

# ETSI TS 127 010 V16.0.0 (2020-08)



**Digital cellular telecommunications system (Phase 2+) (GSM);  
Universal Mobile Telecommunications System (UMTS);  
LTE;  
Terminal Equipment to User Equipment (TE-UE)  
multiplexer protocol  
(3GPP TS 27.010 version 16.0.0 Release 16)**



---

**Reference**

RTS/TSGC-0327010vg00

---

**Keywords**

GSM,LTE,UMTS

**ETSI**

650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° 7803/88

---

**Important notice**

The present document can be downloaded from:

<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at [www.etsi.org/deliver](http://www.etsi.org/deliver).

Users of the present document should be aware that the document may be subject to revision or change of status.

Information on the current status of this and other ETSI documents is available at

<https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx>

If you find errors in the present document, please send your comment to one of the following services:

<https://portal.etsi.org/People/CommitteeSupportStaff.aspx>

---

**Copyright Notification**

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2020.

All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members.

**3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

**oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners.

**GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

---

## Intellectual Property Rights

### Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<https://ipr.etsi.org/>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

### Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

---

## Legal Notice

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under <http://webapp.etsi.org/key/queryform.asp>.

---

## Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

# Contents

Intellectual Property Rights .....	2
Legal Notice .....	2
Modal verbs terminology.....	2
Foreword.....	6
Introduction .....	6
1 Scope .....	7
2 References .....	7
3 Abbreviations .....	7
4 Overview of Multiplexing System .....	8
5 Non Error Recovery mode Options.....	9
5.1 Service Interface Definition .....	9
5.1.1 Service Definition Model.....	9
5.1.2 Start up services .....	10
5.1.3 DLC establishment services.....	11
5.1.4 Data services .....	12
5.1.5 Power Control services .....	12
5.1.5.1 Sleep services .....	12
5.1.5.2 Wakeup services .....	12
5.1.6 DLC Release services .....	12
5.1.7 Close down services.....	13
5.1.8 Control Services.....	13
5.1.8.1 27.010 Services .....	13
5.1.8.1.1 DLC parameter negotiation .....	13
5.1.8.1.2 DLC Service Negotiation service .....	14
5.1.8.1.3 Test service .....	14
5.1.8.1.4 Flow control services.....	14
5.1.8.2 Port Emulation Services .....	15
5.1.8.2.1 Remote DLC parameter negotiation service.....	15
5.1.8.2.2 DLC Control Parameter service .....	15
5.1.8.2.3 DLC Line status indication service.....	16
5.2 Frame Structure .....	16
5.2.1 Frame Fields .....	16
5.2.1.1 Flag Sequence Field .....	16
5.2.1.2 Address Field .....	16
5.2.1.3 Control Field .....	17
5.2.1.4 Information Field .....	17
5.2.1.5 Length Indicator .....	17
5.2.1.6 Frame Checking Sequence Field (FCS).....	18
5.2.2 Format Conventions.....	18
5.2.3 Frame Validity .....	19
5.2.4 Frame Abort.....	19
5.2.5 Inter-frame Fill.....	19
5.2.6 Basic Option .....	19
5.2.6.1 Constraint.....	19
5.2.7 Advanced Option.....	20
5.2.7.1 Control-octet transparency .....	20
5.2.7.2 Start/stop transmission - extended transparency .....	20
5.2.7.3 Flow-control transparency .....	20
5.2.7.4 Frame Structure.....	20
5.3 Frame Types.....	21
5.3.1 Set Asynchronous Balanced Mode (SABM) command.....	21
5.3.2 Unnumbered Acknowledgement (UA) response .....	21

5.3.3	Disconnected Mode (DM) response .....	21
5.3.4	Disconnect (DISC) command .....	21
5.3.5	Unnumbered information with header check (UIH) command and response .....	21
5.3.6	Unnumbered Information (UI) command and response .....	22
5.4	Procedures and States .....	22
5.4.1	DLC Establishment .....	22
5.4.2	DLC Release .....	22
5.4.3	Information Transfer .....	22
5.4.3.1	Information Data .....	22
5.4.3.2	Priority .....	23
5.4.4	Frame Variables .....	23
5.4.4.1	Functions of the poll bit .....	23
5.4.4.2	Functions of the final bit .....	24
5.4.5	Time-out considerations .....	24
5.4.6	Multiplexer Control Channel .....	24
5.4.6.1	Message format .....	24
5.4.6.2	Operating procedures .....	25
5.4.6.3	Message Type and Actions .....	25
5.4.6.3.1	DLC parameter negotiation (PN) .....	25
5.4.6.3.2	Power Saving Control (PSC) .....	27
5.4.6.3.3	Multiplexer close down (CLD) .....	27
5.4.6.3.4	Test Command (Test) .....	28
5.4.6.3.5	Flow Control On Command (FCon) .....	28
5.4.6.3.6	Flow Control Off Command (FCoff) .....	28
5.4.6.3.7	Modem Status Command (MSC) .....	28
5.4.6.3.8	Non Supported Command Response (NSC) .....	30
5.4.6.3.9	Remote Port Negotiation Command (RPN) .....	31
5.4.6.3.10	Remote Line Status Command(RLS) .....	33
5.4.6.3.11	Service Negotiation Command (SNC) .....	34
5.4.7	Power Control and Wake-up Mechanisms .....	36
5.4.8	Flow Control .....	37
5.4.8.1	RTR Flow Control .....	37
5.4.8.2	XON/XOFF Flow Control .....	37
5.5	Convergence Layers .....	38
5.5.1	Type 1 - Unstructured Octet Stream .....	38
5.5.2	Type 2 - Unstructured Octet Stream with flow control, break signal handling and transmission of V.24 signal states .....	38
5.5.3	Type 3 - Uninterruptible Framed Data .....	40
5.5.4	Type 4 - Interruptible Framed Data .....	40
5.6	DLCI Values .....	41
5.7	System Parameters .....	41
5.7.1	Acknowledgement Timer (T1) .....	42
5.7.2	Maximum Frame Size (N1) .....	42
5.7.3	Maximum number of retransmissions (N2) .....	42
5.7.4	Window Size (k) .....	42
5.7.5	Response Timer for multiplexer control channel (T2) .....	42
5.7.6	Response Timer for wake-up procedure(T3) .....	42
5.8	Start-up and close-down of multiplexer .....	42
5.8.1	Start-up procedure .....	42
5.8.2	Close-down procedure .....	43
6	Error Recovery Mode Option .....	43
6.1	Frame Types .....	43
6.1.1	Information transfer, I, command and response .....	43
6.1.2	Receive ready, RR, command and response .....	43
6.1.3	Receive not ready, RNR, command and response .....	44
6.1.4	Reject, REJ, command and response .....	44
6.2	Procedure and State .....	44
6.2.1	Frame state variables and sequence numbers .....	44
6.2.1.1	General .....	44
6.2.1.2	Send state variable V(S) .....	44
6.2.1.3	Send sequence number N(S) .....	44

6.2.1.4	Receive state variable V(R).....	44
6.2.1.5	Receive sequence number N(R).....	44
6.2.1.6	Use of the P/F bit to assist in error recovery .....	45
6.2.2	Exchange of information (I) frames.....	45
6.2.2.1	Sending I frames .....	45
6.2.2.2	Receiving I frames .....	45
6.2.2.3	Reception of incorrect frames .....	46
6.2.2.4	Station receiving acknowledgements .....	46
6.2.2.5	Exception conditions and recovery .....	46
6.2.2.5.1	Busy.....	46
6.2.2.5.2	N(S) sequence error .....	46
6.2.2.5.3	Poll/final (P/F) bit (checkpoint) recovery .....	46
6.2.2.5.4	REJ recovery .....	47
6.2.2.5.5	SABM Command .....	47
6.2.2.5.6	DISC Command .....	47
<b>Annex A (informative): Advice to TE software implementers .....</b>		<b>48</b>
<b>Annex B (informative): Explanatory notes on the CRC Calculation .....</b>		<b>49</b>
B.1	Example.....	49
B.2	Reflected bits.....	49
B.3	Implementation.....	50
B.3.1	Calculate FCS for the example given earlier .....	50
B.3.2	Check FCS for the example given earlier .....	50
B.3.3	The transmitter code.....	50
B.3.4	The receiver code .....	50
B.3.5	Reversed CRC table .....	51
<b>Annex C (informative): Change History .....</b>		<b>52</b>
History .....		53

---

# Foreword

This Technical Specification (TS) has been produced by the 3<sup>rd</sup> Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

---

# Introduction

The multiplexer protocol described in the present document operates between an UE and a TE and allows a number of simultaneous sessions over a normal serial asynchronous interface. Each session consists of a stream of bytes transferring various kinds of data; for instance, voice, fax, data, SMS, CBS, phonebook maintenance, battery status, GPRS, USSD etc. This permits, for example, SMS and CBS to be transferred to a TE when a data connection is in progress. Many other combinations are possible including digital voice. It is, for instance, possible to transfer digital voice in combination with SMS. The multiplexer allows a complete system to be partitioned in a flexible way between a UE and TE.

The design of the multiplexer is flexible and independent of UE/TE platforms, and allows existing applications to work without any modifications.

The multiplexer is designed, with special care for battery-powered devices, to include very important functionality such as power saving control and priorities. It is also specially designed to require minimum processing power and memory consumption.

The multiplexer is defined as a single mode with different options based on the ISO HDLC standard (ISO/IEC 13239) although the basic option is not in accordance with HDLC.

In the basic option, the multiplexer does not make use of any transparency mechanism or error recovery method. The advanced option uses the ISO HDLC standard transparency mechanism and gives the multiplexer an easy re-synchronisation method and the ability to operate over links which use DC1/DC3 (XON/XOFF) flow control. The advanced option also may include error-recovery for links subject to errors.

In its basic option, the multiplexer is intended for use in situations where the link between UE and TE is of a very good quality and where the HDLC transparency mechanism (byte stuffing) can not be implemented in the UE. If an UE supports the HDLC transparency mechanism, it shall be used by the multiplexer. The ISO HDLC transparency mechanism must be used if loss of synchronisation may occur caused by, for example, data over-runs or under-runs. The error-recovery option should be used in situations where the link is subject to errors.

The multiplexer is based on a control channel. On this channel, management information is exchanged, such as parameter negotiation, power saving control information, testing, flow control, close down etc.

The multiplexer is optional, but when supported, it is activated with the AT+CMUX command described in 3GPP TS 27.007 [4].

---

# 1 Scope

The scope of the present document is to define a multiplexing protocol between a UE and a TE. The multiplexing protocol can be used to send any data, for instance voice, SMS, USSD, fax etc.

The present document describes the protocol, but not the commands or data transported with it.

---

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

- [1] Void.
- [2] ISO/IEC 13239 (1997): "Information technology - Telecommunications and information exchange between systems - High-level data link control (HDLC) procedures".
- [3] 3GPP TS 27.005: "Use of Data Terminal Equipment - Data Circuit terminating Equipment (DTE - DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (CBS)".
- [4] 3GPP TS 27.007: "AT command set for User Equipment (UE)".
- [5] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".
- [6] 3GPP TS 46.021: "Half rate speech; Substitution and muting of lost frames for half rate speech traffic channels".
- [7] ISO/IEC 646 (1991): "Information technology - ISO 7-bit coded character set for information interchange".

---

# 3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

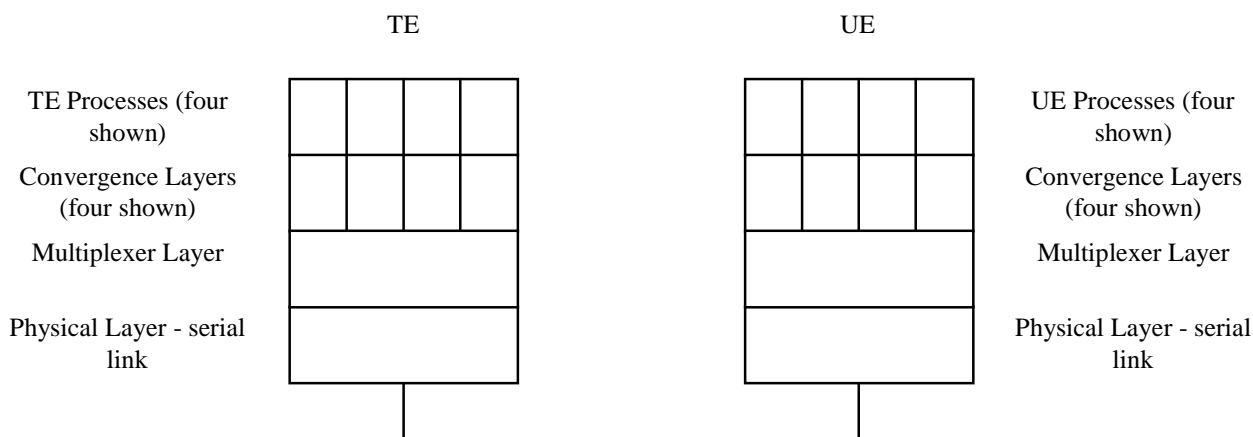
ABM	Asynchronous Balanced Mode
DLC	Data Link Connection
DM	Disconnected Mode
ERM	Error-Recovery Mode
FCS	Frame Check Sequence
MSC	Modem Status Command
PSC	Power Saving Control
SABM	Set Asynchronous Balanced Mode
UAU	Unnumbered Acknowledgement
UIH	Unnumbered Information with header Check
UI	Unnumbered Information

Additional abbreviations can be found in 3GPP TR 21.905 [5].



## 4 Overview of Multiplexing System

The multiplexer provides mechanisms for conveying streams of data between TE and UE over a single start-stop framed, serial link. Figure 1 shows the arrangement of the various protocol levels and functions. The multiplexer layer provides multiplexing of data arranged in octet streams with no other framing; if the structure of the data has to be conveyed, a convergence layer may be necessary. The present document defines some convergence layers, others may be added later.



**Figure 1: Protocol Stacks**

The multiplexer provides a virtual connection between a process in the TE and a similar process in the UE. For example, a PC application supporting SMS functions could be connected to the SMS handler in the UE via a multiplexer channel.

The present document uses start-stop transmission with eight-bit characters. Communication between the two multiplexing entities takes place using frames constructed as defined below.

Each channel between TE and UE is called a Data Link Connection (DLC) and is established separately and sequentially.

Each DLC may have individual flow control procedures for buffer management purposes and the aggregate link also has overall flow control mechanisms.

DLCs have two modes of operation; Error-Recovery Mode (ERM) and non-error-recovery mode (non-ERM), the choice of mode is made when a DLC is established. DLCs using error recovery mode may exist on the same link as DLCs using non-error recovery mode. If the error-recovery mode (ERM) is to be used at least on one DLC, then the multiplexer must be configured with the ISO HDLC transparency mechanism. The use of error recovery mode is optional. Non-error recovery mode uses the UI frame or UIH frame to carry user data; error recovery mode uses the I frame.

The multiplexer has three operating options, basic, advanced without error recovery and advanced with error recovery. The characteristics of the options are:

### Basic:

- length indicator used instead of the HDLC transparency mechanism;
- different flag octet from that used by HDLC;
- can not be used on links which use XON/XOFF flow control;
- may have longer recovery procedure from loss of synchronisation.

### Advanced without error recovery:

- asynchronous HDLC procedures in accordance with ISO/IEC 13239;
- can be used on links which use XON/XOFF flow control;

- recovers quickly from loss of synchronisation.

Advanced with error recovery:

- Uses HDLC error-recovery procedures.

---

## 5 Non Error Recovery mode Options

This clause describes the non-error-recovery options (basic and advanced) of the multiplexer. The main are given below:

- a simple set of procedures with no error recovery mechanism, for use on reliable connections;
- data transparency is provided by the HDLC mechanism (advanced option only);
- a multiplexer control channel which conveys management and control information between the UE and TE;
- a mechanism that permits either UE or TE to enter power-saving modes without compromising the integrity of the multiplexer;
- a comprehensive set of convergence layers which enables many types of data to be carried while preserving the structure of the original data.

The use of the transparency mechanism must be set up at the beginning of the multiplexing session. It is a characteristic for the entire multiplexing session.

The simple set of procedures uses UIH frames to transmit information; these frames are easy to process because their structure permits the HDLC Frame Check Sequence (FCS) to be pre-calculated rather than being constructed on a character-by character basis. The procedures used are very straightforward and it is not necessary to implement the usual HDLC state machines.

UI frames or UIH frames may be used for those channels where the timely delivery of the information is more important than its reliability because erroneous frames will be discarded. UI frames would be used in those cases where it is important that the data delivered is accurate.

### 5.1 Service Interface Definition

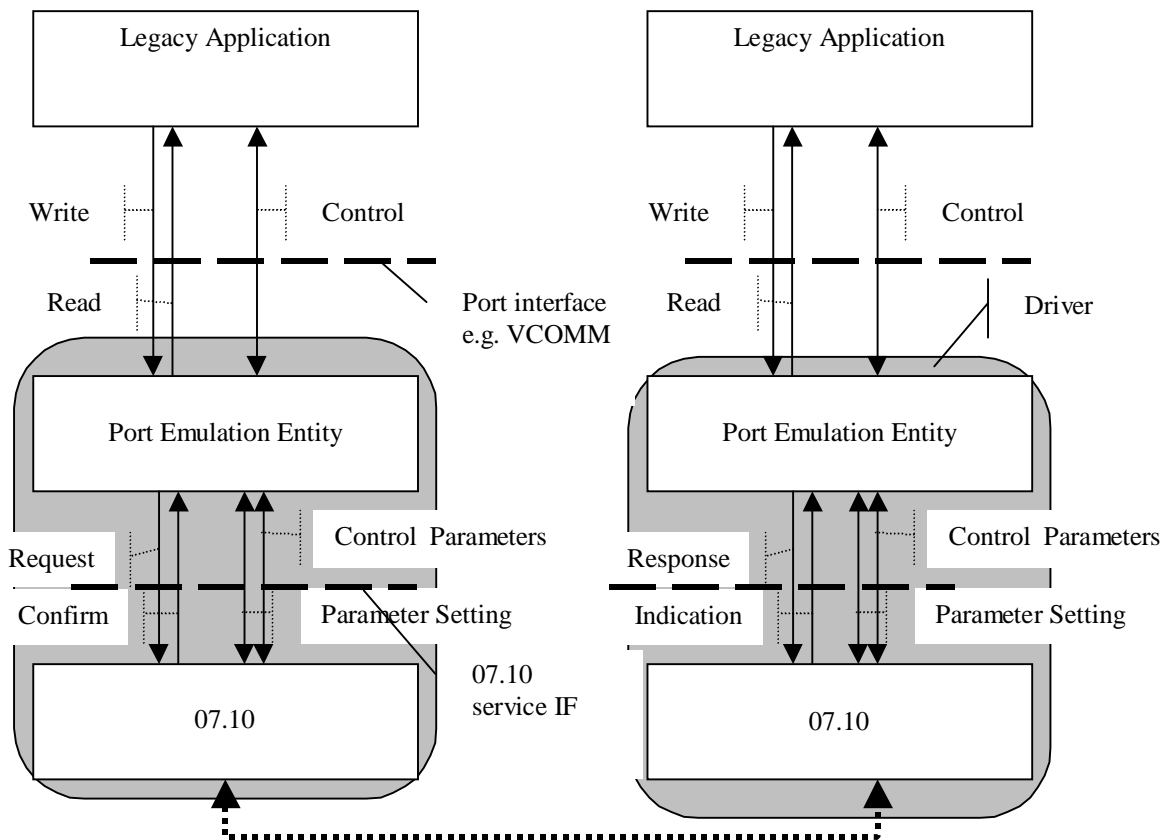
This clause describes the services provided by the 3GPP TS 27.010 data link layer to the upper layer. The interface is specified in terms of primitives and parameters.

NOTE: This clause is only for information, detailed description of the parameters is found in the following clauses.

#### 5.1.1 Service Definition Model

The present document is intended to define a protocol that can be used to emulate a serial port. In most systems the 27.010 will be a part of a port driver which includes a port emulation entity that must support existing communication APIs. The communication APIs vary from operating system to operating system and from device to device. The present document does not specify how 27.010 is used by a port driver to emulate an existing API but instead focus on a set of services that can be used by all port drivers. Port drivers are not required to use all the services of 27.010.

The figure below shows a model of how 27.010 fits into a typical system.



The legacy application utilises a conventional serial port communication interface. The port emulation entity maps a system specific communication interface to 27.010 services. The 27.010 provides several transparent data stream channels and a control channel. The port interface is the application programmers interface for communication. It varies from system to system and one example is Virtual comm ports in windows.

## 5.1.2 Start up services

These services are used to start the TS 27.010 multiplexer operation over a serial channel. The following services are provided:

```

TS0710_START.request (mode, system_parameters);
TS0710_START.indication (mode, system_parameters);
TS0710_START.response (mode, system_parameters, accept);
TS0710_START.confirm (mode, system_parameters, accept).

```

**Description:** the *request* primitive is used to request that the multiplexer mode to be turned on in the desired mode and system parameters. The *indication* primitive transfers the request to start multiplexer operation along with the desired mode and system parameters to the upper layer of the target device. If the target device accepts the request by issuing an affirmative *response* primitive, the suggested mode and system parameters will become valid. The *confirm* primitive is returned to the upper layer of the requesting device. A successful establishment of the multiplexer mode is indicated by the *accept* parameter being set to "true". If the *accept* parameter is set to "false" the returned values for the other parameters are those suggested by the responding device.

Parameters:

*mode* = [Basic | HDLC - UIH frames | HDLC - UI frames | HDLC - frames]. (Note that the frame type for HDLC mode refers to the multiplexer control channel. For subsequently opened DLCs this parameter can be negotiated.

*system\_parameters* = Port speed [9,6 | 19,2 | 38,4 | 57,6 | 115,2 | 230,4 kBit/s],  
 Maximum Frame Size [1 – 128 in Basic mode,  
 1 – 512 in HDLC modes  
 default: 31 for the basic  
 option and 64 for the  
 advanced option]  
 Acknowledgement Timer [0,01s-2,55s,  
 default: 0,1s]  
 Maximum number of retransmissions [0 – 100,  
 default : 3]  
 Response timer for the multiplexer control  
 channel [0,01s-2,55s, default: 0,3s]  
 Wake up response timer [1s – 255s, default 10s]  
 Window size for error recovery mode [1 – 7,  
 default : 2]

*accept* = [true | false]

Support of the mode parameter is optional. If the mode parameter is omitted, Basic mode is implied. Note that some of the above system parameters can be redefined for the individual DLCs, see below under DLC establishment services.

### 5.1.3 DLC establishment services

The DLC establishment services are used to open DLC's on the multiplexer channel. The following services are provided:

```
TS_0710_DLC_ESTABLISHMENT.request(DLCI, system_parameters);
TS_0710_DLC_ESTABLISHMENT.indication(DLCI, system_parameters);
TS_0710_DLC_ESTABLISHMENT.response(DLCI, system_parameters, accept);
TS_0710_DLC_ESTABLISHMENT.confirm(DLCI, system_parameters, accept).
```

**Description:** The transmitting device uses the *request* primitive initiate the establishment of a new DLC with a desired set of system parameters on the multiplexer channel. The *indication* primitive is passed to the upper layer by the TS 27.010 layer of the receiving device on reception of the DLC establishment request. The receiving device uses the *response* primitive to either accept or reject the proposed DLCI with its system parameters. On rejection, it is possible to suggest a modified set of system parameters. The *confirm* primitive is passed to the upper layer of the transmitting device on reception of the response from the receiving device.

#### Parameters:

*DLCI* = 1-63 (DLCI number)

*System parameters* = Type of frame [UIH | UI | I, default: UIH],  
 Convergence layer [1 - 4, default: 1]  
 Priority [0-63]  
 Acknowledgement Timer [0,01s-2,55s,  
 default: 0,1s]  
 Maximum Frame Size [1 – 32768,  
 default: 31 for the basic option and  
 64 for the advanced option]  
 Maximum number of retransmissions [0 – 255,  
 default : 3]  
 Window size for error recovery mode [1 – 7,  
 default : 2]

*Accept* = [true | false]

All entries in the system parameters parameter are optional. The entries not implemented assumes the default values.

## 5.1.4 Data services

The data services provided are:

TS\_0710\_DATA.request(DLCI, User\_data);

TS\_0710\_DATA.indication(DLCI, User\_data).

**Description:** the transmitting unit initiates transmission of data using the frame type specified for the chosen DLCI by means of the request primitive. The transmitted data is delivered to the upper layer of the receiving by the *indication* primitive. No confirmation primitive exists even for the error recovery mode. In this mode TS 27.010 will take care of all mechanisms involved in the error checking and thus deliver data error free.

### Parameters:

*DLCI* = [1 – 63] DLC over which the data is to be transmitted.

*User\_data* = Data to be transferred organised in accordance with the convergence layer of the DLC

## 5.1.5 Power Control services

In some application it might be desirable for either the DTE or the DCE to enter a power saving mode with a minimum of communication activities taking place. Services that support this functionality are the Sleep services and the Wakeup services.

### 5.1.5.1 Sleep services

TS\_0710\_SLEEP.request;

TS\_0710\_SLEEP.indication;

TS\_0710\_SLEEP.confirm.

**Description:** the *request* primitive is used to advise the receiving device that the transmitter wishes to enter a low power state. The TS 27.010 layer of the receiving unit sends an *indication* primitive to the upper layer in order to inform that the transmitting unit has entered the power saving state. The TS 27.010 layer will automatically transmit an acknowledge message to the transmitting device, thus no *response* primitive is required. The *confirm* primitive is sent to the upper layer of the transmitting device when the low power request has been received, and indicates that the TS 27.010 layer has entered the low power mode. Note that the Receiving device is not required to enter a low power mode, but it will be considered to have done so by the TS 27.010 layer.

### 5.1.5.2 Wakeup services

TS\_0710\_WAKEUP.indication;

TS\_0710\_WAKEUP.response.

**Description:** the *indication* primitive is sent to the upper layer when the TS 27.010 layer of the receiving unit receives a request to wake up from the power saving state. When the receiving device is ready to resume operation on the multiplexer channel this is indicated to the TS 27.010 layer in the receiving unit by means of the *response* primitive. Sins the wakeup routine is initiated by the transmitting device attempting to communicate, neither *request* nor *confirm* primitives are provided for the wakeup service. The transmitting device instead uses the Data services described below.

## 5.1.6 DLC Release services

The DLC release services are used to disconnect a DLC. The following services are provided:

TS\_0710\_DLC\_RELEASE.request(DLCI);

TS\_0710\_DLC\_RELEASE.indication(DLCI).

**Description:** The *request* primitive is used by the upper layer in the transmitting device to initiate close down of the selected DLC in TS 27.010. The TS 27.010 layer of the receiving device uses the *indication* primitive to inform the upper layer that the DLC has been closed down.

**Parameters:**

*DLCI* = [1 – 63] Number of the DLC to be released.

## 5.1.7 Close down services

The Close down services are used to terminate multiplexer operation on the serial channel and resume AT mode. The services provided are:

TS\_0710\_CLOSE.*request*;

TS\_0710\_CLOSE.*indication*.

**Description:** when the request primitive is passed to the TS 27.010 layer of the transmitting device close down of the multiplexer mode is initiated and a close down command is sent to the receiving device. On reception of the close down command the TS 27.010 layer of the receiving device sends the indication primitive to the upper layer and the multiplexer mode is terminated.

## 5.1.8 Control Services

### 5.1.8.1 27.010 Services

#### 5.1.8.1.1 DLC parameter negotiation

These services are used to negotiate and set parameters for a specific DLC. The following services are provided:

TS0710\_PARNEG.*request* (*DLC*, *DLC parameters*);

TS0710\_PARNEG.*indication* (*DLC*, *DLC parameters*);

TS0710\_PARNEG.*response* (*DLC*, *DLC parameters*, *accept*);

TS0710\_PARNEG.*confirm* (*DLC*, *DLC parameters*, *accept*).

**Description:** the *request* primitive is used to request that the remote 27.010 entity changes a specific DLC connection parameters. An indication is sent to the remote port emulation entity. The remote emulation entity replies with a response which is forwarded as a confirmation to the originating port emulating entity.

*DLC\_parameters* = frame type [ UIH | UI | I ,  
 default: UIH ]  
 Convergence Layer Type [ Type 1 | Type 2 | Type 3 | Type 4,  
 default: Type 1 ]  
 Priority [1-63,  
 default: according to table in clause 5.6]  
 Acknowledgement timer [10 ms - 25.5 sec,  
 default: 100 ms]  
 Maximum Frame Size [1 – 32768,  
 default: 31 for the basic option and  
 64 for the advanced option]  
 Maximum number of retransmissions [0 – 100,  
 default : 3]  
 Response timer for the multiplexor control  
 channel [0,01s-2,55s, default: 0,3s]  
 Wake up response timer [1s – 255s, default 10s]  
 Window size for error recovery mode [1 – 7,  
 default : 2]

*accept* = [true | false]

### 5.1.8.1.2 DLC Service Negotiation service

These services are used to negotiate and set a specific service on a DLC. The following services are provided:

TS0710\_SERVNEG.request (*DLC, Service\_parameters*);  
 TS0710\_SERVNEG.indication (*DLC, Service\_parameters*);  
 TS0710\_SERVNEG.response (*DLC, Service parameters, accept*);  
 TS0710\_SERVNEG.confirm (*DLC, Service\_parameters, accept*).

**Description:** the *request* primitive is used to request a specific service on a DLC. The indication is sent to the other port emulation. The remote port emulation entity replies with a response containing accepted or possible services. The originating port emulation entity receives a confirm on the request with either an accept or a possible service list.

*service\_parameters* = Service [ data | voice 64kbit/s A-law PCM | reserved 1 | reserved 2 ],  
 voice codec [ 3GPP TS 46.021 | 64kbit/s u-law PCM | coded ADPCM 32kbit/s | coded half rate  
 | 128 kbit/s PCM | reserved ]

### 5.1.8.1.3 Test service

These services are used to test the communication link between two 27.010 entities. The following services are provided:

TS0710\_TEST.request (*Test data*);  
 TS0710\_TEST.confirm (*Test data*).

**Description:** the *request* primitive is used to request a test of the communication link. The data is sent to the remote entity, which loops it back. The confirmation is sent to the originating port emulation entity containing the looped data.

*Test Data* = Data to be transferred as a test pattern, organised in accordance with the convergence layer of the 27.010 control channel.

### 5.1.8.1.4 Flow control services

The flow control services provided are:

TS\_0710\_FLOW.request(DLCI,State);  
 TS\_0710\_FLOW.indication(DLCI, State).

#### **Description:**

The request primitive with *State = disable* disables the issuing of TS\_0710\_DATA.indications by the 27.010 entity. The request primitive with *State = enable* enables the issuing of TS\_0710\_DATA.indications by the 27.010 entity. These requests may or may not result in the remote 27.010 entity issuing a TS\_0710\_FLOW.indication to the remote service user, depending on the states of the buffers in the 27.010 entities.

The indication primitive with *State = disable* disables the issuing of TS\_0710\_DATA.requests by the service user. The indication primitive with *State = enable* enables the issuing of TS\_0710\_DATA.requests by the service user. These indications may or may not have resulted from the receipt by the remote 27.010 entity of a TS\_0710\_FLOW.request from the remote service user. They may have been issued by the local 27.010 entity as a result of its buffer state.

The initial state of the 27.010 entity is with data flow enabled.

#### **Parameters:**

*DLCI* = [1 – 63] DLC over which the data is to be transmitted.

*State* = *enabled* (data may be transferred), *disabled* (data may not be transferred)

## 5.1.8.2 Port Emulation Services

### 5.1.8.2.1 Remote DLC parameter negotiation service

These services are used to negotiate and set of parameters for a remote communication port. The following services are provided:

```
TS0710_PORTNEG.request (DLC, Port_parameters);
TS0710_PORTNEG.indication (DLC, Port_parameters);
TS0710_PORTNEG.response (DLC, Port parameters, accept);
TS0710_PORTNEG.confirm (DLC, Port_parameters, accept).
```

**Description:** The *request* primitive is used to request that the remote port changes its parameters. The indication is sent to the other port emulation entity. The remote port emulation entity replies with a response. A confirm is sent to the originating port entity.

```
port_parameters = Port speed [2,4 | 4,8 | 7,2 | 9,6 | 19,2 | 38,4 | 57,6 | 115,2 |
230,4 kBit/s],
Data bits [ 5 | 6 | 7 | 8,
default: 8 bits |
Stop bits [ 1 | 1,5,
default: 1 bit |
Parity [ no parity | parity,
default: no parity |
Parity Type [ odd | even | mark | space]
```

```
accept = [true | false]
```

### 5.1.8.2.2 DLC Control Parameter service

The DLC Control Parameter service is used to convey control parameters between Port Emulation Entities. Default values should be assumed if no control parameter has been designated since the DLC has been made. This service is to control a specific DLC. It includes such as flow control, Modem signals, Break. The following services are provided:

```
TS0710_CONTROL.request (DLC, Control_parameters);
TS0710_CONTROL.indication (DLC, Contol_parameters);
TS0710_CONTROL.response (DLC, Contro_parameters);
TS0710_CONTROL.confirm (DLC, Control_parameters).
```

**Description:** the *request* primitive is used to convey control information to the remote port. The indication is sent to the other port emulation entity. The remote port emulation entity replies with a response which is sent to the originating 27.010 entity. A confirm is sent back to the port emulation entity.

```
system_parameters = Modem Signal [DTR/DSR | RTS/CTS | RI | DCD ],
Break Signal [0—3 s in steps of 200 ms,
default 0ms ],
Buffers [do not discard buffers, discard buffer
default: do not discard buffers],
Break signal sequence [ as soon as possible | in sequence,
default: in sequence]
```



### 5.1.8.2.3 DLC Line status indication service

These services are used to indicate a DLC line status to a remote port emulation entity.. The following services are provided:

TS0710\_PORTNEG.request (DLC, *Line Status parameter*);

TS0710\_PORTNEG.indication (DLC, *Line Status parameter*).

**Description:** the *request* primitive is used to send the line status to the remote device. The indication is sent to the other port emulation entity. The remote port emulation does not reply.

*Line status parameter* = Port speed [no errors, overrun error, parity error, framing error]

## 5.2 Frame Structure

All information transmitted between the TE and UE is conveyed in frames.

### 5.2.1 Frame Fields

The frame structure is composed of an opening and a closing flag, an Address field, a Control field, an Information field and FCS field. A length indication field is present in each frame if no transparency mechanism is used for the multiplexing session.

#### 5.2.1.1 Flag Sequence Field

Each frame begins and ends with a flag sequence octet which is defined as a constant bit pattern.

#### 5.2.1.2 Address Field

The address field consists of a single octet. It contains the Data Link Connection Identifier (DLCI), the C/R bit and the address field extension bit as shown in figure 2.

Bit No.	1	2	3	4	5	6	7	8
	EA	C/R	D L		C I			

**Figure 2: Format of Address Field**

The DLCI is used to identify an individual user information stream as well as to identify connections between TE and UE. Multiple DLCIs shall be supported but the number is implementation-specific. The DLCIs are dynamically assigned. The values used for specific DLCIs are given in clause 5.6.

The C/R (command/response) bit identifies the frame as either a command or a response. In conformance with HDLC rules, a command frame contains the address of the data link connection entity to which it is transmitted while a response frame contains the address of the data link connection entity transmitting the frame. For a given DLC, the DLCI value of the address field remains the same but the C/R bit changes, as shown in table 1.

**Table 1: Command/response bit usage**

Command/response	Direction		C/R value	
Command	Initiator	————→	Responder	1
	Responder	————→	Initiator	0
Response	Initiator	————→	Responder	0
	Responder	————→	Initiator	1

Initiator is the station that take the initiative to initialize the multiplexer (i.e. sends the SABM command at DLCI 0 ) and the responder is the station that accepts the initialization of the multiplexer (i.e. sends the UA response at DLCI 0)

See clause 5.4.3.1 for more details about C/R bit.

According to the rules of ISO/IEC 13239, the range of the address field may be extended by use of the EA bit. When the EA bit is set to 1 in an octet, it signifies that this octet is the last octet of the address field. When the EA bit is set to 0, it signifies that another octet of the address field follows. In the present document there is only one address octet so the EA bit is always set to 1. Note that future amendments to the present document may extend the address field and use the EA bit.

### 5.2.1.3 Control Field

The content of the control field defines the type of frame. The control fields of the frames used in the present document are described in table 2.

**Table 2: Coding of Control Field**

Frame Type	1	2	3	4	5	6	7	8	Notes
SABM (Set Asynchronous Balanced Mode)	1	1	1	1	P/F	1	0	0	
UA (Unnumbered Acknowledgement)	1	1	0	0	P/F	1	1	0	
DM (Disconnected Mode)	1	1	1	1	P/F	0	0	0	
DISC (Disconnect)	1	1	0	0	P/F	0	1	0	
UIH (Unnumbered Information with Header check)	1	1	1	1	P/F	1	1	1	
UI (Unnumbered Information)	1	1	0	0	P/F	0	0	0	Optional

In table 2, P/F is the Poll/Final bit. The functions of these bits are described later.

### 5.2.1.4 Information Field

The information field is the payload of the frame and carries the user data and any convergence layer information. The field is octet structured. The information field is only present in I frames, UI frames and UIH frames.

### 5.2.1.5 Length Indicator

This field is present only in case when basic option is activated.

It has the following format:

Bit	1	2	3	4	5	6	7	8
	E/A	L1	L2	L3	L4	L5	L6	L7

**Figure 3: Length field, first byte**

The L1 to L7 bits indicates the length of the following data field. The default length is 31 bytes.

According to the rule of ISO/IEC 13239, the range of the length field may be extended by use of the EA bit. When the EA bit is set to 1 in an octet, it signifies that this octet is the last octet of the length field. When the EA bit is set to 0, it signifies that a second octet of the length field follows. The total length of the length field is in that case 15bits, L1-L15.

The second octet of the length field (only present when the EA field in the first byte is set to 0) format:

Bit	1	2	3	4	5	6	7	8
	L8	L9	L10	L11	L12	L13	L14	L15

**Figure 4: Length field, second byte**

The length field shall always be present, even if the data field is empty.

### 5.2.1.6 Frame Checking Sequence Field (FCS)

The FCS shall be the ones complement of the sum (modulo 2) of:

- a) the remainder of

$$x^k (x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x^1 + 1)$$

divided (modulo 2) by the generator polynomial

$$x^8 + x^2 + x + 1,$$

where  $k$  is the number of bits in the frame existing between, but not including, the final bit of the opening flag and the first bit of the FCS, excluding start and stop elements (start/stop transmission), and bits (synchronous transmission) and octets (start/stop transmission) inserted for transparency, and

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

$$x^8 + x^2 + x + 1$$

of the product of  $x^8$  by the content of the frame existing between, but not including, the final bit of the opening flag and the first bit of the FCS, excluding start and stop elements (start/stop transmission), and bits (synchronous transmission) and octets (start/stop transmission) inserted for transparency.

As a typical implementation, at the transmitter, the initial content of the register of the device computing the remainder of the division is preset to all ones and is then modified by division by the generator polynomial (as described above) of the address, control and information fields; the ones complement of the resulting remainder is transmitted as the 8-bit FCS.

At the receiver, the initial content of the register of the device computing the remainder is preset to all ones. The final remainder after multiplication by  $x^8$  and then division (modulo 2) by the generator polynomial

$$x^8 + x^2 + x + 1$$

of the serial incoming protected bits and the FCS, will be 1111 0011 ( $x^7$  through  $x^0$ , respectively) in the absence of transmission errors.

In the case of the UIH frame, the contents of the I-field shall not be included in the FCS calculation. FCS is calculated on the contents of the address, control and length fields only. This means that only the delivery to the correct DLCI is protected, but not the information. The FCS is calculated in the normal manner for all other frames in Table 2.

## 5.2.2 Format Conventions

All transmitted characters will be sent using one start bit, eight data bits, no parity bit and one stop bit.

In the field descriptions, bit 1 is transmitted first.

Addresses, commands, responses and sequence numbers shall be transmitted low-order bit first (for example, the first bit of the sequence number that is transmitted shall have the weight  $2^0$ ).

The FCS shall be transmitted to the line commencing with the coefficient of the highest term.

**NOTE:** The use of these conventions in the present document means that octet values are often expressed in the reverse bit order from conventions used in many other standards. The conventions are used here because of the importance of the correct order of bit transmission; care should be taken during implementation.

### 5.2.3 Frame Validity

An invalid frame in the frame format is one which meets any one (or more) of the following conditions:

- is not properly bounded by two flags;
- does not have at least three octets between flags after removal of characters inserted for transparency;
- indicates presence of a transmission error in that the FCS check fails;
- contains an address field with more than one octet.

Invalid frames shall be discarded without notification to the sender. Actions taken by the multiplexer to indicate reception of an invalid frame to the UE or TE are left to implementers. However, an indication that a frame with an FCS error has been received may be of use when supporting DLCs for voice/audio.

As an optional procedure in response to an invalid frame in error recovery mode, a receiver may transmit an REJ frame.

### 5.2.4 Frame Abort

Aborting a frame is not supported.

### 5.2.5 Inter-frame Fill

The time between frames shall be filled by sending continuous stop-polarity except in the case of the wake-up procedure (see clause 5.4.7). The receiver shall also operate correctly if the time between frames is filled with flag characters. If a receiver receives more than three consecutive flags it shall begin to transmit continuous flags at the first available time (see clause 5.4.7).

### 5.2.6 Basic Option

In this case, opening flag and closing flags may appear in the Information field of the frame. The flags cannot be used to determine beginning and end of a frame. A length indication for the frame must be given instead. The frame structure is then as follows:

Flag	Address	Control	Length Indicator	Information	FCS	Flag
1 octet	1 octet	1 octet	1or2 octets	Unspecified length but integral number of octets	1 octet	1 octet

**Figure 5: Frame Structure for Basic option**

The flag field in basic option has the following format.

Bit	1	2	3	4	5	6	7	8
	1	0	0	1	1	1	1	1

**Figure 6: Flag field in basic option**

#### 5.2.6.1 Constraint

The closing flag may also be the opening flag of the following frame.

The flag value is different from the one used when the advanced option is activated.

Operation on link using DC1/XON and DC3/XOFF control characters defined in ISO/IEC 646 is not supported.

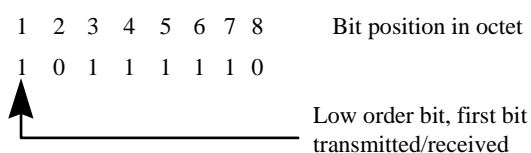
## 5.2.7 Advanced Option

If the advanced option is activated at the beginning of the multiplexing session, then it is used for all frames. This mechanism is based on a control octet transparency. It is based on a unique appearance of the opening and the closing flag in each frame. These flags will never appear in the information field of the frame. This mechanism allows a very quick synchronisation if a loss of synchronisation has occurred on the TE-UE link.

### 5.2.7.1 Control-octet transparency

The following transparency mechanism shall be applied to each frame from address field to FCS field inclusive.

The control escape octet is a transparency identifier that identifies an octet occurring within a frame to which the following transparency procedure is applied. The encoding of the control escape octet is:



The transmitter shall examine the frame between the opening and closing flag sequences including the address, control, and FCS fields and, following completion of the FCS calculation, shall:

- Upon the occurrence of the flag or a control escape octet, complement the 6<sup>th</sup> bit of the octet; and
- Insert a control escape octet immediately preceding the octet resulting from the above prior to transmission.

The receiver shall examine the frame between the two flag octets and shall, upon receipt of a control escape octet and prior to FCS calculation:

- Discard the control escape octet; and
- Restore the immediately following octet by complementing its 6<sup>th</sup> bit.

Other octet values may optionally be included in the transparency procedure by the transmitter. Such inclusion shall be subject to prior system/application agreement.

### 5.2.7.2 Start/stop transmission - extended transparency

The transmitter may apply the above transparency procedure to other octets in addition to the flag and control escape octets. At present, the only other octets are flow-control characters. The procedure is described in clause 5.2.6.3.

### 5.2.7.3 Flow-control transparency

The flow-control transparency option provides transparency processing for the DC1/XON and DC3/XOFF control characters defined in ISO/IEC 646 (i.e., 1000100x and 1100100x, respectively, where x may be either 0 or 1). This has the effect of assuring that the octet stream does not contain values which could be interpreted by intermediate equipment as flow control characters (regardless of parity).

### 5.2.7.4 Frame Structure

The frame structure is shown in figure 7. Note that this structure does not include information added for synchronisation (i.e. Start and stop bits) or transparency purposes. The order of transmission is from left to right.

In case the Transparency mechanism is activated, the frame structure is as follows.

Flag	Address	Control	Information	FCS	Flag
1 octet	1 octet	1 octet	Unspecified length but integral number of octets	1 octet	1 octet

**Figure 7: Frame Structure for Advanced option**

The flag field in advanced option has the following format.

Bit	1	2	3	4	5	6	7	8
	0	1	1	1	1	1	1	0

**Figure 8: Flag field in advanced option**

NOTE: The closing flag may also be the opening flag of the following frame.

## 5.3 Frame Types

### 5.3.1 Set Asynchronous Balanced Mode (SABM) command

The SABM command shall be used to place the addressed station in the Asynchronous Balanced Mode (ABM) where all control fields shall be one octet in length. The station shall confirm acceptance of the SABM command by transmission of a UA response at the first opportunity. Upon acceptance of this command, the DLC send and receive state variables shall be set to zero.

### 5.3.2 Unnumbered Acknowledgement (UA) response

The UA response shall be used by the station to acknowledge the receipt and acceptance of SABM and DISC commands.

### 5.3.3 Disconnected Mode (DM) response

The DM response shall be used to report a status where the station is logically disconnected from the data link. When in disconnected mode no commands are accepted until the disconnected mode is terminated by the receipt of a SABM command. If a DISC command is received while in disconnected mode a DM response should be sent.

### 5.3.4 Disconnect (DISC) command

The DISC command shall be used to terminate an operational or initialization mode previously set by a command. It shall be used to inform one station that the other station is suspending operation and that the station should assume a logically disconnected mode. Prior to actioning the command, the receiving station shall confirm the acceptance of the DISC command by the transmission of a UA response.

DISC command sent at DLCI 0 have the same meaning as the Multiplexer Close Down command (see clause 5.4.6.3.3). See also clause 5.8.2 for more information about the Close-down procedure.

### 5.3.5 Unnumbered information with header check (UIH) command and response

The UIH command/response shall be used to send information without affecting the V(S) or V(R) variables at either station. UIH is used where the integrity of the information being transferred is of lesser importance than its delivery to the correct DLCI. For the UIH frame, the FCS shall be calculated over only the address, control and length fields.

Reception of the UIH command/response is not sequence number verified by the data link procedures; therefore, the UIH frame may be lost if a data link exception occurs during transmission of the protected portion of the command, or duplicated if an exception condition occurs during any reply to the command. There is no specified response to the UIH command/response.

### 5.3.6 Unnumbered Information (UI) command and response

The UI command/response shall be used to send information without affecting the V(S) or V(R) variables at either station. Reception of the UI command/response is not sequence number verified by the data link procedures; therefore, the UI frame may be lost if a data link exception occurs during transmission of the protected portion of the command, or duplicated if an exception condition occurs during any reply to the command. There is no specified response to the UI command/response.

For the UI frame, the FCS shall be calculated over all fields (Address, Control, Length Indicator, Information).

Support of UI frames is optional.

## 5.4 Procedures and States

### 5.4.1 DLC Establishment

In most cases the establishment of a DLC will be initiated by the TE, however, the protocol is balanced and the initiation may come from the UE. The action taken by the higher layers of the TE upon the initiation of the establishment of a DLC from the UE is outside the scope of the present document.

The station wishing to establish a DLC transmits a SABM frame with the P-bit set to 1. The address field contains the DLCI value associated with the desired connection. If the responding station is ready to establish the connection it will reply with a UA frame with the F-bit set to 1. If the responding station is not ready or unwilling to establish the particular DLC it will reply with a DM frame with the F-bit set to 1.

Once a DLC has been established the stations are both said to be in a connected mode, for the particular DLC, and transfer of information may commence.

If no UA or DM response has been received after T1 the initiating station may retransmit the SABM. This action may be repeated until a response is obtained or action is taken by a higher layer.

If no negotiation procedure is used, DLC parameters are the default one.

### 5.4.2 DLC Release

The release of a DLC may be initiated by either station by the transmission of a DISC frame with the P-bit set to one. Confirmation of the DLC release is signalled by the other station sending a UA frame with the F-bit set to 1. Once the DLC has been released the stations enter disconnected mode for that particular DLC.

If the station receiving the DISC command is already in a disconnected mode it will send a DM response.

If no UA or DM response has been received after T1 the initiating station may retransmit the DISC. This action may be repeated until a response is obtained or action is taken by a higher layer.

### 5.4.3 Information Transfer

#### 5.4.3.1 Information Data

Information is conveyed using UI or UIH frames. Support of UIH frames is mandatory and support of UI frames is optional. UI frames are used when it is important to know that data received is correct. An example of the use of UI frames is in carrying IP (Internet Protocol) traffic where error recovery procedures are performed, if necessary, by a higher layer. The use of UIH frames is appropriate if the link is not subject to errors. UI or UIH frames may also be used for data in situations where the delays inherent in error-recovery procedures are unacceptable, such as transmission of voice data.

The transmitter takes information from the convergence layer for the particular DLC and places it in the I-field of the transmitted frame. Once a UI or UIH frame has been correctly received, the contents of its I-field is passed to the convergence layer.

The frames sent by the initiating station have the C/R bit set to 1 and those sent by the responding station have the C/R bit set to 0. Both stations set the P-bit to 0.

See clause 5.2.1.2 for more details about C/R bit.

The maximum length of the information field in UI or UIH frames shall be parameter N1 (see clause 5.7.2).

### 5.4.3.2 Priority

Each data stream has a priority associated with it. The priority is a number in the range 0-63 with lower numbers having higher priority. The TE assigns a priority to each DLC and informs the UE of the priority by means of the multiplexer control channel (see clause 5.4.6.3.1). In the absence of a message assigning priorities DLCs shall be given the priority according to the DLCI Assignment table in clause 5.6. The transmitter is in control of which frames are transmitted and how they are constructed and it is not intended to specify precisely how this task is performed. If data of higher priority than that currently being transmitted is waiting the transmitter has several options available:

- a) complete the transmission of the current frame; or
- b) terminate the assembly of the current frame by sending the current FCS and terminating flag (only for Advanced option);

and start sending the frame of higher-priority data.

Handling of DLC with equal priorities should not favour one over the others. The DLC with the highest priority shall not block any of the lower priorities DLC. Interleaving of higher priority and lower priority frames is necessary in order to avoid permanent blocking of lower priority channels. Optimization of this interleaving for a specific implementation may have a significant impact on the application in use and therefore implementers are encouraged to consider this carefully.

## 5.4.4 Frame Variables

The poll (P) bit set to 1 shall be used by a station to solicit (poll) a response or sequence of responses from the other station.

The final (F) bit set to 1 shall be used by a station to indicate the response frame transmitted as the result of a soliciting (poll) command.

The poll/final (P/F) bit shall serve a function in both command frames and response frames. (In command frames, the P/F bit is referred to as the P bit. In response frames, it is referred to as the F bit).

### 5.4.4.1 Functions of the poll bit

The P bit set to 1 shall be used to solicit a response frame with the F bit set to 1 from the other station at the earliest opportunity.

On a particular DLCI, only one frame with a P bit set to 1 shall be outstanding in a given direction at a given time.

In the case where a SABM or DISC command with the P bit set to 0 is received then the received frame shall be discarded.

If a unsolicited DM response is received then the frame shall be processed irrespective of the P/F setting.

Before a station issues another frame with the P bit set to 1, it shall have received a response frame from the other station with the F bit set to 1. If no valid response frame is obtained within a system-defined time-out period, the retransmission of a command with the P bit set to 1 for error recovery purposes shall be permitted.



### 5.4.4.2 Functions of the final bit

A response frame with the F bit set to 1 shall be used by a station to acknowledge the receipt of a command frame with the P bit set to 1. The response shall be made at the earliest opportunity.

The station may transmit response frames with the F bit set to 0 at any opportunity on an asynchronous basis. However, in the case where a UA response is received with the F bit set to 0 then the received frame shall be discarded.

If an unsolicited DM response is received then the frame shall be processed irrespective of the P/F setting.

If a station receives a command with the P bit set to 1, transmission of a response with the F bit set to 1 shall take precedence over transmission of other commands, with the exception of the mode setting commands (SABM or DISC).

### 5.4.5 Time-out considerations

In order to detect a no-reply or lost-reply condition, each station shall provide a response time-out function (T1). The expiry of the time-out function shall be used to initiate appropriate error recovery procedures.

The duration of the time-out function in the two stations shall be unequal in order to resolve contention situations.

The time-out function shall be started whenever a station has transmitted a frame for which a reply is required. When the expected reply is received, the time-out function shall be stopped. If, during the interval that the time-out function is running, other frames are sent for which acknowledgements are required, the time-out function may have to be restarted.

If the response time-out function runs out, a command with the P bit set to 1 may be (re)transmitted, and the response time-out function restarted.

### 5.4.6 Multiplexer Control Channel

At the initiation of communication between the TE and UE a control channel is set up with DLCI 0 using the procedures of clause 5.8.1. This channel is used to convey information between the two multiplexers. The control channel may use either error recovery mode or non-error recovery mode procedures as defined by the +CMUX command (3GPP TS 27.007 [4]). If error recovery mode procedures are available they should be used.

#### 5.4.6.1 Message format

All messages sent between the multiplexers conform to the following type, length, value format:

Type	Length	Value 1	Value2	...	Value n
------	--------	---------	--------	-----	---------

Each box in the diagram represents a field of minimum size one octet. The type and length octets have extension bits so those fields may contain more than one octet.

The first type field octet has the following format:

Bit	1	2	3	4	5	6	7	8
	EA	C/R	T1	T2	T3	T4	T5	T6

Subsequent type field octets have the following format:

Bit	1	2	3	4	5	6	7	8
	EA	T7	T8	T9	T10	T11	T12	T13

The EA bit is an extension bit. The EA bit is set to 1 in the last octet of the sequence; in other octets EA is set to 0. If only one octet is transmitted EA is set to 1.

The C/R bit indicates whether the message is a command or a response.

The T bits indicate the type coding. Each command has a unique pattern of bits. This means that a single-octet type field can encode 63 different message types. Only single octet message types are defined in the present document.

The first length field octet has the following structure:

Bit	1	2	3	4	5	6	7	8
	EA	L1	L2	L3	L4	L5	L6	L7

Subsequent length field octets have the same structure.

The EA bit is an extension bit. The EA bit is set to 1 in the last octet of the sequence; in other octets EA is set to 0. If only one octet is transmitted EA is set to 1.

The L bits define the number of value octets that follow. In the case of a single-octet length field L1 is the LSB and L7 is the MSB; this permits messages with up to 127 value octets to be constructed.

The contents of the value octets are defined for each message type in clause 5.4.6.3.

Multiple messages may be included in the same frame (as long as the maximum frame size is not exceeded). Messages may not be split over more than one frame.

## 5.4.6.2 Operating procedures

Messages always exist in pairs; a command message and a corresponding response message. If the C/R bit is set to 1 the message is a command, if it is set to 0 the message is a response. A response message has the same T bits as the command that provoked it.

If a command does not produce a response within a time T2 the command may be sent again up to N2 times. If no response is received on the N2 transmissions, the multiplexer control channel should be considered faulty and an alarm raised. Resolution of the error situation is implementation dependent.

**NOTE:** Notice that when UIH frames are used to convey information on DLCI 0 there are at least two different fields that contain a C/R bit, and the bits are set of different form. The C/R bit in the Type field shall be set as it is stated above, while the C/R bit in the Address field (see clause 5.2.1.2) shall be set as it is described in clause 5.4.3.1.

## 5.4.6.3 Message Type and Actions

### 5.4.6.3.1 DLC parameter negotiation (PN)

This procedure is optional. If this command is not supported, default values are applied to each DLC.

Before a DLC is set up using the mechanism in clause 5.4.1, the TE and UE must agree on the parameters to be used for that DLC. These parameters are determined by parameter negotiation.

The parameter negotiation uses the following type field octet:

Bit	1	2	3	4	5	6	7	8
	EA	C/R	0	0	0	0	0	1

The length field octet contains the value 8 and there follow eight value octets. The value octets contain the information in table 3.

**Table 3: Parameter Negotiation**

Value Octet	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
1	D1	D2	D3	D4	D5	D6	0	0
2	I1	I2	I3	I4	CL1	CL2	CL3	CL4
3	P1	P2	P3	P4	P5	P6	0	0
4	T1	T2	T3	T4	T5	T6	T7	T8
5	N1	N2	N3	N4	N5	N6	N7	N8
6	N9	N10	N11	N12	N13	N14	N15	N16
7	NA1	NA2	NA3	NA4	NA5	NA6	NA7	NA8
8	K1	K2	K3	0	0	0	0	0

The various fields are coded as follows:

- the D-bits define the DLCI that the other information refers to; Bit D1 is the least significant;
- the I-bits define the type of frames used for carrying information in the particular DLC - See table 4.

**Table 4: Meaning of I-bits**

Meaning	I1	I2	I3	I4
Use UIH frames	0	0	0	0
Use UI frames	1	0	0	0
Use I frames (note)	0	1	0	0
NOTE: Refer to clause 6.				

Other values are reserved. Default value is 0000.

In the absence of negotiation the frame type used (for DLCI>0) is the same as used by the multiplexer control channel.

The CL-bits define the type of convergence layer to be used on the particular DLCI - see table 5.

**Table 5: Meaning of CL-bits**

Meaning	CL1	CL2	CL3	CL4
Type 1	0	0	0	0
Type 2	1	0	0	0
Type 3	0	1	0	0
Type 4	1	1	0	0

Other values are reserved. Default value is 0000.

The P-bits define the priority to be assigned to the particular DLC. The range of values is 0 to 63 with 0 being the lowest priority. P1 is the least significant bit. Default value for P-bits are given by the DLCI values. See clause 5.6.

The T-bits define the value of the acknowledgement timer (T1) - see clause 5.7.1. The units are hundredths of a second and T1 is the least significant bit.

The N-bits define the maximum frame size (N1) - see clause 5.7.2. The parameter is a sixteen-bit number with N1 as the least significant bit.

The NA-bits define the maximum number of retransmissions (N2) - see clause 5.7.3. The parameter is an eight-bit number with NA1 as the least significant bit.

The K-bits define the window size for error recovery mode (k) - see clause 5.7.4. The parameter is a three-bit number with K1 as the least significant bit.

The TE transmits a parameter negotiation command to the UE with the fields set to the values that the TE intends to use for the particular DLCI. The UE replies with a parameter negotiation response with its proposed set of values. The following rules must be observed by the UE in constructing its response:

- The DLCI value may not be changed.
- The use of I frames or UI frames is optional so an UE may respond with a UIH choice if it does not implement UI frames or I frames.
- The UE may not change the convergence layer proposed by the TE.
- The UE may not change the priority proposed by the TE.
- The T1 value is the one that the TE will use and is not negotiable; the UE will insert its own T1 value. It is advisable that different T1s are used in each direction.
- The UE may propose a smaller value for maximum frame size (N1) if it has insufficient memory to handle the size proposed.
- The N2 value is the one that the TE will use and is not negotiable; the UE will insert its own N2 value.
- The UE may propose a smaller value for window size (k) if it has insufficient memory to handle the size proposed.

If the TE considers the response from the UE to be acceptable the TE will start to establish the DLC in accordance with the procedures of clause 5.3.1. If the response is not acceptable the TE may initiate another parameter negotiation command with revised parameters or pass the failure information to a higher layer.

If an incoming call arrives at the UE from the network for which no DLC has been set up, the UE may initiate the parameter negotiation procedure and set up a DLC. This situation should not occur in practice since a TE will generally set up DLCs for all the functions it shares with the UE after the capabilities exchange. The indication of an incoming call will be through an 07.07 or 07.05 result code.

#### 5.4.6.3.2 Power Saving Control (PSC)

(see also clause 5.4.7).

The power saving control messages use the following type field octet:

Bit	1	2	3	4	5	6	7	8
	EA	C/R	0	0	0	0	1	0

The length byte contains the value 0 and there are no value octets.

If a station wishes to enter a low-power state it transmits a power saving control command; the other station replies with a power saving control response.

If a station wishes to request that the other station enter a low-power state it sends a power saving control command; the other station replies with a power saving control response. The responding station may enter a low-power state but need not do so.

#### 5.4.6.3.3 Multiplexer close down (CLD)

(see also clause 5.8.2).

The multiplexer close down command is used to reset the link into normal AT command mode without multiplexing.

The multiplexer close down messages use the following type field octet:

Bit	1	2	3	4	5	6	7	8
	EA	C/R	0	0	0	0	1	1

The length byte contains the value 0 and there are no value octets.

#### 5.4.6.3.4 Test Command (Test)

The test command is used to test the connection between UE and the TE. The length byte describes the number of values bytes, which are used as a verification pattern. The opposite entity shall respond with exactly the same value bytes.

The type field octet has the following format:

Bit	1	2	3	4	5	6	7	8
	EA	C/R	0	0	0	1	0	0

#### 5.4.6.3.5 Flow Control On Command (FCon)

The flow control command is used to handle the aggregate flow. When either entity is able to receive new information it transmits this command.

The length byte contains the value 0 and there are no value octets.

The type field octet has the following format:

Bit	1	2	3	4	5	6	7	8
	EA	C/R	0	0	0	1	0	1

#### 5.4.6.3.6 Flow Control Off Command (FCoff)

The flow control command is used to handle the aggregate flow. When either entity is not able to receive information it transmits the FCoff command. The opposite entity is not allowed to transmit frames except on the control channel (DLC=0).

The length byte contains the value 0 and there are no value octets.

The type field octet has the following format:

Bit	1	2	3	4	5	6	7	8
	EA	C/R	0	0	0	1	1	0

#### 5.4.6.3.7 Modem Status Command (MSC)

It is desired to convey virtual V.24 control signals to a data stream, this is done by sending the MSC command. The MSC command has one mandatory control signal byte and an optional break signal byte. This command is only relevant when the basic option is chosen.

This command shall be sent prior to any user data after a creation of a DLC.

Command	Length	DLCI	V.24 signals	Break Signals (optional)
---------	--------	------	--------------	--------------------------

The length byte contains the value 2 or 3 and there are 2 or 3 value octets.

Both the DTE and DCE uses this command to notify each other of the status of their own V.24 control signals. The length of the Modem Status Command is either 4 or 5 bytes depending on the break signal.

The command field octet has the following format:

Bit	1	2	3	4	5	6	7	8
	EA	C/R	0	0	0	1	1	1

The C/R bit is used to indicate if it is a Modem Status Command or Modem Status Response.

Every time the signals change, the DTE or DCE sends this command to indicate the current status of each signal. When a DTE or DCE receives a Modem Command it always sends a Response back. The mappings of the V.24 signals to the bits in the control signal octet for the receiver and sender are given in tables 6 and 7, respectively.

In a Modem Status Command it is the status of the sender's own V.24 signals that shall be sent, but in a Response it is copy of the V.24 signals that are received from the Command frame that shall be returned.

The DLCI field identifies the specific DLC to which the command applies. Bit 2 is always set to 1 and the EA bit is set according to the description in clause 5.2.1.2.

Bit No.	1	2	3	4	5	6	7	8
	EA	1	D L		C		I	

**Figure 9: Format of Address Field**

The DLCI field is followed by the control signals field which contains a representation of the state of the signals in accordance with figure 10. The use of the extension bit allows other octets to be added to cater for other circumstances. At present, an optional second octet is defined for handling the transmission of break signals.

Bit No	1	2	3	4	5	6	7	8
Signal	EA	FC	RTC	RTR	reserved 0	reserved 0	IC	DV

**Figure 10: Format of control signal octet**

Description of the control signal byte:

Bit 1. The EA bit is set to 1 in the last octet of the sequence; in other octets EA is set to 0. If only one octet is transmitted EA is set to 1.

Bit 2. Flow Control (FC). The bit is set to 1 (one) when the device is unable to accept frames.

Bit 3. Ready To Communicate (RTC). The bit is set to 1 when the device is ready to communicate.

Bit 4. Ready To Receive (RTR). The bit is set to 1 when the device is ready to receive data.

Bit 5. Reserved for future use. Set to zero by the sender, ignored by the receiver.

Bit 6. Reserved for future use. Set to zero by the sender, ignored by the receiver.

Bit 7. Incoming call indicator (IC). The bit is set to 1 to indicate an incoming call.

Bit 8. Data Valid (DV). The bit is set to 1 to indicate that valid data is being sent.

The control byte is mapped to V.24 signals according to the tables below:

**Table 6: Mapping from the control signal octet by a receiving entity**

Control Signal Byte bit number, name	DTE receiving		DCE receiving	
	signal	V.24 circuit	signal	V.24 circuit
3, RTC	DSR	107	DTR	108/2
4, RTR	CTS	106	RFR (note)	133
7, IC	RI	125	-ignored	-
8, DV	DCD	109	-ignored	-

NOTE: Circuit 133, RFR (Ready for Receiving) is commonly assigned to the connector pin that is alternatively used for circuit 105, RTS. It is sometimes referred to by that name.

**Table 7: Mapping to the control signal octet by a sending entity**

Control Signal Byte bit number, name	DTE sending		DCE sending	
	signal	V.24 circuit	signal	V.24 circuit
3, RTC	DTR	108/2	DSR	107
4, RTR	RFR (note)	133	CTS	106
7, IC	always 0-	-	RI	125
8, DV	always 1-	-	DCD	109
NOTE	Circuit 133, RFR (Ready for Receiving) is commonly assigned to the connector pin that is alternatively used for circuit 105, RTS. It is sometimes referred to by that name.			

If a station is unable to transmit frames because of flow control but wishes to stop accepting further frames itself, it may still send frames containing no user data (i.e. Only the control signal octet and, optionally, the break signal octet) in order to signal flow control.

The EA bit is set to 1 in the last octet of the sequence; in other octets EA is set to 0. If only one octet is transmitted EA is set to 1.

Bit No	1	2	3	4	5	6	7	8
Signal	EA	B1	B2	B3	L1	L2	L3	L4

**Figure 11: Format of Break signal octet (Optional)**

The break signal octet carries information about a break signal detected in the data stream for the DLC. The meanings of the bits are shown in table 8.

**Table 8: Break signal octet meanings**

Bit	Value	Meaning
B1	1	Octet encodes a break signal
	0	Octet does not encode a break signal
B2	0	reserved: set to 0 by sender, ignored by receiver
B3	0	reserved: set to 0 by sender, ignored by receiver
L1-L4	4-bit value	Length of break in units of 200ms

L1 is the least significant bit and L4 is the most significant bit of the break length.

When a station receives a break octet it shall process the information and pass it on in an appropriate way. This is outside the scope of the present document.

#### 5.4.6.3.8 Non Supported Command Response (NSC)

This response is sent whenever a command type is not supported by the receiving entity.

The length byte contains the value 1 and there is one value octets.

The type field octet has the following format:

Bit	1	2	3	4	5	6	7	8
	EA	C/R	0	0	1	0	0	0

The value octet contains the Command Type of the non supported command.

The value octet has the following format:

Bit	1	2	3	4	5	6	7	8
	EA	C/R	Command Type					

The C/R bit (in the value octet above) shall be set to the same value as the C/R bit in the type field octet of the not supported command frame.

#### 5.4.6.3.9 Remote Port Negotiation Command (RPN)

This command is optional.

This command is used for set the remote port communication settings.

All devices must assure that the communication settings are correctly set, prior sending data. There are default values assigned on all parameters, if no negotiation is performed, the default value is chosen.

During a connection, a device must send the RPN whenever the communication settings are changed. The same applies for the Port Line Status.

Command RPN	Length 1 or 8	Value octet1 optional (DLCI)	Value octet2 optional	Value octet3 optional	Value octet4 optional	Value octet5 optional	Value octet6 optional	Value octet7 optional	Value octet8 optional
-------------	---------------	------------------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------

The Remote Port Negotiation Command use the following type field octet:

**Table 9: Type field octet**

Bit	1	2	3	4	5	6	7	8
EA	C/R	0	0	1	0	0	1	

The length byte contains the value 1 or 8 and there are one or eight value octets.

**Table 10: DLCI octet**

Bit No.	1	2	3	4	5	6	7	8
EA	1		D	L	C	I		

Bit 2 in the DLCI octet is not used and always set to 1, the EA bit is according to the description in clause 5.2.1.2. The DLCI field indicated which DLC the command is applied to.

**Table 11: Port Value Octets**

Value Octet	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
2	B1	B2	B3	B4	B5	B6	B7	B8
3	D1	D2	S	P	PT1	PT2	res	res
4	FLC1	FLC2	FLC3	FLC4	FLC5	FLC6	res	res
5	XON1	XON2	XON3	XON4	XON5	XON6	XON7	XON8
6	XOF1	XOF2	XOF3	XOF4	XOF5	XOF6	XOF7	XOF8
7	PM1	PM2	PM3	PM4	PM5	PM6	PM7	PM8
8	PM9	PM10	PM11	PM12	PM13	PM14	PM15	PM16

A device transmits a remote port negotiation command to the other device with the fields set to the desired values with the parameter mask indicating which parameters are set.

When the remote port negotiation command is received, the responding station replies according to the following rules:

The DLCI value may not be changed.

The receiver may accept the Port Value Octet bits proposed by the sender, and reply with a responds with the parameter mask set to 1 for all the parameters accepted. If the receiver does not support any of the proposed values, it replies with the parameter mask set to zero for the parameters not supported. For those parameters with the parameters mask set to 1, the new value is accepted and used.



If the receiver does not support any of the proposed values indicated by the parameter mask, the receiver replies with the Remote Parameter Negotiation response with the parameter mask set to zero.

If only one value byte is included in the command, it is interpreted as a request, and the receiver shall respond with the current Port Values setting.

If the sender considers the response to be acceptable, that is, the bits match, the sender will start to use the DLC according to the Port Value Octets. If the response is not acceptable the sender may initiate another remote port negotiation command with revised parameters until a final agreement is reached or pass the failure information to a higher layer.

The B1-B8 indicates the baudrate, see table 12.

**Table 12: Meaning of B-bits**

Meaning	B1	B2	B3	B4	B5	B6	B7	B8
2 400 bit/s	0	0	0	0	0	0	0	0
4 800 bit/s	1	0	0	0	0	0	0	0
7 200 bit/s	0	1	0	0	0	0	0	0
9 600 bit/s	1	1	0	0	0	0	0	0
19 200 bit/s	0	0	1	0	0	0	0	0
38 400 bit/s	1	0	1	0	0	0	0	0
57 600 bit/s	0	1	1	0	0	0	0	0
115 200 bit/s	1	1	1	0	0	0	0	0
230 400 bit/s	0	0	0	1	0	0	0	0

All other values of the B-bits are reserved.

The default value is 1100 0000 (9600).

The D1-D2 indicates the number of data bits:

D1, D2

00 5 bits

01 6 bits

10 7 bits

11 8 bits - default

The S bit indicate number of stop bits: S=0: 1 stop bit, S=1: 1,5 stop bits. Default value = 0 (1 stop bit).

The P bit indicate the parity. P=0: no parity, P=1: parity. Default value = 0 (no parity).

The PT1 - PT2 indicates the parity type:

PT1,PT2

00 odd parity

01 even parity

10 mark parity

11 space parity

FLC1-FLC6: (Default value=0, no flow control)

Bit1 XON/XOFF on input

Bit2 XON/XOFF on output

Bit3 RTR on input

Bit4 RTR on output

Bit5 RTC on input

Bit6 RTC on output

NOTE: The RTR is mapped to either CTS (circuit 106) or RFR (circuit 133). The RTC is mapped to either DTR (circuit 108/2) or DSR (circuit 107). (Circuit 133, RFR(Ready for Receiving) is commonly assigned to the connector pin that is alternatively used for circuit 105, RTS. It is sometimes referred to by that name).

XON1-XON8: XON character (default DC1)

XOF1-XOF8: XOFF character.(default DC3)

PM1-PM8: Parameter mask

The parameter mask is used to indicate which parameters in the Remote Port Negotiation command are negotiated. For a command, the parameter mask shall be interpreted as follows:

0=no change

1= change.

For a response the following values applies:

0=not accepted proposal

1= accepted proposal, and the new values are used.

The bit mask for the value octets 7 and 8 are shown below:

Bit1 bit rate

Bit2 data bits

Bit3 stop bits

Bit4 Parity

Bit5 parity type

Bit6 XON character

Bit7 XOF character

Bit8 reserved

PM9-PM16: Parameter mask continued

Bit1 XON/XOFF on input

Bit2 XON/XOFF on output

Bit3 RTR on input

Bit4 RTR on output

Bit5 RTC on input

Bit6 RTC on output

All reserved values are set to 0 (zero) by the sender and ignored by the receiver.

#### 5.4.6.3.10 Remote Line Status Command(RLS)

This command is optional.

This command is used for indicate remote port line status.

During a connection, a device must send the RLS whenever the Remote Port Line Status are changed.

The Remote Line Status command use the following type field octet.

**Table 13: Type field octet**

Bit	1	2	3	4	5	6	7	8
	EA	C/R	0	0	1	0	1	0

The length byte contains the value 2 and there are two value octets.

**Table 14: DLCI octet**

Bit No.	1	2	3	4	5	6	7	8
	EA	C/R	D		L	C	I	

The C/R bit in the DLCI octet is not used and always set to 1, the EA bit is according to the description in clause 5.2.1.2.

**Table 15: Remote Line Status Octets**

Value Octet	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
1	L1	L2	L3	L4	res	res	res	res

A device transmits a remote line status command to the other device with the fields set to the desired value. When the remote line status command is received, the remote device must respond with a Remote Line Status Response containing the values that it received.

The L1-L4 bits indicates the Line Status. If L1 is set to 0, no error have occurred. L1 = 1 indicates the following errors:

L2-L4:

- 100      Overrun Error - Received character overwrote an unread character
- 010      Parity Error - Received character's parity was incorrect
- 001      Framing Error - a character did not terminate with a stop bit

**The res bits are set to zero for the sender and ignored by the reciever.**

#### 5.4.6.3.11 Service Negotiation Command (SNC)

This command is used to query and set a specific service on a specific DLC. It is for instance used to set specific digital voice types.

In some situations it is not very suitable to mix AT commands and raw data on the same DLC. For those situations, special DLCs can be established and converted to carry a specific data type. Examples of situation where this is especially useful is for voice transportation, where the AT commands controlling the connection (for instance for multiparty) are transported on one DLC and voice data carried by another DLC. This mechanism can be seen as an alternative to sending escape sequences with AT commands in the data flow. If this command is not used, the DLC is by default set to normal AT command mode. If this command is used, the DLC indicated in the DLCI octet, is converted to carry the specific data type. The originator of this command may also query the specific service on each DLCI.

The service negotiation messages use the following format.

**Table 16: Service Negotiation Command format**

Byte No.	1	2	3	4	5
	Type field code	Length	DLCI	Service Value octet (optional)	Voice Codec Value octet (optional)

The type field octet:

**Table 17: Type field octet**

Bit	1	2	3	4	5	6	7	8
	EA	C/R	0	0	1	0	1	1

The length byte contains the value 1 or 3 and there are one or three value octets.

**Table 18: DLCI octet**

Bit No.	1	2	3	4	5	6	7	8
	EA	C/R	D		L	C	I	

The C/R bit in the DLCI octet is always set to 1 and the EA bit is according to the description in clause 5.2.1.2.

**Table 19: Service Value Octets**

Value Octet	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
1	EA	S1	S2	S3	S4	S5	S6	S7

The EA bit is according to the description in clause 5.2.1.2.

**Table 20: Voice Codec Value Octets**

Value Octet	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
1	EA	V1	V2	V3	V4	V5	V6	V7

The EA bit is according to the description in clause 5.2.1.2.

**Table 21: Service Value Bits**

Value Bit	Service
S1	Data
S2	Voice
S3	Reserved
S4	Reserved
S5	Reserved
S6	Reserved
S7	Reserved

Table 22: Voice Codec Value Bits

Value Bit	Service
V1	Voice (coded – 3GPP TS 46.021)
V2	Voice (coded - PCM 64 kbit/s U-law)
V3	Voice (coded ADPCM 32kbit/s) ITU-T G.726
V4	Voice (coded halfrate)
V5	Voice (coded - PCM 64kbit/s A-law)
V6	Voice (coded PCM 128kbit/s)
V7	Reserved

The sender transmits a service negotiation command with the fields set to the values for all possible services it may use for the particular DLC. The receiver replies with a service negotiation response with its selected set of values. The following rules must be observed by the receiver in constructing its response:

- the DLCI value may not be changed;
- the receiver may select a subset of the services proposed by the sender, but not a superset. It is the receiver's selection that is the valid one. If the receiver does not support any of the proposed services, it replies with the service byte set to zero;
- The Voice Codec Value Octet is always present even though data service is chosen. In this case, the Voice Codec Value Octet V1-V7 bits are set to zero.

A zero value means standard AT command mode.

If no value bytes are included in the command, it is interpreted as a request, and the receiver shall respond with all possible services.

If the sender considers the response to be acceptable, that is, the service bits match, the sender will start to use the DLC according to the services. If the response is not acceptable the sender may initiate another parameter negotiation command with revised parameters until a final agreement is reached or pass the failure information to a higher layer.

If no service negotiation has been performed on the DLCs, it operates in standard AT mode. In this case, an incoming call is indicated on that DLC.

The sender shall set a reserved bit to value 0. The receiver shall ignore a reserved bit.

## 5.4.7 Power Control and Wake-up Mechanisms

It is very important in many types of UE and some TE that the power consumed by the equipment is minimised. This aim is often achieved by entering various power-saving states under conditions of inactivity, for example. The multiplexer system must be able to close down cleanly if either TE or UE wish to enter a low-power state. This is achieved by the use of the multiplexer control channel.

If either TE or UE wishes to enter a low-power state a power saving control command message is sent to the other station on the multiplexer control channel. The station receiving the message will complete the transmission of any frames in progress, report a busy or power-down condition to higher layers, freeze all timers and transmit the power saving control response message. When the station that initiated the power saving control message receives the acknowledgement it is then free to enter the reduced-power state.

Either station may send a power saving control command requesting that the other station enters a low power state. The responding station must acknowledge this command with a power saving control response message but need not obey the command to enter a low-power state. If no response is received the commanding station may repeat the command but must first use the wake-up procedure.

Either station may initiate a wake-up from the reduced power state by the transmission of the wake-up signal which consists of continuous flag characters. When the other station receives the flag characters it will wake-up (if necessary) and start sending flag characters. When the first station receives these flag characters it will stop sending flags and start transmitting the first frame. When the second station detects a valid frame it stops sending flags. The stations unfreeze their timers and continue operation as before.

If no response (continuous flags) is received to the wake-up procedure within time T3, an alarm should be raised to the higher layers and transmission of flags stops.

## 5.4.8 Flow Control

### 5.4.8.1 RTR Flow Control

Figure 12 shows a DTE connection to a DCE. The flow control scheme defined in this clause also applies to DTE - DTE connections. Both 27.010 entities are configured to use RTR (RFR/CTS) flow control. The flow control signal to the local application is a combination of the RTR signal from the opposite device together with three other flow control signals. The flow control signals labelled FCS1 - FCS3 are defined below:

FCS1	Bit 2 in the Modem Status Command or in the control signal octet in convergence layer type 2. Flow control per DLCI
FCS2	Aggregate flow control in 27.010 via the control channel commands Fcon and Fcoff (basic option) or XON/XOFF in the advance option.
FCS3	27.010 internal buffer management (implementation specific)

The flow control signals FCS1-3 are combined with the RTR signal from the opposite 27.010 instance to create the local RTR input signal. E.g. the expression for the CTS signal for the emulated DTE serial port is:

$$DTE.CTS = DCE.RTR \text{ AND } FCS1 \text{ AND } FCS2 \text{ AND } FCS3$$

The flow control emulator duplicates the outgoing RTR signal in bit 2 (FC) and bit 4 (RTR) in the modem status command (when convergence layer 1,3 and 4 is used) or in bit 2 (FC) and bit 4 (RTR) in the control signal octet (when the convergence layer 2 is used).

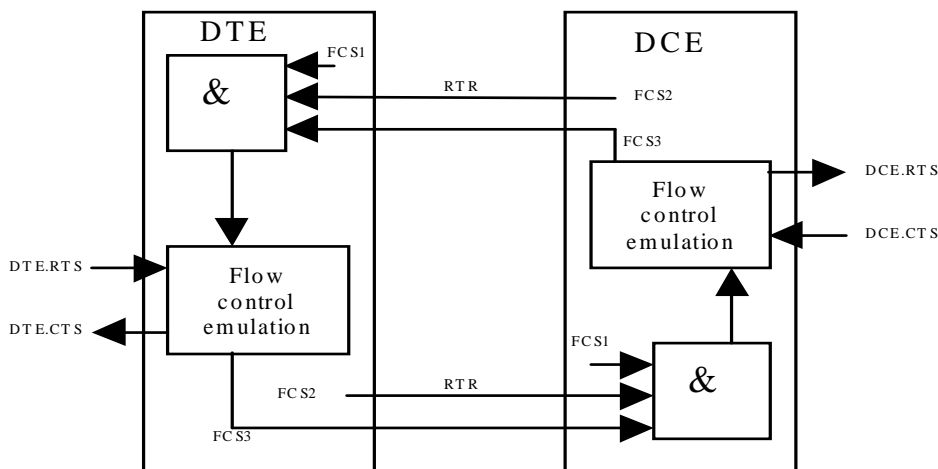


Figure 12: RTR Flow Control

### 5.4.8.2 XON/XOFF Flow Control

Some 27.010 instance may detect XON/OFF characters coming from the local application when XON/XOFF is enabled. In this case the characters are acted upon, but not forwarded to the opposite 27.010 instance i.e. the XON/XOFF characters are filtered out and the flow control signal is transferred as a 27.010 flow signal, see figure 13.

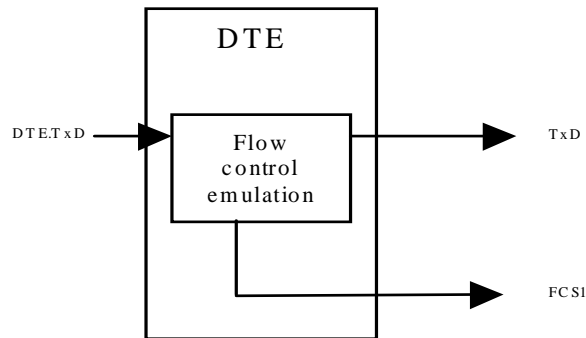


Figure 13: XON/XOFF Flow Control on input

if XON/XOFF flow control is enabled on output then the 27.010 should use XON/XOFF characters to exercise flow control to the local application i.e XON/XOFF characters are inserted into the data stream depending on the 27.010 flow control signals, see figure 14.

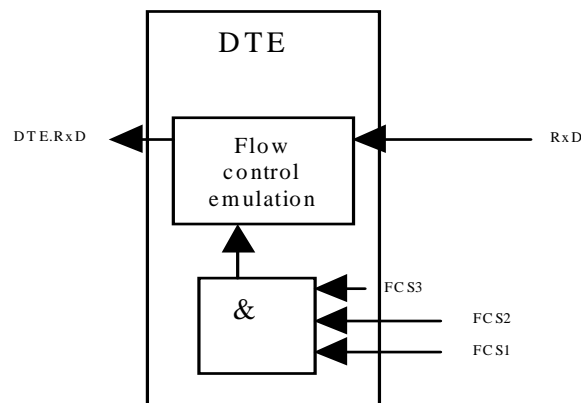


Figure 14: XON/XOFF Flow Control on input

## 5.5 Convergence Layers

Convergence layers are defined to permit data which has implied structure to be conveyed through the multiplexer without losing the structure or other parameters which are associated with the data stream. Common uses of convergence layers are to carry the state of V.24 control signals through a DLC or to ensure that the boundaries of a coded voice frame are preserved.

Convergence layers apply to data whether it is carried by error recovery mode or non-error recovery mode procedures.

The use of particular convergence layers is implicitly linked to the DLCIs used but may be negotiated away from these defaults by the use of the multiplexer control channel.

### 5.5.1 Type 1 - Unstructured Octet Stream

Data which consists of an unstructured sequence of octets, for example, 64 kbit/s uncoded voice or normal asynchronous data without V.24 control signals, is inserted directly into the I-field. In this case, it could be said that the convergence layer is null.

Type 1 is the default convergence layer for each DLC.

### 5.5.2 Type 2 - Unstructured Octet Stream with flow control, break signal handling and transmission of V.24 signal states

If it is desired to convey virtual V.24 control signals associated with a data stream the first octet of each I-field contains a representation of the state of the signals in accordance with figure 15. The use of the extension bit allows other octets

to be added to cater for other circumstances. At present, an optional second octet is defined for handling the transmission of break signals.

The mappings of the V.24 signals to the bits in the control signal octet for the receiver and sender are given in tables 23 and 24, respectively.

Bit No	1	2	3	4	5	6	7	8
Signal	EA	FC	RTC	RTR	reserved 0	reserved 0	IC	DV

**Figure 15: Format of control signal octet**

Description of the control signal byte:

Bit 1. The EA bit is set to 1 in the last octet of the sequence; in other octets EA is set to 0. If only one octet is transmitted EA is set to 1.

Bit 2. DLC (frame) Flow Control (FC). In non-ERM, it is set to 1 when the device is unable to accept frames. In ERM, it is always set to 0 by the sender and ignored by the receiver.

Bit 3. Ready To Communicate (RTC). The bit is set to 1 when the device is ready to communicate.

Bit 4. Ready To Receive (RTR). The bit is set to 1 when the device is ready to receive data.

Bit 5. Reserved for future use. Set to zero by the sender, ignored by the receiver.

Bit 6. Reserved for future use. Set to zero by the sender, ignored by the receiver.

Bit 7. Incoming call indicator (IC). The bit is set to 1 to indicate an incoming call.

Bit 8. Data valid (DV). The bit is set to 1 to indicate that valid data is being sent.

The control byte is mapped to V.24 signals according to the tables below:

**Table 23: Mapping from the control signal octet by a receiving entity**

Control Signal Byte	DTE receiving		DCE receiving	
	signal	V.24 circuit	signal	V.24 circuit
2, FC	DLC flow control	-	DLC flow control	-
3, RTC	DSR	107	DTR	108/2
4, RTR	CTS	106	RFR (note)	133
7, IC	RI	125	ignored	-
8, DV	DCD	109	ignored	-

NOTE: Circuit 133, RFR (Ready for Receiving) is commonly assigned to the connector pin that is alternatively used for circuit 105, RTS. It is sometimes referred to by that name.

**Table 24: Mapping to the control signal octet by a sending entity**

Control Signal Byte	DTE sending		DCE sending	
	signal	V.24 circuit	signal	V.24 circuit
2, FC	frame flow control	-	frame flow control	-
3, RTC	DTR	108/2	DSR	107
4, RTR	RFR (note)	133	CTS	106
7, IC	always 0	-	RI	125
8, DV	always 1	-	DCD	109

NOTE: Circuit 133, RFR (Ready for Receiving) is commonly assigned to the connector pin that is alternatively used for circuit 105, RTS. It is sometimes referred to by that name.

In non-error recovery mode, the FC bit provides frame flow control for the DLC. If a station is unable to accept further frames, for example, because of buffer management issues, it shall set FC to 1. When the station is again able to receive frames it shall set FC to 0. If a station receives a frame with FC set to 1 it shall cease transmitting frames until it receives a frame with FC set to 0.



If a station is unable to transmit frames because of flow control but wishes to stop accepting further frames itself, or to signal a change in the control signals or a break condition it may still send frames containing no user data (i.e. containing only the control signal octet and, optionally, the break signal octet).

In error recovery mode the FC bit is not used and shall be set to 0 by the sender and ignored by the receiver. If a station has been prevented from sending I frames, for example, by receiving a RNR frame, it may still send a UI or UIH frame containing no user data if it wishes to signal a change in the control signals or a break condition.

The EA bit is set to 1 in the last octet of the sequence; in other octets EA is set to 0. If only one octet is transmitted EA is set to 1.

Bit No	1	2	3	4	5	6	7	8
Signal	EA	B1	B2	B3	L1	L2	L3	L4

**Figure 16: Format of Break signal octet**

The break signal octet carries information about a break signal detected in the data stream for the DLC. The meanings of the bits are shown in table 25.

**Table 25: Break signal octet meanings**

Bit	Value	Meaning
B1	1	Octet encodes a break signal
	0	Octet does not encode a break signal
B2	1	Discard data in buffers
	0	Do not discard data in buffers
B3	1	Transmit break signal onwards as soon as possible
	0	Transmit break signal onwards in sequence
L1-L4	4-bit value	Length of break in units of 200ms

L1 is the least significant bit and L4 is the most significant bit of the break length.

When a station receives a break octet it shall process the information and pass it on in an appropriate way. This is outside the scope of the present document.

The remaining octets of the I-field contain data for that DLC.

### 5.5.3 Type 3 - Uninterruptible Framed Data

An example of uninterruptible framed data is coded voice data made up of a sequence of voice frames. It is important that coded voice frames reach the voice decoder with the frame structure intact and with the shortest possible delay. The simplest way of ensuring this is to map one complete voice frame into one I-field. This frame shall not be shortened during transmission even if data of higher priority is waiting.

At the transmitter each frame of data is inserted into the I field of an I frame, UI frame or UIH frame. The data shall not be spread over more than one frame and data from other frames must not be included in the I field. The receiver handles the data as a complete frame and passes it on as a complete frame.

Coded voice data should be transmitted using UI frames or UIH frames because the delays associated with re-transmissions are usually unacceptable.

### 5.5.4 Type 4 - Interruptible Framed Data

This type of convergence layer is used might be used to convey data which has an implied structure but where the delay is not as important as Type 3. The structured data may be segmented across several frames and re-assembled at the other station. PPP-framed IP data is an example of the type of data that could be carried over a Type 4 convergence layer.

The first octet of every Type 4 frame is a control octet and is defined in figure 17.

Bit No	1	2	3	4	5	6	7	8
Signal	EA	-	-	-	-	-	B	F

**Figure 17: Format of Type 4 octet**

The EA bit is for future expansion if more than one octet is needed. It is set to 1 in the case described here.

The B and F bits are used to indicate whether the frame is the first frame in a sequence, a middle frame or the last frame. The meanings of the bits are as shown in table 26.

**Table 26: Meaning of B and F bits**

B	F	Meaning
1	0	First frame of a sequence
0	0	Middle frame of a sequence
0	1	Last frame of a sequence
1	1	Data is contained entirely within one frame

NOTE 1: PPP-framed IP data can be carried using a Type 1 convergence layer if other framing structures, such as a layer 2 protocol, have already been included in the data stream.

NOTE 2: If a frame is coded as being the last frame or that all the data is contained within one frame, that frame may then not be shortened because shortening would make the meaning of the header incorrect. It may be advisable to construct the last frame of a sequence such that it contains no data and avoid the use of the 11 code if frame shortening is desired.

## 5.6 DLCI Values

All DLCs use a type 1 convergence layer by default; the use of other layers may be negotiated using the multiplexer control channel.

**Table 27: DLCI Assignments**

Usage	DLCI number (decimal)	Priority
Multiplexer control channel	0	0
AT commands (07.07 and 07.05)	1-7	7
AT commands (07.07 and 07.05)	8-15	15
AT commands (07.07 and 07.05)	16-23	23
AT commands (07.07 and 07.05)	24-31	31
AT commands (07.07 and 07.05)	32-39	39
AT commands (07.07 and 07.05)	40-47	47
AT commands (07.07 and 07.05)	48-55	55
AT commands (07.07 and 07.05)	56-61	61
Reserved	62-63	

DLCI value 62 is reserved and must be allocated for ETSI purposes of the BOFC and the EOFC in Basic option.

DLCI value 63 is reserved and must not be allocated for ETSI purposes because of the special meaning in HDLC.

DLCI value 0 is reserved for the control channel.

The priority values in Table 27 apply in the absence of an explicit DLC priority assignment message.

## 5.7 System Parameters

The following system parameters are defined for the multiplexer. T1, N1, N2 and k may be negotiated by use of the multiplexer control channel or the default values given here should be used. T2 and T3 are set with the AT+CMUX command.

### 5.7.1 Acknowledgement Timer (T1)

The acknowledgement timer governs the amount of time that a station will wait for an acknowledgement before resorting to other action (e.g. transmitting a frame). The two stations may operate with different values of T1.

The units are hundredths of a second. Times of up to 2,55 s may be used.

The default value is 100 ms and the minimum is 10 ms.

### 5.7.2 Maximum Frame Size (N1)

N1 defines the maximum number of octets that may be contained in an information field. It does not include octets added for transparency purposes.

The default value is 64 octets when the advanced option activated and 31 octets when it is not activated. The range is 1 to 32 768 octets.

NOTE: The maximum frame size should be chosen carefully particularly if a Type 2 convergence layer is being used. The frame must be large enough to contain a complete protocol frame.

### 5.7.3 Maximum number of retransmissions (N2)

N2 defines the maximum number of times that a station re-attempt a procedure requiring a response. The two stations may operate with a different value of N2.

The default value is 3 and the range is 0-255.

### 5.7.4 Window Size (k)

The window size parameter (k) defines the maximum number of I frames that a DLC can have outstanding (i.e. unacknowledged). Identical values need not be used for each direction. The window size may not be larger than 7.

This parameter is only applicable when Error Recovery Option is activated. See clause 6.

The default value is 2 and the range is 1-7.

### 5.7.5 Response Timer for multiplexer control channel (T2)

The T2 timer is the amount of time the multiplexer control channel waits before re-transmitting a command.

T2 must be greater than T1. The units are hundredths of a second. Times of up to 2,55 s may be used.

The default value is 300 ms and minimum 20 ms.

### 5.7.6 Response Timer for wake-up procedure (T3)

The T3 timer is the amount of time the transmitting station of a power wake-up command waits before raising an alarm when no response is received. The units are seconds. Times of up to 255 seconds may be used.

The default value is 10 s and minimum 1 s.

## 5.8 Start-up and close-down of multiplexer

### 5.8.1 Start-up procedure

Multiplexer operation is started by the use of the +CMUX command (see 3GPP TS 27.007 [4]). This command instructs the multiplexer to start up the multiplexer control channel (see clause 5.8.1) using either error recovery mode or non-error recovery mode. The TE multiplexer initiates the establishment of the multiplexer control channel by sending a SABM frame on DLCI 0 using the procedures of clause 5.4.1.

Once the multiplexer channel is established other DLCs may be established using the procedures of clause 5.4.1. The multiplexers may negotiate the parameters associated with each DLC prior to establishment of a DLC or may use the defaults.

## 5.8.2 Close-down procedure

Initiation of the close-down will come from higher layers in either the TE or UE and is outside the scope of the present document. Once the command to close down is received the multiplexer will close down each DLC in turn using the procedures of clause 5.4.2. When all DLCs (except the multiplexer control channel - DLCI 0) are closed down (disconnected mode) the multiplexer that initiated close-down procedure will send a close-down message on the multiplexer control channel. When this message is acknowledged both stations will revert to the non-multiplexed mode. If no response is received to the close-down command within time T2, the initiating station may retransmit it but must close down if no response message is received in time T3.

After closing of the multiplexer protocol, the UE and TE should revert to normal AT mode.

---

# 6 Error Recovery Mode Option

When the Advanced option is selected an error recovery mechanism may be used for better security. The error-recovery mode is optional and is intended for those cases where the quality of the TE-UE link can not be guaranteed and/or when the integrity of the data to be transmitted is critical. Some DLCs may use error recovery mode and some use non-error recovery mode on the same link.

error recovery mode uses I frames to carry the data; the procedures used are defined below.

New frames types and procedures are added.

## 6.1 Frame Types

**Table 28: Coding of Control Field**

Frame Type	1	2	3	4	5	6	7	8
I (Information)	0	N(S)			P/F	N(R)		
RR (Receive Ready)	1	0	0	0	P/F	N(R)		
RNR (Receive Not Ready)	1	0	1	0	P/F	N(R)		
REJ (Reject)	1	0	0	1	P/F	N(R)		

N(R) and N(S) are receive and send sequence numbers, respectively. They are described later in the present document.

### 6.1.1 Information transfer, I, command and response

The function of the information, I, command and response shall be to transfer sequentially numbered frames, each containing an information field, across a data link.

The encoding of the I command/response control field shall be as shown in table 28.

The I frame control field shall contain two sequence numbers:

- N(S), send sequence number, which shall indicate the sequence number associated with the I frame; and
- N(R), receive sequence number, which shall indicate the sequence number (as of the time of transmission) of the next expected I frame to be received, and consequently shall indicate that the I frames numbered up to N(R) - 1 inclusive have been received correctly.

### 6.1.2 Receive ready, RR, command and response

The receive ready, RR, frame shall be used by a station to:

- a) indicate that it is ready to receive an I frame; and
- b) acknowledge previously received I frames numbered up to  $N(R) - 1$  inclusive.

When transmitted, the RR frame shall indicate the clearance of any busy condition that was initiated by the earlier transmission of an RNR frame by the same data station. See clause 6.2.2.5.1.

### 6.1.3 Receive not ready, RNR, command and response

The receive not ready, RNR, frame shall be used by a station to indicate a busy condition; i.e., temporary inability to accept subsequent I frames. I frames numbered up to  $N(R) - 1$  inclusive shall be considered as acknowledged. The I frame numbered  $N(R)$  and any subsequent I frames received, if any, shall not be considered as acknowledged; the acceptance status of these frames shall be indicated in subsequent exchanges.

### 6.1.4 Reject, REJ, command and response

The reject, REJ, frame shall be used by a station to request retransmission of I frames starting with the frame numbered  $N(R)$ . I frames numbered  $N(R) - 1$  and below shall be considered as acknowledged. Additional I frames awaiting initial transmission may be transmitted following the retransmitted I frame(s).

With respect to each direction of transmission on the data link, only one REJ exception condition from a given station to another station shall be established at any given time. A REJ frame shall not be transmitted until an earlier REJ exception condition has been cleared as indicated in clause 6.3.3.5.2.2.

The REJ exception condition shall be cleared (reset) upon the receipt of an I frame with an  $N(S)$  equal to the  $N(R)$  of the REJ frame.

## 6.2 Procedure and State

### 6.2.1 Frame state variables and sequence numbers

#### 6.2.1.1 General

Each station shall maintain an independent send state variable  $V(S)$  for the frames it transmits and an independent receive state variable  $V(R)$  for the I frames it sends to and receives from another station.

#### 6.2.1.2 Send state variable $V(S)$

The send state variable denotes the sequence number of the next in-sequence I frame to be transmitted. The send state variable can take the value 0 to 7 inclusive. The value of the send state variable shall be incremented by one with each successive I frame transmission, but shall not exceed  $N(R)$  of the last received frame by more than 7.

#### 6.2.1.3 Send sequence number $N(S)$

Only I frames shall contain  $N(S)$ , the send sequence number of transmitted frames. Prior to transmission of an I frame,  $N(S)$  shall be set equal to the value of the send state variable.

#### 6.2.1.4 Receive state variable $V(R)$

The receive state variable denotes the sequence number of the next I frame expected to be received. The receive state variable can take the value 0 to 7 inclusive. The value of the receive state variable shall be incremented by one on receipt of an error-free I frame whose send sequence number  $N(S)$  equals the receive state variable.

#### 6.2.1.5 Receive sequence number $N(R)$

All frames which contain  $N(R)$  (see table 28) shall indicate the  $N(S)$  sequence number of the next expected I frame. Prior to transmission of a frame containing  $N(R)$ , the  $N(R)$  shall be set equal to the current value of the receive state

variable. The  $N(R)$  indicates that the station transmitting the  $N(R)$  has correctly received all I frames numbered up to  $N(R) - 1$  inclusive.

### 6.2.1.6 Use of the P/F bit to assist in error recovery

As the P and F bits set to 1 are always exchanged as a pair (for every P bit there shall be one F bit, and another P bit shall not be issued until the previous P bit has been matched with an F bit, and, similarly, another F bit shall not be issued until another P bit is received), the  $N(R)$  contained in a received frame with a P bit or F bit set to 1 can be used to detect that I frame retransmission is required. This capability provides early detection of I frames not received by the remote data station and indicates the frame sequence number where retransmission shall begin. This capability is referred to as checkpointing. The  $N(R)$  of a correctly received frame shall acknowledge previously transmitted I frames to  $N(R) - 1$  inclusive.

The  $N(R)$  of a received I, RR or RNR response frame which has the F bit set to 1 shall cause the receiving station to initiate appropriate error recovery if the  $N(R)$  does not acknowledge at least all I frames transmitted by the receiving station previous to, and concurrent with, the last frame which was transmitted by the receiving station with the P bit set to 1.

## 6.2.2 Exchange of information (I) frames

### 6.2.2.1 Sending I frames

The control field format shall be as defined in clause 6.1 for an I frame, with  $N(S)$  set to the value of the send state variable  $V(S)$  and with  $N(R)$  set to the value of the receive state variable  $V(R)$ . Following data link set-up, both  $V(S)$  and  $V(R)$  shall be set to zero. The maximum length of the information field in I frames shall be parameter  $N1$  (see clause 5.7.2). The default value of  $N1$  is 256 octets; other values may be negotiated by use of the multiplexer control channel.

If a station is ready to send an I frame numbered  $N(S)$ , where  $N(S)$  is equal to the last received acknowledgement plus 7, the station shall not send the I frame but shall follow the procedures described in clause 5.4.5.

The decision whether to send an I frame as a command or as a response, i.e., to use the C/R bit to indicate a P or an F bit, respectively, shall depend on the need to acknowledge a received P bit set to 1 by transmitting a response with the F bit set to 1.

### 6.2.2.2 Receiving I frames

After a station receives correctly an I frame [i.e.,  $N(S)$  equals the value of the receive state variable  $V(R)$ ] that it can accept, the station shall increment its receive state variable  $V(R)$ , and, at its next opportunity to send, take one of the following actions:

- if information is available for transmission and the other station is ready to receive, it shall act as described in clause 6.2.2.1 and acknowledge the received I frame(s) by setting  $N(R)$  in the control field of the next transmitted I frame to the value of  $V(R)$ ; or
- if information is not available for transmission, but the station is ready to receive I frames, the station shall send an RR frame and acknowledge the received I frame(s) by setting  $N(R)$  to the value of  $V(R)$ ; or
- if the station is not ready to receive further I frames, the station may send an RNR frame and acknowledge the received I frame(s) by setting the  $N(R)$  to the value of  $V(R)$ .

If the station is unable to accept the correctly received I frame(s),  $V(R)$  shall not be incremented. The station may send an RNR frame with the  $N(R)$  set to the value of  $V(R)$ .

The I or supervisory frame transmitted will be either a command or a response depending on whether a P bit set to 1 or an F bit set to 1 transmission, respectively, is required. If the transmission of a P bit or F bit set to 1 is not required, the acknowledgement frames may be either commands or responses.

### 6.2.2.3 Reception of incorrect frames

If a frame is received with an incorrect FCS, it shall be discarded.

If an I frame is received with a correct FCS but with an incorrect N(S), the receiving station shall ignore the N(S) field and discard the information field in that frame. This shall continue until the expected I frame is received correctly. The receiving station shall, however, use the P/F and N(R) indications in the discarded I frames. The station shall then acknowledge the expected I frame, when received correctly, as described in clause 5.4.4.2.

The P/F recovery (checkpointing) shall cause the retransmission of the I frames received incorrectly, as described in clause 6.2.1.6.

### 6.2.2.4 Station receiving acknowledgements

A station receiving an I, RR, or RNR frame with a valid N(R) =  $x$  shall treat as acknowledged all previously transmitted I frames up to and including the I frame transmitted with N(S) equal to  $x - 1$ .

### 6.2.2.5 Exception conditions and recovery

The following procedures are available to effect recovery following the detection/occurrence of an exception condition at the data link layer. The exception conditions described are those situations which may occur as the result of transmission errors, data station malfunction or operational situations.

#### 6.2.2.5.1 Busy

The busy condition shall result when a station is temporarily unable to receive, or unable to continue to receive, I frames due to internal constraints; for example, receive buffering limitations. In this case, an RNR frame shall be transmitted with the N(R) number of the next I frame that is expected. Traffic awaiting transmission may be transmitted from the busy data station prior to, or following, the RNR frame. The continued existence of a busy condition shall be reported by retransmission of an RNR frame at each P/F frame exchange.

A data station receiving an RNR frame, when in the process of transmitting, shall stop transmitting I frames at the earliest possible time. The station shall wait for a duration of T<sub>2</sub> before resuming transmission.

Indication that a busy condition has cleared and that I frames will now be acceptable shall be reported by the transmission of an RR, REJ, SABM, or UA frame with or without the P/F bit set to 1. Clearance of a busy condition at a station shall also be indicated by the transmission of an I frame with the F bit set to 1.

#### 6.2.2.5.2 N(S) sequence error

An N(S) sequence error exception condition shall occur in the receiver when an I frame that is received error free (no FCS error) contains an N(S) that is not equal to the receive state variable at the receiver. The receiver shall not acknowledge (i.e., not increment its receive state variable) the frame causing the sequence error or any I frames which may follow until an I frame with the correct N(S) is received.

A station which receives one or more I frames having sequence errors, but which are otherwise error free, shall accept the control information contained in the N(R) field and the P/F bit to perform data link control functions; for example, to receive acknowledgement of previously transmitted I frames, or to cause a station to respond (P bit set to 1). Therefore, the retransmitted I frame may contain an N(R) field and/or P/F bit information that are updated and different from those contained in the originally transmitted I frame.

Following the occurrence of a sequence error, the following means are available for initiating the retransmission of lost I frames or those with errors.

#### 6.2.2.5.3 Poll/final (P/F) bit (checkpoint) recovery

When a data station receives a frame with the P/F bit set to 1, it shall initiate retransmission of unacknowledged I frames previously transmitted with sequence numbers that are less than the V(S), send state variable, value that was current at the time of transmission of the last frame with the P/F bit, respectively, set to 1. Retransmission shall start with the oldest numbered unacknowledged I frame. I frames shall be retransmitted sequentially. New I frames may be transmitted if they become available. Such retransmission of I frames as a result of an exchange of P/F bits set to 1 is known as checkpoint retransmission.

Checkpoint retransmission shall not be initiated under the following conditions:

If a REJ command with the P bit set to 0 or 1, or a REJ response with the F bit set to 0, has been received and actioned while a P bit set to 1 was unanswered, checkpoint retransmission shall be inhibited on the next frame received with the F bit set to 1, if it would cause retransmission of the same particular I frame; i.e., same N(R) in same numbering cycle.

If a P/F bit set to 1 is received in an unnumbered format frame (SABM, UA, DISC, DM, UI or UIH), checkpoint retransmission shall be inhibited.

If, after sending a frame with the P/F bit set to 1, a data station receives an acknowledgement to that frame before receiving the corresponding frame with the P/F bit set to 1, checkpoint retransmission on the frame with the P/F bit set to 1 shall be inhibited.

If any frame with the P bit set to 1 is received, checkpoint retransmission shall be inhibited.

#### 6.2.2.5.4 REJ recovery

The REJ command/response shall be used primarily to initiate an exception recovery (retransmission), following the detection of a sequence error, earlier than is possible by checkpoint (P/F bit) recovery; for example, if a REJ frame is immediately transmitted upon detection of a sequence error, then there is no requirement to wait for a frame with the P/F bit set to 1 in order to update V(R).

With respect to each direction of transmission on the data link, only one "sent REJ" exception condition from one station to another data station shall be established at a time. A "sent REJ" exception condition shall be cleared when the requested I frame is received or when the response/command time-out function runs out. When the data station perceives by time-out that the requested I frame has not been received, because either the requested I frame or the REJ frame was in error or lost, the REJ frame may be repeated.

A station receiving a REJ frame shall initiate sequential transmission (or retransmission) of I frames starting with the I frame indicated by the N(R) contained in the REJ frame. New I frames may be transmitted subsequently if they become available.

If retransmission beginning with a particular frame occurs due to checkpointing (see clause 6.2.2.5.3), and a REJ frame is received which would also start retransmission with the same particular I frame [as identified by the N(R) in the REJ frame], the retransmission resulting from the REJ frame shall be inhibited.

#### 6.2.2.5.5 SABM Command

When this command is actioned, the responsibility for all unacknowledged I frames assigned to data link control reverts to a higher layer. Whether the content of the information field of such unacknowledged I frames is reassigned to data link control for transmission or not is decided at a higher layer.

#### 6.2.2.5.6 DISC Command

When this command is actioned, the responsibility for all unacknowledged I frames assigned to data link control reverts to a higher layer. Whether the content of the information field of such unacknowledged I frames is reassigned to data link control for transmission or not is decided at a higher layer.



---

## Annex A (informative): Advice to TE software implementers

The multiplexing protocol allows a number of virtual channels to be established between a TE and an MT. The TE is normally responsible for establishing and clearing down each virtual channel, although the MT may autonomously clear the entire multiplexing session.

Each channel will start life as an instance of 3GPP TS 27.007 [4], and will allow the normal AT command procedures for both 3GPP TS 27.007 [4] and 3GPP TS 27.005 [3]. Any changes made to the AT register settings will be valid within the virtual channel only. If registers are saved to non-volatile memory then the changes will apply to the defaults for the AT registers and will affect new channels from that point on. Such changes will only affect other active channels if they are reset with ATZ. Changes to the phonebook or to SMS messages will, of course, be on a global basis.

The software in the TE sitting above the multiplexing protocol is recommended to establish a virtual channel and leave it "idle" for the reception of incoming calls according to responses as defined in 3GPP TS 27.007 [4]. When an incoming call arrives the software may then cause an appropriate application to become active in order to receive the incoming call. The previously "idle" channel will then be occupied with this call and so the TE software is recommended to establish an additional channel to take over from the previously-idle channel in preparation for other incoming calls or indications.

Because the ISO HDLC transparency mechanism must be used if it is supported by the UE, the TE shall always try to configure the multiplexer with this transparency mechanism. If it is not supported by the UE, the TE shall configure the multiplexer in the basic option.

## Annex B (informative): Explanatory notes on the CRC Calculation

$R(p)$ = remainder of p.

Message is k bits long.

$$FCS = \text{OnesComplement} \left( R \left( \frac{(x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x^1 + 1)x^k}{x^8 + x^2 + x^1 + 1} \right) \oplus R \left( \frac{(message)x^8}{x^8 + x^2 + x^1 + 1} \right) \right)$$

### B.1 Example

A SABM frame on DLCI 1. Note that bits are written as they are sent on the serial port, LSB bit first and MSB bit last. No start stop bits, transparency bytes, BOFC or EOFC are included in the message. (The length octet is only included in the FCS for UI frames).

BOFC	DLC	Ctrl	FCS	EOFC
10011111	11100000	11111100	To be calculated	10011111

$$k=8*2=16$$

message=11100000 11111100

$$FCS = \text{OnesComplement} \left( R \left( \frac{11111111'00000000'00000000}{100000111} \right) \oplus R \left( \frac{11100000'11111100'00000000}{100000111} \right) \right) =$$

$$\text{OnesComplement}(11010111 \oplus 10111001) = \text{OnesComplement}(01101110) = 10010001$$

### B.2 Reflected bits

In the example the bits were shown as they were sent on the serial line, this is however not the way the application sees the octets, it will see MSB first and LSB last, so before calculating the FCS the octets bit order must be reversed.

BOFC	DLC	Ctrl	FCS	EOFC
0xF9	0x07	0x3F	To be calculated	0xF9
11111001	00000111	00111111	To be calculated	11111001

- 1 Reverse all bit in octets.
- 2 Calculate FCS.
- 3 Reverse all bits in FCS.
- 4 Send the reversed FCS.

Fortunately there is an easier way of doing the reversing of the bits, when implementing the CRC calculation using table lookup the table can be reversed.

## B.3 Implementation

Implementation is very simple because the FCS will be as wide as the lookup table (8 bits). To avoid having to reverse all bits in the octets all the octets in the crc table is reversed instead.

The term  $R\left(\frac{(x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x^1 + 1)x^k}{x^8 + x^2 + x^1 + 1}\right)$  corresponds to initialising the FCS with 0xFF.

### B.3.1 Calculate FCS for the example given earlier

First initialize the crc:                   FCS=0xFF  
 Add first byte:                         FCS=table[0xFF^0x07]=table[0xF8]=0xBA  
 Add second byte:                       FCS=table[0xBA^0x3F]=table[0x85]=0x76  
 Ones complement the FCS:               FCS=0xFF-FCS=0xFF-0x76=0x89 (10001001)  
 Transmit this FCS, this will be the same as the one calculated previous after the uart has reversed the bits.

### B.3.2 Check FCS for the example given earlier

First initialize the crc:                   FCS=0xFF  
 Add first byte:                         FCS=table[0xFF^0x07]=table[0xF8]=0xBA  
 Add second byte:                       FCS=table[0xBA^0x3F]=table[0x85]=0x76  
 Add FCS:                                FCS=table[0x76^0x89]=table[0xFF]=0xCF  
 0xCF is the reversed order of 11110011, the checksum is valid

### B.3.3 The transmitter code

```

/*Init*/
unsigned char FCS=0xFF;
unsigned char len;
unsigned char p;
/*len is the number of bytes in the message, p points to message*/
while (len--) {
FCS=crctable[FCS^*p++];
}
/*Ones complement*/
FCS=0xFF-FCS;

```

### B.3.4 The receiver code

```

/*Init*/
unsigned char FCS=0xFF;
unsigned char len;
unsigned char p;
/*len is the number of bytes in the message, p points to message*/
while (len--) {
FCS=crctable[FCS^*p++];
}
/*Ones complement*/
FCS=crctable[FCS^"received FCS"];
/*0xCF is the reversed order of 11110011.*/
if (FCS==0xCF) {
/*FCS is OK*/
}
else {
/*FCS is not OK*/
}

```

## B.3.5 Reversed CRC table

```

const unsigned char crctable[256] = { //reversed, 8-bit, poly=0x07
    0x00, 0x91, 0xE3, 0x72, 0x07, 0x96, 0xE4, 0x75, 0x0E, 0x9F, 0xED, 0x7C, 0x09, 0x98, 0xEA, 0x7B,
    0x1C, 0x8D, 0xFF, 0x6E, 0x1B, 0x8A, 0xF8, 0x69, 0x12, 0x83, 0xF1, 0x60, 0x15, 0x84, 0xF6, 0x67,
    0x38, 0xA9, 0xDB, 0x4A, 0x3F, 0xAE, 0xDC, 0x4D, 0x36, 0xA7, 0xD5, 0x44, 0x31, 0xA0, 0xD2, 0x43,
    0x24, 0xB5, 0xC7, 0x56, 0x23, 0xB2, 0xC0, 0x51, 0x2A, 0xBB, 0xC9, 0x58, 0x2D, 0xBC, 0xCE, 0x5F,

    0x70, 0xE1, 0x93, 0x02, 0x77, 0xE6, 0x94, 0x05, 0x7E, 0xEF, 0x9D, 0x0C, 0x79, 0xE8, 0x9A, 0x0B,
    0x6C, 0xFD, 0x8F, 0x1E, 0x6B, 0xFA, 0x88, 0x19, 0x62, 0xF3, 0x81, 0x10, 0x65, 0xF4, 0x86, 0x17,
    0x48, 0xD9, 0xAB, 0x3A, 0x4F, 0xDE, 0xAC, 0x3D, 0x46, 0xD7, 0xA5, 0x34, 0x41, 0xD0, 0xA2, 0x33,
    0x54, 0xC5, 0xB7, 0x26, 0x53, 0xC2, 0xB0, 0x21, 0x5A, 0xCB, 0xB9, 0x28, 0x5D, 0xCC, 0xBE, 0x2F,

    0xE0, 0x71, 0x03, 0x92, 0xE7, 0x76, 0x04, 0x95, 0xEE, 0x7F, 0x0D, 0x9C, 0xE9, 0x78, 0x0A, 0x9B,
    0xFC, 0x6D, 0x1F, 0x8E, 0xFB, 0x6A, 0x18, 0x89, 0xF2, 0x63, 0x11, 0x80, 0xF5, 0x64, 0x16, 0x87,
    0xD8, 0x49, 0x3B, 0xAA, 0xDF, 0x4E, 0x3C, 0xAD, 0xD6, 0x47, 0x35, 0xA4, 0xD1, 0x40, 0x32, 0xA3,
    0xC4, 0x55, 0x27, 0xB6, 0xC3, 0x52, 0x20, 0xB1, 0xCA, 0x5B, 0x29, 0xB8, 0xCD, 0x5C, 0x2E, 0xBF,

    0x90, 0x01, 0x73, 0xE2, 0x97, 0x06, 0x74, 0xE5, 0x9E, 0x0F, 0x7D, 0xEC, 0x99, 0x08, 0x7A, 0xEB,
    0x8C, 0x1D, 0x6F, 0xFE, 0x8B, 0x1A, 0x68, 0xF9, 0x82, 0x13, 0x61, 0xF0, 0x85, 0x14, 0x66, 0xF7,
    0xA8, 0x39, 0x4B, 0xDA, 0xAF, 0x3E, 0x4C, 0xDD, 0xA6, 0x37, 0x45, 0xD4, 0xA1, 0x30, 0x42, 0xD3,
    0xB4, 0x25, 0x57, 0xC6, 0xB3, 0x22, 0x50, 0xC1, 0xBA, 0x2B, 0x59, 0xC8, 0xBD, 0x2C, 0x5E, 0xCF
};

```

## Annex C (informative): Change History

TSG	TDoc	VERS	NEW_VERS	CR	REV	REL	CAT	WORKITEM	SUBJECT
T#4	TP-99119	2.0.0	3.0.0	New		R99			Creation of 3GPP 27.010 v3.0.0 out of GSM 07.10 v6.3.0
T#4	TP-99124	3.0.0	3.1.0	001		R99	A	MUX MS-TE	Clarification of how to handle the length field in basic mode
T#4	TP-99146	3.0.0	3.1.0	002		R99	A	MUX MS-TE	Editorial corrections
T#5	TP-99177	3.1.0	3.2.0	003		R99	A	TEI	Clarification of CR bit
T#5	TP-99177	3.1.0	3.2.0	004		R99	A	TEI	Correction of the bits in the start and close flags of the frame in the example on Annex B
T#7	TP-000024	3.2.0	3.3.0	005		R99	F	TEI	Adaptations for UMTS
T#11	-	3.3.0	4.0.0			Rel4			Upgrade to Rel-4
T#13	TP-010212	4.0.0	4.1.0	006		Rel4	F	TEI4	Conversion of GSM to 3GPP references
T#15	TP-020014	4.1.0	4.2.0	008		Rel4	A	TEI	Incorrect explanation of length indicator bit
T#16	-	4.2.0	5.0.0	-		Rel5			Upgrade to Rel-5
T#26	-	5.0.0	6.0.0	-		Rel6			Upgrade to Rel-6
CT#36	-	6.0.0	7.0.0	-		Rel7			Upgrade to Rel-7
CT#42		7.0.0	8.0.0			Rel8			Upgraded to v8.0.0 due to simple upgrade without no technical change
CT#46		8.0.0	9.0.0			Rel9			Automatic upgrade from previous Release
CT#50	CP-100779	9.0.0	9.1.0	010		Rel9	A	TEI8	Correcting non-specific external references
CT#51		9.1.0	10.0.0						Automatic upgrade from previous Release version 9.1.0
CT#57		10.0.0	11.0.0						Automatic upgrade from previous Release version 10.0.0
		11.0.0	12.0.0						Automatic upgrade from previous Release
		12.0.0	13.0.0						Automatic upgrade from previous Release
<b>Change history</b>									
Date	TSG #	TSG Doc.	CR	Rev	Cat	Subject/Comment			New
2017-03	CT#75					Automatic upgrade from previous Release			14.0.0
2018-06	CT#80					Automatic upgrade from previous Release			15.0.0
2020-06	CT#88e	-	-	-		Update to Rel-16 version (MCC)			16.0.0

---

# History

<b>Document history</b>		
V16.0.0	August 2020	Publication