Technical Document - Confidential

GSM FAX & DATA SERVICES

MESSAGE SEQUENCE CHARTS

FAD

Document Number:	8411.203.98.200
Version:	0.3
Status:	Draft
Approval Authority:	
Creation Date:	1998-Apr-30
Last changed:	2015-Mar-08 by XGUTTEFE
File Name:	fad.doc

Important Notice

Change History

Date	Changed by	Approved by	Version	Status	Notes
1998-Apr-30	David J.G. Lewis		0.1		1
1998-Jul-28	David J.G. Lewis		0.2	Modified	
2003-May-13	XGUTTEFE		0.3	Draft	

Notes:

1. Initial version

Table of Contents

List of Figures and Tables

List of References

[ISO 9000:2000] International Organization for Standardization. Quality management systems - Fundamentals and vocabulary. December 2000

1.1 References

[1] Rec. T.4 Standardisation of group 3 facsimile apparatus for document transmission;
(CCITT-T.4, 1984)

[2] ITU-T Recommendation T.30; Series T: Terminal equipments and protocols for telematic services;
Procedures for document facsimile transmission in the general switched telephone network;
(ITU-T.30, 1996)

[3] ITU-T Recommendation T.31; Terminals for telematic services;
Asynchronous facsimile DCE control - service class 1
(ITU-T.31, 1995)

[4] ITU-T Recommendation T.32; Terminals for telematic services;
Asynchronous facsimile DCE control - service class 2
(ITU-T.32, 1995)

[5] Rec. T.35; Terminal equipment and protocols for telematic services;
Procedures for the allocation of CCITT definde codes for non-standard facilities;
(CCITT-T.35, 1991)

[6] ITU-T Recommendation V.25 ter; Series V: data communication over the telephone network;
Interfaces and voiceband modems; Serial asynchronous automatic dialling and control
(ITU-T V.25 ter, 1997)

[7] Rec. V.42 bis Data compression procedures for data circuit terminating equipment (DCE) using error correction procedures;
(CCITT-V.42 bis, 1990)

[8] Rec. V.110 (Blue book, Vol. VIII, Fascicle VIII.1) Support of data terminal equipments (DTEs) with V-series type interfaces by an integrated services digital network (ISDN);
(CCITT-V.110, 1988)

[9] European digital cellular telecommunications system (Phase 2);
GSM Public Land Mobile Network (PLMN) connection types;
(GSM 3.10, September 1994, version 4.3.1)

[10] European digital cellular telecommunications system (Phase 2);
Technical realisation of facsimile group 3 transparent;
(GSM 3.45, September 1995, version 4.5.0)

[11] Digital cellular telecommunications system (Phase 2);
Mobile radio interface layer 3 specification;
(GSM 4.08, November 1996, version 4.17.0)

[12] European digital cellular telecommunications system (Phase 2);
Rate adaptation on the Mobile Station - Base Station System (MS - BSS) Interface;
(GSM 4.21, May 1995, version 4.6.0)

[13] European digital cellular telecommunications system (Phase 2);
Radio Link Protocol (RLP) for data and telematic services on the Mobile Station - Base Station System (MS - BSS) interface and the Base Station System - Mobile-service Switching Centre (BSS - MSC) interface
(GSM 4.22, September 1994, version 4.3.0)

[14] European digital cellular telecommunications system (Phase 2);
Radio Link Protocol (RLP) for data and telematic services on the Mobile Station - Base Station System (MS - BSS) interface and the Base Station System - Mobile-service Switching Centre (BSS - MSC) interface
(Amendment prA1 for GSM 4.22, version 4.3.0)
(GSM 4.22, March 1995, version 4.4.0)

[15] European digital cellular telecommunications system (Phase 2);
General on Terminal Adaptation Functions (TAF) for Mobile Stations (MS);
(GSM 7.01, December 1995, version 4.10.0)

[16] European digital cellular telecommunications system (Phase 2);
Terminal Adaptation Functions (TAF) for services using asynchronous bearer capabilities;
(GSM 7.02, September 1994, version 4.5.1)

[17] European digital cellular telecommunications system (Phase 2);
Terminal Adaptation Functions (TAF) for services using synchronous bearer capabilities;
(GSM 7.03, September 1994, version 4.5.1)
[18] Digital cellular telecommunications system (Phase 2);
Use of Data Terminal Equipment - Data Circuit terminating Equipment (DTE - DCE) interface for Short Message Service (SMS) and Cell Broadcast Services (CBS);
(GSM 7.05, November 1996, version 4.8.0)
[19] Digital cellular telecommunications system (Phase 2);
AT command set for GSM Mobile Equipment (ME)
(GSM 7.07, May 1996, version 4.1.0)
[20] Digital cellular telecommunication system (Phase 2);
Mobile Station (MS) conformance specification;
Part 1: Conformance specification
(GSM 11.10-1, November 1996, version 4.17.0)
[21] Digital cellular telecommunications system (Phase 2);
Mobile Station (MS) conformance specification;
Part 2: Protocol Implementation Conformance Statement (PICS)
proforma specification
(GSM 11.10-2, May 1996, version 4.15.0)
[22] Digital cellular telecommunications system (Phase 2);
Mobile Station (MS) conformance specification;
Part 3: Layer 3 (L3) Abstract Test Suite (ATS)
(GSM 11.10-3, November 1996, version 4.17.0)
[23] Proposal for Rate Adaptation implemented on a DSP;
(C. Bianconi, Texas Instruments, January 1998, version 1.0)
[24] MCU-DSP Interfaces for Data Applications;
Specification S844
(C. Bianconi, Texas Instruments, March 1998, version 0.1)
[25] Users Guide
6147.300.96.100; Condat GmbH
[26] Service Access Point RA
8411.100.98.100; Condat GmbH
[27] Service Access Point RLP
8411.101.98.100; Condat GmbH
[28] Service Access Point L2R
8411.102.98.100; Condat GmbH
[29] Service Access Point FAD
8411.103.98.100; Condat GmbH
[30] Service Access Point T30
8411.104.98.100; Condat GmbH
[31] Service Access Point ACI
8411.105.98.100; Condat GmbH
[32] Message Sequence Charts RLP
8411.201.98.100; Condat GmbH
[33] Message Sequence Charts L2R
8411.202.98.100; Condat GmbH
[34] Message Sequence Charts FAD
8411.203.98.100; Condat GmbH
[35] Message Sequence Charts T30
8411.204.98.100; Condat GmbH
[36] Message Sequence Charts ACI
8411.205.98.100; Condat GmbH
[37] Proposal for Fax & Data Integration; March 1998
8411.300.98.100; Condat GmbH
[38] Test Specification RLP
8411.401.98.100; Condat GmbH
[39] Test Specification L2R
8411.402.98.100; Condat GmbH

[40] Test Specification FAD
8411.403.98.100; Condat GmbH
[41] Test Specification T30
8411.404.98.100; Condat GmbH
[42] Test Specification ACI
8411.405.98.100; Condat GmbH
[43] SDL Specification RLP
8411.501.98.100; Condat GmbH
[44] SDL Specification L2R
8411.502.98.100; Condat GmbH
[45] SDL Specification FAD
8411.503.98.100; Condat GmbH
[46] SDL Specification T30
8411.504.98.100; Condat GmbH
[47] SDL Specification ACI
8411.505.98.100; Condat GmbH
[48] Technical Documentation RLP
8411.701.98.100; Condat GmbH
[49] Technical Documentation L2R
8411.702.98.100; Condat GmbH
[50] Technical Documentation FAD
8411.703.98.100; Condat GmbH
[51] Technical Documentation T30
8411.704.98.100; Condat GmbH
[52] Technical Documentation ACI
8411.705.98.100; Condat GmbH

1.2 Abbreviations

ACI	AT Command Interpreter
AGCH	Access Grant Channel
AT	Attention sequence "AT" to indicate valid commands of the ACI
BCCH	Broadcast Control Channel
BCS	Binary Coded Signals
BS	Base Station
BSIC	Base Station Identification Code
C/R	Command/Response
C1	Path Loss Criterion
C2	Reselection Criterion
CBCH	Cell Broadcast Channel
CBQ	Cell Bar Qualify
CC	Call Control
CCCH	Common Control Channel
CCD	Condat Coder Decoder
CKSN	Ciphering Key Sequence Number
CRC	Cyclic Redundancy Check
DCCH	Dedicated Control Channel
DISC	Disconnect Frame
DL	Data Link Layer
DM	Disconnected Mode Frame
DTX	Discontinuous Transmission
EA	Extension Bit Address Field
EL	Extension Bit Length Field
EMMI	Electrical Man Machine Interface
EOL	End Of Line
F	Final Bit
F&D	Fax and Data Protocol Stack
FACCH	Fast Associated Control Channel
FHO	Forced Handover
GP	Guard Period
GSM	Global System for Mobile Communication
HDLC	High level Data Link Control
HISR	High level Interrupt Service Routine
HPLMN	Home Public Land Mobile Network
I	Information Frame
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
ITU	International Telecommunication Union
IWF	Interworking Function
Kc	Authentication Key
L	Length Indicator
LAI	Location Area Information
LISR	Low level Interrupt Service Routine
LPD	Link Protocol Discriminator
M	More Data Bit
MCC	Mobile Country Code
MM	Mobility Management
MMI	Man Machine Interface
MNC	Mobile Network Code

MS	Mobile Station
MSG	Message phase in the GSM 3.45 protocol
N(R)	Receive Number
N(S)	Send Number
NCC	National Colour Code
NECI	New Establishment Causes included
OTD	Observed Time Difference
P	Poll Bit
P/F	Poll/Final Bit
PCH	Paging Channel
PCO	Point of Control and Observation
PDU	Protocol Description Unit
PL	Physical Layer
PLMN	Public Land Mobile Network
RACH	Random Access Channel
REJ	Reject Frame
RNR	Receive Not Ready Frame
RR	Radio Resource Management
RR	Receive Ready Frame
RTD	Real Time Difference
RTOS	Real Time Operating System
SABM	Set Asynchronous Balanced Mode
SACCH	Slow Associated Control Channel
SAP	Service Access Point
SAPI	Service Access Point Identifier
SDCCH	Slow Dedicated Control Channel
SIM	Subscriber Identity Module
SMS	Short Message Service
SMSCB	Short Message Service Cell Broadcast
SS	Supplementary Services
T.4	CCITT Standardisation for Document coding of Group 3 Facsimile Apparatus
TAP	Test Application Program
TCH	Traffic Channel
TCH/F	Traffic Channel Full Rate
TCH/H	Traffic Channel Half Rate
TDMA	Time Division Multiple Access
TE	Terminal Equipment - e. g. a PC
TMSI	Temporary Mobile Subscriber Identity
UA	Unnumbered Acknowledgement Frame
UI	Unnumbered Information Frame
V(A)	Acknowledgement State Variable
V(R)	Receive State Variable
V(S)	Send State Variable
VPLMN	Visiting Public Land Mobile Network

1.3 Terms

Entity:	Program which executes the functions of a layer
Message:	A message is a data unit which is transferred between the entities of the same layer (peer-to-peer) of the mobile and infrastructure side. Message is used as a synonym to protocol data unit (PDU). A message may contain several information elements.
Primitive:	A primitive is a data unit which is transferred between layers on one component (mobile station or infrastructure). The primitive has an operation code which identifies the primitive and its parameters.
Service Access Point:	A Service Access Point is a data interface between two layers on one component (mobile station or infrastructure).

2 Overview

The Protocol Stacks are used to define the functionality of the GSM protocols for interfaces. The GSM specifications are normative when used to describe the functionality of interfaces, but the stacks and the subdivision of protocol layers does not imply or restrict any implementation.

The protocol stack for fax and data transmission consists of several entities. Each entity has one ore more service access points, over which the entity provides a service for the upper entity. The entity, which is described in this document, is coloured grey in the following figure :

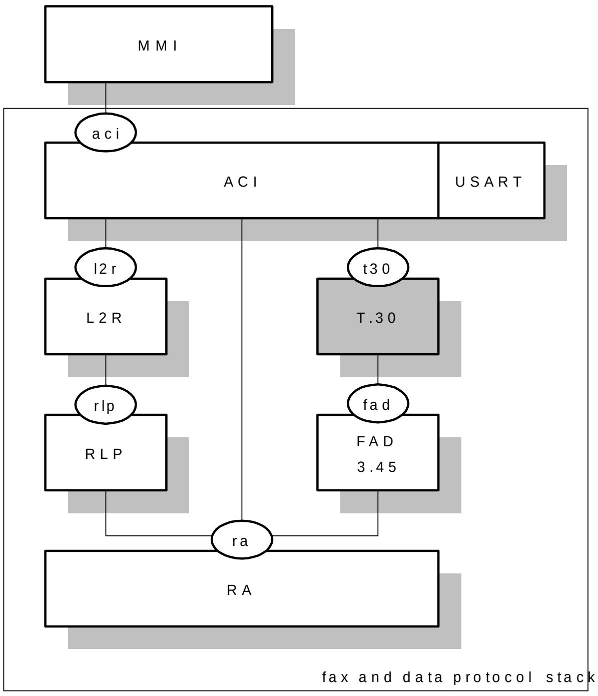

Figure 2-1: Architecture of the fax and data protocol stack

The information units passed via the SAPs are called primitives and consists of an operation code and several parameters. See the Users Guide for details.

The entities of the fax and data protocol stack are:

2.1 RA - Rate Adaptation

This entity performs an adaptation between an asynchronous or synchronous data stream with several bit rates on to the fixed bit rate used at the TCH. This is performed by the rate adaptation functions RA1' and RA0 described in GSM 04.21.

2.2 RLP - Radio Link Protocol

This entity provides a Layer 2 protocol for asynchronous reliable data transfer as specified in GSM 04.22. It includes error correction, sequence numbers and a mechanism for repeating corrupted and lost messages.

2.3 L2R - Layer 2 Relay Functionality

The L2R provides relay functions in order to adapt the character-oriented data received from the TE via USART to the bit-oriented RLP protocol.

2.4 FAD 03.45 - Fax Adaptation Protocol

The fax adaptation protocol, as specified in GSM 03.45, provides synchronisation with the BCS and MSG modems of the peer entity. It uses byte repetition in conjunction with a voting algorithm to handle corruption on the TCH data stream. The non-transparent fax protocol in accordance with GSM 03.46 is not part of this implementation.

The fax adapter enables T.30 to send BCS at 300 BPS and T.4 MSG in 2400, 4800, 7200 and 9600 BPS.

2.5 T.30 - Fax Protocol Entity

The protocol uses binary coded signals packed in HDLC frames to set up and release a connection in the message phase of the FAX transmission. This entity is specified in the ITU-T.30. The main tasks of this unit are:

- Building the HDLC frames with CRC.
- Performing bit stuffing/de-stuffing.
- Executing a sequence of 5 phases: 1.) set up, 2.) pre-message procedures, 3.) transmission/reception, 4.) post message procedures, 5.) waiting for call release.

2.6 ACI - AT Command Interpreter

The ACI is specified in GSM 07.07. It is responsible for call establishment via the GSM voice protocol stack and terminal adaptation for asynchronous transparent character-oriented data transmission. The ACI is able to receive AT commands and send the replies over the USART driver to a remote PC. This makes it possible to control the voice and data protocol stack from a remote application running on a PC. The ACI also provides a unique interface for an internal MMI in the MS.

2.7 USART - Universal Synchronous Asynchronous Receiver Transmitter Driver

The USART is a hardware component that facilitates a connection between the mobile station and terminal equipment (e.g. a PC). This interface uses some of the circuits described in V.24.

The data exchange provided by this unit is serial and asynchronous (synchronous communication is not in the scope of this document). A driver that uses interrupts to manage a circular buffer for the sending and receiving direction is necessary in order to use this component in the F&D. The driver has to be able to perform flow control.

3 Facsimile Group 3 Transparent ([10] §3)

3.1 Fax Adaptor functionality ([10] §3.1)

The functionally of Fax Adaptor is as follows:

- monitoring and where necessary, manipulation of the T.30 protocol
- overall control of the adaptor
- connection over the synchronous V.24 interface to the MT as described in GSM 07.03
- where necessary, autocalling and autoanswering functions according to V.25bis

3.2 Connection types ([10] §4)

Figure 3-1 shows the connection elements attributes applicable to this Teleservice (note), adapted from GSM 03.10.

Protocol type of figure 6 of GSM 03.10	Access to TAF of the MS	Radio interface connection element	Intermediate rate RA1 to RA2	BS-MSC/IWF connection element
Model 6:Speech	----	Speech/GSM	----	CCITT A-law
Model 5:	C/D/S UDI	C/D/S UDI	C/D/S UDI	C/D/S UDI
Facsimile	-14.4 kbit/s	-14.5 kbit/s	-16 kbit/s	-64 kbit/s
Group 3	-9.6 kbit/s	-12.0 kbit/s	-16 kbit/s	-64 kbit/s
	-4.8 kbit/s	-6.0 kbit/s	-8 kbit/s	-64 kbit/s`
	-2.4 kbit/s	-3.6 kbit/s	-8 kbit/s	-64 kbit/s
Model 5b	C/D/S UDI	C/D/S UDI	C/D/S UDI	C/D/S UDI
Facsimile	-14.4 kbit/s	14.5 or 2X12 kbit/s	16 or 2X16 kbit/s	-64 kbit/s
Group 3	-9.6 kbit/s	12.0 or 2X6 kbit/s	16 or 2X8 kbit/s	-64 kbit/s
	-4.8 kbit/s	6.0 kbit/s	8 kbit/s	-64 kbit/s`
	-2.4 kbit/s	6.0 kbit/s	8 kbit/s	-64 kbit/s

Figure 3-1: Elements of connection types

NOTE: The highest Access Rate actually supported in this teleservice will be consistent with the highest Access Rate of the Transparent Bearer Service provided by the Network Operator

C = Circuit switched S = Synchronous

D = Full-duplex UDI = Unrestricted Digital Information

Fax modem rates	No. of sub-streams	Radio i/f rate	Intermediate	Padding Scheme
14.4 kbit/s	1 2	14.5 kbit/s 12 kbit/s	16 kbit/s 16 kbit/s	
12.0 kbit/s	1 2	14.5 kbit/s 12 kbit/s	16 kbit/s 16 kbit/s	FA(5D + S) FA(5D + S)
9.6 kbit/s	1 2	12 kbit/s 6 kbit/s	16 kbit/s 8 kbit/s	
7.2 kbit/s	1 2	12 kbit/s 6 kbit/s	16 kbit/s 8 kbit/s	FA(3D + S) FA(3D + S)
4.8 kbit/s	1	6 kbit/s	8 kbit/s	
2.4 kbit/s	1	6 kbit/s	8 kbit/s	

Figure 3-2

FA = Padding is performed in the FA

(nD + mS) means that m SYNC frames will be added every n'th DATA frame

The MS must support the combinations in Figure 3-1, restricted to what has been negotiated between the MS and the network.

All transitions from one combination in Figure 3-2 to another combination in the same table, must be supported by the MS, as long as the fax modem rate is kept constant or the transition of fax modem rate

is performed in steps of 2.4 kbit/s.

To grant full support to the CCITT Recommendation T.30, requiring different transmission speeds, the following strategy shall be implemented:

- PLMN provides for three Access Rate only, that is 9 600 bit/s, 4 800 bit/s, 2 400 bit/s
- radio channel modification procedures are used for switching between Speech phase and facsimile phase, as well as to select the suitable Access Rate (9 600/4 800/2 400 bit/s) within the facsimile phase, as resulting from the preliminary end-to-end negotiation between the terminals
- to transport Binary Coded Signalling (BCS) requiring a synchronous 300 bit/s bit-rate, speed conversion will be used at both the PLMN ends.

3.2.1 Interactions with T.30 ([10] §4.2)

The philosophy of this specification is to allow the T.30 protocol to pass transparently wherever possible, through the Fax Adaptors at both ends of the PLMN channel. Manipulations are only made to the protocol where necessary to overcome problems resulting from the differences between the PSTN and the GSM system. Basically, these problems fall into four categories:

- supporting facsimile on a digital connection type
- bit errors during transfer of BCS frames
- the need to change speed to reduce the impact of bit errors during transfer of Fax encoded Messages
- inability to support some features of T.30

3.2.1.1 Link control strategy ([10] §4.2.1)

Though the T.30 procedure is plain half-duplex, between the FAs at both the GSM PLMN ends a full-duplex mode connection will be established. Information transfer is structured in fixed length blocks, carrying either data received from the local modem, or idle synchronization patterns.

Establishment, maintenance and release of the data circuit is performed autonomously by each FA, by properly handling the physical interface to its local modem.

Due to the rigid timeout constrains in T.30 protocol procedure, which will be actually concatenated to the procedure defined here for the GSM connection part, a strictly forward only approach is adopted for data link control; that is no retransmission means are explicitly introduced to recover corrupted information blocks.

Although in the majority of cases the T.30 protocol can take care of lost frames and easily recover, there are some particularly critical points where two consecutive segments in the procedure, without any interposed acknowledgement, require different transmission speeds (BCS speed/Message speed).

In these cases, to grant that a correct change-over from one modem type to another takes place at both the PLMN ends, some additional protection is necessary in the GSM environment. This protection, actually at the beginning and at the end of the Fax page transmission, is given by means of a confirmation mechanism, whereby the originating FA waits while transmitting a fixed pattern until an explicit acknowledgement is received from the terminating FA. In these cases an interruption of the normal T.30 information flow may eventually result.

Aiming at minimizing any possible delay in the end-to-end transmission between the two facsimile apparatuses, the FAs will pass on information without error control of the message integrity (note). To this regard, however, it is worth underlining that an essential requirement in this procedure is the detection of some key messages exchanged between the terminals, carrying basic control information. Hence, in parallel to the primary information passing process, a continuous monitoring of the information flowing across the FA is necessary, to detect these messages, fully checked in their logical integrity.

NOTE: All V.21 modems in the FAs shall use the faster response time provided by CCITT Rec. V.21 (see table 2/V.21) for OFF to ON transition of CT109 and CT106.

In addition, as detailed below, some fields in some frames may need to be changed.

3.2.1.2 Speed conversion for BCS phases ([10] §4.2.2)

Given the signalling load caused by the Channel Mode Modify procedure, rather than changing the radio channel speed to 300 bit/s to carry the BCS frames, a speed conversion mechanism is exploited at both the PLMN ends, allowing to maintain the Channel rate of the Message phase (9600/4800/2400 bit/s) during BCS phases.

A speed conversion factor can be defined as:

- Fax Message speed (9600 or 4800 or 2400 bit/s)
- Standard BCS speed (300 bit/s)

Standard BCS speed assumes the value 8 or 16 or 32, depending on the actual Message speed negotiated between the terminals.

On the basis of the above speed conversion factor, in the originating FA up-conversion to Message speed will be performed according to the repetition algorithm specified for the BCS phase (see Section 3.3.1.3).

In the terminating FA, down-conversion to the BCS speed will require a decimation algorithm (see Section 3.3.1.4); this algorithm is assumed implementation dependent, and hence its actual definition is beyond the scope of this ETS.

As the actual Access Rate over the GSM-TCH may change throughout the Call, speed conversion factor needs to be updated in both FAs.

The FA/MT which does not have direct access to the GSM signalling channel (except for MT0 configuration), shall estimate the actual Access Rate established over the GSM-TCH from the effective Access Rate of the MT data interface (or any other means in case of MT0); this check shall be performed whenever a BCS frame is sent towards the Radio path after which a change for GSM-TCH Access Rate may be expected (e.g., DIS and DCS frames for Normal Fax mode, PPR and CTC frames for Error Correction Mode).

12000 and 7200 bit/s Facsimile Document Transfer ([10] §4.2.3.4)

End-to-end transfer of T.4 information at 12 000 and 7 200 bit/s can be supported, even if this Bearer Service is not provided by a GSM PLMN.
To this purpose a 14 400 bit/s or 9 600 bit/s Access Rate shall be used in the PLMN, by exploiting only part of the available bandwidth. The originating FA will add padding information to data received from the
MSG modem, before transmission over the GSM-TCH; the terminating FA will remove the same information to restore the net 12 000 or 7 200 bit/s continuous data stream to be transmitted over its local
MSG modem.
The relevant procedures are fully described in sections 5.2.3.4 and 5.2.3.5.

3.3 Terminal adaptation functions ([10] §5)

The Fax Adaptor is split in its main functional blocks:

- a BCS (300 bit/s) modem capability (half duplex);
- a MSG (14400/12000/9600/7200/4800/2400 bit/s) modem capability (half duplex);
- Mux/Demux capability, to connect the GSM traffic channel (full duplex) to either of these modems.

Looking at the overall Facsimile service architecture in GSM (Figure 5/03.45), half-duplex transmission is initiated in either side of the connection by the actual request on the physical interface (CT109 ON on the local modem), is then continued on the PLMN traffic channel (TCH), and finally terminated on the appropriate modem at the remote side.

On the basis of the above assumptions, considering the Fax Adaptor a finite state machine, the whole Adaptation process can be described by a set of 5 states:

- IDLE state, when the FA is just connected to the GSM-TCH, sending synchronizing patterns over the radio path
- BCS-REC state, when the FA is receiving data from the V.21 modem (BCS phases)
- BCS-TRA state, when the FA is transmitting data over the V.21 modem (BCS phases)
- MSG-REC state, when the FA is receiving data from the MSG modem (2400/4800/7200/9600 bit/s)
- MSG-TRA state, when the FA is transmitting data over the MSG modem

3.3.1 Basic protocol structure ([10] §5.2.2)

The protocol structure is based on a strictly synchronous approach, using 64 bit fixed length frames; that is each FA actually sends/receives information as 64 bit frames, in sequence and without interruption, during the whole duration of a call; the content of each frame depends on the specific state currently implemented.

The following set of frame types encompasses the full range of capabilities required:

- SYNC frame, explicitly designed to allow synchronization at the remote end, even in the adverse transmission environment like the cellular radio channel; it is a unique frame, used even as idle frame whenever there is no information to be sent over the radio path
- STATUS frame, intended to carry both state identification codes, along with state specific information; this frame has a unique structured format to allow synchronization checking at remote side; the actual information content is related to the specific state
- DATA frame which is fully unstructured, and carries Fax coded information during MSG phases

3.3.1.1 Frame formats ([10] §5.2.2.1)

To reduce complexity of finding synchronization over a 64 bit pattern, SYNC frames are obtained by concatenating two 32 bit sync codewords, the second one being the 1's complement notation of the first one.

The following codes shall be used for sync codewords:

- hex code 3E375096: first sync codeword
- hex code CIC8AF69: second sync codeword

To improve the probability of detecting synchronization, up to 3 errors are tolerated in each single sync codeword; even in this conditions, the false sync probability is quite negligible.

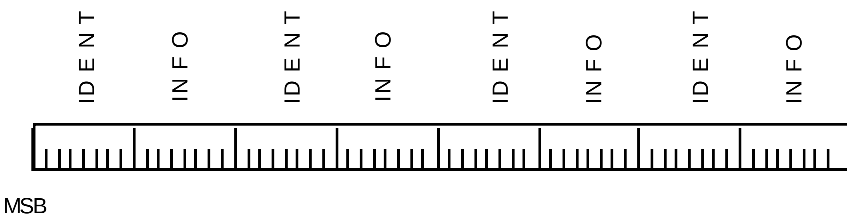

Figure 3-3: STATUS frame format

STATUS frames have an internal 8 bit modularity, where odd octets contain state identification codes (IDENT), and even octets contain status specific information (INFO). IDENT octets are split in two (four bit) fields, each one repeating the same code for status identification:

- hex code 1: BCS-REC state
- hex code 3: MSG-REC state
- hex code 4: MSG-TRA state

To improve the probability of detecting the correct state, up to 3 corrupted codes are tolerate in the total 8 instances repeated within a single STATUS frame. INFO octets contain the same code repeated 4 times within one frame; in particular:

- for BCS-REC state this octet contains 8 bits received from the BCS modem
- in all other states an idle code (hex code 0F) is inserted

DATA frames are 64 bit fully unstructured information blocks; the last DATA frame in a Message phase will be truncated to the actual length of the bit stream to be transmitted.

A general rule concerning all the above frames is that most significant bits are transmitted first; in addition, a basic First-In First-Out functionality will be implemented in the information passing process.

3.3.1.2 IDLE state ([10] §5.2.3.1)

In each FA this state implies a continuous transmission of SYNC frames towards the radio path, to allow frame synchronization at the remote end. This state is entered into immediately after the end of the synchronization process over GSM-TCH, and will be returned to whenever loss of synchronization is assumed.

3.3.1.3 BCS-REC state ([10] §5.2.3.2)

The basic function of FA in this state is transferring BCS information from local modem to GSM-TCH.

Transition to this state is triggered by CT109 ON condition of V.21 modem. Towards the radio path continuous transmission of STATUS frames is performed, according to the format described in Section 3.3.1.1, as soon as 8 bits at least are available from the modem; information received from GSM-TCH is ignored while CT109 (local modem) is in the ON condition. Every 8 bits received from the V.21 modem, the appropriate number of STATUS frames (1 or 2 or 4 depending on the current TCH access rate) will be originated. Following CT109 (local modem) OFF condition, padding bits (e.g. ls) will be inserted to assemble the last octet.

Monitoring the content of certain BCS frames is required; TCF and DCS frame requires a specific procedure (see Section 3.3.2).

3.3.1.4 BCS-TRA state ([10] §5.2.3.3)

The basic function of FA in this state is transferring BCS information from GSM-TCH to its local V.21 modem.

Transition to this state is triggered by reception of BCS-REC code from TCH. Towards the radio path continuous transmission of SYNC frames is performed, according to the format described in Section 3.3.1.1.

The information received from GSM-TCH, after proper decimation (likely a voting algorithm, implementation dependent), is passed to the V.21 modem. Upon detecting again a SYNC frame, the modem is turned OFF and a transition to the IDLE state is performed.

Monitoring the content of certain BCS frames is required; TCF and DCS frame require a specific procedure, described in Section 3.3.2.

In addition a supervisoring function is required to check maintenance of synchronization, by examining the inherently structured STATUS frame format. Loss of synchronization will be assumed in case of reception of unstructured STATUS frames, that is, when the repetition mechanism for both the IDENT and the INFO octets does not allow unambiguous result to the voting algorithm. In this case an estimate of the INFO octet value in the received STATUS frames is anyway passed on to the V.21 modem, up to a maximum BCS-TRA duration of 2.5 sec.; at this point, if the correct synchronization has not been recovered yet, the modem is turned OFF and a transition to the IDLE state is performed.

In case of loss of V.110 synchronization on the PLMN side, the FA shall transmit 0s towards the analogue interface, as long as no data is available.

3.3.1.5 MSG-REC state ([10] §5.2.3.4)

The basic function of FA in this state is transferring MSG information from its local modem to GSM-TCH.

Transition to this state is triggered by the MSG modem being trained. Towards the radio path continuous transmission of STATUS frames interleaved with SYNC frames is performed. Actual transfer of Fax coded data over the radio path can be initiated only after the specific acknowledgement is received from the remote side, that is reception of MSG-TRA indication in a STATUS frame, signifying the correct state transition. All data received from the MSG modem will be stored in the FA buffer, to be passed on to GSM-TCH (First-In First-Out mechanism) as soon as this confirmation message is received. Transmission is performed by means of unstructured DATA frames, aligned to the last SYNC or STATUS frame; in this phase, information received from GSM-TCH is ignored.

Following CT109 OFF condition of MSG local modem, after all buffered data are transmitted (the last DATA frame is truncated if necessary), a transition to the IDLE state is performed. In this state the FA waits (meanwhile BCS data received from the local modem are ignored) for a minimum of 5 transmitted SYNC frames, in order to indicate to the remote FA the end of the message.

While waiting for the acknowledgement re-synchronization shall be performed if necessary, following reception of unrecognisable SYNC frames or unstructured STATUS frames.

In case of 7.2 kbit/s MSG speed, a SYNC frame will be stuffed every 3 DATA frames, to produce the data stream at 9.6 kbit/s. The overall protocol structure will result in multi-frame entities (3 DATA frames followed by a single SYNC frame), continuously sent over the radio path.

3.3.1.6 MSG-TRA state ([10] §5.2.3.5)

The basic function of FA in this state is transferring MSG information from GSM-TCH to its local MSG modem.

Transition to this state is triggered by reception of MSG-REC code from TCH; towards the radio path continuous transmission of STATUS frames interleaved with SYNC frames is performed.

The MSG modem is trained and a timer (300 m/sec) corresponding to the round trip time over the GSM-TCH is started. After timeout, loss of synchronization in the information received from TCH, will be assumed as the first Fax coded DATA frame. From the receipt of Message on, continuous transmission of SYNC frames is performed.

All data received from the GSM-TCH will be stored in the FA buffer, to be passed on to the MSG modem (First-In First-Out mechanism) as soon as the modem training terminates (CT106 ON).
From this time on, re-synchronization will be attempted continuously; when an IDLE state is recognized again in the data stream received from the radio path, end of MSG phase will be assumed; then a transition to the IDLE state will be executed, where the FA will wait (ignoring data received from GSM-TCH) until the buffered information has been fully transmitted to the local MSG modem; the procedure will then proceed in the normal way.

In case of 7.2 kbit/s MSG speed, the above general rule applies as well (note). However multi-frame synchronization shall be checked in addition, to remove the SYNC frame stuffed by the originating FA to match the 9.6 kbit/s Access Rate over the PLMN. If necessary multi-frame re-synchronization shall be performed.

NOTE: In this case, no longer an isolate SYNC frame can be interpreted as the end of MSG phase (transition to the IDLE state of the originating FA at the remote side).

In case of loss of V.110 synchronization on the PLMN side, the FA shall transmit 0s towards the analogue interface, as long as no data is available.

3.3.2 DCS and TCF processing ([10] §5.2.4)

Transmission of TCF is performed end-to-end between the two Fax apparatuses, and requires in both FAs a specific routine triggered by DCS command.

As far as the originating FA is concerned, the general procedure as described above for MSG phase (MSG-REC state) applies, but no acknowledgement is required, both at the beginning and at the end, and so no buffering is necessary. That is, just following CT 109 ON condition of the MSG modem, unstructured DATA frames are sent over the radio path, aligned to previous frames; upon CT 109 OFF condition the last frame is sent (truncated if necessary), and a transition to the IDLE state is performed.

As far as the terminating FA is concerned, after passing DCS command and waiting for the appropriate delay (75 m/sec), transition to the MSG-TRA is executed; here modem training shall be pre-empted and, as soon as the modem is ready for sending (CT 106 ON), loss of synchronization on TCH will be assumed as the first DATA frame containing TCF information and will be passed to the MSG modem. After 1.5 sec. timeout (standard TCF duration), the MSG modem will be turned OFF and the IDLE state entered as usual. Fill information (i.e. logical 0s) will be sent on the local modem if real TCF bit stream is not available.

To ensure that the time gap between the DCS and TCF is within 75 +/-20 ms period as specified in CCITT/T.30, the training shall be pre-empted in the terminating FA, as defined above.

4 Protocol - Overview

4.1 Send Fax (9600, 4800, 2400), best case

```
        ACI                     T30     FAD             RA
        ||                        |     |
        ||                        |     |
        ||                        |     |
1       | RA_ACTIVATE_REQ         |     |                |
        *=============================================================================================>*
2       | RA_ACTIVATE_CNF         |     |                |
        *<=============================================================================================*
3       | T30_ACTIVATE_REQ (configs)    |                |              |
        *==============================>*                |              |
4       || FAD_ACTIVATE_REQ (configs)   |                |
        |*==============================>*               |
5       ||              | RA_DATA_IND (SYNC)             |              \
        ||        *<==============================*      |
6       ||                 | RA_READY_IND                |              | Nx
        ||        *<==============================*      |
7       ||              | RA_DATA_REQ (SYNC)             |              |
        ||        *==============================>*      /
        ||                        |     |
8       ||            | RA_DATA_IND (PREAMBLE)           |              \
        ||        *<==============================*      |
9       ||                 | RA_READY_IND                |              | Nx for
        ||        *<==============================*   | 1 sec
10      ||              | RA_DATA_REQ (SYNC)             |              |
        ||        *==============================>*      /
        ||                        |     |
11      ||         | RA_DATA_IND (STATUS[NSF...])        |              \ optional
        ||        *<==============================*      |
12      ||                 | RA_READY_IND                |              |Nx
        ||        *<==============================*      |
13      ||              | RA_DATA_REQ (SYNC)             |              |
        ||        *==============================>*      /
        ||                        |     |
14      ||          | RA_DATA_IND (STATUS[DIS])          |              \ mandatory
        ||        *<==============================*      |
15      ||                 | RA_READY_IND                |              | Nx
        ||        *<==============================*      |
16      ||              | RA_DATA_REQ (SYNC)             |              |
        ||        *==============================>*      /
        ||                        |     |
17      ||              | RA_DATA_IND (SYNC)             |
        ||        *<==============================*      |
18      ||                 | RA_READY_IND                |
        ||        *<==============================*      |
19      ||              | RA_DATA_REQ (SYNC)             |
        ||        *==============================>*
20      || FAD_DATA_IND (HDLC[NSF..DIS])                 |              |
        |*<=============================*                |
21      | T30_CAP_IND (caps, FMOD_SND)  |                |              |
        *<=============================*                 |              |
22      ||              | RA_DATA_IND (SYNC)             |              \
        ||        *<==============================*      |
23      ||                 | RA_READY_IND                |              |
        ||        *<==============================*      |
24      ||              | RA_DATA_REQ (SYNC)             |              |
        ||        *==============================>*      |
25      | T30_CAP_RES (caps, FMOD_SND)  |                |              |        |
        *==============================>*                |              |     | Nx for
26      || FAD_DATA_REQ (bcs, HDLC[NSS..DCS])            |              | 75 msec
        |*==============================>*               |              |
27      ||              | RA_DATA_IND (SYNC)             |              |
        ||        *<==============================*      |
28      ||                 | RA_READY_IND                |              |
        ||        *<==============================*      |
29      ||              | RA_DATA_REQ (SYNC)             |              |
        ||        *==============================>*      /
        ||                        |     |
30      ||              | RA_DATA_IND (SYNC)             |              \
        ||        *<==============================*      |
31      ||                 | RA_READY_IND                |              | Nx for
        ||        *<==============================*   | 1 sec
32      ||            | RA_DATA_REQ (PREAMBLE)           |              |
        ||        *==============================>*      /
        ||                        |     |
33      ||              | RA_DATA_IND (SYNC)             |              \ optional
        ||        *<==============================*      |
34      ||                 | RA_READY_IND                |              | Nx
        ||        *<==============================*      |
35      ||         | RA_DATA_REQ (STATUS[NSS...])        |              |
        ||        *==============================>*      /
        ||                        |     |
36      ||              | RA_DATA_IND (SYNC)             |              \ mandatory
        ||        *<==============================*      |
37      ||                 | RA_READY_IND                |              | Nx
        ||        *<==============================*      |
38      ||          | RA_DATA_REQ (STATUS[DCS])          |              |
        ||        *==============================>*      /
39      || FAD_DATA_CNF           |     |
        |*<=============================*                |
```

```
40      ||             | RA_DATA_IND (SYNC)            |            \
        ||       *<===========================*        |
41      ||                | RA_READY_IND               |            |
        ||       *<===========================*        |
42      ||             | RA_DATA_REQ (SYNC)            |            |
        ||       *===========================>*        |
43      || FAD_SND_TCF_REQ       |     |               |
        |*===========================>*                |            | Nx for
44      ||             | RA_DATA_IND (SYNC)            |            | 75 msec
        ||       *<===========================*        |
45      ||                | RA_READY_IND               |            |
        ||       *<===========================*        |
46      ||             | RA_DATA_REQ (SYNC)            |            |
        ||       *===========================>*        /
        ||                       |     |
47      ||             | RA_DATA_IND (SYNC)            |            \
        ||       *<===========================*        |
48      ||                | RA_READY_IND               |            | Nx for
        ||       *<===========================*    | 1,5 sec
49      ||          | RA_DATA_REQ (DATA[TCF])          |            |
        ||       *===========================>*        /
50      || FAD_SND_TCF_CNF       |     |
        |*<===========================*                |
51      ||             | RA_DATA_IND (SYNC)            |            \
        ||       *<===========================*        |
52      ||                | RA_READY_IND               |            | Nx
        ||       *<===========================*        |
53      ||             | RA_DATA_REQ (SYNC)            |            |
        ||       *===========================>*        /
        ||                       |     |
54      ||           | RA_DATA_IND (PREAMBLE)          |            \
        ||       *<===========================*        |
55      ||                | RA_READY_IND               |            | Nx for
        ||       *<===========================*     | 1 sec
56      ||             | RA_DATA_REQ (SYNC)            |            |
        ||       *===========================>*        /
        ||                       |     |
57      ||         | RA_DATA_IND (STATUS[CFR])         |            \
        ||       *<===========================*        |
58      ||                | RA_READY_IND               |            | Nx
        ||       *<===========================*        |
59      ||             | RA_DATA_REQ (SYNC)            |            |
        ||       *===========================>*        /
60      || FAD_DATA_IND (HDLC[CFR])    |               |
        |*<===========================*                |
61      | T30_READY_IND          |     |               |
        *<===========================*                 |            |
62      ||             | RA_DATA_IND (SYNC)            |            \
        ||       *<===========================*        |
63      ||                | RA_READY_IND               |            | Nx
        ||       *<===========================*        |
64      ||             | RA_DATA_REQ (SYNC)            |            |
        ||       *===========================>*        /
65      | T30_DATA_REQ (fax data, not-final)           |            |
        *===========================>*                 |            |
66      || FAD_DATA_REQ (msg, train, fax data, not-final)|
        |*===========================>*                |
67      || FAD_READY_IND         |     |               |
        |*<===========================*                |
68      | T30_READY_IND          |     |               |
        *<===========================*                 |            |
69      | T30_DATA_REQ (fax data, not-final)           |            |
        *===========================>*                 |            |
70      || FAD_DATA_REQ (msg, fax data, not-final)     |
        |*===========================>*                |
71      ||             | RA_DATA_IND (SYNC)            |                     \
        ||       *<===========================*                     |
72      ||                | RA_READY_IND               |                     |
        ||       *<===========================*                     |
73      ||           | RA_DATA_REQ (STATUS)            |                     |
        ||       *===========================>*                     |
        ||                       |     |                            | Nx
74      ||             | RA_DATA_IND (SYNC)            |                     |
        ||       *<===========================*                     |
75      ||                | RA_READY_IND               |                     |
        ||       *<===========================*                     |
76      ||             | RA_DATA_REQ (SYNC)            |                     |
        ||       *===========================>*                     /
        ||                       |     |
77      ||           | RA_DATA_IND (STATUS)            |
        ||       *<===========================*
78      ||                | RA_READY_IND               |
        ||       *<===========================*
79      ||        | RA_DATA_REQ (SYNC | STATUS)        |
        ||       *===========================>*
        ||                       |     |
80      ||           | RA_DATA_IND (STATUS)            |
        ||       *<===========================*
81      ||                | RA_READY_IND               |
        ||       *<===========================*
82      ||             | RA_DATA_REQ (DATA)            |
        ||       *===========================>*
```

```
        ||                      |     |
83      ||             | RA_DATA_IND (SYNC)          |             \        \
        ||       *<============================*      |             |
84      ||                | RA_READY_IND              |             | Nx     |
        ||       *<============================*      |             |
85      ||             | RA_DATA_REQ (DATA)          |             |        |
        ||       *============================>*      /             |
86      || FAD_READY_IND        |     |           if buffer     |
        |*<============================*              |            is low get                    | Nx
87      | T30_READY_IND         |     |               |            next fax-|
        *<============================*               |             |      data               |
↓ ----- |---------          |-------- |--------    |--------      |
88      | T30_DATA_REQ (fax data, not-final)          |             |                             |
        *============================>*               |             |                             |
        ||                      |     |                             |
89      || FAD_DATA_REQ (msg, fax data, not-final)    |                      |
        |*============================>*              |                      /
or ---- |---------          |-------- |--------    |--------
90      | T30_DATA_REQ (fax data, final)              |             |
        *============================>*               |             |
91      || FAD_DATA_REQ (msg, fax data, final)        |
        |*============================>*              |
92      ||             | RA_DATA_IND (SYNC)          |             \
        ||       *<============================*      |
93      ||                | RA_READY_IND              |             | Nx
        ||       *<============================*      |
94      ||             | RA_DATA_REQ (DATA)          |             |
        ||       *============================>*      /
        ||                      |     |
95      ||             | RA_DATA_IND (SYNC)          |             \
        ||       *<============================*      |
96      ||                | RA_READY_IND              |             | 5x min
        ||       *<============================*      |
97      ||             | RA_DATA_REQ (SYNC)          |             |
        ||       *============================>*      /
98      || FAD_DATA_CNF         |     |
        |*<============================*              |
99      | T30_SGN_IND (SGN_NOT_USED)  |               |             |
        *<============================*               |             |
↑ ----- |---------          |-------- |--------    |--------
↓ ----- |---------          |-------- |--------    |--------
100     ||             | RA_DATA_IND (SYNC)          |             \
        ||       *<============================*      |
101     ||                | RA_READY_IND              |             |
        ||       *<============================*      |
102     ||             | RA_DATA_REQ (SYNC)          |             |
        ||       *============================>*      |
103     | T30_SGN_RES (SGN_MPS)  |     |               |             |
        *============================>*               |             |        |
104     || FAD_DATA_REQ (bcs, HDLC[MPS])              |             |        |
        |*============================>*              |             | Nx for
105     ||             | RA_DATA_IND (SYNC)          |             | 75 msec
        ||       *<============================*      |
106     ||                | RA_READY_IND              |             |
        ||       *<============================*      |
107     ||             | RA_DATA_REQ (SYNC)          |             |
        ||       *============================>*      /
        ||                      |     |
108     ||             | RA_DATA_IND (SYNC)          |             \
        ||       *<============================*      |
109     ||                | RA_READY_IND              |             | Nx for
        ||       *<============================*   | 1 sec
110     ||           | RA_DATA_REQ (PREAMBLE)        |             |
        ||       *============================>*      /
        ||                      |     |
111     ||             | RA_DATA_IND (SYNC)          |             \
        ||       *<============================*      |
112     ||                | RA_READY_IND              |             | Nx
        ||       *<============================*      |
113     ||          | RA_DATA_REQ (STATUS[MPS])      |             |
        ||       *============================>*      /
114     || FAD_DATA_CNF         |     |
        |*<============================*              |
115     ||             | RA_DATA_IND (SYNC)          |             \
        ||       *<============================*      |
116     ||                | RA_READY_IND              |             | Nx
        ||       *<============================*      |
117     ||             | RA_DATA_REQ (SYNC)          |             |
        ||       *============================>*      /
        ||                      |     |
118     ||           | RA_DATA_IND (PREAMBLE)        |             \
        ||       *<============================*      |
119     ||                | RA_READY_IND              |             | Nx for
        ||       *<============================*   | 1 sec
120     ||             | RA_DATA_REQ (SYNC)          |             |
        ||       *============================>*      /
        ||                      |     |
121     ||            | RA_DATA_IND (STATUS)         |             \
        ||       *<============================*      |
122     ||                | RA_READY_IND              |             |
        ||       *<============================*     | Nx
123     ||             | RA_DATA_REQ (SYNC)          |             |
        ||       *============================>*      /
        ||                      |     |
```

```
124     ||             | RA_DATA_IND (SYNC)             |
        ||       *<=============================*       |
125     ||                | RA_READY_IND                |
        ||       *<=============================*       |
126     ||             | RA_DATA_REQ (SYNC)             |
        ||       *=============================>*       |
127     || FAD_DATA_IND (HDLC[MCF])    |                |
        |*<============================*                |
128     | T30_READY_IND           |     |                |
        *<=============================*                |             | ... continue at 69
or ---- |---------           |-------- |--------    |--------
129     ||             | RA_DATA_IND (SYNC)             |             \
        ||       *<=============================*       |
130     ||                | RA_READY_IND                |             |
        ||       *<=============================*       |
131     ||             | RA_DATA_REQ (SYNC)             |             |
        ||       *=============================>*       |
132     | T30_SGN_RES (SGN_EOP)   |     |                |             |
        *=============================>*                |             |        |
133     || FAD_DATA_REQ (bcs, HDLC[EOP])                |             |        |
        |*============================>*                |             | Nx for
134     ||             | RA_DATA_IND (SYNC)             |             | 75 msec
        ||       *<=============================*       |
135     ||                | RA_READY_IND                |             |
        ||       *<=============================*       |
136     ||             | RA_DATA_REQ (SYNC)             |             |
        ||       *=============================>*       /
        ||                        |     |
137     ||             | RA_DATA_IND (SYNC)             |             \
        ||       *<=============================*       |
138     ||                | RA_READY_IND                |             | Nx for
        ||       *<=============================*    | 1 sec
139     ||           | RA_DATA_REQ (PREAMBLE)           |             |
        ||       *=============================>*       /
        ||                        |     |
140     ||             | RA_DATA_IND (SYNC)             |             \
        ||       *<=============================*       |
141     ||                | RA_READY_IND                |             | Nx
        ||       *<=============================*       |
142     ||          | RA_DATA_REQ (STATUS[EOP])         |             |
        ||       *=============================>*       /
143     || FAD_DATA_CNF           |     |
        |*<============================*                |
144     ||             | RA_DATA_IND (SYNC)             |             \
        ||       *<=============================*       |
145     ||                | RA_READY_IND                |             | Nx
        ||       *<=============================*       |
146     ||             | RA_DATA_REQ (SYNC)             |             |
        ||       *=============================>*       /
        ||                        |     |
147     ||           | RA_DATA_IND (PREAMBLE)           |             \
        ||       *<=============================*       |
148     ||                | RA_READY_IND                |             | Nx for
        ||       *<=============================*    | 1 sec
149     ||             | RA_DATA_REQ (SYNC)             |             |
        ||       *=============================>*       /
        ||                        |     |
150     ||            | RA_DATA_IND (STATUS)            |             \
        ||       *<=============================*       |
151     ||                | RA_READY_IND                |             |
        ||       *<=============================*      | Nx
152     ||             | RA_DATA_REQ (SYNC)             |             |
        ||       *=============================>*       /
        ||                        |     |
153     ||             | RA_DATA_IND (SYNC)             |
        ||       *<=============================*       |
154     ||                | RA_READY_IND                |
        ||       *<=============================*       |
155     ||             | RA_DATA_REQ (SYNC)             |
        ||       *=============================>*       |
156     || FAD_DATA_IND (HDLC[MCF])    |                |
        |*<============================*                |
157     || FAD_DATA_REQ (bcs, HDLC[DCN])|               |
        |*============================>*                |
158     ||             | RA_DATA_IND (SYNC)             |             \
        ||       *<=============================*       |
159     ||                | RA_READY_IND                |             | Nx for
        ||       *<=============================*    | 1 sec
160     ||           | RA_DATA_REQ (PREAMBLE)           |             |
        ||       *=============================>*       /
        ||                        |     |
161     ||             | RA_DATA_IND (SYNC)             |             \
        ||       *<=============================*       |
162     ||                | RA_READY_IND                |             | Nx
        ||       *<=============================*       |
163     ||          | RA_DATA_REQ (STATUS[DCN])         |             |
        ||       *=============================>*       /
164     || FAD_DATA_CNF           |     |
        |*<============================*                |
165     | T30_CMPL_IND (CMPL_EOP)|     |                |
        *<=============================*                |             |
or ---- |---------           |-------- |--------    |--------
        ||                        |     |
```

```
166     ||             | RA_DATA_IND (SYNC)            |             \
        ||       *<============================*      |
167     ||                | RA_READY_IND               |             |
        ||       *<============================*      |
168     ||             | RA_DATA_REQ (SYNC)            |             |
        ||       *============================>*      |
169     | T30_SGN_RES (SGN_EOM)  |     |               |             |
        *===========================>*                 |             |        |
170     || FAD_DATA_REQ (bcs, HDLC[EOM])               |             |        |
        |*===========================>*                |             | Nx for
171     ||             | RA_DATA_IND (SYNC)            |             | 75 msec
        ||       *<============================*      |
172     ||                | RA_READY_IND               |             |
        ||       *<============================*      |
173     ||             | RA_DATA_REQ (SYNC)            |             |
        ||       *============================>*      /
        ||                      |     |
174     ||             | RA_DATA_IND (SYNC)            |             \
        ||       *<============================*      |
175     ||                | RA_READY_IND               |             | Nx for
        ||       *<============================*   | 1 sec
176     ||           | RA_DATA_REQ (PREAMBLE)          |             |
        ||       *============================>*      /
        ||                      |     |
177     ||             | RA_DATA_IND (SYNC)            |             \
        ||       *<============================*      |
178     ||                | RA_READY_IND               |             | Nx
        ||       *<============================*      |
179     ||          | RA_DATA_REQ (STATUS[EOM])        |             |
        ||       *============================>*      /
180     || FAD_DATA_CNF         |     |                |
        |*<===========================*                |
181     ||             | RA_DATA_IND (SYNC)            |             \
        ||       *<============================*      |
182     ||                | RA_READY_IND               |             | Nx
        ||       *<============================*      |
183     ||             | RA_DATA_REQ (SYNC)            |             |
        ||       *============================>*      /
        ||                      |     |
184     ||           | RA_DATA_IND (PREAMBLE)          |             \
        ||       *<============================*      |
185     ||                | RA_READY_IND               |             | Nx for
        ||       *<============================*   | 1 sec
186     ||             | RA_DATA_REQ (SYNC)            |             |
        ||       *============================>*      /
        ||                      |     |
187     ||            | RA_DATA_IND (STATUS)           |             \
        ||       *<============================*      |
188     ||                | RA_READY_IND               |             |
        ||       *<============================*     | Nx
189     ||             | RA_DATA_REQ (SYNC)            |             |
        ||       *============================>*      /
        ||                      |     |
190     ||             | RA_DATA_IND (SYNC)            |
        ||       *<============================*      |
191     ||                | RA_READY_IND               |
        ||       *<============================*      |
192     ||             | RA_DATA_REQ (SYNC)            |
        ||       *============================>*      |
193     || FAD_DATA_IND (HDLC[MCF])    |               |
        |*<===========================*                |
194     | T30_CMPL_IND (CMPL_EOM)|     |               |
        *<===========================*                 |             |
↑ ----- |---------          |-------- |--------    |--------
195     ||             | RA_DATA_IND (SYNC)            |             \
        ||       *<============================*      |
196     ||                | RA_READY_IND               |             | Nx
        ||       *<============================*      |
197     ||             | RA_DATA_REQ (SYNC)            |             |
        ||       *============================>*      /
198     | T30_DEACTIVATE_REQ    |     |                |
        *===========================>*                  |             |
199     || FAD_DEACTIVATE_REQ   |     |                |
        |*===========================>*                |
200     | RA_DEACTIVATE_REQ     |     |                |
        *==========================================================================================>*
        ||                      |     |
```

4.2 Receive Fax (9600, 4800, 2400), best case

```
         ACI                    T30     FAD             RA
        ||                       |      |
        ||                       |      |
        ||                       |      |
1       | RA_ACTIVATE_REQ        |      |                |
        *=======================================================================================>*
2       | RA_ACTIVATE_CNF        |      |                |
        *<=======================================================================================*
3       | T30_ACTIVATE_REQ (configs)    |                |             |
        *========================>*                      |             |
4       || FAD_ACTIVATE_REQ (configs)   |                |
        |*========================>*                     |
5       ||             | RA_DATA_IND (SYNC)              |             \
        ||        *<==========================*          |
6       ||                | RA_READY_IND                 |             | Nx
        ||        *<==========================*          |
7       ||             | RA_DATA_REQ (SYNC)              |             |
        ||        *==========================>*          /
8       | T30_CAP_REQ (caps)     |      |                |
        *==========================>*                    |             |
9       || FAD_DATA_REQ (bcs, HDLC[NSF..DIS])            |
        |*========================>*                     |
10      ||             | RA_DATA_IND (SYNC)              |             \
        ||        *<==========================*          |
11      ||                | RA_READY_IND                 |             | Nx for
        ||        *<==========================*     | 1 sec
12      ||           | RA_DATA_REQ (PREAMBLE)            |             |
        ||        *==========================>*          /
        ||                       |      |
13      ||             | RA_DATA_IND (SYNC)              |             \ optional
        ||        *<==========================*          |
14      ||                | RA_READY_IND                 |             | Nx
        ||        *<==========================*          |
15      ||         | RA_DATA_REQ (STATUS[NSF...])        |             |
        ||        *==========================>*          /
        ||                       |      |
16      ||             | RA_DATA_IND (SYNC)              |             \ mandatory
        ||        *<==========================*          |
17      ||                | RA_READY_IND                 |             | Nx
        ||        *<==========================*          |
18      ||          | RA_DATA_REQ (STATUS[DIS])          |             |
        ||        *==========================>*          /
19      || FAD_DATA_CNF          |      |
        |*<==========================*                   |
20      ||             | RA_DATA_IND (SYNC)              |             \
        ||        *<==========================*          |
21      ||                | RA_READY_IND                 |             | Nx
        ||        *<==========================*          |
22      ||             | RA_DATA_REQ (SYNC)              |             |
        ||        *==========================>*          /
        ||                       |      |
23      ||           | RA_DATA_IND (PREAMBLE)            |             \
        ||        *<==========================*          |
24      ||                | RA_READY_IND                 |             | Nx for
        ||        *<==========================*     | 1 sec
25      ||             | RA_DATA_REQ (SYNC)              |             |
        ||        *==========================>*          /
26      || FAD_MUX_IND (MUX_BCS) |      |
        |*<==========================*                   |
27      ||         | RA_DATA_IND (STATUS[NSS...])        |             \ optional
        ||        *<==========================*          |
28      ||                | RA_READY_IND                 |             | Nx
        ||        *<==========================*          |
29      ||             | RA_DATA_REQ (SYNC)              |             |
        ||        *==========================>*          /
        ||                       |      |
30      ||          | RA_DATA_IND (STATUS[DCS])          |             \ mandatory
        ||        *<==========================*          |
31      ||                | RA_READY_IND                 |             | Nx
        ||        *<==========================*          |
32      ||             | RA_DATA_REQ (SYNC)              |             |
        ||        *==========================>*          /
        ||                       |      |
33      ||             | RA_DATA_IND (SYNC)              |
        ||        *<==========================*          |
34      ||                | RA_READY_IND                 |
        ||        *<==========================*          |
35      ||             | RA_DATA_REQ (SYNC)              |
        ||        *==========================>*          |
36      || FAD_DATA_IND (HDLC[NSS..DCS])                 |             |
        |*<==========================*                   |
37      | T30_CAP_CNF (caps, FMOD_RCV) |                 |             |
        *<=========================*                     |             |
38      || FAD_RCV_TCF_REQ       |      |
        |*==========================>*                   |
39      ||             | RA_DATA_IND (SYNC)              |             \
        ||        *<==========================*          |
40      ||                | RA_READY_IND                 |             | Nx
        ||        *<==========================*          |
41      ||             | RA_DATA_REQ (SYNC)              |             |
        ||        *==========================>*          /
        ||                       |      |
```

```
42      ||          | RA_DATA_IND (DATA[TCF])       |             \
        ||       *<============================*     |
43      ||                | RA_READY_IND            |             | Nx for
        ||       *<============================*  | 1,5 sec
44      ||             | RA_DATA_REQ (SYNC)         |             |
        ||       *============================>*     /
        ||                   |     |
45      ||             | RA_DATA_IND (SYNC)         |
        ||       *<============================*
46      ||                | RA_READY_IND            |
        ||       *<============================*
47      ||             | RA_DATA_REQ (SYNC)         |
        ||       *============================>*
48      || FAD_RCV_TCF_CNF (ratio)     |            |
        |*<============================*            |
49      ||             | RA_DATA_IND (SYNC)         |             \
        ||       *<============================*     |
50      ||                | RA_READY_IND            |             | Nx
        ||       *<============================*     |
51      ||             | RA_DATA_REQ (SYNC)         |             |
        ||       *============================>*     /
52      || FAD_DATA_REQ (bcs, HDLC[CFR])            |             |
        |*============================>*            |
53      ||             | RA_DATA_IND (SYNC)         |             \
        ||       *<============================*     |
54      ||                | RA_READY_IND            |             | Nx for
        ||       *<============================*   | 1 sec
55      ||           | RA_DATA_REQ (PREAMBLE)       |             |
        ||       *============================>*     /
        ||                   |     |
56      ||             | RA_DATA_IND (SYNC)         |             \
        ||       *<============================*     |
57      ||                | RA_READY_IND            |             | Nx
        ||       *<============================*     |
58      ||          | RA_DATA_REQ (STATUS[CFR])     |             |
        ||       *============================>*     /
59      || FAD_DATA_CNF      |     |
        |*<============================*            |
60      ||             | RA_DATA_IND (SYNC)         |             \
        ||       *<============================*     |
61      ||                | RA_READY_IND            |             | Nx
        ||       *<============================*     |
62      ||             | RA_DATA_REQ (SYNC)         |             |
        ||       *============================>*     /
        ||                   |     |
63      ||            | RA_DATA_IND (STATUS)        |
        ||       *<============================*
64      ||                | RA_READY_IND            |
        ||       *<============================*
65      ||             | RA_DATA_REQ (SYNC)         |
        ||       *============================>*
        ||                   |     |
66      ||         | RA_DATA_IND (SYNC | STATUS)    |             \        \
        ||       *<============================*     |             |
67      ||                | RA_READY_IND            |             | Nx for |
        ||       *<============================*   | 1 sec        |
68      ||           | RA_DATA_REQ (PREAMBLE)       |             |        |
        ||       *============================>*     /             |
        ||                   |     |                                | Nx
69      ||         | RA_DATA_IND (SYNC | STATUS)    |                      |
        ||       *<============================*                   |
70      ||                | RA_READY_IND            |                      |
        ||       *<============================*                   |
71      ||            | RA_DATA_REQ (STATUS)        |                      |
        ||       *============================>*                   /
72      || FAD_MUX_IND (MUX_MSG) |     |
        |*<============================*            |
        ||                   |     |
73      ||             | RA_DATA_IND (DATA)         |             \        \
        ||       *<============================*     |             |
74      ||                | RA_READY_IND            |             | Nx     |
        ||       *<============================*     |             |
75      ||             | RA_DATA_REQ (SYNC)         |             |        |
        ||       *============================>*     /             |
76      || FAD_DATA_IND (fax data, not-final)       |             if buffer|
        |*<============================*            |             is full | Nx
77      | T30_DATA_IND (fax data, not-final)        |             |     send t4-            |
        *<============================*             |             |      data               |
78      | T30_READY_REQ          |     |            |                      |
        *============================>*             |             |                         |
79      || FAD_READY_REQ         |     |            |             |
        |*============================>*            |                      /
80      ||             | RA_DATA_IND (SYNC)         |
        ||       *<============================*
81      ||                | RA_READY_IND            |
        ||       *<============================*
82      ||             | RA_DATA_REQ (SYNC)         |
        ||       *============================>*
83      || FAD_DATA_IND (fax data, final)           |
        |*<============================*            |
84      | T30_DATA_IND (fax data, final)            |             |
        *<============================*             |             |
```

```
85      ||             | RA_DATA_IND (SYNC)           |            \
        ||       *<============================*      |
86      ||                | RA_READY_IND              |            | Nx
        ||       *<============================*      |
87      ||             | RA_DATA_REQ (SYNC)           |            |
        ||       *============================>*      /
        ||                      |     |
88      ||           | RA_DATA_IND (PREAMBLE)         |            \
        ||       *<============================*      |
89      ||                | RA_READY_IND              |            | Nx for
        ||       *<============================*   | 1 sec
90      ||             | RA_DATA_REQ (SYNC)           |            |
        ||       *============================>*      /
91      || FAD_MUX_IND (MUX_BCS) |     |
        |*<===========================*               |
↓ ----- |------             |------  |------      |-------
        ||                      |     |
92      ||         | RA_DATA_IND (STATUS, MPS)        |            \
        ||       *<============================*      |
93      ||                | RA_READY_IND              |            | Nx
        ||       *<============================*      |
94      ||             | RA_DATA_REQ (SYNC)           |            |
        ||       *============================>*      /
        ||                      |     |
95      ||             | RA_DATA_IND (SYNC)           |
        ||       *<============================*      |
96      ||                | RA_READY_IND              |
        ||       *<============================*      |
97      ||             | RA_DATA_REQ (SYNC)           |
        ||       *============================>*
98      || FAD_DATA_IND (HDLC[MPS])    |              |
        |*<===========================*               |
99      | T30_SGN_IND (SGN_MPS)  |     |              |
        *<===========================*                |            |
100     | T30_SGN_RES (SGN_MCF)  |     |              |
        *===========================>*                |            |
101     || FAD_DATA_REQ (HDLC[MCF])    |              |
        |*===========================>*               |
102     ||             | RA_DATA_IND (SYNC)           |            \
        ||       *<============================*      |
103     ||                | RA_READY_IND              |            | Nx for
        ||       *<============================*   | 1 sec
104     ||           | RA_DATA_REQ (PREAMBLE)         |            |
        ||       *============================>*      /
        ||                      |     |
105     ||             | RA_DATA_IND (SYNC)           |            \
        ||       *<============================*      |
106     ||                | RA_READY_IND              |            | Nx
        ||       *<============================*      |
107     ||          | RA_DATA_REQ(STATUS[MCF])        |            |
        ||       *============================>*      /
        ||                      |     |
108     ||             | RA_DATA_IND (SYNC)           |
        ||       *<============================*      |
109     ||                | RA_READY_IND              |
        ||       *<============================*      |
110     ||             | RA_DATA_REQ (SYNC)           |
        ||       *============================>*
111     || FAD_DATA_CNF           |     |
        |*<===========================*     | ... continue at 58
or ---- |------             |------  |------      |-------
        ||                      |     |
112     ||         | RA_DATA_IND (STATUS, EOP)        |            \
        ||       *<============================*      |
113     ||                | RA_READY_IND              |            | Nx
        ||       *<============================*      |
114     ||             | RA_DATA_REQ (SYNC)           |            |
        ||       *============================>*      /
        ||                      |     |
115     ||             | RA_DATA_IND (SYNC)           |
        ||       *<============================*      |
116     ||                | RA_READY_IND              |
        ||       *<============================*      |
117     ||             | RA_DATA_REQ (SYNC)           |
        ||       *============================>*
118     || FAD_DATA_IND (HDLC[EOP])    |              |
        |*<===========================*               |
119     | T30_SGN_IND (SGN_EOP)  |     |              |
        *<===========================*                |            |
120     | T30_SGN_RES (SGN_MCF)  |     |              |
        *===========================>*                |            |
121     || FAD_DATA_REQ (HDLC[MCF])    |              |
        |*===========================>*               |
122     ||             | RA_DATA_IND (SYNC)           |            \
        ||       *<============================*      |
123     ||                | RA_READY_IND              |            | Nx for
        ||       *<============================*   | 1 sec
124     ||           | RA_DATA_REQ (PREAMBLE)         |            |
        ||       *============================>*      /
        ||                      |     |
125     ||             | RA_DATA_IND (SYNC)           |            \
        ||       *<============================*      |
126     ||                | RA_READY_IND              |            | Nx
        ||       *<============================*      |
127     ||          | RA_DATA_REQ(STATUS[MCF])        |            |
        ||       *============================>*      /
        ||                      |     |
```

```
128     ||              | RA_DATA_IND (SYNC)              |
        ||        *<==============================*
129     ||                 | RA_READY_IND                 |
        ||        *<==============================*
130     ||              | RA_DATA_REQ (SYNC)              |
        ||        *==============================>*
131     || FAD_DATA_CNF           |     |
        |*<==============================*                |
132     ||            | RA_DATA_IND (PREAMBLE)            |             \
        ||        *<==============================*       |
133     ||                 | RA_READY_IND                 |             | Nx for
        ||        *<==============================*   | 1 sec
134     ||              | RA_DATA_REQ (SYNC)              |             |
        ||        *==============================>*       /
        ||                        |     |
135     ||          | RA_DATA_IND (STATUS[DCN])           |             \
        ||        *<==============================*       |
136     ||                 | RA_READY_IND                 |             | Nx
        ||        *<==============================*       |
137     ||              | RA_DATA_REQ (SYNC)              |             |
        ||        *==============================>*       /
        ||                        |     |
138     ||              | RA_DATA_IND (SYNC)              |
        ||        *<==============================*
139     ||                 | RA_READY_IND                 |
        ||        *<==============================*
140     ||              | RA_DATA_REQ (SYNC)              |
        ||        *==============================>*
141     || FAD_DATA_IND (HDLC[DCN])     |                 |
        |*<==============================*                |
142     | T30_CMPL_IND (CMPL_EOP)|      |                 |
        *<==============================*                 |             |
or ---- |------               |------   |------       |-------
143     ||          | RA_DATA_IND (STATUS, EOM)           |             \
        ||        *<==============================*       |
144     ||                 | RA_READY_IND                 |             | Nx
        ||        *<==============================*       |
145     ||              | RA_DATA_REQ (SYNC)              |             |
        ||        *==============================>*       /
        ||                        |     |
146     ||              | RA_DATA_IND (SYNC)              |
        ||        *<==============================*
147     ||                 | RA_READY_IND                 |
        ||        *<==============================*
148     ||              | RA_DATA_REQ (SYNC)              |
        ||        *==============================>*
149     || FAD_DATA_IND (HDLC[EOM])     |                 |
        |*<==============================*                |
150     | T30_SGN_IND (SGN_EOM)   |     |                 |
        *<==============================*                 |             |
151     | T30_SGN_RES (SGN_MCF)   |     |                 |
        *==============================>*                 |             |
152     || FAD_DATA_REQ (HDLC[MCF])     |                 |
        |*==============================>*                |
153     ||              | RA_DATA_IND (SYNC)              |             \
        ||        *<==============================*       |
154     ||                 | RA_READY_IND                 |             | Nx for
        ||        *<==============================*   | 1 sec
155     ||            | RA_DATA_REQ (PREAMBLE)            |             |
        ||        *==============================>*       /
        ||                        |     |
156     ||              | RA_DATA_IND (SYNC)              |             \
        ||        *<==============================*       |
157     ||                 | RA_READY_IND                 |             | Nx
        ||        *<==============================*       |
158     ||           | RA_DATA_REQ(STATUS[MCF])           |             |
        ||        *==============================>*       /
        ||                        |     |
159     ||              | RA_DATA_IND (SYNC)              |
        ||        *<==============================*
160     ||                 | RA_READY_IND                 |
        ||        *<==============================*
161     ||              | RA_DATA_REQ (SYNC)              |
        ||        *==============================>*
162     || FAD_DATA_CNF           |     |
        |*<==============================*                |
163     | T30_CMPL_IND (CMPL_EOM)|      |                 |
        *<==============================*                 |             |
↑ ----- |---------            |-------- |--------    |--------
164     ||              | RA_DATA_IND (SYNC)              |             \
        ||        *<==============================*       |
165     ||                 | RA_READY_IND                 |             | Nx
        ||        *<==============================*       |
166     ||              | RA_DATA_REQ (SYNC)              |             |
        ||        *==============================>*       /
        ||                        |     |
167     | T30_DEACTIVATE_REQ      |     |                 |
        *==============================>*                 |             |
168     || FAD_DEACTIVATE_REQ     |     |
        |*==============================>*                |
169     | RA_DEACTIVATE_REQ       |     |                 |
        *==============================================================================================>*
        ||                        |     |
```

5 Protocol

5.1 FAD State NULL

5.1.1 Activate FAD

```
T30                            FAD                            RA
 |                              |                              |
 |       FAD_ACTIVATE_REQ       |                              |
 *============================>*                              |
 |                           (FAD 1)                           |
 |                              |                              |
```

(FAD 1)
FAD receives a FAD-ACTIVATE request primitive while in the state NULL. The primitive contains the transmission rate for the TCH and whether or not half-rate is to be set. FAD changes to the state IDLE.

5.2 FAD State IDLE

5.2.1 Set Transmission Rate

```
T30                            FAD                            RA
 |                              |                              |
 |        FAD_MODIFY_REQ        |                              |
 *============================>*                              |
 |                           (FAD 1)                           |
 |                              |                              |
```

(FAD 1)
FAD receives a FAD-MODIFY request primitive while in the state IDLE. The primitive indicates contains the transmission rate for the TCH.

5.2.2 Synchronization in IDLE state (SYNC Frames from Remote Station)

```
T30                         FAD                          RA
 |                           |                            |
 |                           |        RA_READY_IND        |
 |                           *<===========================*
 |                           |                          (RA 1)
 |                           |        RA_DATA_REQ         |
 |                           |           (SYNC)           |
 |                           *===========================>*
 |                           |                          (RA 2)
 |                           |                            |
```

(RA 1)

Receipt of a RA-READY indication primitive indicates that FAD may send a frame.

(RA 2)

A RA-DATA request primitive containing a SYNC frame is issued by FAD.

The continuous transmission of SYNC frames between local and remote station continues for as long as FAD remains in the state IDLE.

5.2.3 Request from Local Station to send BCS data (not final) - change to state BCS-REC

```
T30                         FAD                          RA
 |                           |                            |
 |        FAD_DATA_REQ       |                            |
 |    (BCS data, not final)  |                            |
 *==========================>*                            |
 |                         (FAD 1)                        |
 |        FAD_READY_IND      |                            |
 *<==========================*                            |
 |                         (FAD 2)                        |
 |                           |                            |
```

(FAD 1)

FAD receives a FAD-DATA request primitive from T30 in which the access rate is set to BCS_RATE and the FINAL flag to FALSE. FAD stores the BCS data for sending and changes to the state BCS-REC.

(FAD 2)

FAD issues a FAD-READY indication primitive to T30 and awaits further BCS data.

5.2.4 Request from Local Station to send BCS data (final) - change to state BCS-REC

```
T30                          FAD                          RA
 |                            |                            |
 |        FAD_DATA_REQ        |                            |
 |      (BCS data, final)     |                            |
 *===========================>*                            |
 |                         (FAD 1)                         |
 |                            |                            |
```

(FAD 1)

FAD receives a FAD-DATA request primitive from T30 in which the access rate is set to BCS_RATE and the FINAL flag to TRUE. FAD stores the BCS data for sending and changes to the state BCS-REC.

5.2.5 Request from Local Station to send TCF - change to state MSG-REC-TCF

```
T30                          FAD                          RA
 |                            |                            |
 |       FAD_SND_TCF_REQ      |                            |
 *===========================>*                            |
 |                         (FAD 1)                         |
 |                            |                            |
```

(FAD 1)

FAD receives a FAD-SND-TCF request primitive from T30. The access rate is set to the rate requested in the primitive and FAD changes to the state MSG-REC-TCF.

5.2.6 Request from Local Station to expect TCF - change to state MSG-TRA-TCF

```
T30                          FAD                          RA
 |                            |                            |
 |       FAD_RCV_TCF_REQ      |                            |
 *===========================>*                            |
 |                         (FAD 1)                         |
 |                            |                            |
```

(FAD 1)

FAD receives a FAD-RCV-TCF request primitive from T30. The access rate is set to the rate requested in the primitive and FAD changes to the state MSG-TRA-TCF.

5.2.7 Request from Local Station to send MSG data (not final) - change to state MSG-REC

```
T30                           FAD                            RA
 |                             |                              |
 |        FAD_DATA_REQ         |                              |
 |    (MSG data, not final)    |                              |
 *===========================>*                              |
 |                          (FAD 1)                           |
 |        FAD_READY_IND        |                              |
 *<===========================*                              |
 |                          (FAD 2)                           |
 |                             |                              |
```

(FAD 1)

FAD receives a FAD-DATA request primitive from T30 containing MSG data; the access rate is set to the rate requested in the primitive and the FINAL flag to FALSE. FAD stores the MSG data for immediate sending and changes to the state MSG-REC.

(FAD 2)

FAD issues a FAD-READY indication primitive to T30 and awaits further MSG data.

5.2.8 Request from Local Station to send MSG data (final) - change to state MSG-REC

```
T30                           FAD                            RA
 |                             |                              |
 |        FAD_DATA_REQ         |                              |
 |      (MSG data, final)      |                              |
 *===========================>*                              |
 |                          (FAD 1)                           |
 |                             |                              |
```

(FAD 1)

FAD receives a FAD-DATA request primitive from T30 containing MSG data; the access rate is set to the rate requested in the primitive and the FINAL flag to TRUE. FAD stores the MSG data for immediate sending and changes to the state MSG-REC.

5.2.9 PREAMBLE frame received from Remote Station - change to state BCS-TRA

```
T30                          FAD                          RA
 |                            |                            |
 |                            |         RA_DATA_IND        |
 |                            |          (PREAMBLE)        |
 |                            *<===========================*
 |                            |                          (RA 1)
 |      FAD_MUX_IND (BCS)     |                            |
 *<===========================*                            |
 |                          (FAD 1)                        |
 |                            |                            |
```

(RA 1)

FAD receives a RA-DATA indication primitive containing a PREAMBLE and with the identifier set to BCS-REC. FAD enters the state BCS-TRA.

(FAD 1)

FAD issues a FAD-MUX indication primitive to notify T30 that BCS data is being received.

5.2.10 STATUS frame received from Remote Station - change to state MSG-TRA

```
T30                          FAD                          RA
 |                            |                            |
 |                            |         RA_DATA_IND        |
 |                            |           (STATUS)         |
 |                            *<===========================*
 |                            |                          (RA 1)
 |      FAD_MUX_IND (MSG)     |                            |
 *<===========================*                            |
 |                          (FAD 1)                        |
 |                            |                            |
```

(RA 1)

FAD receives a A RA-DATA indication primitive containing a STATUS frame and with the identifier set to MSG-REC. FAD enters the state MSG-TRA and awaits training.

(FAD 1)

FAD issues a FAD-MUX indication primitive to notify T30 that training prior to reception of MSG data has been started.

5.3 Binary Coded Signalling Phase (FAD State BCS-REC)

5.3.1 Request from Local Station to send BCS data

```
  T30                          FAD                          RA
   |                            |                            |
   |         FAD_DATA_REQ       |                            |
   |          (BCS data)        |                            |
   *==========================>*                            |
   |                         (FAD 1)                         |
   |                            |                            |
```

(FAD 1)

FAD receives a FAD-DATA request primitive from T30 in which the access rate is set to BCS_RATE. FAD stores the BCS data for sending as well as the value of the FINAL flag contained in the primitive.

5.3.2 Send BCS data

```
  T30                          FAD                          RA
   |                            |                            |
   |                            |         RA_DATA_IND        |
   |                            |           (SYNC)           |
   |                            *<==========================*
   |                            |                         (RA 1)
   |                            |        RA_READY_IND        |
   |                            *<==========================*
   |                            |                         (RA 2)
   |                            |         RA_DATA_REQ        |
   |                            |          (STATUS)          |
   |                            *==========================>*
   |                            |                         (RA 3)
   |             ...            |             ...            |
   |        FAD_READY_IND       |                            |
   *<==========================*                            |
   |                         (FAD 1)                         |
   |                            |                            |
```

(RA 1)

A RA-DATA indication primitive with a SYNC frame is received.

(RA 2)

Receipt of a RA-READY indication primitive indicates that FAD may send data.

(RA 3)

A RA-DATA request primitive containing a STATUS frame is issued by FAD.

Steps RA 1-3 are performed 1, 2 or 4 times according to how the TCH access rate has been set by the most recent FAD-CONFIG request primitive.

(FAD 1)

The FINAL flag has been set to FALSE in the last FAD-DATA request primitive received. FAD therefore issues a FAD-READY indication primitive to T30 and awaits further data.

5.3.3 Sending of BCS data concluded - return to IDLE State

```
T30                          FAD                         RA
 |                            |                           |
 |                            |        RA_DATA_IND        |
 |                            |          (SYNC)           |
 |                            *<==========================*
 |                            |                         (RA 1)
 |                            |       RA_READY_IND        |
 |                            *<==========================*
 |                            |                         (RA 2)
 |                            |        RA_DATA_REQ        |
 |                            |          (SYNC)           |
 |                            *==========================>*
 |                            |                         (RA 3)
 |        FAD_DATA_CNF        |                           |
 *<===========================*                           |
 |                         (FAD 1)                        |
 |                            |                           |
```

(RA 1)

A RA-DATA indication primitive with a SYNC frame is received by FAD.

(RA 2)

Receipt of a RA-READY indication primitive indicates that FAD may send a frame.

(RA 3)

The FINAL flag has been set to TRUE in the last FAD-DATA request primitive received. FAD ensures that all BCS data has been sent and then issues a RA-DATA request primitive containing a SYNC frame.

(FAD 1)

FAD issues a FAD-DATA confirmation primitive and enters the state IDLE.

5.4 Binary Coded Signalling Phase (FAD State BCS-TRA)

5.4.1 Synchronization in BCS-TRA state

```
T30                           FAD                           RA
 |                             |                             |
 |                             |         RA_DATA_IND         |
 |                             |         (BCS data)          |
 |                             *<===========================*
 |                             |                          (RA 1)
 |                             |        RA_READY_IND         |
 |                             *<===========================*
 |                             |                          (RA 2)
 |                             |         RA_DATA_REQ         |
 |                             |           (SYNC)            |
 |                             *===========================>*
 |            ....             |             ...             |
 |                             |                          (RA 3)
 |        FAD_DATA_IND         |                             |
 |         (BCS data)          |                             |
 *<===========================*                             |
 |                          (FAD 1)                          |
 |                             |                             |
```

(RA 1)

A RA-DATA indication primitive with BCS data is received. The data contained in the primitive is stored by FAD.

(RA 2)

Receipt of a RA-READY indication primitive indicates that FAD may send a frame.

(RA 3)

A RA-DATA request primitive containing a SYNC frame is issued by FAD.

Steps RA 1-3 are performed so long as the remote site has BCS data to be sent.

(FAD 1)
If the receive buffer is found to be full before receipt of BCS data is complete FAD issues a FAD-DATA indication primitive containing the BCS data (modified in order that it can be correctly interpreted by the higher-layer entities) and with the FINAL flag set to FALSE.

5.4.2 Conclusion of BCS-TRA Sychronization - return to IDLE State

```
T30                        FAD                         RA
 |                          |                           |
 |                          |        RA_DATA_IND        |
 |                          |          (SYNC)           |
 |                          *<==========================*
 |                          |                        (RA 1)
 |                          |       RA_READY_IND        |
 |                          *<==========================*
 |                          |                        (RA 2)
 |                          |        RA_DATA_REQ        |
 |                          |          (SYNC)           |
 |                          *==========================>*
 |                          |                        (RA 3)
 |           ....           |            ...            |
 |       FAD_DATA_IND       |                           |
 |        (BCS data)        |                           |
 *<==========================*                          |
 |                       (FAD 1)                        |
 |                          |                           |
 |                          |                           |
 |                          |                           |
```

(RA 1)

A RA-DATA indication primitive with a SYNC frame is received.

(RA 2)

Receipt of a RA-READY indication primitive indicates that FAD may send a frame.

(RA 3)

A RA-DATA request primitive containing a SYNC frame is issued by FAD.

Steps RA 1-3 are performed 1, 2 or 4 times according to how the access rate has been set by the most recent FAD-CONFIG request primitive.

(FAD 1)

FAD issues a FAD-DATA indication primitive containing the BCS data (modified in order that it can be correctly interpreted by the higher-layer entities) and with the FINAL flag set to TRUE. FAD then returns to the IDLE state.

5.5 Message Training Phase (FAD State MSG-REC-TCF)

5.5.1 Training in MSG-REC-TCF Phase

```
  T30                          FAD                          RA
   |                            |                            |
   |                            |         RA_DATA_IND        |
   |                            |           (SYNC)           |
   |                            *<===========================*
   |                            |                          (RA 1)
   |                            |        RA_READY_IND        |
   |                            *<===========================*
   |                            |                          (RA 2)
   |                            |         RA_DATA_REQ        |
   |                            |           (TCF)            |
   |                            *===========================>*
   |             ...            |            ...             |
   |                            |                          (RA 3)
   |       FAD_SND_TCF_CNF      |                            |
   <============================*                            |
   |                          (FAD 2)                        |
   |                            |                            |
```

(RA 1)

A RA-DATA indication primitive containing a SYNC frame is received by FAD.

(RA 2)

Receipt of a RA-READY indication primitive indicates that FAD may send data.

(RA 3)

A RA-DATA request primitive with TCF data is issued by FAD.

Steps RA 1-RA 3 are repeated for 1.5 seconds.

(FAD 1)

FAD issues a FAD-SND-TCF confirmation primitive and returns to the state IDLE.

5.6 Message Training Phase (FAD State MSG-TRA-TCF)

5.6.1 Training in MSG-TRA-TCF Phase

```
 T30                         FAD                         RA
  |                           |                           |
  |                           |        RA_DATA_IND        |
  |                           |           (TCF)           |
  |                           *<==========================*
  |                           |                         (RA 1)
  |                           |        RA_READY_IND       |
  |                           *<==========================*
  |                           |                         (RA 2)
  |                           |        RA_DATA_REQ        |
  |                           |          (SYNC)           |
  |                           *==========================>*
  |                           |                         (RA 3)
  |            ...            |            ...            |
  |                           |        RA_DATA_IND        |
  |                           |          (SYNC)           |
  |                           *<==========================*
  |                           |                         (RA 4)
  |      FAD_RCV_TCF_CNF      |                           |
  |       (error ratio)       |                           |
  <===========================*                           |
  |                        (FAD 1)                        |
  |                           |                           |
```

(RA 1)

A RA-DATA indication primitive containing a TCF frame is received by FAD.

(RA 2)

Receipt of a RA-READY indication primitive indicates that FAD may send a frame.

(RA 3)

FAD issues a RA-DATA request primitive with a SYNC frame.

Steps RA 1-RA 3 are repeated for 1.5 seconds, during which period FAD calculates the ratio of correctly-set TCF data (i.e., no. of TCF bytes (= 0) / total bytes received). After this period has elapsed FAD then awaits reception of a SYNC frame from the remote station.

(RA 1)

A RA-DATA indication primitive containing a SYNC frame is received by FAD.

(FAD 1)

FAD issues a FAD-RCV-TCF confirmation primitive containing the error ratio calculated during reception of TCF data. FAD then returns to the state IDLE.

5.7 Message Data Phase (FAD State MSG-REC)

5.7.1 Request from Local Station to send MSG data

```
T30                        FAD                        RA
 |                          |                          |
 |        FAD_DATA_REQ      |                          |
 |         (MSG data)       |                          |
 *=========================>*                          |
 |                       (FAD 1)                       |
 |                          |                          |
```

(FAD 1)

FAD receives a FAD-DATA request primitive from T30. The access rate is set to the rate requested in the primitive . FAD stores the MSG data for sending as well as the value of the FINAL flag contained in the primitive.

5.7.2 Training in MSG-REC Phase

```
 T30                           FAD                           RA
  |                             |                             |
  |                             |        RA_DATA_IND          |
  |                             |          (SYNC)             |
  |                             *<============================*
  |                             |                          (RA 1)
  |                             |       RA_READY_IND          |
  |                             *<============================*
  |                             |                          (RA 2)
  |                             |        RA_DATA_REQ          |
  |                             |         (STATUS)            |
  |                             *============================>*
  |                             |                          (RA 3)
  |                             |        RA_DATA_IND          |
  |                             |          (SYNC)             |
  |                             *<============================*
  |                             |                          (RA 4)
  |                             |       RA_READY_IND          |
  |                             *<============================*
  |                             |                          (RA 5)
  |                             |        RA_DATA_REQ          |
  |                             |          (SYNC)             |
  |                             *============================>*
  |                             |                          (RA 6)
  |                             |                             |
```

(RA 1)

A RA-DATA indication primitive containing a SYNC frame is received by FAD.

(RA 2)

Receipt of a RA-READY indication primitive indicates that FAD may send data.

(RA 3)

A RA-DATA request primitive with a STATUS frame is issued by FAD.

(RA 4)

A RA-DATA indication primitive containing a SYNC frame is received by FAD.

(RA 5)

Receipt of a RA-READY indication primitive indicates that FAD may send data.

(RA 6)

A RA-DATA request primitive with a SYNC frame is issued by FAD.

Steps RA 1-RA 6 are repeated until a STATUS frame is received from the remote station.

5.7.3 Start sending MSG data

```
T30                          FAD                          RA
 |                            |                            |
 |                            |        RA_DATA_IND         |
 |                            |          (STATUS)          |
 |                            *<===========================*
 |                            |                         (RA 1)
 |                            |       RA_READY_IND         |
 |                            *<===========================*
 |                            |                         (RA 2)
 |            ...             |            ...             |
 |                            |        RA_DATA_REQ         |
 |                            |         (MSG data)         |
 |                            *===========================>*
 |       FAD_READY_IND        |                         (RA 3)
 *<===========================*                            |
 |                         (FAD 1)                         |
 |                            |                            |
```

(RA 1)

A RA-DATA indication primitive containing a STATUS frame is received by FAD.

(RA 2)

Receipt of a RA-READY indication primitive indicates that FAD may send a frame.

(RA 3)

A RA-DATA request primitive containing the stored MSG data is issued by FAD.

(FAD 1)

FAD issues a FAD-READY indication primitive to T30 and awaits further data.

Steps RA 3 and FAD 1 are repeated for as long as the local station wishes to send MSG data, i.e. for as long as the the FINAL flag is set to FALSE in the last FAD-DATA request primitive received.

5.7.4 Conclusion of MSG-REC Phase

```
  T30                           FAD                           RA
   |                             |                             |
   |                             |         RA_DATA_IND         |
   |                             |           (SYNC)            |
   |                             *<============================*
   |                             |                          (RA 1)
   |                             |        RA_READY_IND         |
   |                             *<============================*
   |                             |                          (RA 2)
   |                             |         RA_DATA_REQ         |
   |                             |           (SYNC)            |
   |                             *============================>*
   |                             |                          (RA 3)
   |         FAD_DATA_CNF        |                             |
   <=============================*                             |
   |                          (FAD 1)                          |
   |                             |                             |
```

(RA 1)

A RA-DATA indication primitive containing a SYNC frame is received by FAD.

(RA 2)

Receipt of a RA-READY indication primitive indicates that FAD may send data.

(RA 3)

The FINAL flag has been set to TRUE in the last FAD-DATA request primitive received. FAD establishes that all stored MSG data has been sent and then issues a RA-DATA request primitive with a SYNC frame, indicating to the remote station that transmission of MSG data is complete.

(FAD 1)

FAD issues a FAD-DATA confirmation primitive and returns to the state IDLE.

5.8 Message Data Phase (FAD State MSG-TRA)

5.8.1 Training in MSG-TRA Phase

```
T30                         FAD                          RA
 |                           |                            |
 |                           |        RA_DATA_IND         |
 |                           |          (STATUS)          |
 |                           *<===========================*
 |                           |                          (RA 1)
 |                           |        RA_DATA_IND         |
 |                           |           (SYNC)           |
 |                           *<===========================*
 |                           |                          (RA 2)
 |             ...           |             ...            |
 |                           |        RA_READY_IND        |
 |                           *<===========================*
 |                           |                          (RA 3)
 |                           |        RA_DATA_REQ         |
 |                           |          (STATUS)          |
 |                           *===========================>*
 |                           |                          (RA 4)
 |                           |                            |
```

(RA 1)

A RA-DATA indication primitive containing a STATUS frame is received by FAD.

(RA 2)

A RA-DATA indication primitive containing a SYNC frame is received by FAD.

Alternative STATUS and SYNC frames continue to be received from the remote station.

(RA 3)

Receipt of a RA-READY indication primitive indicates that FAD may send data.

(RA 4)

FAD issues a RA-DATA request primitive with a STATUS frame and awaits reception of MSG data from the remote station.

5.8.2 Receive Fax Data

```
T30                           FAD                           RA
 |                             |                             |
 |                             |         RA_DATA_IND         |
 |                             |          (MSG data)         |
 |                             *<===========================*
 |                             |                         (RA 1)
 |                             |        RA_READY_IND         |
 |                             *<===========================*
 |                             |                         (RA 2)
 |                             |         RA_DATA_REQ         |
 |                             |            (SYNC)           |
 |                             *===========================>*
 |                             |                         (RA 3)
 |                             |                             |
```

(RA 1)
FAD receives a RA-DATA indication primitive containing MSG data. FAD stores the data prior to passing it to the higher-layer entities.

(RA 2)

Receipt of a RA-READY indication primitive indicates that FAD may send a frame.

(RA 3)

FAD issues a RA-DATA request primitive with a SYNC frame .

Steps RA 1-RA 3 are repeated for as long as the remote station continues to send MSG data.

5.8.3 Request for MSG Data from local station

```
T30                           FAD                           RA
 |                             |                             |
 |        FAD_READY_REQ        |                             |
 *===========================>*                              |
 |                          (FAD 1)                          |
 |         FAD_DATA_IND        |                             |
 |    (MSG data, not final)    |                             |
 *<===========================*                              |
 |                          (FAD 2)                          |
 |                             |                             |
```

(RA 1)
A request to send more MSG data is received by FAD in the form of a FAD-READY request primitive.

(RA 2)

FAD issues a FAD-DATA indication primitive containing MSG data received from the remote station. The FINAL flag is set to FALSE.

5.8.4 Request to stop passing received MSG data to the higher-layer entities

```
T30                            FAD                            RA
 |                              |                              |
 |       FAD_IGNORE_REQ         |                              |
 *============================>*                              |
 |                            (FAD 1)                          |
 |                              |                              |
```

(RA 1)
A request to stop passing received MSG data to the higher-layer entities is received by FAD in the form of a FAD-IGNORE request primitive.

FAD subsequently discards all MSG data received from the remote station.

5.8.5 Stop Receiving Fax Data

```
T30                            FAD                            RA
 |                              |                              |
 |                              |         RA_DATA_IND          |
 |                              |            (SYNC)            |
 |                              *<============================*
 |                              |                            (RA 1)
 |                              |         RA_READY_IND         |
 |                              *<============================*
 |                              |                            (RA 2)
 |                              |         RA_DATA_REQ          |
 |                              |            (SYNC)            |
 |                              *============================>*
 |                              |                            (RA 3)
 |             ...              |             ...              |
 |         FAD_DATA_IND         |                              |
 |    (empty buffer, final)     |                              |
 *<============================*                              |
 |                            (FAD 2)                          |
 |                              |                              |
```

(RA 1)

A RA-DATA request primitive with a SYNC frame is received by FAD, indicating that there is no more data to be received from the remote side.

(RA 2)

Receipt of a RA-READY indication primitive indicates that FAD may send a frame.

(RA 3)

A RA-DATA request primitive containing a SYNC frame is issued by FAD. The T_POSTMSG timer (75 msec) is then started.

Steps RA 1-3 are repeated at least five times.

(FAD 1)
FAD ensures that all data received has been forwarded and issues a FAD-DATA confirmation primitive with the FINAL flag parameter set to TRUE. FAD then returns to the state IDLE.

5.9 All States

5.9.1 Deactivate FAD

```
T30                           FAD                           RA
 |                             |                             |
 |     FAD_DEACTIVATE_REQ      |                             |
 *===========================>*                              |
 |                          (FAD 1)                          |
 |                             |                             |
```

(FAD 1)
When FAD receives a FAD-DEACTIVATE request primitive in any state it changes to the state NULL.

5.9.2 Set Transmission Rate

```
T30                           FAD                           RA
 |                             |                             |
 |       FAD_MODIFY_REQ        |                             |
 *===========================>*                              |
 |                          (FAD 1)                          |
 |                             |                             |
```

(FAD 1)
FAD receives a FAD-MODIFY request primitive containing the transmission rate for the TCH. This primitive is ignored in the NULL state. In any other state FAD returns to the IDLE state.

5.9.3 Error handling

```
T30                           FAD                           RA
 |                             |                             |
 |        FAD_ERROR_IND        |                             |
 <============================*                              |
 |                          (FAD 1)                          |
 |                             |                             |
```

(FAD 1)
FAD issues a FAD-ERROR indication primitive containing the cause of the error. This primitive is ignored in the NULL state. In any other state FAD returns to the IDLE state.

Appendices

A. Acronyms

DS-WCDMA	Direct Sequence/Spread Wideband Code Division Multiple Access

B. Glossary

International Mobile Telecommunication 2000 (IMT-2000/ITU-2000)	Formerly referred to as FPLMTS (Future Public Land-Mobile Telephone System), this is the ITU's specification/family of standards for 3G. This initiative provides a global infrastructure through both satellite and terrestrial systems, for fixed and mobile phone users. The family of standards is a framework comprising a mix/blend of systems providing global roaming. <URL: http://www.imt-2000.org/>