	1	1	← **window size information element.**
0	0	0	0	0	0	0	1	← one octet follows this octet.
0	0	0	0	0	1	1	1	← k = 7.
0	0	0	0	0	1	0	0	← **inital round trip time info element**
0	0	0	0	0	0	0	1	← one octet follows this octet
0	0	1	1	0	0	0	0	← round trip time = 4.8 seconds
0	0	0	0	0	1	0	1	← **transmission speed info element**
0	0	0	0	0	0	1	1	← three octets follow this octet
1	0	0	0	1	1	1	1	← 56 kbit/s is preferred speed.
1	0	0	0	0	0	1	1	← 2.4 kbit/s is second choice.
1	0	0	0	0	0	1	0	← 1.2 kbit/s is third choice.
0	0	0	0	0	0	0	1	← **digipeater compatibility info element**
0	0	0	0	0	1	1	0	← six octets follow this octet.
1	0	0	0	0	0	1	1	← only window size and N1 was understood and analyzed
0	1	0	0	1	0	1	1	← K3NA digipeater has checked for compatibility.
0	0	1	1	0	0	1	1	
0	1	0	0	1	1	1	0	
0	1	0	0	0	0	0	1	
0	1	1	0	0	0	0	1	
0	0	0	0	0	0	0	1	← **digipeater compatibility info element**
0	0	0	0	1	0	0	0	← eight octets follow this octet.
1	0	0	0	0	0	1	1	← only window size and N1 was understood and analyzed.
0	1	0	1	0	1	1	1	← WB4JFI-1 digipeater has checked for compatibility.
0	1	0	0	0	0	1	0	
0	0	1	1	0	1	0	0	
0	1	0	0	1	0	1	0	
0	1	0	0	0	1	1	0	
0	1	0	0	1	0	0	1	
0	1	1	0	0	0	1	1	
(frame check sequence)								

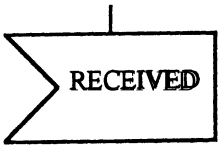
order of octet transmission ↓

Figure 9 -- Example of an XID frame.

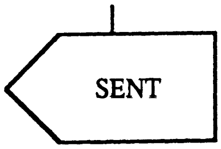
SOURCE/DESTINATION DATA LINK MANAGEMENT

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

MDL Primitives



MDL Negotiate Request

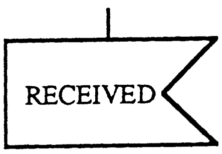


MDL Negotiate Confirm
MDL Error

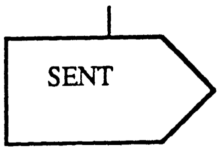
Error Codes

A -- XID command received without P bit set to ‘1’.
B -- unexepected XID response received.
C -- NM201 retries attempted; no valid response received.
D -- XID response received without F bit set to ‘1’.
E -- changeover failed.

LM Primitives



XID command
XID response



XID command
XID response

Flags & Parameters

changeover -- set when changeover procedure is to be executed next.
NM201 -- retry limit.
RC -- retry counter.

States

o-ready
1 -- negotiating
2 -- changeover

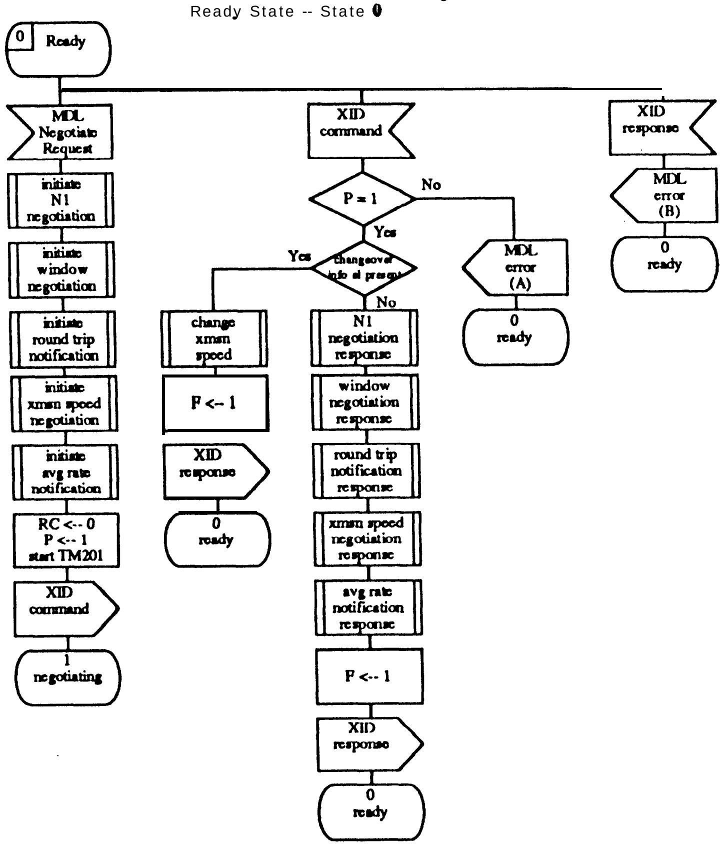
Timers

TM201 -- supervises exchange of XID frames for negotiation.
TM202 -- supervises exchange of XID frames for changeover.

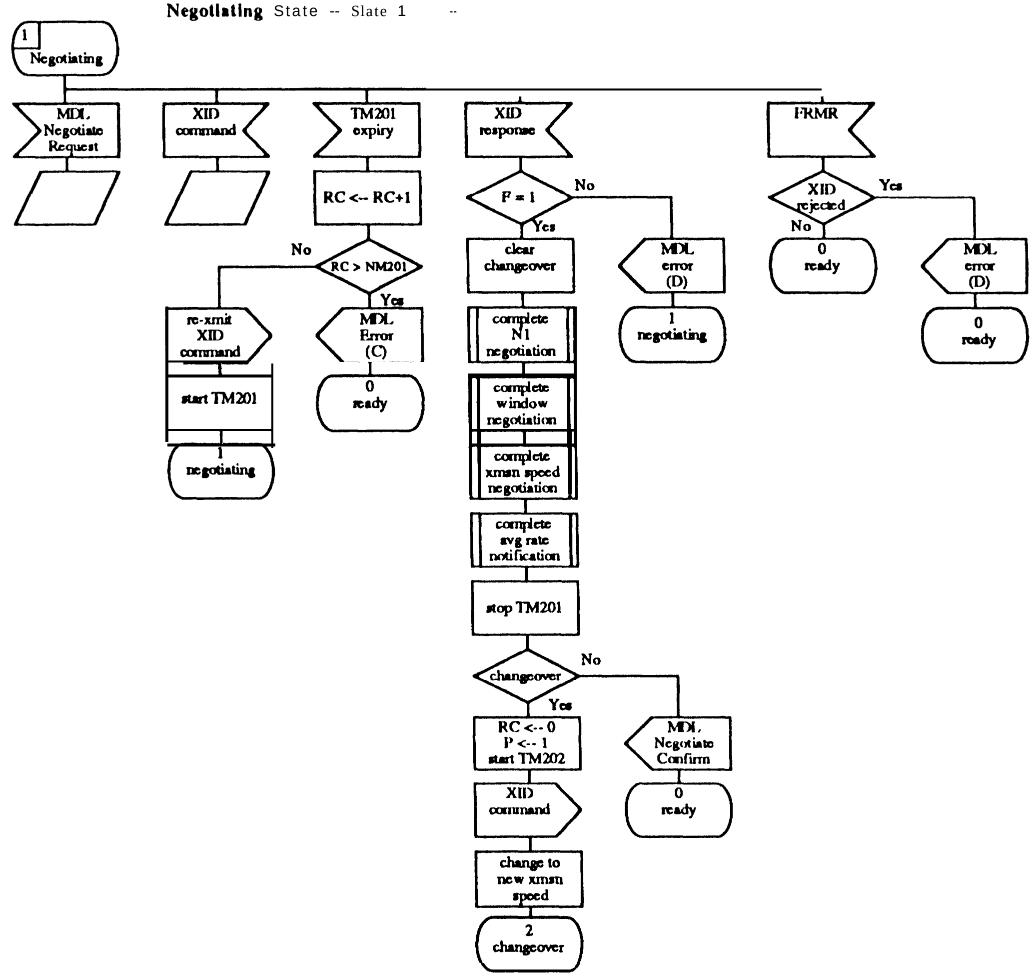
Queues

none.

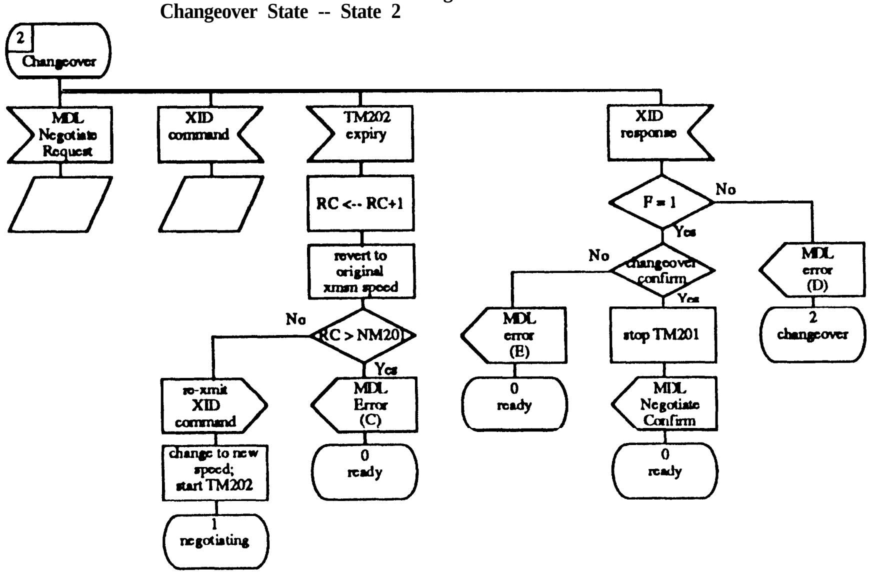
Source/Destination Station Data Link Manrgement
Ready State -- State 0



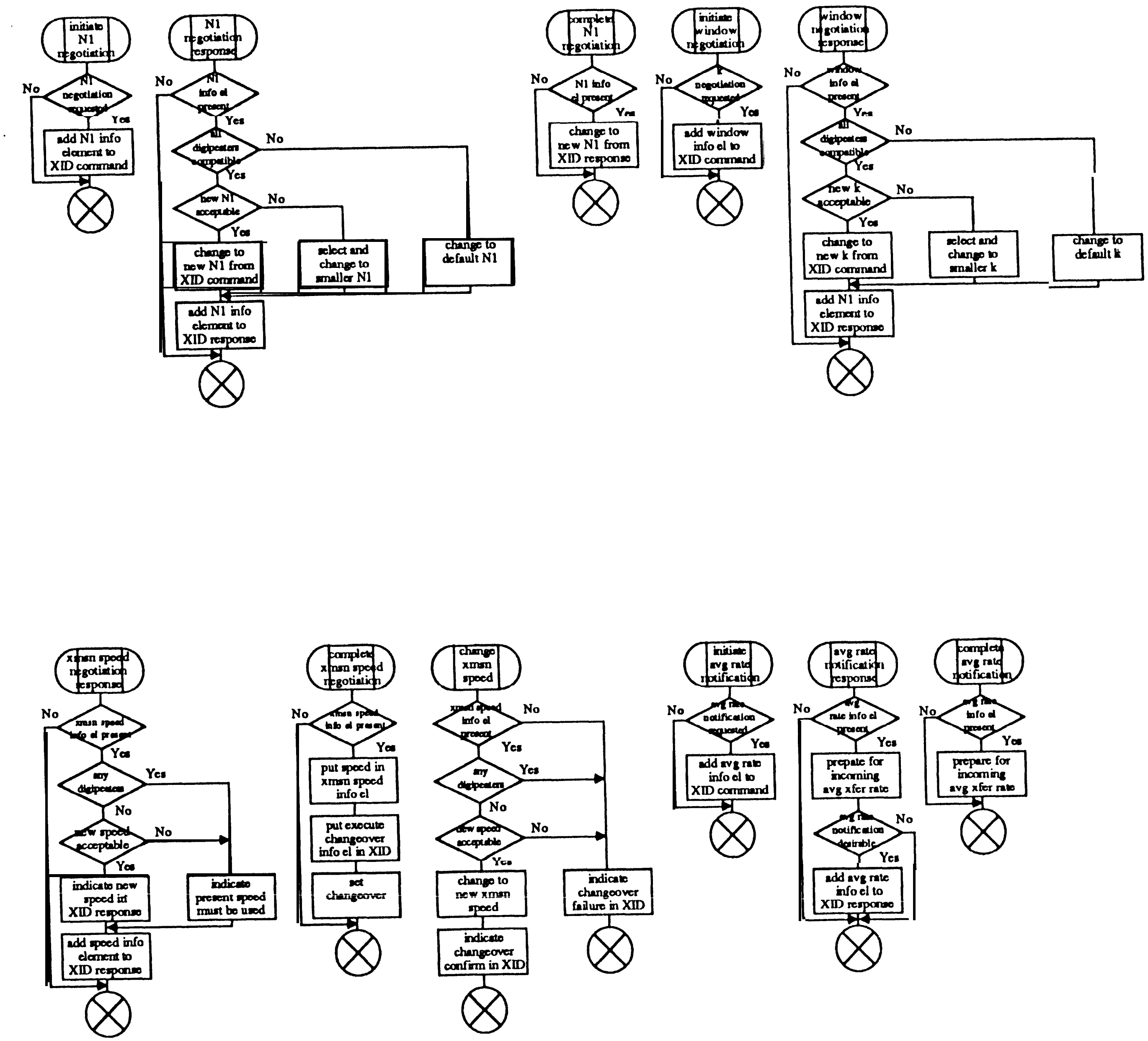
Source/Destination Station Data Link Management
Negotiating State -- Slate 1 --



Source/Destination Station Data Link Management
Changeover State -- State 2



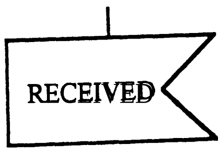
Source/Destination Station Data Link Management
Subroutine8



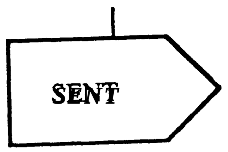
DIGIPEATER DATA LINK MANAGEMENT

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

LM Primitives



XID command
XID response



XID command
XID response

States

0 -- ready

Queues

none.

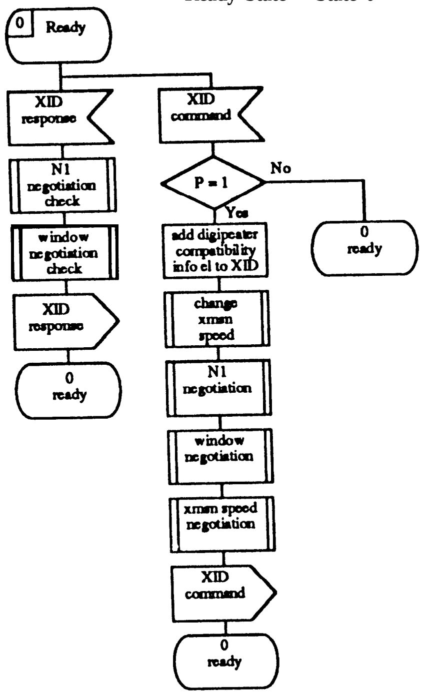
Flags & Parameters

none.

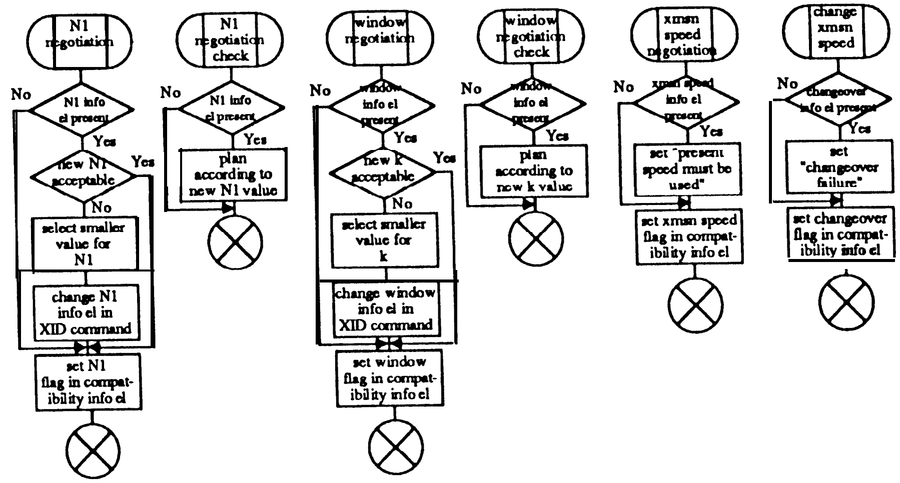
Timers

none.

Digipeater Station Data Link Management
Ready State -- State 0



Source/Destination Station Data Link Management
Subroutines



Segmenter
State Machine

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

0. Summary

This paper is part of a series of papers which provide extended finite state machine representations for AX.25 and related protocols. The state machines are depicted using state description language (SDL) graphic conventions from the Z.100 series of Recommendations developed by the International Telegraph and Telephone Consultative Committee (CCITT) of the International Telecommunications Union (ITU). An extended finite state machine representation of a communications protocol such as AX.25 avoids the ambiguities associated with prose descriptions. These descriptions also compell the protocol designer to confront many of the error scenarios which arise on a communications path, and simplify the implementor's task of producing correct solutions which will interwork with solutions created by others.

This particular paper describes an extended finite state machine to permit long units of data to be transmitted over an X.25 data link, even when that data link is restricted to smaller data units. The state machine has two parts:

-- a segmenter, which divides lengthy data units into a chain of segments for transmission; and,
-- a reassembler, which accumulates segments back together into the original long data unit.

For simplicity, these state machines are referred together as a "segmenter".

1. Status of Proposal

The SDL description here is a draft. The ARRL Digital Committee intends to include this machine as an Annex of the upcoming publication of AX.25 Revision 2.1.

AX.25 Revision 2.0 permits units of data up to 256 octets in length to be transmitted in a single AX.25 I or UI frame. From time to time, applications wish to exchange units of data which exceed this limit. Heretofore, there has been only one solution: to use a higher layer protocol to divide the units of data into smaller units for transmission. The drawback of this approach, particularly in a TCP/IP environment, is that the higher layer protocol headers must be inserted on each subdivided unit. This results in increased overhead on the link.

The situation degrades even *further* when links of poorer transmission quality, such as those experienced on HF radio, are encountered. On these poorer links, it is desirable to keep the AX.25 frames even shorter than normal. Higher layer protocol headers can easily consume more than 50% of a short AX.25 frame on such links.

This proposal remedies these limitations. The segmentor is a simple process which 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, and steps are taken to prevent deadly embrace situations from arising in the buffer managements of both stations on the link.

The following material is still in a draft state. You are invited to review and comment on this material. Comments are desired so that the final publication is as useful as possible to its readers.

2. Features of the Segmentor SDL Machine

The simplex physical SDL machine includes the following features:

a) no overhead when segmentation is not required.

b) low overhead when segmentation is required.

c) provisions to prevent resource deadlocks (deadly embraces) on the data link during segmentation.

d) minimum delay impact during segmentation.

e) adaptability to any value of AX.25 data link N1 (maximum number of octets in the information field of an I or UI frame, excluding the flags, address, control, and frame check sequence fields, as well as 0-bits inserted for transparency).

f) maximum of 128 segments. For the standard AX.25 data link N1 value of 256 octets, this means that a single higher-layer protocol unit, such as an IP datagram, of 32k bytes can be transmitted under a single higher-layer protocol header.

g) protection to ensure that a loss of one or more segments is detected and reported to the AX.25 higher layer user.

h) useable on both connectionless (UI frame) and connection-oriented (I frame) AX.25 transfer modes; however, the use of I frames is *strongly* recommended to avoid aborting a segmented data unit.

3. Location in Overall Model

This SDL machine resides within the data link layer of the standard Open Systems Interconnection reference model. It interacts with the AX.25 user above *it,* and with the data link state machine below it. Neither the AX.25 user nor the data link SDL machine need be aware of the presence of the segmenter SDL machine.

In fact, to simplify the representation of the logic, there are two independent SDL machines: the segmenter and the reassembler.

3.1 Segmenter SDL Machine

The AX.25 user assumes that it is directing the operation of the data link SDL machine through the use of DL primitives described in an accompanying paper. However, when the segmenter SDL machine exists, it intercepts DL primitives and examines them. Only the following DL primitives will be candidates for modification by the segmenter SDL machine:

DL Data Request -- The AX.25 user employs this primitive to provide information to be transmitted using connection-oriented procedures.; i.e., using I frames The segmenter SDL machine examines the quantity of data to be transmitted. If the quantity of data to be transmitted exceeds the data link parameter N1, the segmenter SDL machine swings into action. The segmentation procedures ruthlessly chop up the data into segments of length N1-2 octets. Each segment is prepended with a two octet header. See Figures 1 and 2. The segments are then turned over to the data link SDL machine for transmission, using multiple DL Data Request primitives. All segments are turned over immediately; therefore the data link SDL machine will transmit them consecutively on the data link.

DL Unit Data Request -- The AX.25 user employs this primitive to provide information to be transmitted using connectionless procedures; i.e., using UI frames. The segmenter SDL machine examines the quantity of data to be transmitted. If the quantity of data to be transmitted exceeds the data link parameter N1, the segmenter SDL machine swings into action again, as described above. The segments are turned over to the data link SDL machine for transmission, using multiple DL Unit Data Request primitives. All segments are turned over immediately; therefore the data link SDL machine will transmit them consecutively on the data link.

DL Data Request and DL Unit Data Request primitives which are accompanied with a quantity of data less than N1 will pass transparently through the segmenter SDL machine. No segment header is prepended to the data.

3.2 Reassembler SDL Machine

The data link SDL machine delivers various primitives to the AX.25 user via the reassembler SDL machine. All primitives from the data link SDL machine are delivered transparently, without modification, exceed for the following:

DL Data Indication -- This primitive is examined by the reassembler SDL machine. If the accompanying received data begins with an octet encoded other than 0000 1000, it is assumed that this octet is a conventional PID (as presently described by AX.25 Revision 2.0). If the received data begins with an octet encoded 0000 1000, the reassembler SDL machine assumes that a segment header is present. 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 large data unit. Upon receipt and aggregation of the last segment, the entire large data unit is delivered to the AX.25 user via a single DL Data Indication primitive.

DL Unit Data Indication -- An identical process is followed for the DL Unit Data Indication primitive; the primitive is examined by the reassembler SDL machine. If the accompanying received data begins with an octet encoded other than 0000 1000, it is assumed that this octet is a conventional PID (as presently described by AX.25 Revision 2.0). If the received data begins with an octet encoded 0000 1000, the reassembler SDL machine assumes that a segment header is present. 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 large data unit. Upon receipt and aggregation of the last segment, the entire large data unit is delivered to the AX.25 user via a single DL Unit Data Indication primitive.

DL Data Indication and DL Unit Data Indication primitives containing a conventional PID will pass transparently through the reassembler SDL machine and be delivered immediately to the AX.25 user.

3.3 Summary

Under normal operation, therefore, the net result is that the sending AX.25 user provides a single DL Data Request primitive, containing a large unit of data, to the overall set of SDL machines. Segmentation occurs in the segmenter SDL machine; the data link, link multiplexor, and physical SDL machines work together to transmit the segments across the link to the receiving station. At the receiving station, the physical, link multiplexor, and data link SDL machines work together to receive the incoming segments; reassembly occurs in the reassembler SDL machine. The receiving AX.25 user then receives a single DL Data Request primitive containing the original large unit of data.

The entire set of SDL machines works together to transparently move large data units across the data link.

If an error in reassembly is detected, that error is reported to the AX.25 user with a DL Error Indication primitive.

4. Internal Operation of the SDL Machines

The internal states, error codes, and timers are summarized on the first page of the SDL diagram.

4.1 Internal Operation of the Segmenter SDL Machine

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

4.2 Internal Operation of the Reassembler SDL Machine

The reassembler SDL 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 2). If *this* is *not* the first segment, then the reassembler SDL machine assumes that the actual first segment was lost somewheres, and signals an error. All segments will be discarded as they are received.

Assume now that the first segment was received correctly. The reassembler SDL machine then allocates sufficient storage to receive all the remaining segments; this prevents deadly embrace (resource deadlock) conditions. The reassembler SDL 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 designated TR210: "R" for reassembler; "2" for level 2, the data link level of the OSI open systems interconnection protocol architecture; and "10" simply to avoid confusion with any other timers in this family of SDL 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 provided to the AX.25 user. No attempt is made by the segmenter and reassembler SDL machines to recover segmented data units; this is left to the higher level AX.25 user. Rather, the reassembler SDL 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 SDL 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:

- to ensure that stations with multiple data links minimize the amount of buffer capacity tied up in partially received or transmitted large data units. This in turns reduces the likelihood of link busy conditions (RNR) on connection-oriented links and of discard on connectionless links; and,
- to minimize the delay in transmission of large data units, once large data unit has reached the top of the queue. This in turn means that the AX.25 users, having sent or received the large wad of information, can then move on with the job of processing it and satisfy the thirst of their human users (local or remote) for knowledge.

5.

As mentioned § 2 (h) above, the use of connection-oriented data link procedures is recommended when segmentation is used across 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 increase the amount of congestion on the physical channel.

The simple, CCITT-standardized segmentation procedure here provide two important advancements in packet radio capabilities:

- the ability to carry large data units across a data link with a minimum of overhead (less than 1% for the standard N1 value of 256 octets).
- the ability to reduce N1 for error-prone data links (such as those *over* HF radio channels) to improve the net successful information transfer rate without having to replicate lengthy higher level protocol headers.

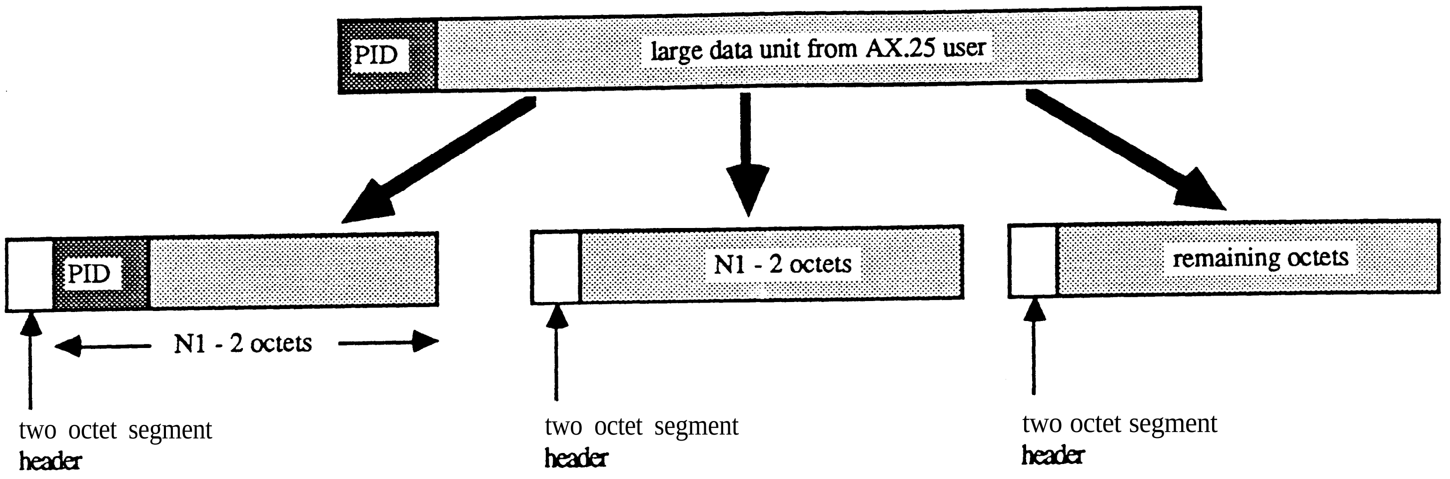


Figure 1 -- Segmentation Process

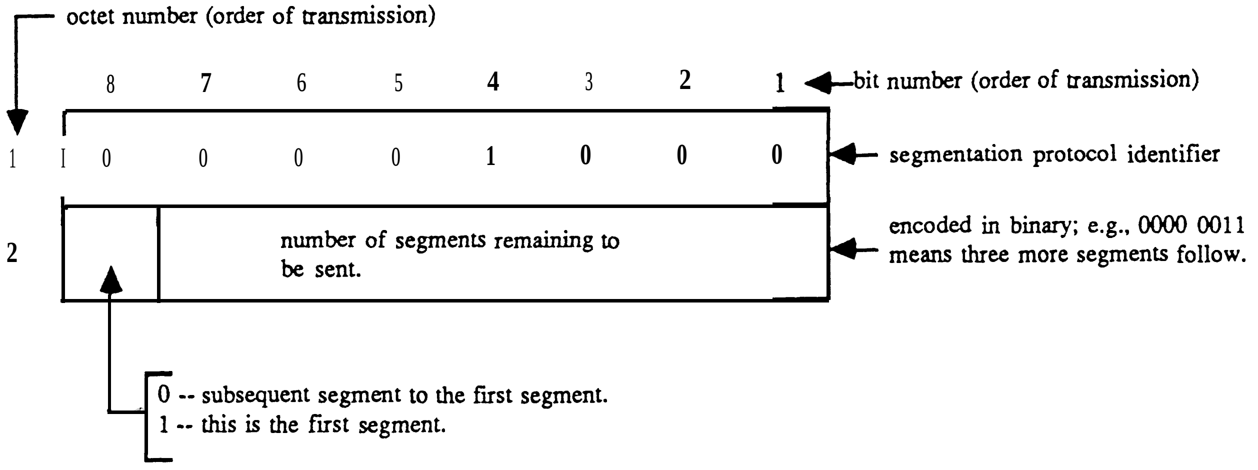
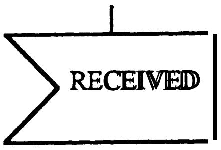


Figure 2 -- Segment Header Format

SEGMENTER

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

DL Primitives



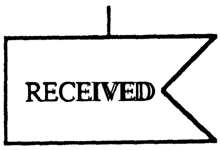
DL Data Request
DL Unit Data Requent
Note -- other DL primitives are passed transparently.

SENT

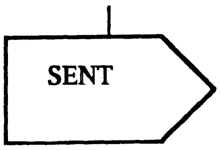
c

DL Data Indication
DL Unit Data Indication
DL Error Indication
Note -- other DL primitives are passed transparently.

DL Primitives



DL Data Indication
DL Unit Data Indication
Note -- all other DL primitives are passed transparently.



DL Data Request
DL Unit Data Request
Note -- all other DL primitives are passed transparently.

Segmenter States

0 -- ready

Reassembler States

0 -- null
1 -- reassembling data
2 -- reassembling unit data

Queues

none.

Error Codes

Z -- reassembly error.

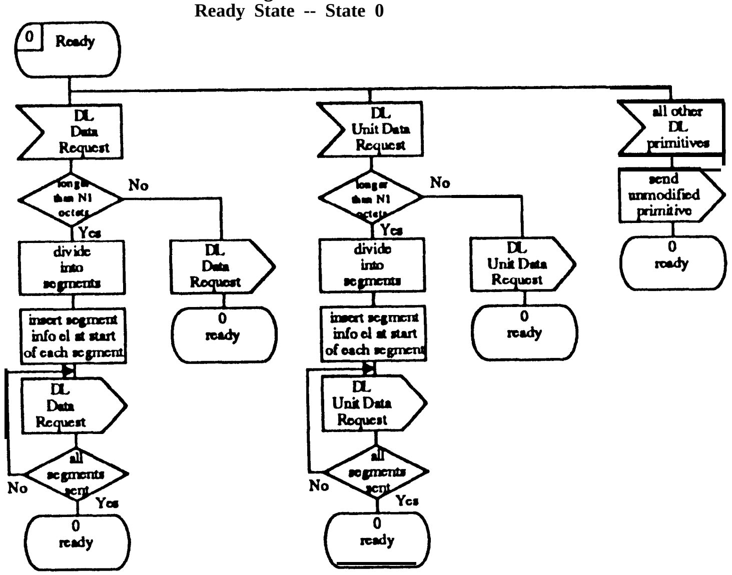
Flags & Parameters

N -- number of segments remaining to be reassembled.

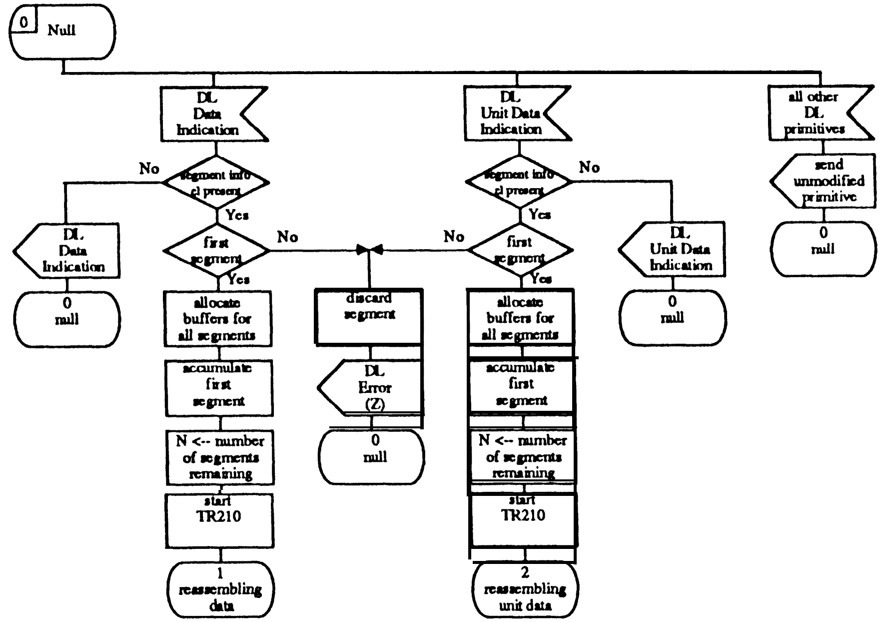
Timers

TR210 -- time limit for receipt of next segment.

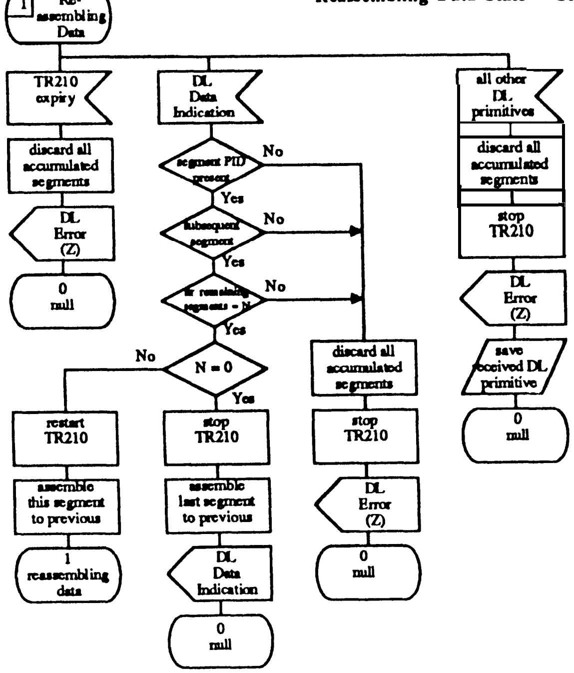
Segmcnter
Ready State -- State 0



Rtrrrembler
Null State -- Slate 0



Reassembler
Reassembling Data State -- State 1



Reassembler
Reassembling Unit Data State -- State 2