MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

GARCHING BEI MÜNCHEN

IMPLEMENTIERUNG EINES X.25-CONTROLLERS

Anton F. Hackl

IPP-R-39 Oct. 1982

Die nachstehende Arbeit wurde im Rahmen des Vertrages zwischen dem Max-Planck-Institut für Plasmaphysik und der Europäischen Atomgemeinschaft über die Zusammenarbeit auf dem Gebiete der Plasmaphysik durchgeführt.

1. Einleitung

Mit dem rapiden Fortschreiten der Technik sind auch die Geraete der Datenverarbeitung, die miteinander kommunizieren koennen und wollen, im Laufe der Jahre immer perfekter und komplexer geworden. Da ist es ganz natuerlich, dass auch altbewaehrte Methoden der Organisation und Ueberwachung von Kommunikation den heutigen Anforderungen an sogenannte "Protokolle" nicht mehr genuegen.

Auch reicht es nicht aus, moderne Loesungen im nationalen Bereich anzustreben, da die Kommunikation in der Datenverarbeitung nicht an Laendergrenzen haltmacht. Es sind statt dessen internationale Standards fuer die Kommunikation von Computer-Systemen zu fordern, weil nur diese eine umfassende Kompatibilitaet von Systemen verschiedener Hersteller gewaehrleisten.

Aufgrund der Komplexitaet heutiger Computer-Systeme ist es leicht einzusehen, dass die Standardisierung in Bezug auf Telekommunikation nicht von heute auf morgen erfolgen kann. Auf dem langen Weg dorthin, hat man jedoch seit Ende 1980 ein wichtiges Hilfsmittel an der Hand: das ISO-Architekturmodell fuer Offene Systeme. (ISO=Internationale Organisation fuer Standardisierung)

Dieses Modell beschreibt das Verstaendnis, das man im Laufe der Diskussionen vom Vorgang der Interkommunikation gewonnen hat, baut eine Architektur aus sieben Ebenen auf, und definiert verbal die Beziehungen dieser Ebenen zueinander. Es muss aber nochmals betont werden, dass es sich hierbei ausschliesslich um eine Modellbeschreibung handelt, die nichts ueber die Implementierung der definierten Aufgaben der einzelnen betroffenen Ebenen aussagt.

Den Normungsgremien bleibt es vorbehalten, sich auf Protokollstandards fuer die angesprochenen sieben Ebenen zu einigen mit der Massgabe, dass diese dann in den logischen Rahmen des Modells passen.

Diese Einigung wird mit Sicherheit noch einige Jahre vergehen lassen, sodass Institute wie das IPP gezwungen sind, Eigenentwicklungen und Pilotimplementierungen zu leisten, da man sehr wohl zum jetzigen Zeitpunkt schon auf Applikationen verweisen kann, die fuer die Realisierung nach dem ISO-Architekturmodell praedestiniert sind.

Ausgehend von der Triplex-Konfiguration des Rechenzentrums Garching - bestehend aus den miteinander gekoppelten CRAY-1, SIEMENS 7880 und AMDAHL 470 V/6 - soll durch "Oeffnung" dieses Dreiecks erreicht werden, auch entfernten Benutzern die Moeglichkeit zu geben, von dieser sehr leistungsfaehigen Einrichtung Gebrauch zu machen. Andererseits sollte natuerlich auch erreicht werden, dass Physiker des Instituts auf wissenswerte Information und

wuenschenswerte Einrichtungen (z.B. in den USA) zugreifen koennen.

Bei der Realisierung des ISO-Schichtenmodells, das eine notwendige Voraussetzung fuer das gesteckte Ziel darstellt, entschieden wir uns fuer eine Zweiteilung:

(1) Die drei Ebenen von X.25 (so ist die Bezeichnung fuer international genormte Protokolle der ersten drei Ebenen des Architekturmodells) werden in einem Minirechner INTERDATA 8/16 implementiert, welcher ueber eine Kanalverbindung mit dem Hauptrechner verbunden ist. Dieser Minicomputer uebernimmt die Rolle eines in dieser Form auf dem Markt nicht verfuegbaren X.25-Telekommunikations-Controllers.

(2) Die hoeheren Protokolle (Ebenen 4-7) werden im Hauptrechner lokalisiert. Diese Protokolle wurden wegen fehlender Normen selbst entwickelt, jedoch modular aufgebaut, sodass sie austauschbar sind.

Im folgenden wird ein Ueberblick ueber die Arbeiten Punkt (1) betreffend gegeben: Implementierung des X.25-Telekommunikations- Controllers (X.25-TC).

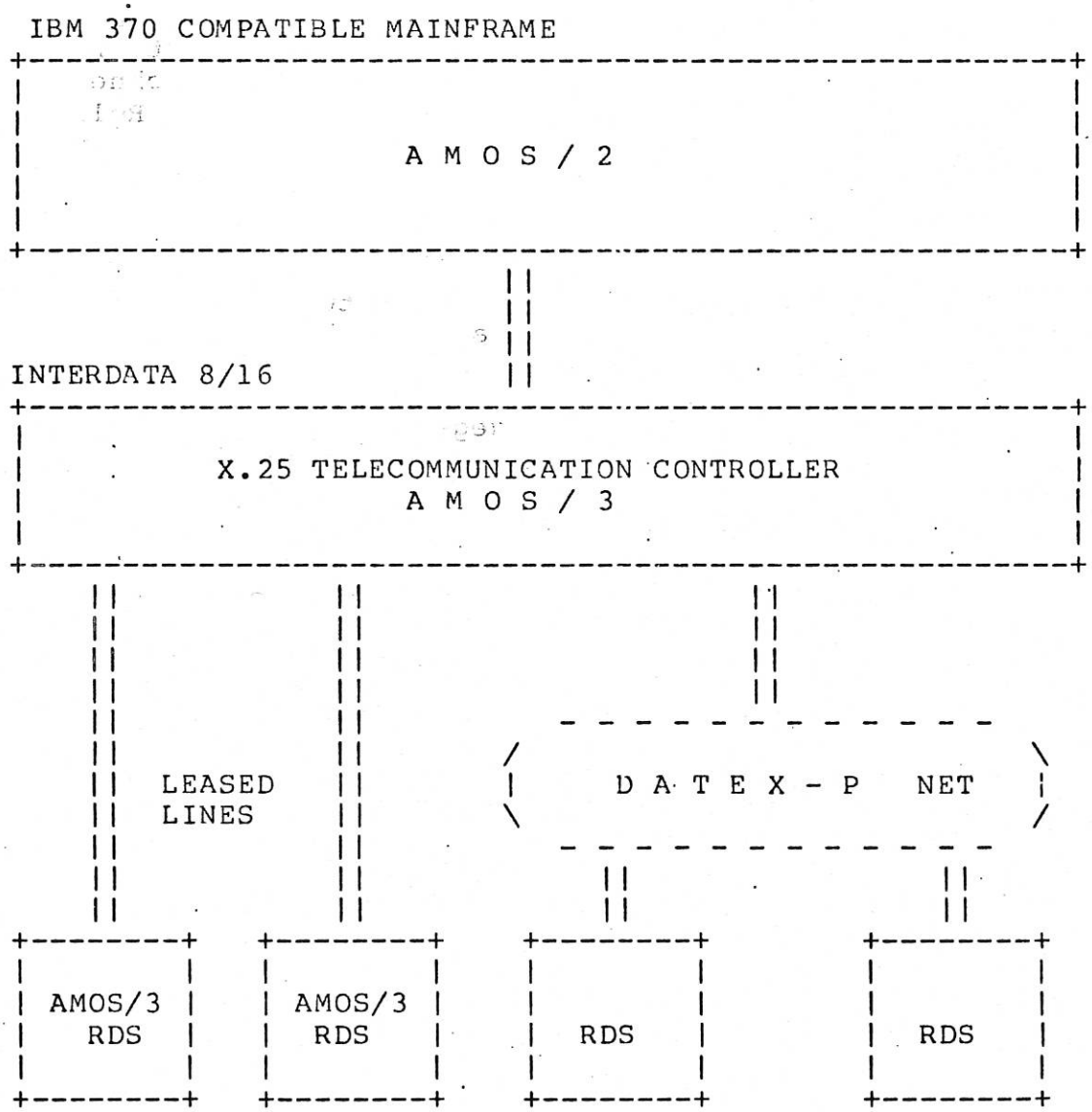
2. Ueberblick X.25-TC:

2.1 Allgemein

Aufgabe des X.25-TC ist es, vom Betriebssystem AMOS/2 aus, das in einem IBM-kompatiblen Hauptrechner residiert, den Anschluss von Terminals und Prozessen (zu diesem Begriff vgl. Dokumentation AMOS/2) an das von der Deutschen Bundespost betriebene Paketvermittlungsnetz DATEX-P zu ermoeglichen. Des weiteren sollten gemietete Leitungen unterstuetzt werden, da sich z.B. bei bestehenden "AMOS/3 Remote Data Stations" (RDS) die Unwirtschaftlichkeit von DATEX-P ergeben kann.

Beides - DATEX-P und gemietete Leitungen - stellt allerdings kein prinzipielles Problem dar, sondern ist ausschliesslich fuer das Protokoll X.25 von Bedeutung, da sich bei gemieteten Leitungen eine Kommunikation von DTE zu DTE ergibt (und nicht wie bei X.25 ueblich DTE/DCE).

Aus dieser Zielsetzung ergibt sich folgende Konfiguration:



AMOS NETWORK

Dieser X.25-TC befreit das Betriebssystem AMOS/2 von (Verwaltungs)arbeiten X.25 betreffend und sorgt mit einer geeigneten Schnittstelle dafuer, dass alle Ereignisse an AMOS/2 gemeldet werden und umgekehrt alle moeglichen Dienstleistungen, die X.25 bietet, von AMOS/2 in Anspruch genommen werden koennen (vgl. dazu den Bericht "Message Link Protocol (MLP) and its Interfaces").

2.2 Hardware

Hardware-Basis ist ein 16-Bit Minicomputer Interdata 8/16 mit folgender Ausstattung:

- CPU
- 64 KByte Memory
- Clock (Timer)
- Consol-Interface + Console
- mehrere Interfaces mit HDLC-Faehigkeit, um die X.25-Leitungen zu betreiben (leased lines + DATEX-P)
- Selektor-Kanal (= DMA-Interface)
- Modell 360/370 W/W Interface, um den TC an den IBM (kompatiblen) Kanal zu haengen

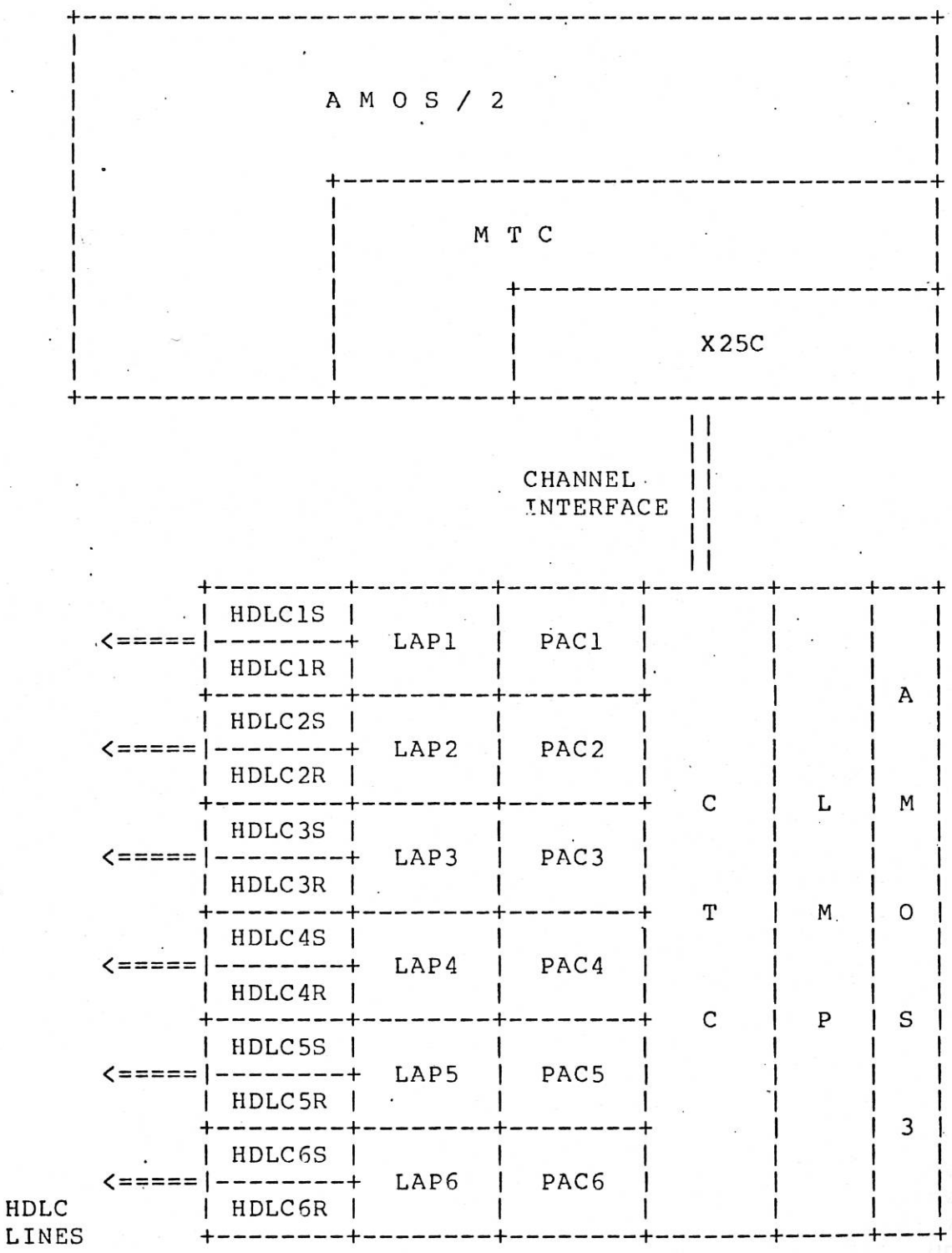
2.3 Software

Die zu entwickelnde Software fuer X.25 und den Kanalanschluss wurde in AMOS/3 integriert, das als Betriebssystem die Interdata 8/16 steuert. (s. Dok. AMOS/3)

Folgende Prozessoren (=AMOS/3-Notation) wurden integriert:

- HDLCSENDER (HDLCS), I/O-Processor, der die Senderichtung einer HDLC-Leitung bedient
- HDLCRECEIVER (HDLCR), I/O-Processor, der die Empfangsrichtung einer HDLC-Leitung bedient
- LAP, CPU-Prozessor, der X.25-Level 2 realisiert
- PAC, CPU-Prozessor, der X.25-Level 3 realisiert
- CTC, I/O-Prozessor, der die Kanalverbindung realisiert
- LMP, CPU-Prozessor, der Ueberwachungsfunktionen ausuebt

Die Prozessoren HDLCSENDER, HDLCRECEIVER, LAP, PAC sind mehrfach (reentrant) vorhanden, abhaengig von der Anzahl der Leitungen. Moeglich sind sechs Leitungen mit zusammen 60 logischen X.25 Kanaelen (die Begrenzung setzt hierbei die Zykluszeit der CPU und die Groesse des Hauptspeichers).



SOFTWARE UEBERBLICK

3. Implementierung von X.25

3.1 HDLCSENDER, HDLCRECEIVER

Die I/O-Prozessoren HDLCSENDER und HDLCRECEIVER realisieren X.25 Ebene 1. Als Hardware-Interface wird ein "Single Line Synchronous Adapter" (SSA) von der Fa. PERKIN-ELMER verwendet.

Technische Daten:

- Maximale Datenrate 2 Mbits/sec
- RS-232 Interface
- Double-Buffering
- Full-Duplex-Faehigkeit
- Automatisches Einfuegen und Entfernen von "Zero Bit" und "Flag"

3.2 LAP

3.2.1 Transaction Processor LAP

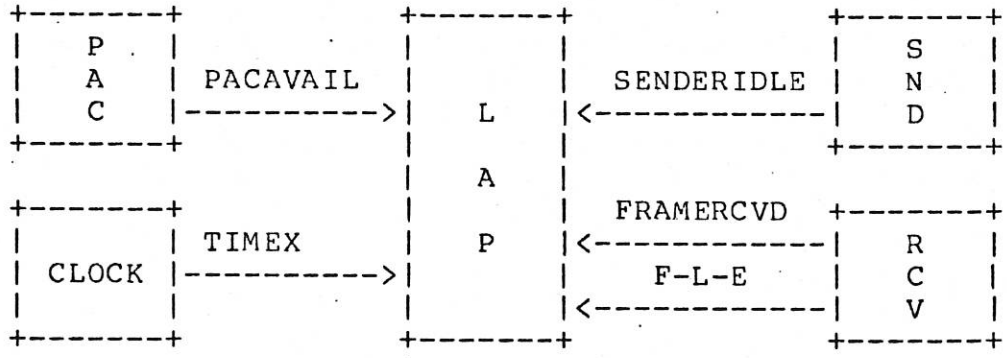
(Fuer Einzelheiten des Begriffes "Transaction Processor" siehe AMOS/3-ADN D322.)

Aufgabe des Transaction Processors LAP ist es, die im "DATEX-P Handbuch" definierte Ebene 2 von X.25 (LAP/B) zu realisieren und dabei Verbindungen zu Level 1 und 3 herzustellen und aufrecht zu erhalten. Diese Verbindungen werden primaer ueber Transaction-Queues und Transaction-Requests organisiert, in Richtung Level 1 (HDLCSENDER) ist jedoch auch noch die I/O-Termination-Queue beteiligt.

3.2.2 Transaction Requests

Auftraege fuer LAP

In diesem Abschnitt werden die Transaction Requests aufgefuehrt und erlaeutert, die von anderen Prozessoren (X.25 Level 1+3) generiert werden und den LAP veranlassen, gezielte Aktionen ablaufen zu lassen.



SENDERIDLE — Mit diesem Request meldet sich der HDLCSENDER, wenn er bereit ist, einen kompletten HDLC-Frame zu uebernehmen und zu senden. LAP notiert diesen Request und befriedigt HDLCSENDER, sobald er dazu in der Lage ist.

PACAVAIL — PAC meldet damit die Existenz eines Level3-Packets, das vom LAP angefordert wurde (mit Transaction Request REQUPACKET s.u.) LAP uebernimmt das Level3-Packet und uebergibt es bei Gelegenheit an den HDLCSENDER.

FRAMERCVD — HDLCRECEIVER zeigt damit das Eintreffen eines

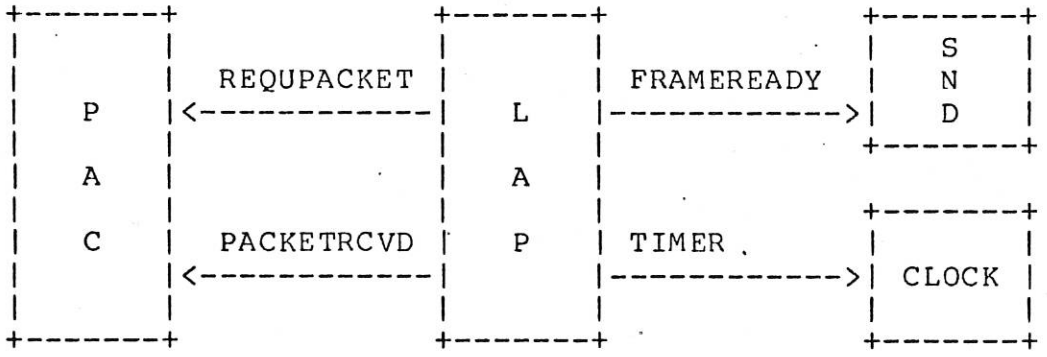
HDLC-Frames an.
LAP uebernimmt den Frame und verarbeitet ihn (CRC-Pruefung, S-Frame Behandlung gemaess Protokoll, I-Frame Weitergabe an PAC).
HDLCRECEIVER wird im Austausch mit dem empfangenen Frame ein neuer leerer Puffer zur Verfuegung gestellt (dessen Queue mit verfuegbaren Puffern duerfte daher niemals leer sein).

TIMEEX — Bei bestimmten zu sendenden Frames (I-Frames, SABM, u.a.) wird ein Timer aufgezogen, der abgelaufen ist, wenn LAP diesen Request in seiner Transaction-Queue vorfindet. LAP laesst Aktionen ablaufen, die abhaengig sind vom Zustand der HDLC-Verbindung und des Wiederholungszaehlers (i.d.R. wird Fehlerbehandlung durch Retransmission angestossen). Dieser Request kann in mehreren Variationen auftreten (TIME1EX, TIME2EX).

F-L-E — Frame-Length-Error: Bei Empfang eines Frames, der die maximal zulaessige Framelaenge von 128 Bytes ueberschreitet, wird diese Transaktion vom RECEIVER fuer den LAP generiert.

Auftraege des LAP

LAP nimmt von vier Prozessoren Auftraege ueber seine Transaction-Queue an, er generiert jedoch auch Auftraege fuer Prozessoren, die mit ihm kommunizieren:



REQUPACKET Dieser Auftrag veranlasst den PAC ein Packet bereitzustellen.

PACKETRCVD LAP uebergibt damit einen I-Frame an PAC, nachdem zahlreiche Pruefungen bestanden wurden.

FRAMEREADY Ueber die I/O-Termination-Queue bekommt damit der HDLCSENDER mitgeteilt, dass ein Frame generiert wurde und gesendet werden kann. Die physikalische Adresse des Frame-Anfangs befindet sich zu diesem Zeitpunkt bereits im PCB (Process Control Block) des HDLCSENDER.

TIMER Mit diesem Auftrag wird eine Zeitueberwachung aktiviert.

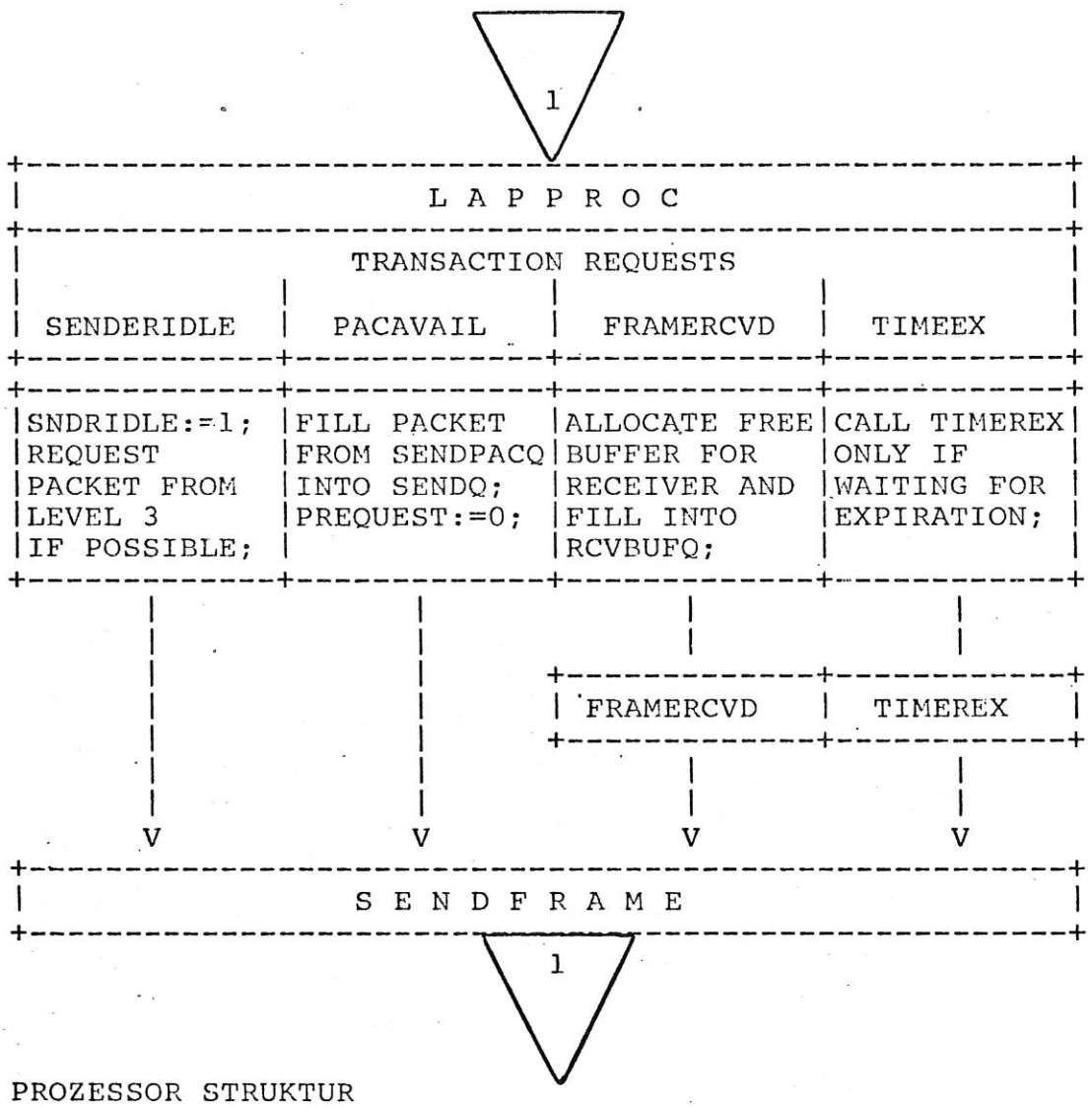
3.2.3 LAP Prozessor Struktur

Aufgrund der Organisation als Transaction-Prozessor ist es moeglich, eine einfache, aber aeusserst effiziente Ablaufstruktur fuer den LAP-Prozessor aufzubauen:

Nach dem Start des Prozessors mit einer notwendigen Initialisierungssequenz wird die Transaction-Queue inspiziert: Ist kein Auftrag vorhanden, wartet LAP solange, bis ein kommunizierender Prozessor ihm einen Auftrag mit der Generierung eines Transaction-Requests erteilt. Dieser soeben generierte, oder ein bereits in der Queue vorhandener Request wird sofort verarbeitet, wobei natuerlich mehrere Prozeduren beteiligt sind (z.B. FRAMERCVD,TIMEREX,...). Sind die mit dem Transaction-Request direkt verknuepften Aktionen abgelaufen, so wird in der "Global Procedure SENDFRAME" ueberprueft, ob die Leitung (HDLCSENDER) einen eventuell vorhandenen Frame uebernehmen kann.

Da diese Prozedur SENDFRAME nach der Behandlung eines jeden Transaction-Requests aufgerufen wird, ist gewaehrleistet, dass der HDLCSENDER immer aktiviert bleibt und Frames in der Regel nur sehr kurzzeitig zwischengespeichert bleiben.

Nach dem Verlassen der Prozedur SENDFRAME wird wieder auf einen neuen Auftrag gewartet bzw. ein bereits in der Transaction-Queue anstehender Request verarbeitet. Auf diese Weise entsteht eine von Transaction-Requests gesteuerte Ablaufschleife, die den LAP nur dann aktiviert, wenn wirklich etwas zu tun ist.

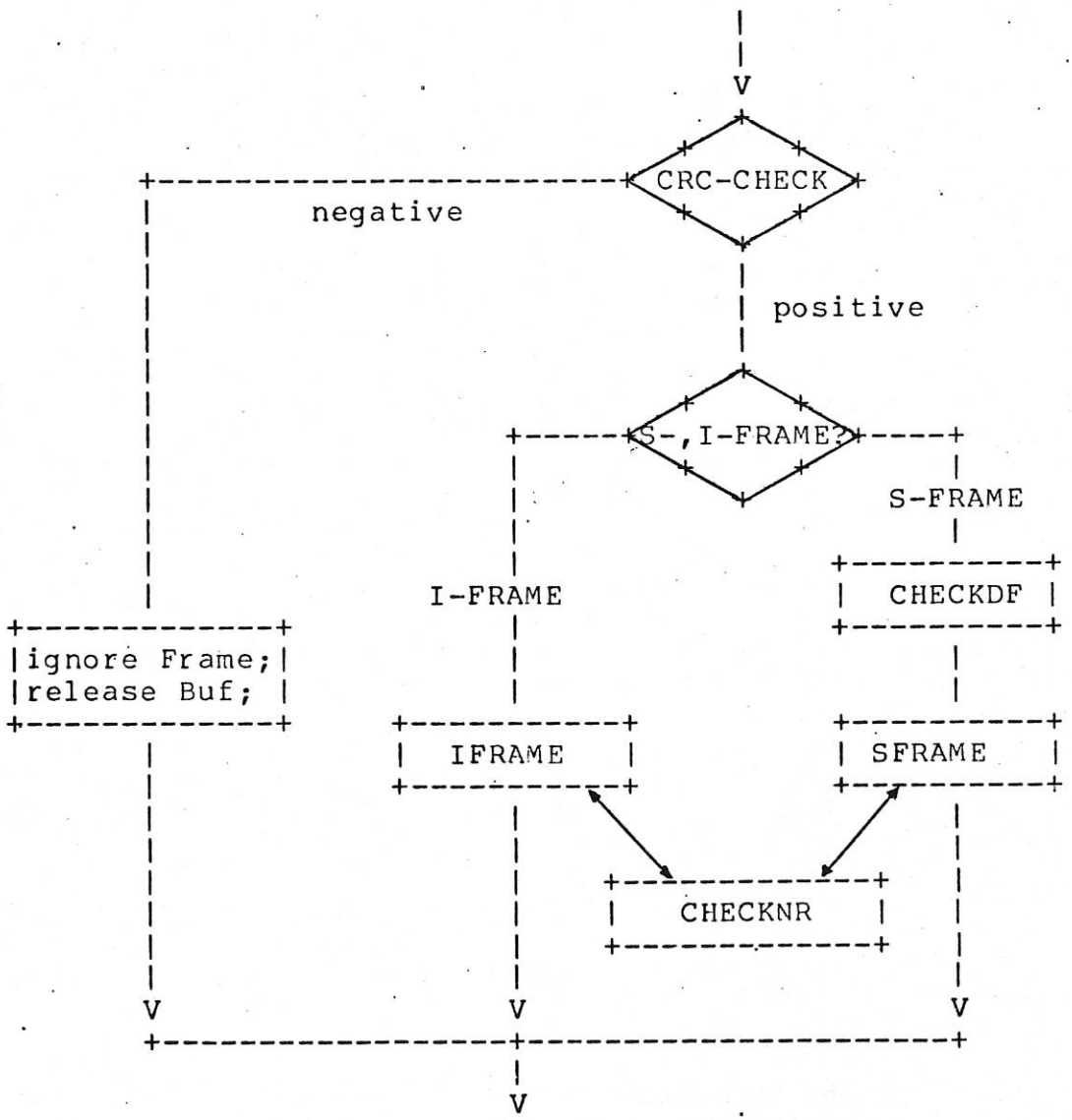


PROZESSOR STRUKTUR

3.2.4 Empfang eines Frames

Mit dem Transaction-Request "FRAMERCVD" meldet der HDLCRECEIVER den Empfang eines Frames (die Pufferadresse dieses Frames befindet sich in der Liste RCVDFRAMEQ). Daraufhin wird aus LAPPROC die Global Procedure FRAMERCVD aufgerufen, in der folgende Aktionen ablaufen:

- CRC-Pruefung
- Entscheidung, ob S- oder I-Frame
- Datenfeld-Pruefung (siehe DATEX-P)
- Aufruf der lokalen Prozeduren SFRAME bzw. IFRAME



Procedure SFRAME

Falls Bit 0 des X.25 Level 2 Control-Byte identisch 1 ist, wird von der Global Procedure FRAMERCVD die Local Procedure SFRAME aufgerufen, da dieses Bit 1=1 einen "Supervisory-Frame" kennzeichnet.

Um eine Struktur aufzubauen, bei der "case-statements" die Abarbeitung eines empfangenen S-Frame in einem gegebenen Zustand des LAP-Prozessors organisieren, wird eine Funktion f(t,s) definiert:

```
f(t,s): = g(t * 6 + (s - 1));
```

t Typ des empfangenen S-Frame
(ausmaskiert mit #0E aus dem Control-Byte)

s Zustand des LAP-Prozessors
(1 = SILENT 4 = SABMCOLL
2 = SENDDM 5 = ABM
3 = SENDSABM 6 = FRMRSENT)

g(x) definiert von den Tabellen CMDS,RESP
(welche Tabelle verwendet wird, haengt vom ADDRESS-Byte von X.25 Level 2 ab).

C M D S							R E S P						
	s												s
t	1	2	3	4	5	6	1	2	3	4	5	6	t
RR	1,	2,	2,	3,	3,	10	1,	1,	1,	2,	3,	1	RR
DISC	4,	4,	5,	5,	5,	5	1,	1,	4,	4,	5,	5	UA
RNR	1,	2,	2,	6,	6,	10	1,	1,	1,	2,	7,	1	RNR
ILL.				6(10)			1,	1,	8,	1,	6,	9	FRMR
REJ	1,	2,	2,	7,	7,	10	1,	1,	1,	2,	10,	1	REJ
ILL.				6(10)						6(11)			ILL.
ILL.				6(10)						6(11)			ILL.
SABM	8,	8,	9,	9,	8,	8	1,	9,	1,	1,	9,	9	DM

Der von der Funktion f eindeutig bestimmte Funktionswert kennzeichnet ein case-statement, in dem die entsprechenden Aktionen ablaufen.

Procedure CHECKNR

Fuer jeden empfangenen Frame, der Flusskontroll-Zaehler enthaelt und der die Vorpruefungen ueberstanden hat (CRC-Check, Datenfeld-Pruefung), wird die lokale Procedure CHECKNR aufgerufen, um den Empfangsfolgezaehler N(R) auszuwerten.

Aktionen:

- Pruefung der Ungleichung VSU $\leq$ N(R) $\leq$ V(S)
- Loeschen des Timers T1, falls N(R) = V(S)
  d.h. der letzte gesendete I-Frame wurde mit dem empfangenen N(R) quittiert.
- Freigeben von Puffern aus der Liste SENTQ, da die darin gespeicherten und bereits gesendeten Frames von N(R) als empfangen bestaetigt werden
  Zahl der freizugebenden Puffern: (N(R)-VSU) mod 8
- Ist N(R) ausserhalb des zulaessigen Bereiches, wird ein FRMR-Frame generiert.

- Setzen des Return-Registers:
  R0 = 0: VSU $\leq$ N(R) $\leq$ V(S)
  R0 = 1: N(R) ausserhalb
- Setze VSU: = N(R), falls Ungleichung erfuellt.

Bem: VSU = V(S) des aeltesten noch nicht bestaetigten
---- Information-Frame: unteres Fenstereck

3.2.5 Senden eines Frames

SENDFRAME ist die Global Procedure, die fuer die Generierung und Uebergabe von generierten Frames an den HDLCSENDER verantwortlich ist.

Aktionen:

- Zuallererst wird die Bereitschaft des HDLCSENDER ueberprueft, einen Frame uebernehmen zu koennen (Flag SNDRIDLE). Bei Nichtbereitschaft wird Aktion gestoppt und die Prozedur verlassen.

- Die Information darueber, ob und wenn ja, welcher Frame zu generieren ist, befindet sich in der Variablen FRAMETYPE. Diese wird bei der Abarbeitung der Transaction-Requests entsprechend gesetzt. Der geforderte S-Frame wird generiert und an den HDLCSENDER uebergeben.

- Ist kein S-Frame zu generieren (nur solche werden in FRAMETYPE angezeigt), so wird nachgesehen, ob ein I-Frame ansteht (Liste SENDQ). Bei Bestehen aller Pruefungen (Zustand = ABM, DCE empfangsbereit, Window nicht ausgeschoepft) wird ein evtl. Vorhandener an den HDLCSENDER uebergeben.

3.3 PAC

3.3.1 Ueberblick

Der Prozess PAC realisiert primaer die Ebene 3 der CCITT-Empfehlung X.25.
Die Funktionen dieser Ebene sind:

- Multiplexen mehrerer logischer Kanaele auf dem Uebermittlungsabschnitt von Ebene 2 (LAP)
- Abwicklung gewaehlter virtueller Verbindungen (virtual calls) und/oder fester virtueller Verbindungen (permanent virtual circuits) auf den logischen Kanaelen
- Fehlerkontrolle und Flusskontrolle
- Unterbrechung des normalen Datenflusses durch Aussenden spezieller Informationen unter Umgehung der Flusskontrolle
- Korrektur von Fehlern

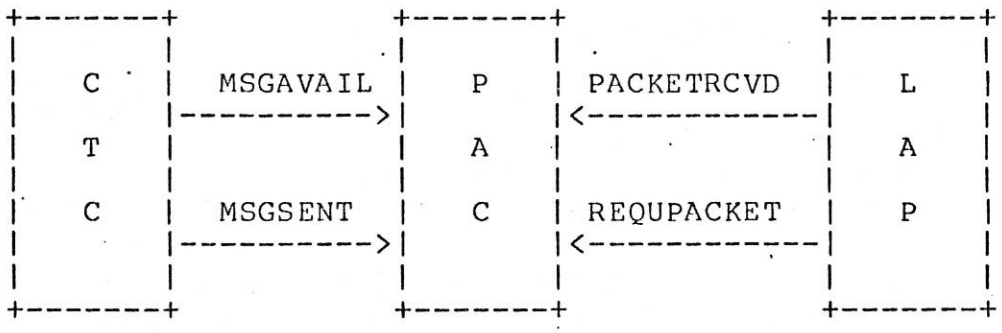
Sekundaer (aber fuer den Betrieb nicht weniger wichtig) baut PAC ueber eine definierte Schnittstelle eine Verbindung zu dem Prozess auf, der fuer die Kanalverbindung zum IBM-kompatiblen Hauptrechner verantwortlich ist (CTC).

Dadurch wird gewaehrleistet, dass - im Zusammenspiel mit dem Interface von Level 4/Level 3 - eine Kommunikation zwischen Benutzer und X.25-TC ueber zwei eigenstaendige Rechner und Betriebssysteme hinweg zustande kommt.

3.3.2 Transaction Requests

Auftraege fuer PAC

Ebenso wie der LAP organisiert als Transaction Prozessor, wird der Prozessor PAC von einer Transaction-Queue und darin enthaltenen Transaction-Requests gesteuert. Diese steuernden Auftraege werden von den Prozessoren LAP und CTC generiert:



Kurzbeschreibung der Aktionen bei den einzelnen Transaction-Requests:

PACKETRCVD

Nachdem ein empfangener Information-Frame alle Pruefungen im LAP positiv durchlaufen hat, wird der Puffer in eine Queue eingetragen (RCVDPACQ) und mit diesem Auftrag PACKETRCVD dem PAC mitgeteilt, dass ein Paket zur Verarbeitung ansteht. PAC holt dieses Paket aus der Queue ab, bestimmt den zugeordneten logischen Kanal und verarbeitet das Paket entsprechend den Bestimmungen von X.25 Level 3.
(Beteiligte Prozeduren: PACRCVD, DATAPAC, X25TERM).

REQUPACKET

Damit zeigt LAP an, dass PAC zu sendende Pakete an den LAP uebergeben kann. PAC speichert diese Indikation (Variable LAPFREQ) und uebergibt - soweit vorhanden - in der Prozedur SENDPAC mit dem Request PACAVAIL ein Paket an den LAP (Puffer ist abgelegt in Queue SENDPACQ).

MSGAVAIL

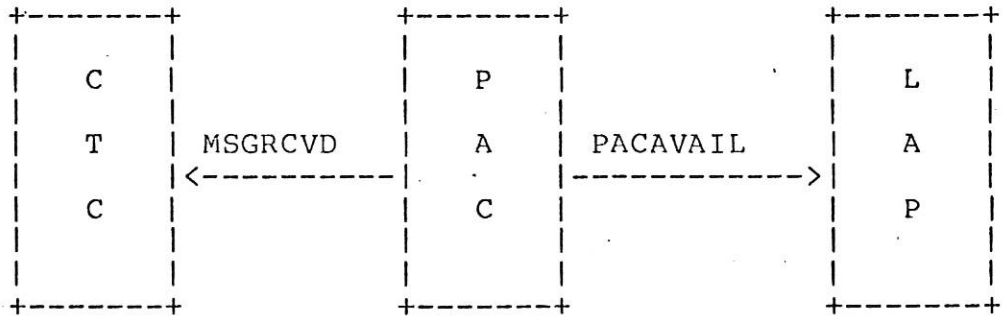
Der Kanaltreiber CTC hat ueber den Kanal eine Nachricht (message) empfangen, die er mit MSGAVAIL an den PAC uebergibt. Die Message befindet sich in der Liste SENDMSGQ. PAC uebernimmt die Message aus SENDMSGQ, bestimmt in der Prozedur CTCMSG den zugehoerigen logischen Kanal und generiert bei Bedarf ein Level 3-Paket.

MSGSENT

Sinn dieses Requests ist es, in die X.25-Flusskontrolle auch den Kanalanschluss miteinzubeziehen: Anstatt sofort fuer empfangene Datenpakete eine Empfangsbestaetigung (RR) zu schicken, wird diese Bestaetigung solange zurueckgehalten, bis vom CTC der Request MSGSENT generiert wird, womit angezeigt ist, dass fuer den in der Liste POSTCHQ angegebenen logischen Kanal ein Datenpaket erfolgreich ueber den Kanalanschluss tranferiert wurde. PAC generiert ein "Receive Ready"-Paket.

Auftraege des PAC

Fuer die Kommunikation des Pac mit CTC und LAP ist es notwendig, nicht nur Auftraege von diesen Prozessoren entgegenzunehmen, sondern ihnen auch Auftraege zu erteilen:

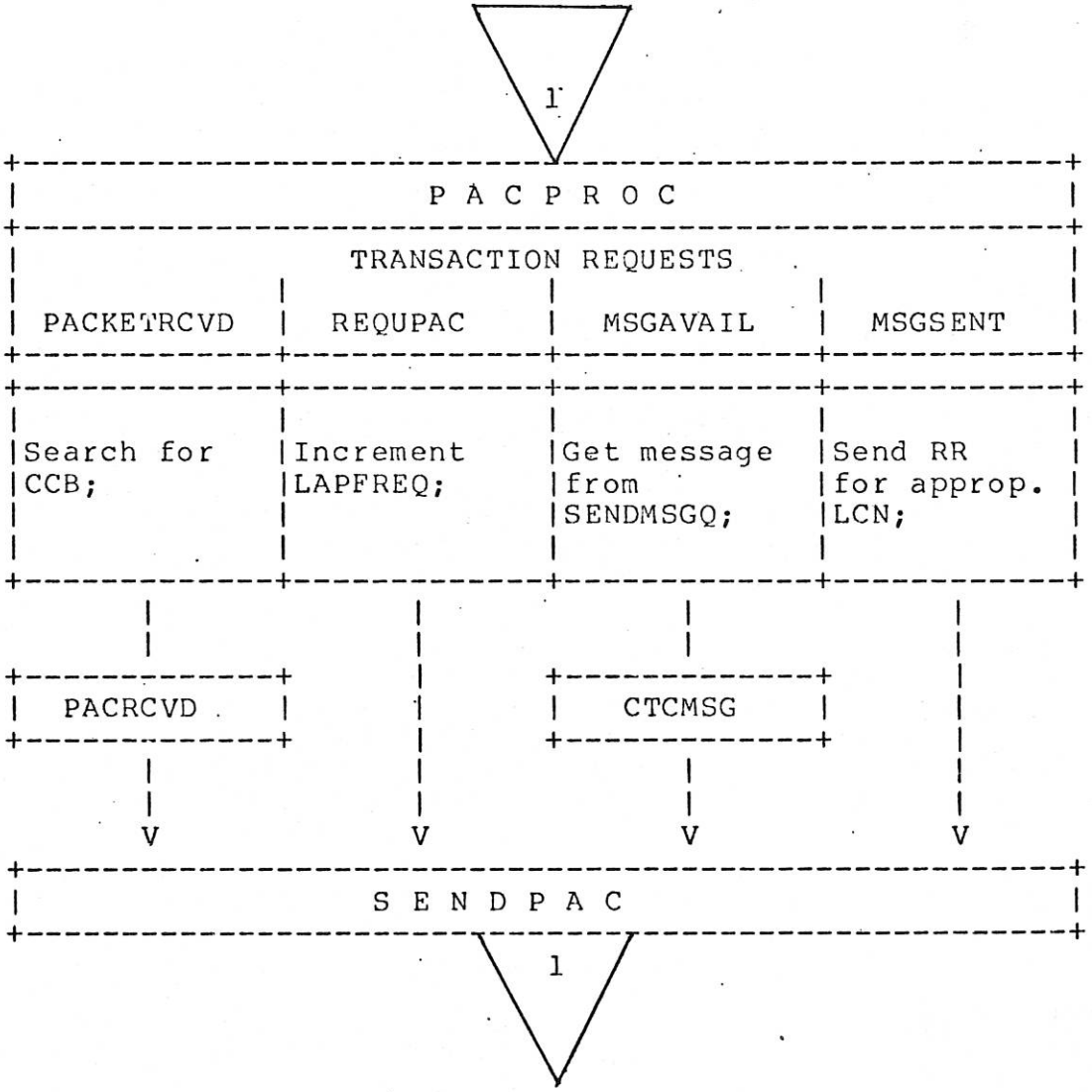


PACAVAIL
Dieser Transaction Request teilt dem LAP mit, dass in der Liste SENDPACQ ein zu sendendes X.25-Level 3-Paket anliegt.

MSGRCVD
In der Liste RCVMSGQ wird eine vom PAC empfangene bzw. generierte Nachricht abgelegt ,und der CTC davon mit MSGRCVD benachrichtigt.

3.3.3 PAC Prozessor Struktur

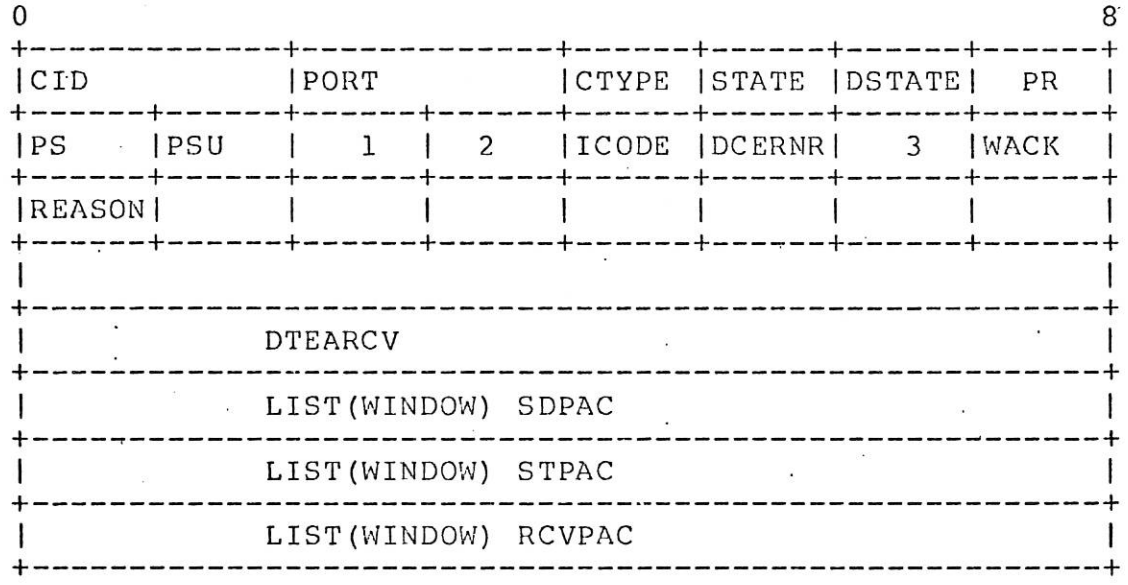
Es wurde versucht, logisch eine aehnliche Struktur aufzubauen wie sie vom LAP bereits bekannt ist. Dieses fiel aufgrund der Implementierung des PAC als Transaction Prozessor nicht schwer. (Fuer Einzelheiten siehe LAP)



3.3.4 Verwaltung der logischen Kanaele

Channel Control Block

Die gesamte fuer Verwaltung und Ueberwachung eines logischen Kanals (geschaltet oder permanent) notwendige Information enthaelt der CHANNEL CONTROL BLOCK CCB, der eine Laenge von 64 Bytes aufweist:



1 = PACTYPE, 2 = IRPTCTRL, 3 = SENDING

CID — Channel-Identifier = log.Kanalgruppennummer (LCGN) + log.Kanal (LCN)

PORT — falls "message link" aufgebaut: PORT gibt Benutzer des Kanals an

CTYPE — Art des Kanals: 0 = permanent, 1 = geschaltet

STATE — Zustand des Kanals:
1 = READY 2 = DTEWAIT
3 = DCEWAIT 4 = DATATRANS
5 = CALLCOLL 6 = DTECLEAR
7 = DCECLEAR 8 = DTEREST
9 = DCEREST

DSTATE — Unterzustand in STATE = DATATRANS
1 = FLOWCTRL 2 = DTERESET
3 = DCERESET

PR — Paket-Empfangsfolgezaehler P(R)

PS — Paket-Sendefolgezaehler P(S)

PSU — Untere Fenstergroesse: letztes nicht bestaetigte P(S)

PACTYPE Art des naechsten zu generierenden Pakets

IRPTCTRL INTERRUPT-Ueberwachung:
0 = es wurde kein INTERRUPT-Paket gesendet
1 = es wurde ein INTERRUPT-Paket gesendet

ICODE nimmt Datenbyte des INTERRUPT-Pakets auf

DCERNR gibt Empfangszustand der DCE an:
0 = DCE empfangsbereit
1 = DCE nicht empfangsbereit

SENDING Sendezustand des Kanals
0 = Kanal sendet nicht
1 = Kanal hat Paket(e) zu schicken

WACK notwendig fuer Steuerung des Interface Level 3/4
1 = sende DATACK
2 = sende RESUME

REASON Grund fuer Ausloesung, Ruecksetzen,...

DTEARCV DTE-Adresse des Empfaengers

Liste SENDQ enthaelt zu sendende Pakete

Liste SENTQ enthaelt gesendete, nicht bestaetigte Pakete

Liste RCVPAC enthaelt empfangene, vom Empfaenger noch nicht uebernommene Pakete

3.3.5 Organisation der CCB

Implementierungskriterien:

- Unabhaengigkeit von Kanalnummern-Vergabe durch die Deutsche Bundespost.
- Anzahl der Kanaele pro PAC maximal 10 (soll aber keine Einschraenkung sein, sondern theoretisch muessen 4096 bedient werden koennen).
- Einfuegen und Loeschen eines SVC (switched virtual circuit) moeglichst schnell.
- Lokalisierung des zu einem empfangenen Paket gehoerenden Kontrollblocks direkt, wenn moeglich. (Direkte Adressierung nur mit grossem Overhead moeglich, daher schnelles, indirektes Verfahren)

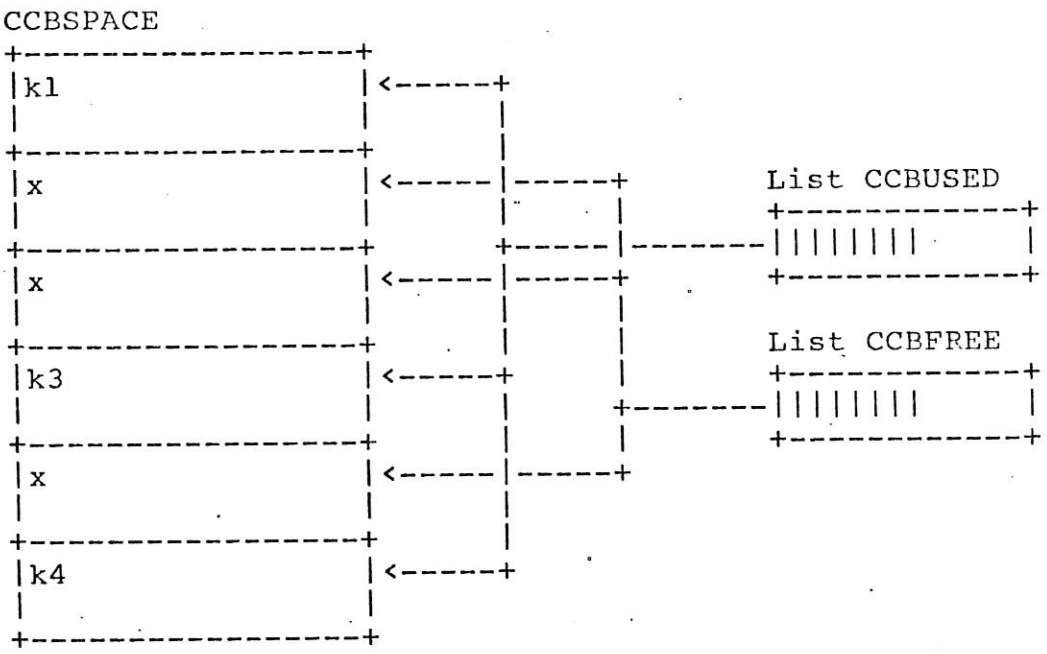
Entscheidung:
Implementierung mit mehreren zyklischen Listen (realisierbar mit List-Instruktion der Interdata 8/16) und einem Pool von Bloecken der Groesse 64 Bytes:

CCBSPACE sequentielle Liste, die 64-Bytes Bloecke enthaelt.

Liste CCBFREE enthaelt Adressen von Bloecken aus CCBSPACE,

verwendbar als CCB.

Liste CCBUSED enthaelt Adressen von benuetzten CCBs.



k_i = Suchschluessel i = (LCGN + LCN)

Daraus ergeben sich sehr einfache Aktionen fuer die Operationen:

(a) Einfuegen eines CCB
Diese Situation tritt auf, wenn ein "Incoming Call"-Paket mit einem SVC im "Ready"-Status empfangen wird.

Aktion:
Wird der angesprochene SVC von der DTE akzeptiert, so hole den Kopf-Eintrag aus der Liste CCBFREE. Dieser Eintrag enthaelt einen Zeiger auf den zu verwendenden CCB. Initialisiere CCB.

(b) Loeschen eines CCB:
Nur wenn ein SVC geloescht wird, muss auch der zugehoerige CCB geloescht werden.

Aktion:
Entferne den Zeiger auf den CCB aus der Liste CBBUSED und bewahre ihn in der Liste CCBFREE auf. (Er kann als CCB-Pointer fuer einen anderen SVC weiterverwendet werden)

(c) Suchen eines zugehoerigen CCB:
Bevor ein empfangenes Paket weiterverarbeitet werden kann, ist es immer erst notwendig, den zugehoerigen CCB

zu suchen (egal ob PVC oder SVC).

Aktion:
Durchsuche die Liste CCBUSED nach einem Zeiger, dessen CCB dem angesprochenen logischen Kanal zugeordnet ist.

Bem.:
Der Aufwand fuer das staendige Suchen in der Liste CCBUSED ist sehr gering, da die Liste kurz ist. (Ein Direktverfahren durch Hash-Codierung duerfte wahrscheinlich mehr Aufwand erfordern)

3.3.6 Multiplex-Algorithmus (Procedure SENDPAC)

Ein einfacher, aber trotzdem effizienter Algorithmus, der dafuer sorgt, dass die einzige physikalische Leistung den zahlreichen virtuellen X.25-Kanaelen zugewiesen wird, ist eine verallgemeinerte Version des "ROUND-ROBIN"-Verfahrens: Ist die physikalische Leitung fuer Datenuebertragung bereit (zu sehen an der Variablen LAPFREQ), kann der an oberster Stelle in der Warteschlange stehende logische Kanal dem LAP die Anfangsadresse eines zu uebertragenden X.25-Paketes in der Liste SENDPACQ mitteilen. Hat dieser virtuelle Kanal noch weitere Pakete zu uebergeben, so wandert er an das Ende der Warteschlange (Liste CHANNEL), um zu warten, bis alle vor ihm stehenden Kanaele aktiviert werden.

Eine weitere Aufgabe des Multiplex-Algorithmus besteht darin, fuer jeden aktiven virtuellen Kanal entscheiden zu muessen, welches Paket an die physikalische Leitung (bzw. an den LAP) zu uebergeben ist. Dabei haben Kontroll-Pakete generell Prioritaet vor Datenpaketen.
Welches Kontroll-Paket zu generieren ist, ersieht man aus der Variablen FRAMETYPE, ob ein Datenpaket zu senden ist aus der Liste SENDQ, die im positiven Fall nicht leer ist und direkt den Zeiger auf das Paket enthaelt.

Sollte dieses Datenpaket aus irgendeinem Grunde nicht uebergeben werden duerfen (moeglich: Fenster voll, DCE nicht empfangsbereit, Zustand nicht Datentransfer), wird dieser virtuelle Kanal deaktiviert und der naechste in der Liste CHANNEL anstehende aktiviert bzw. der Algorithmus stoppt, falls die Liste mit den Kanaelen leer ist.

Bem.: Der Algorithmus gewaehrleistet, dass jeder Kanal in der Liste CHANNEL hoechstens einmal pro Aufruf aktiviert wird. Das bedeutet, dass in den Faellen, dass einige Kanaele aus o.g. Gruenden keine Datenpakete uebertragen duerfen, der Algorithmus endet, sobald die Liste CHANNEL einmal abgearbeitet wurde.

3.3.7 Empfang eines X25/3 Kontroll-Paketes

Ist Bit 0 des X.25 Level 3 Typ-Byte identisch 1, wird von der Global Procedure PACRCVD die Local Procedure X25TERM aufgerufen, da dieses Bit 0 = 1 ein Level 3 Kontrollpaket kennzeichnet. Um eine Struktur aufzubauen, bei der "case-statements" die Abarbeitung eines empfangenen Kontrollpakets in einem bekannten Zustand des betreffenden

virtuellen Kanals organisieren, wird - analog zum LAP-Prozessor - eine Funktion f(t,s) definiert:

f(t,s): = g(t * 9 + (s-1))

t Typ des empfangenen Kontrollpakets (da t aus dem Type-Byte mit #1E ausmaskiert wird, treten Ueberlappungen fuer RESET und RESTART-Pakete auf, die durch Zuweisung eines geeigneten t geloest werden)

s Zustand des virtuellen Kanals
(1 = READY
2 = DTEWAIT
3 = DCEWAIT
4 = DATATRANS
5 = CALLCOLL
6 = DTECLEAR
7 = DCECLEAR
8 = DTEREST
9 = DCEREST)

g(x) definiert von der Tabelle ACTION

Der von der Funktion eindeutig bestimmte Funktionswert kennzeichnet ein case-statement, in dem die entsprechenden Aktionen ablaufen.

A C T I O N

type \ state	1	2	3	4	5	6	7	8	9
RR	3,	3,	3,	10,	3,	11,	3,	12,	13
INTERRUPT	3,	3,	3,	18,	3,	11,	3,	12,	13
RNR	3,	3,	3,	19,	3,	11,	3,	12,	13
IRPT CONF.	3,	3,	3,	20,	3,	11,	3,	12,	13
REJ	3,	3,	3,	21,	3,	11,	3,	12,	13
INC. CALL	1,	2,	3,	3,	3,	4,	3,	12,	13
RESET IND.	3,	3,	3,	22,	3,	11,	3,	12,	13
CALL CONN.	3,	5,	3,	3,	5,	4,	3,	12,	13
RESET CONF	3,	3,	3,	23,	3,	11,	3,	12,	13
CLEAR IND.	6,	6,	6,	6,	6,	7,	8,	12,	13
REST. IND.	14,	14,	14,	14,	14,	14,	14,	15,	16
CLEAR CONF	9,	9,	9,	9,	9,	7,	9,	12,	13
REST. CONF	17,	17,	17,	17,	17,	17,	17,	15,	17
ILLEGAL	9(24)								

3.3.8 X.25-DTE-Version

Da im "Benutzerhandbuch DATEX-P" nicht explizit das DTE-Verhalten gemaess X.25 beschrieben ist, soll das an dieser Stelle anhand von Tabellen nachgeholt werden, wobei auf Protokoll-Details nicht eingegangen wird.

Konvention:
Das Symbol

```
+--------+
|      CR|   ist zu lesen:
|        |      Sende "Clear-Request"-
|p6      |      Paket, gehe ueber in den
+--------+      Zustand p6;
```

STATE OF DTE / PACKET FROM DCE	READY p1	DTE WAITING p2	DCE WAITING p3	DATA TRANSFER p4	CALL COLLISION p5	DTE CLEAR REQUEST p6	DCE CLEAR INDICATION p7
INCOMING CALL	CA p4 / CR p6	* / CR p5	CR p6	CR p6	CR p6	--	CR p6
CALL CONNECTED	CR p6	CR p4	CR p6	CR p6	CR p4	--	CR p6
CLEAR INDICATION	CC p1	CC p1	CC p1	CC p1	CC p1	* / -- p1	
DCE CLEAR CONFIRMATION	CR p6	CR p6	CR p6	CR p6	CR p6	p1	CR p6
DATA, RESET, IRPT, FLOW CONTROL	CR p6	CR p6	CR p6	see Table 2	CR p6	--	CR p6

Fig. 1 : Actions taken by the DTE on receipt of packets in a given state of the DTE/DCE-Interface: Call Setup and Clearing

	p4		
STATE OF DTE / PACKET FROM DCE	FLOW CONTROL READY d1	DTE RESET REQUEST d2	DCE RESET REQUEST d3
DCE RESET INDICATION	RESC d1	* d1	RESC d1
DCE RESET CONFIRM-ATION	RESR d2	 d1	RESR d2
DATA, INTERRUPT, FLOW CONTROL	 d1	--- 	RESR d2

Fig. 2 : Actions taken by the DTE on receipt of packets in a given state of the DTE/DCE-Interface: Flow Control, Data Tranfer

STATE OF DTE / PACKET FROM DCE	p1,...,p7 d1,d2,d3	DTE RESTART REQUEST r1	DCE RESTART INDICATION r2
DATA,RESET INTERRUPT, FLOW CONTROL Packets	see Table 1,2	---	RESTR r1
CALL Setup CLEARING Packets	see Table 1	---	RESTR r1
RESTART INDICATION	RESTC p1(SVC) d1(PVC)	* p1(SVC) d1(PVC)	---
RESTART CONFIRM-ATION	RESTR r1	 p1(SVC) d1(PVC)	RESTR r1

Fig. 3 : Actions taken by the DTE on receipt of packets in a given state of the DTE/DCE-Interface: Restart

Legende:

CA =Call accepted	RESTR=Restart request
CR =Clear request	RESTC=Restart confirmation
CC =Clear confirmation	* =Collision
RESC=Reset confirmation	-- =ignore packet
RESR=Reset request	

3.4 Puffermanagement

3.4.1 Ueberblick

Puffer sind exklusiv verwendbare Betriebsmittel, die in AMOS/3 nur von Prozessoren angefordert werden (CTC,PAC,LAP). Das Betriebssystem stellt dafuer zwei "Supervisor Calls" zur Verfuegung:
ALLOCATE uebergibt dem anfordernden Prozessor einen Puffer (bei temporaerer Nichtverfuegbarkeit wird der Prozessor in den Wartezustand versetzt bis Verfuegbarkeit vorliegt). Mit RELEASE gibt der Processor nicht mehr benoetigte Puffer an das System zurueck.
(siehe AMOS/3 ADN D317)

Aus Gruenden der Effizienz (kein Kopieren von Pufferinhalten noetig) wurde hier im X.25-TC auf die exklusive Verwendbarkeit von Puffern verzichtet: Prozessoren muessen angeforderte und verwendete Puffer nicht notwendigerweise auch selbst wieder an den Systempool zurueckgeben, sondern es ist (zumindest bei Datenframes) die Regel, dass gefuellte Puffer an kommunizierende Prozessoren weitergereicht werden.

Um auch bei diesem Verfahren einen einwandfreien Betrieb gewaehrleisten zu koennen, ist aeusserste Disziplin im Umgang mit Puffern von den Prozessoren zu fordern. Ausserdem muessen gemeinsam verfuegbare Listen fuer den Pufferaustausch definiert werden, und die Beteiligten muessen genau wissen, wann ein Puffer mit RELEASE freizugeben ist.

3.4.2 Ablauf der Pufferverwaltung

(siehe Fig. 3.4.2)

Sendeseite

(a) PAC uebernimmt eine Nachricht vom CTC (die entsprechende Pufferadresse findet er in der Liste SENDMSGQ) bzw. besorgt sich mit ALLOCATE einen freien Puffer, falls er ein Kontrollpaket zu generieren hat. Die Adresse des gefuellten Puffers legt er in der Liste SENDPACQ ab und benachrichtigt LAP mit dem Transaction-Request PACAVAIL.

(b) LAP holt daraufhin die Paketadresse aus SENDPACQ ab und uebertraegt sie nach SENDQ.

(c) Wird in der Procedure SENDFRAME festgestellt, dass ein I-Frame gesendet werden darf, wird die Pufferadresse aus der Liste SENDQ in den PCB des HDLCSENDER kopiert, der fuer die physikalische Uebertragung des Frames sorgt. Fuer zu generierende Supervisory-Frames wird

mittels ALLOCATE ein freier Puffer besorgt.

(d) I-Frames aus der Liste SENDQ (es sind nur solche eingetragen) werden in die Liste SENTQ eingetragen, bevor sie gesendet werden (um damit die Pufferadressen fuer Bestaetigungen und evtl. Wiederholungen noch verfuegbar zu haben).

(e) Durch den Empfangsfolgezaehler N(R) bestaetigte I-Frames werden aus der Liste SENTQ entfernt und die zugehoerigen Puffer mit RELEASE freigegeben.

Empfangsseite:

(f) Die Pufferadressse eines empfangenen Frame traegt der HDLCRECEIVER in seine Liste RCVDFRAMEQ ein und informiert LAP mit dem Transaction-Request FRAMERCVD. Dieser leistet daraufhin Ersatz fuer den gefuellten Puffer (mittels ALLOCATE wird ein freier Puffer besorgt und die Liste RCVBUFQ des HDLCRECEIVER damit aufgefuellt) und verarbeitet den uebergebenen Frame. Handelte es sich um einen S-Frame, wird Puffer mit RELEASE dem System zurueckgegeben.

(g) Empfangene Information-Frames werden nach Ueberpruefung von Level2-Information an das Puffer -Interface zwischen LAP und PAC weitergereicht: die Pufferadresse wird in die Liste RCVDPACQ eingetragen.
Gleichzeitig wird PAC davon mit dem Transaction-Request PACKETRCVD benachrichtigt.

(h) PAC holt das Paket aus RCVDPAQ ab, verarbeitet es und gibt die Pufferadresse eines Kontrollpaketes mit RELEASE frei, waehrend er die eines Datenpaketes in die Liste RCVPAC uebergibt und den CTC benachrichtigt.
Diese Pufferadresse wird erst dann vom CTC freigegeben, wenn der Inhalt erfolgreich ueber die Kanalverbindung uebertragen wurde.

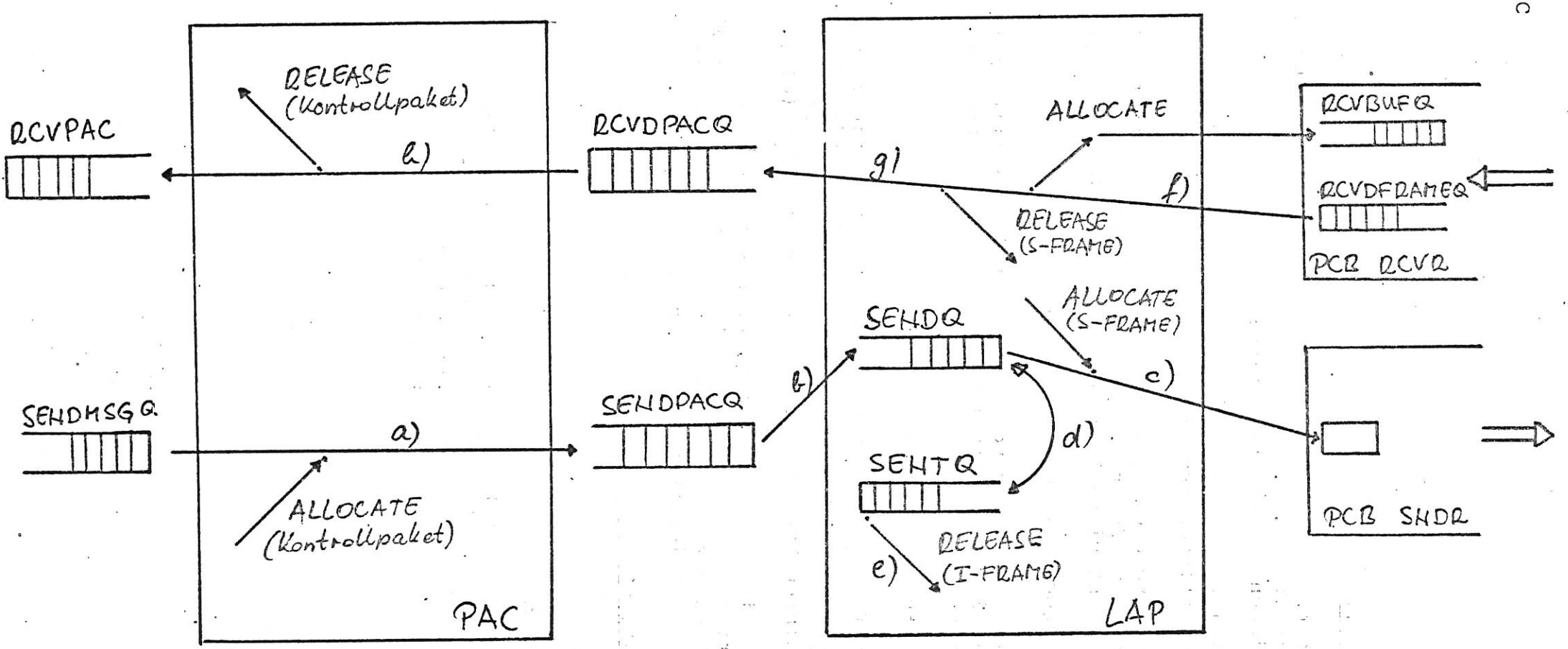


Fig. 3.4.2: PUFFER-MANAGEMENT PAC/LAP/DRIVER

4. FTZ-Zulassung

Mit Wirkung vom 02.08.1982 wurde dem hier beschriebenen X.25-Controller vom Zentralamt fuer Zulassungen im Fernmeldewesen die Zulassung als Datenendeinrichtung zum Datex-P-Netz der Deutschen Bundespost unter der FTZ-Nummer FTZ 02493 D erteilt.

Appendix 1: X.25-LEVEL 2

Format	Befehle	Meldungen	8 7 6 5 4 3 2 1 Codierung
I:Information Transfer-Datenübermittlung	I(Information -Daten)		N(R) P N(S) 0
S:Supervisory-Steuerung	RR(Receive Ready-Empfangsbereit)	RR(Receive Ready-Empfangsbereit)	N(R) P/F 000 1
	RNR(Receive Not Ready - Nicht empfangsbereit)	RNR(Receive Not Ready - Nicht emfangsbereit	N(R) P/F 010 1
	REJ(Reject-Wiederholungsaufforderung)	REJ(Reject - Wiederholungsaufforderung)	N(R) P/F 100 1
U:Unnumbered - Ohne Folgenummern		DM(Disconnected Mode - Abgebrochen)	000 F 111 1
	SABM(Set Asynchronous Balanced Mode - beginne gleichberechtigten Spontanbetrieb)		001 P 111 1
	DISC (Disconnect-Abbrechen)		010 P 001 1
		UA(Unnumbered Acknowledge - Bestätigung ohne Folgenummer)	011 F 001 1
		CMDR (Command Reject - Rückweisung des Befehls); FRMR (Frame Reject-Rückweisung des Blocks)	100 F 011 1

Tabelle 1.1.2.2.3.4-1: Befehle und Meldungen

The six major states of LAP B (symmetric) are:

0) SILENT - it is the other end's responsibility to initialize the link.

1) SENDING DM - the other end is being notified of a problem using the techniques allowed by LAP B.

2) SENDING SABM - the other end must be informed of a problem and the methods allowed by LAP B have been exhausted, so SABM is send.

3) SABM COLLISION - SABMs have crossed, it must be remembered that a UA is valid.

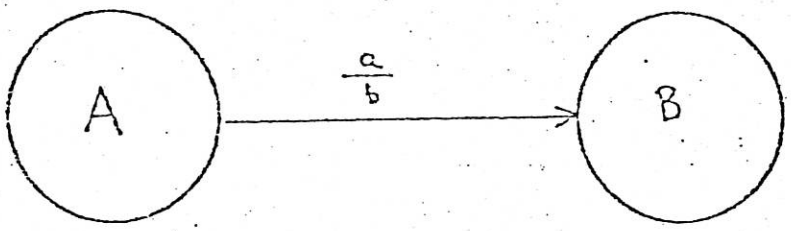
4) ABM - link is established.

5) FRMR SENT - an FRMR has been sent, so no further data can be.

STATE DIAGRAM CONVENTIONS

1) States are represented by circles containing a descriptive name and state number.

2) Input a in state A causing output b and transition to state B is represented by



3) $\frac{a}{-}$ indicates input a causes no output.

4) It is not intended that the state diagram be as complete as the state table.

5) T is a T1 timeout.

TG is more than N2 * T1 timeouts.

6) NEC - not expected command

NER - not expected response

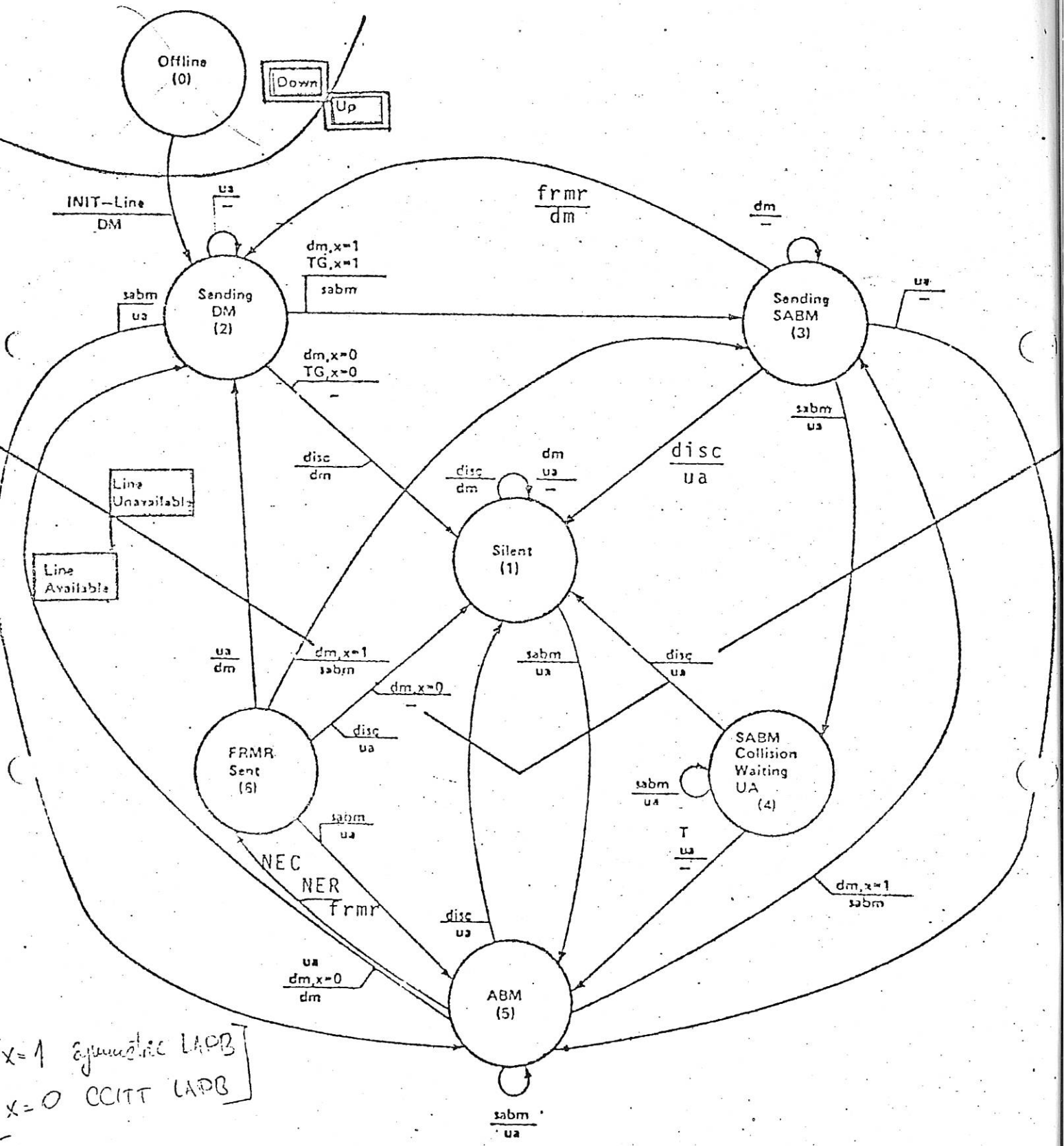


Figure 1.3.2.1-1: LAP-B Major State Transitions

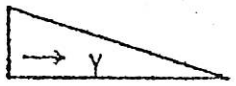
Table 1.3.2.1-1:

Actions taken by the DTE on receipt of frames
(LAPB, symmetric version)

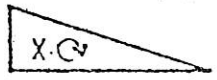
FRAME FROM THE DTE / STATE OF THE DTE, LEVEL 2	SILENT (1)	SENDING DM (2)	SENDING SABM (3)	SABM COL. WAITING UA (4)	ADM (5)	FRMR SENT (6)	NOTES ON THE P/F-BIT ⓐ
Command SABM	UA → 5	UA → 5	UA → 4	UA 4↻	UA 5↻	UA → 5	P-Bit = 1 ⇒ F-Bit = 1 P-Bit = 0 ⇒ F-Bit = 0
Command DISC	DM 1↻	DM → 1	UA → 1 DM →2	UA → 1 ⓑ ⓓ [process]	UA → 1 ⓑ ⓓ [process]	UA → 1 ⓓ FRMR 6↻	P-Bit = 1 ⇒ F-Bit = 1 P-Bit = 0 ⇒ F-Bit = 0
Command RR	DM 1↻ Ignore	DM 2↻ Ignore	Ignore	DM →2 [process]	ⓔ [process]	ⓔ Ignore 6↻	F-Bit = 1 F-Bit = ✗
Response RR	ⓔ Ignore	ⓔ Ignore	ⓔ Ignore	DM →2 [process]	ⓔ [process]	ⓔ Ignore	F-Bit = 1 F-Bit = 0
Command RNR	DM 1↻ Ignore	DM 2↻ Ignore	DM → 2 Ignore	ⓑ ⓓ [process]	ⓑ ⓓ [process]	ⓓ FRMR 6↻	P-Bit = 1 ⇒ F-Bit = 1 P-Bit = 0 ⇒ F-Bit = 0
Response RNR	ⓔ Ignore	ⓔ Ignore	ⓔ Ignore	DM →2 [process]	ⓔ [process]	ⓔ Ignore	F-Bit = 1 F-Bit = 0
Command REJ	DM 1↻ Ignore	DM 2↻ Ignore	DM → 2 Ignore	ⓑ ⓓ [process]	ⓑ ⓓ [process]	ⓓ FRMR 6↻	P-Bit = 1 ⇒ F-Bit = 1 P-Bit = 0 ⇒ F-Bit = 0
Response REJ	ⓔ Ignore	ⓔ Ignore	ⓔ Ignore	DM →2 [process]	ⓔ [process]	ⓔ Ignore	F-Bit = 1 F-Bit = 0

The state tables (Table 1.3.2.1-1, Table 1.3.2.1-2, Table 1.3.2.1-2a) describe the reactions of the DTE in more detail.

Conventions: in state X,

"  "

means the transition from X to Y (Y≠X),

"  "

means the transition from X to X.

STATE OF THE DTE LEVEL 2 / FRAME FROM THE DTE	SILENT (1)	SENDING DM (2)	SENDING SABM (3)	SABM COL. WAITING UA (4)	ABM (5)	FRMR SENT (6)	NOTES ON THE P/F-BIT (a)
Command I	DM → 1 ↺ ignore	DM → 2 ↺ ignore	DM → 2 ignore	(b) (d) [process]	[process] (c) (d)	FRMR (d) → 4 ↺	P-Bit=1 ⇒ F-Bit=1 P-Bit=0 ⇒ F-Bit=0
Response FRMR	ignore	ignore	DM → 2	ignore	SABM (P=1) (f) → 5 ↺	SABM (P=1) → 3	no difference between FRMR (F=1) and FRMR (F=0)
Response DM	ignore	SABM (P=1) → 3	ignore	ignore	SABM (P=1) → 3	SABM (P=1) → 3	no difference between DM (F = 1) and DM (F = 0)
Response UA	ignore	ignore	→ 5	→ 5	(f1) DM → 2	DM → 2	F-Bit = 1 (UA-Response should be for polled commands only)

TIME-OUTS						
T1	Timer not active	DM response retransmitted → 2 ↺	SABM (P=1) → 3 ↺	→ 5	I-frame retransmitted. (P=1) (f2) → 5 ↺	
N2*T1	Timer not active	SABM (P=1) → 3	SABM (P=1) → 3 ↺		DM → 2	

(Continued)

(continued)

Explanations:

(a) P/F-Bit correctly used, i.e. F-Bit set to 1 only received after having sent a frame with P-Bit set to 1

(b) Command processed and answered by a RR or RNR

(c) I-frame with P-Bit set to 1 processed and answered by a RR, RNR or REJ;
I-frame (P-Bit = 0) processed and answered as above, but additonally answered by an I-frame

(d) Independent of wether P-Bit = 1 or P-Bit = Ø

(e) Independent of wether F-Bit = 1 or F-Bit = Ø,
but F-Bit must be used correctly (see (a))

(f) For handling FRMR in state 'ABM' there is a special substate called 'FRMR RECEIVED': The DTE resets the link by sending SABM. Note that the DTE does not send a ' RESTART Indication' as it occurs in the transition from state 'SENDING SABM' to 'ABM'.

(f_1) Being in substate 'FRMR RECEIVED', UA only leeds to '"NORMAL" ABM'.

(f_2) Being in substate 'FRMR RECEIVED', another SABM will be transmitted to the DTE.

Table 1.3.2.1-2:

Actions taken by the DTE in receipt of incorrect frames (LAPB, symmetric Version)

STATE OF THE DTE LEVEL 2 / FRAME FROM THE DTE	SILENT (1)	SENDING DM (2)	SENDING SABM (3)	SABM COL. WAITING UA (4)	ADM (5)	FRMR SENT (6)	ERROR CONDITIONS
R [F=0]	not possible (DCE not polling)		FRMR → 2 rest: ignore	DM → 2; UA: FRMR DM; rest: ign.	N(R) evaluated	not possible	response with F-Bit=0 after have being polled
R [F=1]	ignore	DM → 3 rest: ign.	SABM (P=1)	not possible (DCE is polling)	FRMR DM → 3; rest → 2 DM; rest: ign.	SABM (P=1)	response with F-Bit=1 without being polled (b1)
Fr	FRMR (valid for all states) → 6						invalid or unknown command or response (d)
Fr [Test] Fr [Addr.]	frame will be discarded						frame with an incorrect FCS or an unknown address
I [xxx] (P=1)	1 Cr	2 Cr DM → 2		DM → 2	FRMR → 6	FRMR	I-frame with an invalid data field (c) (b2)
I [xxx] (P=0)	ignore	ignore	ignore	ignore → 6	FRMR → 6	FRMR	frame (except I-frame) with an invalid data field
Command [xxx]	see Table 1.3.2.1-2a						
Response [xxx]	see Table 1.3.2.1-2a						
I [N(S)]	N(S) not evaluated				REJ → 5 Cr	N(S) not evaluated	incorrect N (S)
Fr [N(R)]	N(R) not evaluated		FRMR → 6	FRMR → 6	FRMR → 6	N(R) not evaluated	frame with an invalid N(R)

(continued)

Explanations:

(a) Receiving FRMR in substate 'FRMR RECEIVED', the DTE will send DM and transit to state 'SENDING DM'

(b_1) (b_2) When both errors occur simultaneously than they are handled as shown by (b_2).

(c) Note that this is an I-frame exceeding the maximum frame size as defined in the Network. There is no relation to level 3 maximum packet length.

(d) An invalid command/response is
a frame containing a command code and an address indicating a response or
a frame containing a response code and an address indicating a command.