



INTERNATIONAL TELECOMMUNICATION UNION

**CCITT**

THE INTERNATIONAL  
TELEGRAPH AND TELEPHONE  
CONSULTATIVE COMMITTEE

**X.224**

(11/1988)

SERIES X: DATA COMMUNICATION NETWORKS:  
OPEN SYSTEMS INTERCONNECTION (OSI) –  
PROTOCOL SPECIFICATIONS, CONFORMANCE  
TESTING

---

**TRANSPORT PROTOCOL SPECIFICATION  
FOR OPEN SYSTEMS INTERCONNECTION  
FOR CCITT APPLICATIONS**

Reedition of CCITT Recommendation X.224 published in  
the Blue Book, Fascicle VIII.5 (1988)

---

## NOTES

- 1 CCITT Recommendation X.224 was published in Fascicle VIII.5 of the *Blue Book*. This file is an extract from the *Blue Book*. While the presentation and layout of the text might be slightly different from the *Blue Book* version, the contents of the file are identical to the *Blue Book* version and copyright conditions remain unchanged (see below).
- 2 In this Recommendation, the expression “Administration” is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

## Recommendation X.224

### TRANSPORT PROTOCOL SPECIFICATION FOR OPEN SYSTEMS INTERCONNECTION FOR CCITT APPLICATIONS<sup>1)</sup>

(Malaga-Torremolinos, 1984; amended at Melbourne, 1988)

The CCITT,

*considering*

- (a) that Recommendation X.200 defines the Reference Model of Open Systems Interconnection for CCITT Applications;
- (b) that Recommendation X.214 is the Transport Service Definition for Open Systems Interconnection for CCITT Applications;
- (c) that Recommendation X.213 is the Network Service Definition for Open Systems Interconnection for CCITT Applications;
- (d) that Recommendation T.70 defines the Network-Independent Basic Transport Service for Teletex,

*unanimously declares*

- (1) that scope, field of application, references, definitions, symbols and abbreviations are given in §§ 1 to 4;
- (2) that the Transport Protocol is overviewed in § 5;
- (3) that the elements of procedure are as specified in § 6;
- (4) that the classes of protocol are as specified in §§ 7 to 12;
- (5) that the structure and encoding of the transport protocol data units is as specified in § 13;
- (6) that the conformance requirements are as specified in § 14;
- (7) that the state table description of the Transport Protocol is contained in Annex A.

---

<sup>1)</sup> Recommendation X.224 and ISO 8073 (Information Processing Systems – Open Systems Interconnection – Transport Protocol Specification) were developed in close collaboration and are technically aligned, except for the differences noted in Appendix II.

## CONTENTS

0	<i>Introduction</i>
1	<i>Scope and field of application</i>
2	<i>References</i>
3	<i>Definitions</i>
4	<i>Symbols and abbreviations</i>
5	<i>Overview of the transport protocol</i>
5.1	Service provided by the transport layer
5.2	Service assumed from the network layer
5.3	Functions of the transport layer
5.4	Classes and options
5.5	Model of the transport layer
6	<i>Elements of procedure</i>
6.1	Assignment to network connection
6.2	Transport protocol data unit (TPDU) transfer
6.3	Segmenting and reassembling
6.4	Concatenation and separation
6.5	Connection establishment
6.6	Connection refusal
6.7	Normal release
6.8	Error release
6.9	Association of TPDU with transport connections
6.10	Data TPDU numbering
6.11	Expedited data transfer
6.12	Reassignment after failure
6.13	Retention until acknowledgment of TPDU
6.14	Resynchronization
6.15	Multiplexing and demultiplexing
6.16	Explicit flow control
6.17	Checksum
6.18	Frozen references
6.19	Retransmission on timeout
6.20	Resequencing
6.21	Inactivity control
6.22	Treatment of protocol errors
6.23	Splitting and recombining

7	<i>Protocol classes</i>
8	<i>Specification for Class 0: Simple Class</i>
8.1	Functions of Class 0
8.2	Procedures for Class 0
9	<i>Specification for Class 1: basic error recovery class</i>
9.1	Functions of Class 1
9.2	Procedures for Class 1
10	<i>Specification for Class 2: multiplexing class</i>
10.1	Functions of Class 2
10.2	Procedures for Class 2
11	<i>Specification for Class 3: error recovery and multiplexing class</i>
11.1	Functions of Class 3
11.2	Procedures for Class 3
12	<i>Specification for Class 4: error detection and recovery class</i>
12.1	Functions of Class 4
12.2	Procedures for Class 4
13	<i>Structure and encoding of TPDU's</i>
13.1	Validity
13.2	Structure
13.3	Connection request (CR) TPDU
13.4	Connection confirm (CC) TPDU
13.5	Disconnect request (DR) TPDU
13.6	Disconnect confirm (DC) TPDU
13.7	Data (DT) TPDU
13.8	Expedited data (ED) TPDU
13.9	Data acknowledgment (AK) TPDU
13.10	Expedited data acknowledgment (EA)
13.11	Reject (RJ) TPDU
13.12	TPDU error (ER) TPDU
14	<i>Conformance</i>
<i>Annex A</i>	– State Tables
<i>Annex B</i>	– Transport Protocol Identification
<i>Appendix I</i>	– Checksum Algorithms
<i>Appendix II</i>	– Differences between Recommendation X.224 and ISO 8073 (1986)

## 0 Introduction

The Transport Protocol is one of a set of Recommendations produced to facilitate the interconnection of computer systems. The set of Recommendations covers the services and protocols required to achieve such interconnection.

The Transport Protocol is positioned with respect to other related Recommendations by the layers defined in the Reference Model of Open Systems Interconnection for CCITT Applications [1]. It is most closely related to, and lies within the field of application of the Transport Service [2]. It also uses and makes reference to the Network Service [3], whose provisions it assumes in order to accomplish the transport protocol's aims. The interrelationship of these Recommendations is depicted in Figure 1/X.224.

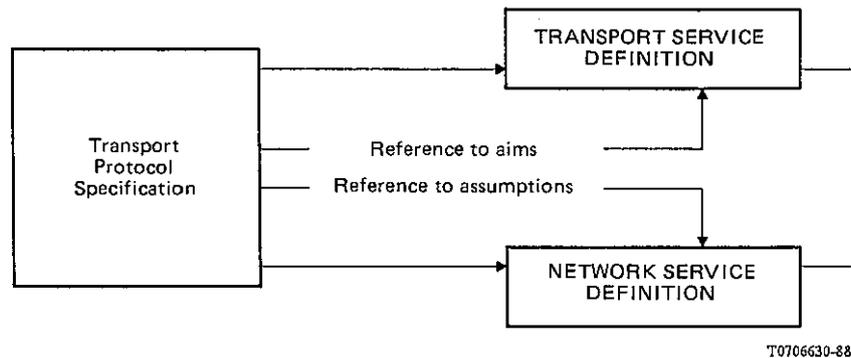


FIGURE 1/X.224

### Relationship between the Transport Protocol and adjacent services

This Recommendation specifies a common encoding and a number of classes of transport protocol procedures to be used with different network qualities of service.

It is intended that the Transport Protocol should be simple but general enough to cater for the total range of Network Service qualities possible, without restricting future extensions.

The protocol is structured to give rise to classes of protocol which are designed to minimize possible incompatibilities and implementation costs.

The classes are selectable with respect to the Transport and Network Services in providing the required quality of service for the interconnection of two session entities (note that each class provides a different set of functions for enhancement of service qualities).

This protocol defines mechanisms that can be used to optimize network tariffs and enhance the following qualities of service:

- a) different throughput;
- b) different error rates;
- c) integrity of data requirements;
- d) reliability requirements.

It does not require an implementation to use all of these mechanisms, nor does it define methods for measuring achieved quality of service or criteria for deciding when to release transport connections following quality of service degradation.

The primary aim of this Recommendation is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer entities at the time of communication. These rules for communication are intended to provide a sound basis for development in order to serve a variety of purposes:

- a) as a guide for implementors and designers;
- b) for use in the testing and procurement of equipment;
- c) as part of an agreement for the admittance of systems into the open systems environment;
- d) as a refinement of the understanding of OSI.

It is expected that the initial users of the Recommendation will be designers and implementors of equipment and the Recommendation contains, in notes or in Annexes, guidance on the implementation of the procedures defined in the Recommendation.

It should be noted that, as the number of valid protocol sequences is very large, it is not possible with current technology to verify that an implementation will operate the protocol defined in this Recommendation correctly under all circumstances. It is possible by means of testing to establish confidence that an implementation correctly operates the protocol in a representative sample of circumstances. It is, however, intended that this Recommendation can be used in circumstances where two implementations fail to communicate in order to determine whether one or both have failed to operate the protocol correctly.

This Recommendation contains a section on conformance of equipment claiming to implement the procedures in this Recommendation. Attention is drawn to the fact that the Recommendation does not contain any tests to demonstrate this conformance.

The variations and options available within this Recommendation are essential to enable a Transport Service to be provided for a wide variety of applications over a variety of network qualities. Thus, a minimally conforming implementation will not be suitable for use in all possible circumstances. It is important therefore to qualify all references to this Recommendation with statements of the options provided or required or with statements of the intended purpose of provision or use.

## **1 Scope and field of application**

1.1 This Recommendation specifies:

a) five classes of procedures:

- 1) Class 0: Simple Class;
- 2) Class 1: Basic Error Recovery Class;
- 3) Class 2: Multiplexing Class;
- 4) Class 3: Error Recovery and Multiplexing Class;
- 5) Class 4: Error Detection and Recovery Class;

for the connection oriented transfer of data and control information from one transport entity to a peer transport entity;

b) the means of negotiating the class of procedures to be used by the transport entities;

c) the structure and encoding of the transport protocol data units used for the transfer of data and control information.

1.2 The procedures are defined in terms of:

- a) the interactions between peer transport entities through the exchange of transport protocol data units;
- b) the interactions between a transport entity and the transport service user in the same system through the exchange of transport service primitives;
- c) the interactions between a transport entity and the network service provider through the exchange of network service primitives.

These procedures are defined in the main text of the standard supplemented by state tables in Annex A.

1.3 These procedures are applicable to instances of communication between systems which support the Transport Layer of the OSI Reference Model and which wish to interconnect in an open systems environment.

1.4 This Recommendation specifies, in § 14, conformance for systems implementing these procedures. It does not contain tests which can be used to demonstrate these conformance requirements.

## **2 References**

- [1] Recommendation X.200 – Reference Model of Open Systems Interconnection for CCITT Applications (see also ISO 7498);
- [2] Recommendation X.214 – Transport Service Definition for Open Systems Interconnection for CCITT Applications (see also ISO 8072);

- [3] Recommendation X.213 – Network Service Definition for Open Systems Interconnection for CCITT Applications (see also ISO 8348);
- [4] Recommendation T.70 – Network-Independent Basic Transport Service for Teletex;

### 3 Definitions

*Note* – The definitions contained in this section make use of abbreviations defined in § 4.

3.1 This Recommendation is based on the concepts developed in the Reference Model of Open Systems Interconnection for CCITT Applications [1] and makes use of the following terms defined in that Recommendation:

- a) concatenation and separation;
- b) segmenting and reassembling;
- c) multiplexing and demultiplexing;
- d) splitting and recombining;
- e) flow control.

3.2 For the purpose of this Recommendation, the following definitions apply:

#### 3.2.1 **equipment**

Hardware or software or a combination of both; it need not be physically distinct within a computer system.

#### 3.2.2 **transport service user**

An abstract representation of the totality of those entities within a single system that make use of the transport service.

#### 3.2.3 **network service provider**

An abstract machine which models the totality of the entities providing the network service, as viewed by a transport entity.

#### 3.2.4 **local matter**

A decision made by a system concerning its behaviour in the Transport Layer that is not subject to the requirements of this protocol.

#### 3.2.5 **initiator**

A transport entity that initiates a CR TPDU.

#### 3.2.6 **responder**

A transport entity with whom an initiator wishes to establish a transport connection.

*Note* – Initiator and responder are defined with respect to a single transport connection. A transport entity can be both an initiator and responder simultaneously.

#### 3.2.7 **sending transport entity**

A transport entity that sends a given TPDU.

#### 3.2.8 **receiving transport entity**

A transport entity that receives a given TPDU.

#### 3.2.9 **preferred class**

The protocol class that the initiator indicates in a CR TPDU as its first choice for use over the transport connection.

### 3.2.10 **alternative class**

A protocol class that the initiator indicates in a CR TPDU as an alternative choice for use over the transport connection.

### 3.2.11 **proposed class**

A preferred class or an alternative class.

### 3.2.12 **selected class**

The protocol class that the responder indicates in a CC TPDU that it has chosen for use over the transport connection.

### 3.2.13 **proposed parameter**

The value for a parameter that the initiator indicates in a CR TPDU that it wishes to use over the transport connection.

### 3.2.14 **selected parameter**

The value for a parameter that the responder indicates in a CC TPDU that it has chosen for use over the transport connection.

### 3.2.15 **error indication**

An N-RESET indication, or an N-DISCONNECT indication with a reason code indicating an error, that a transport entity receives from the NS-provider.

### 3.2.16 **invalid TPDU**

A TPDU which does not comply with the requirements of this Recommendation for structure and encoding.

### 3.2.17 **protocol error**

A TPDU whose use does not comply with the procedures for the class.

### 3.2.18 **sequence number**

- a) The number in the TPDU-NR field of a DT TPDU which indicates the order in which the DT TPDU was transmitted by a transport entity.
- b) The number in the YR-TU-NR field of an AK or RJ TPDU which indicates the sequence number of the next DT TPDU expected to be received by a transport entity.

### 3.2.19 **transmit window**

The set of consecutive sequence numbers which a transport entity has been authorised by its peer entity to send at a given time on a given transport connection.

### 3.2.20 **lower window edge**

The lowest sequence number in a transmit window.

### 3.2.21 **upper window edge**

The sequence number which is one greater than the highest sequence number in the transmit window.

### 3.2.22 **upper window edge allocated to the peer entity**

The value that a transport entity communicates to its peer entity to be interpreted as its new upper window edge.

### 3.2.23 **closed window**

A transmit window which contains no sequence number.

### 3.2.24 **window information**

Information contained in a TPDU relating to the upper and the lower window edges.

### 3.2.25 **frozen reference**

A reference which is not available for assignment to a connection because of the requirements of § 6.18.

### 3.2.26 **unassigned reference**

A reference that is neither currently in use for identifying a transport connection nor in a frozen state.

### 3.2.27 **transparent (data)**

TS-user data which is transferred intact between transport entities and which is unavailable for use by the transport entities.

### 3.2.28 **owner (of a network connection)**

The transport entity that issued the N-CONNECT request leading to the creation of that network connection.

### 3.2.29 **retained TPDU**

A TPDU which is subject to the retransmission procedure or retention until acknowledgment procedure and is available for possible retransmission.

## **4 Symbols and abbreviations**

### 4.1 *Data units*

TPDU	Transport Protocol Data Unit
TSDU	Transport Service Data Unit
NSDU	Network Service Data Unit

### 4.2 *Types of transport protocol data unit*

CR TPDU	Connection Request TPDU
CC TPDU	Connection Confirm TPDU
DR TPDU	Disconnect Request TPDU
DC TPDU	Disconnect Confirm TPDU
DT TPDU	Data TPDU
ED TPDU	Expedited Data TPDU
AK TPDU	Data Acknowledge TPDU
EA TPDU	Expedited Acknowledge TPDU
RJ TPDU	Reject TPDU
ER TPDU	Error TPDU

### 4.3 *TPDU fields*

LI	Length Indicator (field)
CDT	Credit (field)
TSAP-ID	Transport Service Access Point Identifier (field)
DST-REF	Destination Reference (field)
SRC-REF	Source Reference (field)
EOT	End of TSDU mark
TPDU-NR	DT TPDU number (field)
ED-TPDU-NR	ED TPDU number (field)

YR-TU-NR	Sequence number response (field)
YR-EDTU-NR	ED TPDU number response (field)

#### 4.4 *Times and associated variables*

T1	Local retransmission time
N	The maximum number of retransmissions
L	Time bound on reference and sequence numbers
I	Inactivity time
W	Window time
TTR	Time to try reassignment/resynchronization
TWR	Time to wait for reassignment/resynchronization
TS1	Supervisory timer 1
TS2	Supervisory timer 2
M <sub>LR</sub>	NSDU lifetime local-to-remote
M <sub>RL</sub>	NSDU lifetime remote-to-local
E <sub>LR</sub>	Expected maximum transit delay local-to-remote
E <sub>RL</sub>	Expected maximum transit delay remote-to-local
R	Persistence time
A <sub>L</sub>	Local acknowledge time
A <sub>R</sub>	Remote acknowledge time

#### 4.5 *Miscellaneous*

TS-user	Transport Service user
TSAP	Transport Service Access Point
NS-provider	Network Service provider
NSAP	Network Service Access Point
QOS	Quality of service

## 5 **Overview of the transport protocol**

*Note* – This overview is not exhaustive and has been provided for guidance to the reader of this Recommendation.

### 5.1 *Service provided by the transport layer*

The protocol specified in this Recommendation supports the transport service defined in [2].

Information is transferred to and from the TS-user in the transport service primitives listed in Table 1/X.224.

### 5.2 *Service assumed from the network layer*

The protocol specified in this Recommendation assumes the use of the network services defined in [3].

Information is transferred to and from the NS-provider in the network service primitives listed in Table 2/X.224.

### 5.3 *Functions of the transport layer*

#### 5.3.1 *Overview of functions*

The functions in the transport layer are those necessary to bridge the gap between the services available from the network layer and those to be offered to the TS-users.

The functions in the transport layer are concerned with the enhancement of quality of service, including aspects of cost optimization.

The functions are grouped below into those used at all times during a transport connection and those concerned with connection establishment, data transfer and release.

*Note* – This Recommendation does not include the following functions which are under consideration for inclusion in future editions of this Recommendation:

- a) encryption;
- b) accounting mechanisms;
- c) status exchanges and monitoring of QOS;
- d) blocking;
- e) temporary release of network connections;
- f) alternative checksum algorithm.

TABLE 1/X.224

**Transport service primitives**

Primitive	Parameters
T-CONNECT request T-CONNECT indication	Called Address, Calling Address, Expedited Data Option, Quality of Service, TS-User data.
T-CONNECT response T-CONNECT confirm	Responding Address, Quality of Service, Expedited Data Option, TS-User data.
T-DATA request T-DATA indication	TS-User data.
T-EXPEDITED DATA request T-EXPEDITED DATA indication	TU-User data.
T-DISCONNECT request	TS-User data.
T-DISCONNECT indication	Disconnect reason, TS-User data.

5.3.1.1 *Functions used at all times*

The following functions, depending on the selected class and options, are used at all times during a transport connection:

- a) *transmission of TPDU*s (see §§ 6.2 and 6.9);
- b) *multiplexing and demultiplexing* (see § 6.15), a function used to share a single network connection between two or more transport connections;
- c) *error detection* (see §§ 6.10, 6.13 and 6.17), a function used to detect the loss, corruption, duplication, misordering or misdelivery of TPDU)s;
- d) *error recovery* (see §§ 6.12, 6.14, 6.18, 6.19, 6.20, 6.21 and 6.22), a function used to recover from detected and signalled errors.

TABLE 2/X.224

## Network service primitives

Primitives	X/Y	Parameters (1, 2)	X/Y/Z
N-CONNECT request	X	Called address,	X
N-CONNECT indication	X	Calling address, Receipt confirmation selection, Expedited data selection, QOS parameter set, NS user data (3).	X Y Y X Z
N-CONNECT response	X	Responding address.	X
N-CONNECT confirmation	X	Receipt confirmation selection, Expedited data selection, QOS parameter set, NS user data (3).	Y Y X Z
N-DATA request	X	NX-user data.	X
N-DATA indication	X	Confirmation request.	Y
N-DATA ACKNOWLEDGE request	Y		
N-DATA ACKNOWLEDGE indication	Y		
N-EXPEDITED DATA request	Y	NS user data.	Y
N-EXPEDITED DATA indication	Y		
N-RESET request	X	Reason.	Z
N-RESET indication	X	Originator, Reason.	Z Z
N-RESET response	X		
N-RESET confirmation	X		
N-DISCONNECT request	X	Reason, NS user data, Responding address	Z Z Z
N-DISCONNECT indication	X	Originator, Reason, NS user data Responding address.	Z Z Z Z

X: The Transport Protocol assumes that this feature is provided by all network service providers.

Y: The Transport Protocol assumes that this feature is provided by some network service providers and a mechanism is provided to optionally use the feature.

Z: The Transport Protocol does not use this parameter.

Note 1 – The parameters listed in this table are those in the network service definition (Reference 3).

Note 2 – The way the parameters are exchanged between the transport entity and the NS-provider is a local matter.

Note 3 – Although not used in the transport protocol *per se*, this parameter may be used for transport protocol identification, as specified in Annex B.

### 5.3.1.2 *Connection establishment*

The purpose of connection establishment is to establish a transport connection between two TS-users. The following functions of the transport layer during this phase match the TS-users' requested quality of service with the services offered by the network layer:

- a) select network service which best matches the requirement of the TS-user taking into account charges for various services (see § 6.5);
- b) decide whether to multiplex multiple transport connections onto a single network connection (see § 6.5);
- c) establish the optimum TPDU size (see § 6.5);
- d) select the functions that will be operational upon entering the data transfer phase (see § 6.5);
- e) map transport addresses onto network addresses;
- f) provide a means to distinguish between two different transport connections (see § 6.5);
- g) transport of TS-user data (see § 6.5).

### 5.3.1.3 *Data transfer*

The purpose of data transfer is to permit duplex transmission of TSDUs between the two TS-users connected by the transport connection. This purpose is achieved by means of two-way simultaneous communication and by the following functions, some of which are used or not used in accordance with the result of the selection performed in connection establishment.

- a) *concatenation and separation* (see § 6.4), a function used to collect several TPDU's into a single NSDU at the sending transport entity and to separate the TPDU's at the receiving transport entity;
- b) *segmenting and reassembling* (see § 6.3), a function used to segment a single TSDU into multiple TPDU's at the sending transport entity and to reassemble them into their original format at the receiving transport entity;
- c) *splitting and recombining* (see § 6.23), a function allowing the simultaneous use of two or more network connections to support the same transport connection;
- d) *flow control* (see § 6.16), a function used to regulate the flow of TPDU's between two transport entities on one transport connection;
- e) *transport connection identification*, a means to uniquely identify a transport connection between the pair of transport entities supporting the connection during the lifetime of the transport connection;
- f) *expedited data* (see § 6.11), a function used to bypass the flow control of normal data TPDU. Expedited data TPDU flow is controlled by separate flow control;
- g) *TSDU delimiting* (see § 6.3), a function used to determine the beginning and ending of a TSDU.

### 5.3.1.4 *Release*

The purpose of release (see §§ 6.7 and 6.8) is to provide disconnection of the transport connection, regardless of the current activity.

## 5.4 *Classes and options*

### 5.4.1 *General*

The functions of the transport layer have been organized into classes and options.

A class defines a set of functions. Options define those functions within a class which may or may not be used.

This Recommendation defines five classes of protocol:

- a) Class 0: Simple Class;
- b) Class 1: Basic Error Recovery Class;
- c) Class 2: Multiplexing Class;
- d) Class 3: Error Recovery and Multiplexing Class;
- e) Class 4: Error Detection and Recovery Class.

*Note 1* – Transport connections of Classes 2, 3 and 4 may be multiplexed together onto the same network connection.

*Note 2* – Classes 0 to 3 do not specify mechanisms to detect unsignalled network transmission failures.

#### 5.4.2 *Negotiation*

The use of classes and options is negotiated during connection establishment. The choice made by the transport entities will depend on:

- a) the TS-users' requirements expressed via T-CONNECT service primitives;
- b) the quality of the available network services;
- c) the user required service versus cost ratio acceptable for the TS-user.

#### 5.4.3 *Choice of network connection*

The following list classifies network services in terms of quality with respect to error behaviour in relation to user requirements; its main purpose is to provide a basis for the decision regarding which class of transport connection should be used in conjunction with a given network connection.

- a) *Type A* – Network connection with acceptable residual error rate (for example not signalled by disconnect or reset) and acceptable rate of signalled errors.
- b) *Type B* – Network connections with acceptable residual error rate (for example not signalled by disconnect or reset) but unacceptable rate of signalled errors.
- c) *Type C* – Network connections with unacceptable residual error rate.

It is assumed that each transport entity is aware of the quality of service provided by particular Network connection.

#### 5.4.4 *Characteristics of Class 0*

Class 0 provides the simplest type of transport connection and is fully compatible with Recommendation T.70 [4].

Class 0 has been designed to be used with type A network connections.

#### 5.4.5 *Characteristics of Class 1*

Class 1 provides a basic transport connection with minimal overheads.

The main purpose of the class is to recover from network disconnect or reset.

Selection of this class is usually based on reliability criteria. Class 1 has been designed to be used with type B network connections.

#### 5.4.6 *Characteristics of Class 2*

##### 5.4.6.1 *General*

Class 2 provides a way to multiplex several transport connections onto a single network connection. This class has been designed to be used with type A network connections.

##### 5.4.6.2 *Use of explicit flow control*

The objective is to provide flow control to help avoid congestion at transport-connection-end-points and on the network connection. Typical use is when traffic is heavy and continuous, or when there is intensive multiplexing. Use of flow control can optimize response times and resource utilization.

##### 5.4.6.3 *Non-use of explicit flow control*

The objective is to provide a basic transport connection with minimal overheads suitable when explicit disconnection of the transport connection is desirable. The option would typically be used for unsophisticated terminals, and when no multiplexing onto network connections is required. Expedited data is never available.

#### 5.4.7 Characteristics of Class 3

Class 3 provides the characteristics of Class 2 plus the ability to recover from network disconnect or reset. Selection of this class is usually based upon reliability criteria. Class 3 has been designed to be used with type B network connections.

#### 5.4.8 Characteristics of Class 4

Class 4 provides the characteristics of Class 3, plus the capability to detect and recover from errors which occur as a result of the low grade of service available from the NS-provider. The kinds of errors to be detected include: TPDU loss, TPDU delivery out of sequence, TPDU duplication and TPDU corruption. These errors may affect control TPDU's as well as data TPDU's.

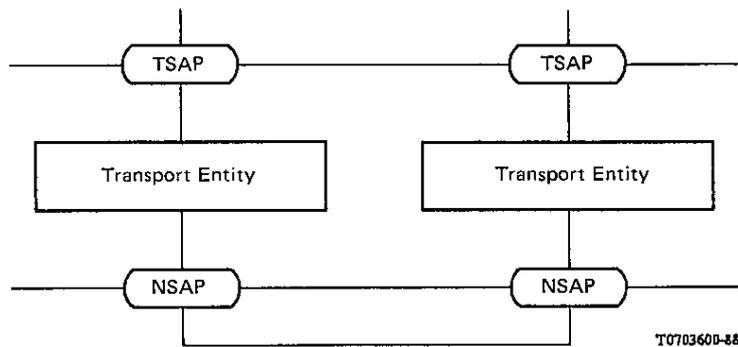
This class also provides for increased throughput capability and additional resilience against network failure.

Class 4 has been designed to be used with type C network connections.

#### 5.5 Model of the transport layer

A transport entity communicates with its TS-users through one or more TSAPs by means of the service primitives as defined by the transport service definition (Reference 2). Service primitives will cause or be the result of transport protocol data unit exchanges between the peer transport entities supporting a transport connection. These protocol exchanges are effected using the services of the network layer as defined by the network service definition [3] through one or more NSAPs.

Transport connection endpoints are identified in end systems by an internal, implementation-dependent mechanism so that the TS-user and the transport entity can refer to each transport connection (see Figure 2/X.224).



*Note* – For the purpose of illustration, this figure shows only one TSAP and one NSAP for each transport entity. In certain instances, more than one TSAP and/or more than one NSAP may be associated with a particular transport entity.

FIGURE 2/X.224

Model of the transport layer

## 6 Elements of procedure

This section contains elements of procedure which are used in the specification of protocol classes in §§ 7 to 12. These elements are not meaningful on their own.

The procedures define the transfer of TPDU's whose structure and coding is specified in § 13. Transport entities shall accept and respond to any TPDU received in a valid NSDU and may issue TPDU's initiating specific elements of procedure specified in this section.

*Note* – Where network service primitives or TPDU's and parameters used are not significant for a particular element of procedure, they have not been included in the specification.

### 6.1 Assignment to network connection

#### 6.1.1 Purpose

The procedure is used in all classes to assign transport connections to network connections.

### 6.1.2 *Network service primitives*

The procedure makes use of the following network service primitives:

- a) N-CONNECT;
- b) N-DISCONNECT.

### 6.1.3 *Procedure*

Each transport connection shall be assigned to a network connection. The initiator may assign the transport connection to an existing network connection of which it is the owner or to a new network connection (see Note 1) which it creates for this purpose.

The initiator shall not assign or reassign the transport connection to an existing network connection if the protocol class(es) proposed or the class in use for the transport connection are incompatible with the current usage of the network connection with respect to multiplexing (see Note 2).

During the resynchronization (see § 6.14) and reassignment after failure (see § 6.12) procedures, a transport entity may reassign a transport connection to another network connection joining the same NSAPs, provided that it is the owner of the network connection and that the transport connection is assigned to only one network connection at any given time.

During the splitting procedure (see § 6.23), a transport entity may assign a transport connection to any additional network connection joining the same NSAPs, provided that it is the owner of the network connection and that multiplexing is possible on the network connection.

The responder becomes aware of the assignment when it receives:

- a) a CR TPDU during the connection establishment procedure (see § 6.5); or
- b) an RJ TPDU or a retransmitted CR or DR TPDU during the resynchronization (see § 6.14) and reassignment after failure (see § 6.12) procedures; or
- c) any TPDU when splitting (see § 6.23) is used.

*Note 1* – When a new network connection is created, the quality of service requested is a local matter, although it will normally be related to the requirements of transport connection(s) expected to be assigned to it.

*Note 2* – An existing network connection may also not be suitable if, for example, the quality of service requested for the transport connection cannot be attained by using or enhancing the network connection.

*Note 3* – A network connection with no transport connection(s) assigned to it, may be available after initial establishment or because all of the transport connections previously assigned to it have been released. It is suggested that only the owner of such a network connection should release it. Furthermore, it is suggested that it not be released immediately after the transmission of the final TPDU of a transport connection – either a DR TPDU in response to a CR TPDU or a DC TPDU in response to a DR TPDU. An appropriate delay will allow the TPDU concerned to reach the other transport entity, allowing the freeing of any resources associated with the transport connection concerned.

*Note 4* – After the failure of a network connection, transport connections which were previously multiplexed together may be assigned to different network connections, and vice versa.

*Note 5* – The transport protocol identification procedures specified in Annex B may need to be considered in conjunction with this procedure.

## 6.2 *Transport protocol data unit (TPDU) transfer*

### 6.2.1 *Purpose*

The TPDU transfer procedure is used in all classes to convey transport protocol data units in user data fields of network service primitives.

### 6.2.2 *Network service primitives*

This procedure uses the following network service primitives:

- a) N-DATA;
- b) N-EXPEDITED DATA.

### 6.2.3 *Procedure*

The transport protocol data units (TPDUs) defined for the protocol are listed in § 4.2.

When the network expedited variant has been selected for Class 1, the transport entities shall transmit and receive ED and EA TPDUs as NS-user data parameters of N-EXPEDITED DATA primitives.

In all other cases, transport entities shall transmit and receive TPDUs as NS-user data parameters of N-DATA primitives.

When a TPDU is put into an NS-user data parameter, the significance of the bits within an octet and the order of octets within a TPDU shall be as defined in § 13.2.

*Note 1* – TPDUs may be concatenated (see § 6.4).

*Note 2* – The transport protocol identification procedures specified in Annex B may need to be considered in conjunction with this procedure.

## 6.3 *Segmenting and reassembling*

### 6.3.1 *Purpose*

The segmenting and reassembling procedure is used in all classes to map TSDUs onto TPDUs.

### 6.3.2 *TPDUs and parameters used*

The procedure makes use of the following TPDU and parameter:

DT TPDU:

- End of TSDU.

### 6.3.3 *Procedure*

A transport entity shall map a TSDU on to an ordered sequence of one or more DT TPDUs. This sequence shall not be interrupted by other DT TPDUs on the same transport connection.

All DT TPDUs except the last DT TPDU in a sequence greater than one shall have a length of data greater than zero.

*Note 1* – The EOT of a DT TPDU indicates whether or not there are subsequent DT TPDUs in the sequence.

*Note 2* – There is no requirement that the DT TPDUs shall be of the maximum length selected during connection establishment.

## 6.4 *Concatenation and separation*

### 6.4.1 *Purpose*

The procedure for concatenation and separation is used in Classes 1, 2, 3 and 4 to convey multiple TPDUs in one NSDU.

### 6.4.2 *Procedure*

A transport entity may concatenate TPDUs from the same or different transport connections while maintaining the order of TPDUs for a given transport connection compatible with the protocol operation.

A valid set of concatenated TPDUs may contain:

- a) any number of TPDUs from the following list: AK, EA, RJ, ER, DC TPDUs provided that these TPDUs come from different transport connections;
- b) no more than one TPDU from the following list: CR, DR, CC, DT, ED TPDUs; if this TPDU is present, it shall be placed last in the set of concatenated TPDUs.

A transport entity shall accept a valid set of concatenated TPDUs.

*Note 1* – The TPDUs within a concatenated set may be distinguished by means of the length indicator parameter.

*Note 2* – The end of a TPDU containing data is indicated by the termination of the NSDU.

*Note 3* – The number of concatenated TPDU's referred to in § 6.4.2 a) is bounded by the maximum number of transport connections which are multiplexed together, except during assignment or reassignment.

## 6.5 *Connection establishment*

### 6.5.1 *Purpose*

The procedure for connection establishment is used in all classes to create a new transport connection.

### 6.5.2 *Network service primitives*

The procedure uses the following network service primitive:

N-DATA.

### 6.5.3 *TPDU's and parameters used*

The procedure uses the following TPDU's and parameters:

- a) CR TPDU;
  - CDT;
  - DST-REF (set to zero);
  - SRC-REF;
  - CLASS and OPTIONS (preferred), i.e.:
    - i) class,
    - ii) use of extended format,
    - iii) non-use of explicit flow control in Class 2;
  - calling TSAP-ID;
  - called TSAP-ID;
  - TPDU size (proposed);
  - version number;
  - protection parameter;
  - checksum;
  - additional option selection (preferred), i.e.:
    - i) use of network expedited in Class 1,
    - ii) use of receipt confirmation in Class 1,
    - iii) non-use of checksums in Class 4,
    - iv) use of transport expedited data transfer service;
  - alternative protocol class(es);
  - acknowledge time;
  - throughput (proposed);
  - residual error rate (proposed);
  - priority (proposed);
  - transit delay (proposed);
  - reassignment time;
  - user data.
- b) CC TPDU;
  - CDT;
  - DST-REF;
  - SRC-REF;
  - CLASS and OPTIONS (selected);
  - calling TSAP-ID;
  - called TSAP-ID;
  - TPDU size (selected);

- protection parameter;
- checksum;
- additional option selection (selected);
- acknowledge time;
- throughput (selected);
- residual error rate (selected);
- priority (selected);
- transit delay (selected);
- user data.

#### 6.5.4 Procedure

A transport connection is established by means of one transport entity (the *initiator*) transmitting a CR TPDU to the other transport entity (the *responder*), which replies with a CC TPDU. Before sending the CR TPDU, the initiator assigns the transport connection being created to one (or more if the splitting procedure is being used) network connection(s). It is this set of network connections over which the TPDU's are sent.

*Note* – Even if the initiator assigns the transport connection to more than one network connection, all the CR TPDU's (if repeated) or DR TPDU's with DST-REF set to zero which are sent prior to the receipt of the CC TPDU, shall be sent on the same network connection, unless an N-DISCONNECT indication is received. (This is necessary because the remote entity may not support Class 4 and therefore may not recognize splitting.) If the initiator has made other assignments, it will use them only after receipt of a Class 4 CC TPDU (see also the splitting procedure § 6.23).

During this exchange, all information and parameters needed for the transport entities to operate shall be exchanged or negotiated.

*Note 1* – The transport protocol identification procedures specified in Annex B may need to be considered in conjunction with this procedure.

*Note 2* – Except in Class 4, it is suggested that the initiator start an optional timer TS1 at the time the CR TPDU is sent. This timer should be stopped when the connection is considered as accepted or refused or unsuccessful. If the timer expires, the initiator should reset or disconnect the network connection and, in Classes 1 and 3, freeze the reference (see § 6.18). For all other transport connection(s) multiplexed on the same network connection, the procedures for reset or disconnect as appropriate should be followed.

When an unexpected duplicated CR TPDU is received (with Class 4 as preferred class), it shall be ignored in Classes 0, 1, 2 and 3 and a CC TPDU shall be returned in Class 4.

After receiving the CC TPDU for a class which includes the procedure for retention until acknowledgment of TPDU's, the initiator shall acknowledge the CC TPDU as defined in Table 5/X.224 (see § 6.13).

When the network expedited variant of expedited data transfer (see § 6.11) has been agreed (possible in Class 1 only), the responder shall not send an ED TPDU before the CC TPDU is acknowledged.

The following information is exchanged:

- a) *references* – Each transport entity chooses a reference to be used by the peer entity which is 16 bits long and which is arbitrary except for the following restrictions:
  - 1) it shall not already be in use or frozen (see § 6.18),
  - 2) it shall not be zero.

This mechanism is symmetrical and provides identification of the transport connection independent of the network connection. The range of references used for transport connections, in a given transport entity, is a local matter.

- b) *calling and called TSAPs-IDs* (optional) – Indicate the calling and called transport service access points. When either network address unambiguously defines the transport address, this information may be omitted.
- c) *initial credit* – Only relevant for classes which include the explicit flow control function.
- d) *user data* – Not available if Class 0 is the preferred class (see Note). Up to 32 octets in other classes.

*Note* – If Class 0 is a valid response according to Table 3/X.224, inclusion of user data in the CR TPDU may cause the responding entity to refuse the connection (e.g. if it only supports Class 0).

- e) *acknowledgment time* – Only in Class 4.
- f) *checksum parameter* – Only in Class 4.
- g) *protection parameter* – This parameter and its semantics are user defined.

TABLE 3/X.224

Valid responses corresponding to preferred and any alternative class proposed in the CR TPDU

Preferred class	Alternative class					
	0	1	2	3	4	none
0	not valid	not valid	not valid	not valid	not valid	class 0
1	class 1 or 0	class 1 or 0	not valid	not valid	not valid	class 1 or 0
2	class 2 or 0	not valid	class 2	not valid	not valid	class 2
3	class 3, 2 or 0	class 3, 2, 1 or 0	class 3 or 2	class 3 or 2	not valid	class 3 or 2
4	class 4, 2 or 0	class 4, 2, 1 or 0	class 4 or 2	class 4, 3 or 2	class 4 or 2	class 4 or 2

*Note 1* – The valid responses indicated in the table result from both *explicit* negotiation, whereby each of the classes proposed is a valid response and *implicit* negotiation, whereby:

- if class 3 or 4 is proposed then class 2 is a valid response;
- if class 1 is proposed then class 0 is a valid response.

*Note 2* – Negotiation from class 2 to class 1 and from any class to a higher-numbered class is not valid.

*Note 3* – Redundant combinations of proposed classes can occur (e.g. due to the implicit negotiation rules). These are not considered as protocol errors.

The following negotiations take place:

- h) *protocol class* – The initiator shall propose a preferred class and any number of alternative classes which permit a valid response as defined in Table 3/X.224. The initiator should assume when it sends the CR TPDU that its preferred class will be agreed to, and commence the procedures associated with that class, except that if Class 0 or Class 1 is an alternative class, multiplexing shall not commence until a CC TPDU selecting the use of Classes 2, 3 or 4 has been received.

*Note* – This means, for example, that when the preferred class includes resynchronization (see § 6.14), the resynchronization will occur if a reset is signalled during connection establishment.

The responder shall select one class defined in Table 3/X.224 as a valid response corresponding to the preferred class and to the class(es), if any, contained in the alternative class parameter of the CR TPDU. It shall indicate the selected class in the CC TPDU and shall follow the procedures for the selected class.

If the preferred class is not selected, then on receipt of the CC TPDU, the initiator shall adjust its operation according to the procedures of the selected class.

- i) *TPDU size* – The initiator may propose a maximum size for TPDU's, and the responder may accept this value or respond with any value between 128 and the proposed value in the set of values available for the class (see § 13.3.4 b)).

*Note* – The length of the CR TPDU does not exceed 128 octets (see § 13.3).

- j) *normal or extended format* – Either normal or extended is available. When extended is used, this applies to CDT, TPDU-NR, ED-TPDU-NR, YR-TU-NR and YR-EDTU-NR parameters.
- k) *checksum selection* – This defines whether or not TPDU's of the connection are to include a checksum.
- l) *quality of service parameters* – This defines the throughput, transit delay, priority, and residual error rate.  
*Note* – The transport service defines transit delay as requiring a previously stated average TSDU size as a basis for any specification. The protocol, as specified in § 13.3.4, l), uses a value of 128 octets. Conversion to and from specifications based upon some other value is a local matter.
- m) *the non-use of explicit flow control* in Class 2.
- n) *the use of network receipt confirmation and network expedited* when Class 1 is to be used.
- o) *the use of expedited data transfer service* – This allows both TS-users to negotiate the use or non-use of the expedited data transport service as defined in the transport service definition [2].

The following information is set only in the CR TPDU:

- p) *version number* – This defines the version of the transport protocol used for this connection.
- q) *reassignment time parameter* – This indicates the time for which the initiator will persist in following the reassignment after failure procedure.

The negotiation rules for the options are such that the initiator may propose either to use or not to use the option. The responder may either accept the proposed choice or select an alternative choice as defined in Table 4/X.224.

When a parameter [which is valid for the proposed class(es)] is absent and a default value is defined in this Recommendation, this is equivalent to the presence of the parameter with the default value.

In Class 2, whenever a transport entity requests or agrees to the transport expedited data transfer service or to the use of extended formats, it shall request or agree (respectively) to the use of explicit flow control.

## 6.6 *Connection refusal*

### 6.6.1 *Purpose*

The connection refusal procedure is used in all classes when a transport entity refuses a transport connection in response to a CR TPDU.

TABLE 4/X.224

## Negotiation of options during connection establishment

Option	Proposal made by the initiator	Valid selection by the responder
Transport expedited data transfer service (Classes 1, 2, 3, 4 only)	Yes No	Yes or No No
Use of receipt confirmation (Class 1 only)	Yes No	Yes or No No
Use of network expedited variant (Class 1 only)	Yes No	Yes or No No
Non-use of checksums (Class 4 only)	Yes No	Yes or No No
Non-use of explicit flow control (Class 2 only)	Yes No	Yes or No No
Use of extended format (Class 2, 3 4 only)	Yes No	Yes or No No

*Note* – Table 4/X.224 defines the procedures for negotiation of options. This negotiation has been designed such that if the initiator proposes the mandatory implementation option specified in § 14, the responder has to accept use of this option over the transport connection, except for the use of the transport expedited data transfer service which may be rejected by the TS-user. If the initiator proposes a non-mandatory implementation option, the responder is entitled to select use of the mandatory implementation option for use over the transport connection.

### 6.6.2 TPDUs and parameters used

The procedure makes use of the following TPDUs and parameters:

- a) DR TPDU:
  - SRC-REF;
  - reason;
  - user data.
- b) ER TPDU:
  - reject cause;
  - invalid TPDU.

### 6.6.3 Procedure

If a transport connection cannot be accepted, the responder shall respond to the CR TPDU with a DR TPDU. The reason shall indicate why the connection was not accepted. The source reference field in the DR TPDU shall be set to zero to indicate an unassigned reference.

If a DR TPDU is received, the initiator shall regard the connection as released.

The responder shall respond to an invalid CR TPDU by sending an ER or DR TPDU. If an ER TPDU is received in response to a CR TPDU, the initiator shall regard the connection as released.

*Note 1* – When the invalid CR TPDU can be identified as having Class 0 as the preferred class, it is suggested to respond with an ER TPDU. For all other invalid CR TPDUs, either an ER TPDU or DR TPDU may be sent.

*Note 2* – If the optional supervisory timer TS1 has been set for this connection, then the initiator should stop the timer on receipt of the DR or ER TPDU.

## 6.7 *Normal release*

### 6.7.1 *Purpose*

The release procedure is used by a transport entity in order to terminate a transport connection. The implicit variant is used only in Class 0. The explicit variant is used in Classes 1, 2, 3 and 4.

*Note 1* – When the implicit variant is used (i.e. in Class 0), the lifetime of the transport connection is directly correlated with the lifetime of the network connection.

*Note 2* – The use of the explicit variant of the release procedure enables the transport connection to be released independently of the underlying network connection.

### 6.7.2 *Network service primitives*

The procedure makes use of the following network service primitives:

- a) N-DISCONNECT (implicit variant only),
- b) N-DATA.

### 6.7.3 *TPDUs and parameters used*

The procedure makes use of the following TPDUs and parameters:

- a) DR TPDU:
  - reason;
  - user data;
  - SRC-REF;
  - DST-REF.
- b) DC TPDU.

### 6.7.4 *Procedure for implicit variant*

In the implicit variant, either transport entity disconnects a transport connection by disconnecting the network connection to which it is assigned. When a transport entity receives an N-DISCONNECT indication, this should be considered as the release of the transport connection.

### 6.7.5 *Procedure for explicit variant*

When the release of a transport connection is to be initiated, a transport entity:

- a) if it has previously sent or received a CC TPDU (see Note 1), shall:
  - 1) send a DR TPDU;
  - 2) discard all subsequently received TPDUs other than a DR or DC TPDU;
  - 3) consider the transport connection released on receipt of a DR or DC TPDU;
- b) if a) is not applicable, it shall:
  - 1) for classes other than Class 4, wait for acknowledgment of the outstanding CR TPDU; if it receives a CC TPDU, it shall follow the procedure in § 6.7.5 a);
  - 2) for Class 4, either send a DR TPDU with a zero value in the DST-REF field, or follow the procedure in § 6.7.5 b) 1).

In the former case, receipt of a CC TPDU specifying Class 4 will be ignored. Receipt of a CC TPDU with another class will be processed as follows: if the class is 0, the network connection shall be disconnected; otherwise, a DR TPDU with the DST-REF field set to the value of the SRC-REF field of the received CC TPDU shall be sent and the release procedure of the class is continued.

A transport entity that receives a DR TPDU shall:

- c) if it has previously sent a DR TPDU for the same transport connection, consider the transport connection released;
- d) if it has previously sent a CR TPDU that has not been acknowledged by a CC TPDU, consider the connection refused (see § 6.6);

If the SRC-REF is not zero, a DC TPDU shall be sent using the SRC-REF as the DST-REF.

*Note* – In this case, the DR TPDU has been associated regardless of its SRC-REF field (see § 6.9.4).

- e) if c) and d) are not applicable, send a DC TPDU and consider the transport connection released. If the received DR has the DST-REF field set to zero, then a DC with SRC-REF set to zero shall be sent, regardless of the local reference. If the entity receiving such a DR TPDU has previously decided to negotiate down the class, this entity is always entitled to consider such a DR TPDU as spurious. Since no association has been made, the transport connection is not released at the responder side but the CC TPDU, when sent, will be answered by a DR TPDU (spurious CC TPDU).

*Note 1* – This requirement ensures that the transport entity is aware of the remote reference for the transport connection.

*Note 2* – When the transport connection is considered as released, the local reference is either available for re-use or is frozen (see § 6.18).

*Note 3* – After the release of a transport connection, the network connection can be released or retained to enable its re-use for the assignment of other transport connections (see § 6.1).

*Note 4* – Except in Class 4, it is suggested that if a transport entity does not receive acknowledgment of a DR TPDU within time TS2, it should either reset or disconnect the network connection, and freeze the reference when appropriate (see § 6.18). For all other transport connection(s) multiplexed on the same network connection, the procedures for reset or disconnect as appropriate should be followed.

*Note 5* – When a transport entity is waiting for a CC TPDU before sending a DR TPDU and the network connection is reset or released, it should consider the transport connection released and, in classes other than Classes 0 and 2, freeze the reference (see § 6.18).

## 6.8 *Error release*

### 6.8.1 *Purpose*

This procedure is used only in Classes 0 and 2 to release a transport connection on the receipt of a N-DISCONNECT or N-RESET indication.

### 6.8.2 *Network service primitives*

The procedure makes use of the following service primitives:

- a) N-DISCONNECT indication;
- b) N-RESET indication.

### 6.8.3 *Procedure*

When, on the network connection to which a transport connection is assigned, an N-DISCONNECT or N-RESET indication is received, both transport entities shall consider that the transport connection is released, and so inform the TS-users.

*Note* – In other classes, since error recovery is used, the receipt of an N-RESET indication or N-DISCONNECT indication will result in the invocation of the error recovery procedure.

## 6.9 *Association of TPDUs with transport connections*

### 6.9.1 *Purpose*

This procedure is used in all classes to interpret a received NSDU as TPDU(s) and, if possible, to associate each such TPDU with a transport connection.

### 6.9.2 *Network service primitives*

This procedure makes use of the following network service primitives:

- a) N-DATA indication;
- b) N-EXPEDITED DATA indication.

### 6.9.3 *TPDUs and parameters used*

This procedure makes use of the following TPDUs and parameters:

- a) in any TPDU except: CR TPDU; DT TPDU in Classes 0 or 1; AK TPDU in Class 1:
  - DST-REF.
- b) CR, CC, DR and DC TPDUs:
  - SRC-REF.
- c) DT TPDU in Classes 0 or 1 and AK TPDU in Class 1.

### 6.9.4 *Procedures*

#### 6.9.4.1 *Identification of TPDUs*

If the received NSDU or expedited NSDU cannot be decoded (i.e. does not contain one or more correct TPDUs) or is corrupted (i.e. contains a TPDU with a wrong checksum) then the transport entity shall:

- a) if the network connection on which the error is detected has a Class 0 or Class 1 transport connection assigned to it, then treat as a protocol error (see § 6.22) for that transport connection;
- b) otherwise:
  - 1) if the NSDU can be decoded but contains corrupted TPDUs, discard the TPDUs (Class 4 only) and optionally apply § 6.9.4.1 b) 2);
  - 2) if the NSDU cannot be decoded, issue an N-RESET (or N-DISCONNECT) request for the network connection and for all of the transport connections assigned to this network connection (if any), apply the procedures defined for handling of network signalled reset or disconnect.

If the NSDU can be decoded and is not corrupted, the transport entity shall:

- c) if the network connection on which the NSDU was received has a Class 0 transport connection assigned to it, then consider the NSDU as forming one TPDU and associate the TPDU with the transport connection (see § 6.9.4.2);
- d) otherwise, invoke the separation procedures and for each of the individual TPDUs in the order in which they appear in the NSDU apply the procedure defined in § 6.9.4.2.

#### 6.9.4.2 *Association of individual TPDUs*

If the received TPDU is a CR TPDU, then, if it is a duplicate as recognized by using the NSAPs of the network connection, and the SRC-REF parameter, then it is associated with the transport connection created by the original copy of the CR TPDU; otherwise, it is processed as requesting the creation of a new transport connection.

If the received TPDU is a DT TPDU and the network connection has a Class 0 or Class 1 transport connection assigned to it, or an AK TPDU where a Class 1 transport connection is assigned, then the TPDU is associated with the transport connection.

Otherwise, the DST-REF parameter of the TPDU is used to identify the transport connection. The following cases are distinguished:

- a) If the DST-REF is not allocated to a transport connection, the transport entity shall respond on the same network connection with a DR TPDU if the TPDU is a CC TPDU, with a DC TPDU if the TPDU is a DR TPDU and shall discard the TPDU if neither a DR TPDU nor CC TPDU. No association with a transport connection is made.

*Note* – If the DR TPDU is carrying an SRC-REF field set to zero, then no DC TPDU shall be sent.

- b) If the DST-REF is allocated to a connection, but the TPDU is received on a network connection to which the connection has not been assigned, then there are three cases:
  - 1) if the transport connection is of Class 4 and if the TPDU is received on a network connection with the same pair of NSAPs as that of the CR TPDU, then the TPDU is associated with this transport connection and considered as performing assignment;
  - 2) if the transport connection is not assigned to any network connection (waiting for reassignment after failure) and if the TPDU is received on a network connection with the same pair of NSAPs as that of the CR TPDU, then the association with that transport connection is made except in the case of DC, DR and CC TPDUs which are respectively described in § 6.9.4.2 c), d) and e);

- 3) otherwise, the TPDU is considered as having a DST-REF not allocated to a transport connection [case a)].
- c) If the TPDU is a DC TPDU, then it is associated with the transport connection to which the DST-REF is allocated, unless the SRC-REF is not the expected one, in which case the DC TPDU is discarded.
- d) If the TPDU is a DR TPDU then there are four cases:
  - 1) if the SRC-REF is not as expected, then a DC TPDU with a DST-REF equal to the SRC-REF of the received DR TPDU is sent back and no association is made;
  - 2) if a CR TPDU is unacknowledged, then the DR TPDU is associated with the transport connection, regardless of the value of its SRC-REF parameter;
  - 3) if the transport entity implements Class 4 and if the DST-REF is zero and there is an unacknowledged CC TPDU or T-CONNECT response is awaited, then the DR TPDU shall be associated with the transport connection holding the SRC-REF as the remote reference;
  - 4) otherwise, the DR TPDU is associated with the transport connection identified by the DST-REF parameter.
- e) If the TPDU is a CC TPDU whose DST-REF parameter identifies an open connection (one for which a CC TPDU has been previously received), and the SRC-REF in the CC TPDU does not match the remote reference, then a DR TPDU is sent back with DST-REF equal to the SRC-REF of the received CC TPDU and no association is made.
- f) If none of the above cases apply, then the TPDU is associated with the transport connection identified by the DST-REF parameter.

## 6.10 *Data TPDU numbering*

### 6.10.1 *Purpose*

Data TPDU numbering is used in Classes 1, 2 (except when the non-use of explicit flow control option is selected), 3 and 4. Its purpose is to enable the use of recovery, flow control and resequencing functions.

### 6.10.2 *TPDUs and parameters used*

This procedure makes use of the following TPDU and parameter:

DT TPDU;

– TPDU-NR.

### 6.10.3 *Procedure*

A transport entity shall allocate the sequence number zero to the TPDU-NR of the first DT TPDU which it transmits for a transport connection. For subsequent DT TPDUs sent on the same transport connection, the transport entity shall allocate a sequence number one greater than the previous one.

When a DT TPDU is retransmitted, the TPDU-NR parameter shall have the same value as in the first transmission of that DT TPDU.

Modulo  $2^7$  arithmetic shall be used when normal formats have been selected and modulo  $2^{31}$  arithmetic shall be used when extended formats have been selected. In this Recommendation, the relationships “greater than” and “less than” apply to a set of contiguous TPDU numbers whose range is less than the modulus and whose starting and finishing numbers are known. The term “less than” means “occurring sooner in the window sequence” and the term “greater than” means “occurring later in the window sequence”.

## 6.11 *Expedited data transfer*

### 6.11.1 *Purpose*

Expedited data transfer procedures are selected during connection establishment. The network normal data variant may be used in Classes 1, 2, 3 and 4. The network expedited variant is only used in Class 1.

### 6.11.2 *Network service primitives*

The procedure makes use of the following network service primitives:

- a) N-DATA;
- b) N-EXPEDITED DATA.

### 6.11.3 *TPDUs and parameters used*

The procedure makes use of the following TPDUs and parameters:

- a) ED TPDU:
  - ED-TPDU-NR.
- b) EA TPDU:
  - YR-EDTU-NR.

### 6.11.4 *Procedure*

The TS-user data parameter of each T-EXPEDITED DATA request shall be conveyed as the data field of an Expedited Data (ED) TPDU.

Each ED TPDU received shall be acknowledged by an Expedited Acknowledge (EA) TPDU.

No more than one ED TPDU shall remain unacknowledged at any time for each direction of a transport connection.

An ED TPDU with a zero length data field shall be treated as a protocol error.

*Note 1* – The network normal data variant is used, except when the network expedited variant (available in Class 1 only), has been agreed, in which case ED and EA TPDUs are conveyed in the data fields of N-EXPEDITED DATA primitives (see § 6.2.3).

*Note 2* – No TPDUs can be transmitted using network expedited until the CC TPDU becomes acknowledged, to prevent the network expedited from overtaking the CC TPDU.

## 6.12 *Reassignment after failure*

### 6.12.1 *Purpose*

The reassignment after failure procedure is used in Classes 1 and 3 to commence recovery from an NS-provider signalled disconnect.

### 6.12.2 *Network service primitives*

The procedure uses the following network service primitive:

N-DISCONNECT indication.

### 6.12.3 *Procedure*

When an N-DISCONNECT indication is received for the network connection to which a transport connection is assigned, the initiator shall apply one of the following alternatives:

- a) if the TTR timer has not already run out and no DR TPDU is retained, then:
  - 1) assign the transport connection to a different network connection (see § 6.1) and start its TTR timer if not already started;
  - 2) while waiting for the completion of assignment if:
    - an N-DISCONNECT indication is received, repeat the procedure from § 6.12.3 a),
    - the TTR timer expires, begin procedure § 6.12.3 b);
  - 3) when reassignment is completed, begin resynchronization (see § 6.14) and:
    - if a valid TPDU is received as the result of the resynchronization, stop the TTR timer, or
    - if TTR runs out, wait for the next event, or
    - if an N-DISCONNECT indication is received, then begin either procedure § 6.12.3 a) or § 6.12.3 b) depending on the TTR timer.

*Note* – After TTR expires and while waiting for the next event, it is suggested that the initiator set a timer with a value equal to TWR. If this timer expires before the next event, the initiator should begin the procedure in § 6.12.3 b);

- b) if the TTR timer has run out, consider the transport connection as released and freeze the reference (see § 6.18);
- c) if a DR TPDU is retained and the TTR timer has not run out, then follow the actions in either § 6.12.3 a) or § 6.12.3 b).

The responder shall start its TWR timer if not already started. The arrival of the first TPDU related to the transport connection (because of resynchronization by the initiator) completes the reassignment after failure procedure. The TWR timer is stopped and the responder shall continue with resynchronization (see § 6.14). If reassignment does not take place within this time, the transport connection is considered released and the reference is frozen (see § 6.18).

If reassignment occurs successfully, both transport entities shall continue with resynchronization.

*Note* – The transport protocol identification procedures specified in Annex B may need to be considered in conjunction with this procedure.

#### 6.12.4 *Timers*

The reassignment after failure procedure uses two timers:

- a) TTR, the time to try reassignment/resynchronization timer;
- b) TWR, the time to wait for reassignment/resynchronization timer.

The TWR timer is used by the initiator. Its value shall not exceed two minutes minus the sum of the maximum disconnect propagation delay and the maximum transit delay of the network connections (see Note 1). The value for the TTR timer may be indicated in the CR TPDU.

The TWR timer is used by the responder. If the reassignment time parameter is present in the CR TPDU, the TWR timer value shall be greater than the sum of the TTR timer plus the maximum disconnect propagation delay plus the maximum transit delay of the network connections.

If the reassignment time parameter is not present in the CR TPDU, a default value of 2 minutes shall be used for the TWR timer.

*Note 1* – Provided that the required quality of service is met, TTR may be set to zero (i.e., no reassignment), for example if the rate of NS-provider generated disconnects is very low.

*Note 2* – Inclusion of the reassignment time parameter in the CR TPDU allows the responder to use a TWR value of less than 2 minutes.

*Note 3* – If the optional TS1 and TS2 timers are used, it is suggested:

- a) to stop TS1 or TS2 if running when TTR or TWR is started;
- b) to restart TS1 or TS2 if necessary when the corresponding TPDU (CR TPDU or DR TPDU respectively) is repeated;
- c) to select for TS1 and TS2 values greater than TTR.

#### 6.13 *Retention until acknowledgment of TPDUs*

##### 6.13.1 *Purpose*

The retention until acknowledgment of TPDUs procedure is used in Classes 1, 3 and 4 to enable and minimize retransmission after possible loss of TPDUs.

The confirmation of receipt variant is used only in Class 1 when it has been agreed during connection establishment (see Note).

The AK variant is used in Classes 3 and 4 and also in Class 1 when the confirmation of receipt variant has not been agreed during connection establishment.

*Note* – Use of confirmation of receipt variant depends on the availability of the network layer receipt confirmation service and the expected cost reduction.

### 6.13.2 *Network service primitives*

The procedure uses the following network service primitives:

- a) N-DATA;
- b) N-DATA ACKNOWLEDGE.

### 6.13.3 *TPDUs and parameters used*

The procedure uses the following TPDUs and parameters:

- a) CR, CC, DR and DC TPDUs.
- b) RJ and AK TPDUs:
  - YR-TU-NR.
- c) DT TPDUs:
  - TPDU-NR.
- d) ED TPDUs:
  - ED-TPDU-NR.
- e) EA TPDUs:
  - YR-EDTU-NR.

### 6.13.4 *Procedure*

Copies of the following TPDUs shall be retained upon transmission to permit their later retransmission:

CR, CC, DR, DT and ED TPDUs

except that if a DR TPDUs is sent in response to a CR TPDUs, there is no need to retain a copy of the DR TPDUs.

A copy of each of these TPDUs shall be retained until:

- a) it is acknowledged, as specified in Table 5/X.224; or
- b) the transport connection is released.

In the confirmation of receipt variant, applicable only in Class 1, transport entities shall:

- a) set the confirmation request parameter only if the data parameter contains a CC or DT TPDUs (see Notes 1 and 2); and
- b) issue an N-DATA ACKNOWLEDGE request when it receives an N-DATA indication with the confirmation request parameter set.

*Note 1* – It is a local matter for each transport entity to decide which N-DATA requests should have the confirmation request parameter set. This decision will normally be related to the amount of storage available for retained copies of the DT TPDUs.

*Note 2* – Use of the confirmation request parameter may affect the quality of network service.

TABLE 5/X.224

## Acknowledgement of TPDU

Retained TPDU	Variant	Retained until acknowledged by
CR	both	CC, DR or ER TPDU.
DR	both	DC or DR (in case of collision) TPDU.
CC	Confirmation of receipt	N-DATA ACKNOWLEDGE indication, RJ, DT, ED or EA TPDU.
CC	AK	RJ, DT, AK, ED or EA TPDU.
DT	Confirmation of receipt	N-DATA ACKNOWLEDGE indication corresponding to an N-DATA request which conveyed, or came after, the DT TPDU.
DT	AK	AK or RJ TPDU for which the YR-TU-NR is greater than TPDU-NR in the DT TPDU.
ED	both	EA TPDU for which the YR-EDTU-NR is equal to the ED-TPDU-NR in the ED TPDU.

6.14 *Resynchronization*6.14.1 *Purpose*

The resynchronization procedures are used in Classes 1 and 3 to restore the transport connection to normal after a reset or during reassignment after failure according to § 6.12.

6.14.2 *Network service primitives*

The procedure makes use of the following network service primitive:

N-RESET indication.

6.14.3 *TPDUs and parameters used*

The procedure uses the following TPDU and parameters:

- a) CR, DR, CC and DC TPDU.
- b) RJ TPDU:
  - YR-TU-NR.
- c) DT TPDU:
  - TPDU-NR.
- d) ED TPDU:
  - ED-TPDU-NR.
- e) EA TPDU:
  - YR-EDTU-NR.

#### 6.14.4 Procedure

A transport entity which is notified of the occurrence of an N-RESET or which is performing reassignment after failure according to § 6.12 shall carry out the active resynchronization procedures (see § 6.14.4.1) *unless* any of the following hold:

- a) the transport entity is the responder. In this case, the passive resynchronization procedure shall be carried out (see § 6.14.4.2);
- b) the transport entity has elected not to reassign (see § 6.12.3 c)). In this case, no resynchronization takes place.

##### 6.14.4.1 Active resynchronization procedures

The transport entity shall carry out one of the following actions:

- a) if the TTR timer has been previously started and has run out (i.e. no valid TPDU has been received), the transport connection is considered as released and the reference is frozen (see § 6.18);
- b) otherwise, the TTR timer shall be started (unless it is already running) and the first applicable one of the following actions shall be taken:
  - 1) if a CR TPDU is unacknowledged, then the transport entity shall retransmit it;
  - 2) if a DR TPDU is unacknowledged, then the transport entity shall retransmit it;
  - 3) otherwise, the transport entity shall carry out the data resynchronization procedures (§ 6.14.4.3).

The TTR timer is stopped when a valid TPDU is received.

##### 6.14.4.2 Passive resynchronization procedures

The transport entity shall not send any TPDU until a TPDU has been received. The transport entity shall start its TWR timer if it was not already started (due to a previous N-DISCONNECT or N-RESET indication). If the timer runs out prior to the receipt of a valid TPDU which commences resynchronization (i.e. CR or DR or ED or RJ TPDU), the transport connection is considered as released and the reference is released (see § 6.18).

When a valid TPDU is received, the transport entity shall stop its TWR timer and carry out the appropriate one of the following actions, depending on the TPDU:

- a) if it is a DR TPDU, then the transport entity shall send a DC TPDU;
- b) if it is a repeated CR TPDU (see Note 1) then the transport entity shall carry out the action which is appropriate from the following:
  - 1) if a CC TPDU has already been sent, and acknowledged: treat as a protocol error;
  - 2) if a DR TPDU is unacknowledged (whether or not a CC TPDU is unacknowledged): retransmit the DR TPDU, but setting the source reference to zero;
  - 3) if the T-CONNECT response has not yet been received from the user: take no action;
  - 4) otherwise: retransmit the CC TPDU followed by any unacknowledged ED TPDU (see Note 2) and any DT TPDU.

*Note 1* – A repeated CR can be identified by being on a network connection with the appropriate network addresses and having a correct source reference.

*Note 2* – The transport entity should not use network expedited until the CC is acknowledged (see § 6.5). This rule prevents the network expedited from overtaking the CC TPDU;

- c) if it is an RJ or ED TPDU, then one of the following actions shall be taken:
  - 1) if a DR TPDU is unacknowledged, then the transport entity shall retransmit it;
  - 2) if a CC TPDU is unacknowledged, the RJ or ED TPDU shall be considered as acknowledging the CC TPDU, and the transport entity shall carry out the data resynchronization procedures (§ 6.14.4.3);
  - 3) if a CC TPDU was never sent, the RJ or ED TPDU should be considered as a protocol error;
  - 4) otherwise, the transport entity shall carry out the data resynchronization procedures (§ 6.14.4.3.)

#### 6.14.4.3 *Data resynchronization procedures*

The transport entity shall carry out the following actions in the following order:

- a) (re)transmit any ED TPDU which is unacknowledged;
- b) transmit an RJ TPDU with YR-TU-NR field set to the TPDU-NR of the next expected DT TPDU;
- c) wait for the next TPDU from the other transport entity, unless it has already been received. If a DR TPDU is received, the transport entity shall send a DC TPDU, freeze the reference, inform the TS-user of the disconnection and take no further action [i.e. it shall not follow the procedures in § 6.14.4.3 d)]. If an RJ TPDU is received, the procedures of § 6.14.4.3 d) shall be followed. If an ED TPDU is received, the procedures as described in § 6.11 shall be followed. If it is a duplicated ED TPDU, the transport entity shall acknowledge it with an EA TPDU, discard the duplicated ED TPDU and wait again for the next TPDU;
- d) (re)transmit any DT TPDUs which are unacknowledged, subject to any applicable flow control procedures (see Note).

*Note* – The RJ TPDU may have reduced the credit.

#### 6.15 *Multiplexing and demultiplexing*

##### 6.15.1 *Purpose*

The multiplexing and demultiplexing procedures are used in Classes 2, 3 and 4 to allow several transport connections to share a network connection at the same time.

##### 6.15.2 *TPDUs and parameters used*

The procedure makes use of the following TPDUs and parameters:

CC, DR, DC, DT, AK, ED, EA, RJ and ER TPDUs:

- DST-REF.

##### 6.15.3 *Procedure*

The transport entities shall be able to send and receive on the same network connection TPDUs belonging to different transport connections.

*Note 1* – When performing demultiplexing, the transport connection to which the TPDUs apply is determined by the procedures defined in § 6.9.

*Note 2* – Multiplexing allows the concatenation of TPDUs belonging to different transport connections to be transferred in the same N-DATA primitive (see § 6.4).

#### 6.16 *Explicit flow control*

##### 6.16.1 *Purpose*

The explicit flow control procedure is used in Classes 2, 3 and 4 to regulate the flow of DT TPDUs independently of the flow control in the other layers.

##### 6.16.2 *TPDUs and parameters used*

The procedure makes use of the following TPDUs and parameters:

- a) CR, CC, AK and RJ TPDUs:
  - CDT.
- b) DT TPDU:
  - TPDU-NR.
- c) AK TPDU:
  - YR-TU-NR;
  - subsequence number;
  - flow control confirmation.

- d) RJ TPDU:
  - YR-TU-NR.

### 6.16.3 Procedure

The procedures differ in different classes. They are defined in the sections specifying the separate classes.

### 6.17 Checksum

#### 6.17.1 Purpose

The checksum procedure is used to detect corruption of TPDU's by the NS-provider.

*Note* – Although a checksum algorithm has to be adapted to the type of errors expected on the network connection, at present only one algorithm is defined.

#### 6.17.2 TPDU's and parameters used

The procedure uses the following TPDU's and parameters:

All TPDU's:

- checksum.

#### 6.17.3 Procedure

The checksum is used only in Class 4. It shall always be used for the CR TPDU, and shall be used for all other TPDU's unless the non-use of the checksum was selected during connection establishment.

The sending transport entity shall transmit TPDU's with the checksum parameter set such that the following formulae are satisfied:

$$\sum_{i=1}^L a_i \equiv 0 \text{ (modulo 255)}$$

$$\sum_{i=1}^L ia_i \equiv 0 \text{ (modulo 255)}$$

where

- $i$  = number (i.e. position) of an octet within the TPDU (see § 13.2).
- $a_i$  = value of octet in position  $i$ .
- $L$  = length of TPDU in octets.

A transport entity which receives a TPDU for a transport connection for which the use of checksum has been agreed and which does not satisfy the above formulae shall discard the TPDU (see also Note 2).

When a spurious TPDU is received and an answer is to be sent the transport entity shall:

- a) if it supports the checksum algorithm and the received TPDU contains a checksum parameter, include a checksum parameter in the answering TPDU; or
- b) in all other cases, not include a checksum parameter in the answering TPDU.

An entity not supporting checksum may always suppose that a CR TPDU with Class 4 proposed is correct and therefore negotiate down to a class lower than 4.

*Note 1* – An efficient algorithm for determining the checksum parameters is given in Appendix I.

*Note 2* – If the checksum is incorrect, it is not possible to know with certainty to which transport connection the TPDU is related; thus, further action may be taken for all the transport connections assigned to the network connection (see § 6.9).

*Note 3* – The checksum proposed is easy to calculate and so will not impose a heavy burden on implementations. However, it will not detect insertion or loss of leading or trailing zeros and will not detect some octets misordering.

*Note 4* – When a TPDU is received on a network connection, it is never possible to know with certainty that only Class 4 transport connections use this network connection because it may be a TPDU performing reassignment.

Therefore the only way to check the validity is the following:

- a) if the network connection is used by a Class 0 or Class 1 transport connection, there is no checksum;
- b) examine the TPDU code;
- c) deduce the fixed part length;
- d) from LI, deduce the variable part;
- e) go through parameters and if the checksum parameter is found, then verify it;
- f) if it is incorrect, then assume that transport connection is Class 4 and drop it;
- g) if it is correct, then associate the TPDU with a transport connection; if the transport connection uses the checksum, it is correct; else, it must be considered as a protocol error.

## 6.18 *Frozen references*

### 6.18.1 *Purpose*

This procedure is used in order to prevent re-use of a reference while TPDU's associated with the old use of the reference may still exist.

### 6.18.2 *Procedure*

When a transport entity determines that a particular connection is released, it shall place the reference which it has allocated to the connection in a frozen state according to the procedures of the class. While frozen, the reference shall not be re-used.

*Note* – The frozen reference procedure is necessary because retransmission or misordering can cause TPDU's bearing a reference to arrive at an entity after it has released the connection for which it allocated the reference. Retransmission, for example, can arise when the class includes either resynchronization (see § 6.14) or retransmission on timeout (see § 6.19).

#### 6.18.2.1 *Procedure for Classes 0 and 2*

This Recommendation does not specify frozen reference procedures for classes 0 and 2.

*Note* – For consistency with the other classes, references may be frozen as a local matter.

#### 6.18.2.2 *Procedure for Classes 1 and 3*

The frozen reference procedure is used except in the following cases (see Note 1):

- a) when the transport entity receives a DC TPDU in response to a DR TPDU which it has sent (see Note 2);
- b) when the transport entity sends a DR TPDU in response to a CR TPDU which it has received (see Note 3);
- c) when the transport entity has considered the connection to be released after the expiration of the TWR timer (see Note 4);
- d) when the transport entity receives a DR or ER TPDU in response to a CR TPDU which it has sent.

The period of time for which the reference remains frozen shall be greater than the TWR time.

*Note 1* – However, even in these cases, for consistency, freezing the reference may be done as a local decision.

*Note 2* – When the DC TPDU is received, it is certain that the other transport entity considers the connection released.

*Note 3* – When the DR or ER TPDU is sent, the peer transport entity has not been informed of any reference assignment and thus cannot possibly make use of a reference (this includes the case where a CC TPDU was sent, but was lost).

*Note 4* – In § 6.18.2 c), the transport entity has already effectively frozen the reference for an adequate period.

#### 6.18.2.3 *Procedure for Class 4*

The frozen reference procedure shall be used in Class 4. The period for which the reference remains frozen shall be greater than  $L$  (see § 12.2.1.1.6).

6.19 *Retransmission on timeout*

6.19.1 *Purpose*

The procedure is used in Class 4 to cope with unsignalled loss of TPDU by the NS-provider.

6.19.2 *TPDUs used*

The procedure makes use of the following TPDU:

CR, CC, DR, DT, ED and AK TPDU.

6.19.3 *Procedure*

The procedure is specified in the procedures for Class 4 (see § 12.2.1.2 i)).

6.20 *Resequencing*

6.20.1 *Purpose*

The resequencing procedure is used in Class 4 to cope with misordering of TPDU by the NS-provider.

6.20.2 *TPDUs and parameters used*

The procedure uses the following TPDU and parameters:

a) DT TPDU:

– TPDU-NR.

b) ED TPDU:

– ED-TPDU-NR.

6.20.3 *Procedure*

The procedure is specified in the procedures for Class 4 (see § 12.2.3.5).

6.21 *Inactivity control*

6.21.1 *Purpose*

The inactivity control procedure is used in Class 4 to cope with unsignalled termination of a network connection.

6.21.2 *Procedure*

The procedure is specified in the procedures for Class 4 (see § 12.2.3.3).

6.22 *Treatment of protocol errors*

6.22.1 *Purpose*

The procedure for treatment of protocol errors is used in all classes to deal with invalid TPDU.

6.22.2 *TPDUs and parameters used*

The procedure uses the following TPDU and parameters:

a) ER TPDU:

– reject cause;  
– invalid TPDU.

b) DR TPDU:

– reason code.

### 6.22.3 Procedure

A transport entity that receives a TPDU that can be associated to a transport connection and is invalid or constitutes a protocol error (see §§ 3.2.16 and 3.2.17) shall take one of the following actions so as not to jeopardize any other transport connections not assigned to that network connection:

- a) transmitting an ER TPDU;
- b) resetting or closing the network connection; or
- c) invoking the release procedures appropriate to the class.

Under certain circumstances it is also possible to discard the TPDU.

If an ER TPDU is sent in Class 0, it shall contain the octets of the invalid TPDU up to and including the octet where the error was detected (see Notes 3, 4 and 5).

If the TPDU cannot be associated with a particular transport connection, the transport entity shall follow the procedure in § 6.9.

*Note 1* – In general, no further action is specified for the receiver of the ER TPDU, but it is suggested that it initiates the release procedure appropriate to the class. If the ER TPDU has been received as an answer to a CR TPDU, then the connection is regarded as released (see § 6.6).

*Note 2* – Care should be taken by a transport entity receiving several invalid TPDU's or ER TPDU's to avoid looping if the error is generated repeatedly.

*Note 3* – If the invalid received TPDU is greater than the selected maximum TPDU size, it is possible that it cannot be included in the invalid TPDU parameter of the ER TPDU.

*Note 4* – It is suggested that the sender of the ER TPDU start a timer TS2 to ensure the release of the connection. If the timer expires, the transport entity shall initiate the release procedures appropriate to the class. The timer should be stopped when a DR TPDU or an N-DISCONNECT indication is received.

*Note 5* – In classes other than 0, it is suggested that the invalid TPDU be also included in the ER TPDU.

## 6.23 Splitting and recombining

### 6.23.1 Purpose

This procedure is used only in Class 4 to allow a transport connection to make use of multiple network connections to provide additional resilience against network failure, to increase throughput, or for other reasons.

### 6.23.2 Procedure

When this function is being used, a transport connection may be assigned (see § 6.1) to multiple network connections (see Note 1). TPDU's for the connection may be sent over any such network connection.

If the use of Class 4 is not accepted by the remote transport entity following the negotiation rules, then no network connection except that over which the CR TPDU was sent may have this transport connection assigned to it.

*Note 1* – The resequencing function of Class 4 (see § 6.20) is used to ensure that TPDU's are processed in the correct sequence.

*Note 2* – Either transport entity may assign the connection to further network connections of which it is the owner at any time during the lifetime of the transport connection, provided the following constraints are respected:

- the initiator does not start splitting before having received the CC TPDU;
- as soon as a new assignment is done, it is recommended to send a TPDU on this network connection in order to make the remote entity aware of this assignment.

*Note 3* – In order to enable the detection of unsignalled network connection failures, a transport entity performing splitting should ensure that TPDU's are sent at intervals on each supporting network connection, for example by sending successive TPDU's on successive network connections, where the set of network connections is used cyclically. By monitoring each network connection, a transport entity may detect unsignalled network connection failures, following the inactivity procedures defined in § 12.2.3.3. Thus, for each network connection, no period I (see § 12.2.3.1) may elapse without the receipt of some TPDU for some transport connection.

## 7 Protocol classes

Table 6/X.224 gives an overview of which elements of procedure are included in each class. In certain cases, the elements of procedure within different classes are not identical, and, for this reason, Table 6/X.224 cannot be considered part of the definitive specification of the protocol.

## 8 Specification for Class 0: simple class

### 8.1 *Functions of Class 0*

Class 0 is designed to have minimum functionality. It provides only the functions needed for connection establishment with negotiation, data transfer with segmenting and protocol error reporting.

Class 0 provides transport connections with flow control based on the network service provided flow control, and disconnection based on the network service disconnection.

### 8.2 *Procedures for Class 0*

#### 8.2.1 *Procedures applicable at all times*

The transport entities shall use the following procedures:

- a) TPDU transfer (see § 6.2);
- b) association of TPDU with transport connections (see § 6.9);
- c) treatment of protocol errors (see § 6.22);
- d) error release (see § 6.8).

TABLE 6/X.224

## Allocation of elements of procedure within classes

Procedure	Cross ref.	Variant	0	1	2	3	4
Assignment to network conn.	6.1		X	X	X	X	X
TPDU Transfer	6.2		X	X	X	X	X
Segmenting and reassembling	6.3		X	X	X	X	X
Concatenation and separation	6.4			X	X	X	X
Connection Establishment	6.5		X	X	X	X	X
Connection Refusal	6.6		X	X	X	X	X
Normal Release	6.7	implicit explicit	X	X	X	X	X
Error Release	6.8		X		X		
Association of TPDU with TCs	6.9		X	X	X	X	X
Data TPDU Numbering	6.10	normal extended		X	m <sup>a)</sup> o <sup>a)</sup>	m o	m o
Expedited Data Transfer	6.11	network normal network express		m ao	X <sup>a)</sup>	X	X
Reassignment after failure	6.12			X		X	c)
Retention until acknowledgement of TPDU	6.13	Conf. receipt AK		ao m		X	X
Resynchronization	6.14			X		X	c)
Multiplexing and Demultiplexing	6.15				X <sup>b)</sup>	X	X
Explicit Flow Control (with) Explicit Flow Control (without)	6.16		X	X	m o	X	X
Checksum (use of) Checksum (non-use of)	6.17		X	X	X	X	X o
Frozen References	6.18			X		X	X
Retransmission on Timeout	6.19						X
Resequencing	6.20						X
Inactivity Control	6.21						X
Treatment of Protocol Errors	6.22		X	X	X	X	X
Splitting and Recombining	6.23						X

X: Procedure always included in class.

empty square: Not applicable.

m: Negotiable procedure whose implementation in equipment is mandatory.

o: Negotiable procedure whose implementation in equipment is optional.

ao: Negotiable procedure whose implementation in equipment is optional and where use depends on availability within the network service.

a) Not applicable in class 2 when non-use of explicit flow control is selected.

b) Multiplexing may lead to degradation of the quality of service if the non-use of explicit flow control has been selected.

c) This function is provided in class 4 using procedures other than those used in the cross reference.

### 8.2.2 *Connection establishment*

The transport entities shall use the following procedures:

- a) assignment to network connection (see § 6.1); then
- b) connection establishment (see § 6.5) and, if appropriate, connection refusal (see § 6.6);

subject to the following constraints:

- c) the CR and CC TPDU shall contain no parameter field other than those for TSAP-ID and maximum TPDU size;
- d) the CR and CC TPDU shall not contain a data field.

### 8.2.3 *Data transfer*

The transport entities shall use the segmenting and reassembling procedure (see § 6.3).

### 8.2.4 *Release*

The transport entities shall use the implicit variant of the normal release procedure (see § 6.7).

*Note* – The lifetime of the transport connection is directly correlated with the lifetime of the network connection.

## 9 **Specification for Class 1: basic error recovery class**

### 9.1 *Functions of Class 1*

Class 1 provides transport connections with flow control based on the network service provided flow control, error recovery, expedited data transfer, disconnection, and also the ability to support consecutive transport connections on a network connection.

This class provides the functionality of Class 0 plus the ability to recover after a failure signalled by the Network Layer, without involving the TS-user.

### 9.2 *Procedures for Class 1*

#### 9.2.1 *Procedures applicable at all times*

The transport entities shall use the following procedures:

- a) TPDU transfer (see § 6.2);
- b) association of TPDU with transport connections (see § 6.9);
- c) treatment of protocol errors (see § 6.22);
- d) reassignment after failure (see § 6.12);
- e) resynchronization (see § 6.14), or reassignment after failure (see § 6.12) together with resynchronization (see § 6.14);
- f) concatenation and separation (see § 6.4);
- g) retention until acknowledgment of TPDU (see § 6.13); the variant used, AK or confirmation of receipt, shall be as selected during connection establishment (see Notes);
- h) frozen references (see § 6.18).

*Note 1* – The negotiation of the variant of retention until acknowledgment of TPDU procedure to be used over the transport connection has been designed such that if the initiator proposes the use of the AK variant (i.e. the mandatory implementation option), the responder has to accept use of this option and if the initiator proposes use of the confirmation of receipt variant the responder is entitled to select use of the AK variant.

*Note 2* – The AK variant makes use of AK TPDU to release copies of retained DT TPDU. The CDT parameter of AK TPDU in Class 1 is not significant, and is set to 1111.

*Note 3* – The confirmation of receipt variant is restricted to this class and its use depends on the availability of the network layer receipt confirmation service, and the expected cost reduction.

### 9.2.2 *Connection establishment*

The transport entities shall use the following procedures:

- a) assignment to network connection (see § 6.1); then
- b) connection establishment (see § 6.5) and, if appropriate, connection refusal (see § 6.6).

### 9.2.3 *Data transfer*

#### 9.2.3.1 *General*

The sending transport entity shall use the following procedures:

- a) segmenting (see § 6.3); then
- b) the normal format variant of DT TPDU numbering (see § 6.10).

The receiving transport entity shall use the following procedures:

- c) the normal format variant of DT TPDU numbering (see § 6.10); then
- d) reassembling (see § 6.3).

*Note 1* – The use of RJ TPDU during resynchronization (see § 6.14) can lead to retransmission. Thus, the receipt of a duplicate DT TPDU is possible; such a DT TPDU is discarded.

*Note 2* – It is possible to decide on a local basis to issue an N-RESET request in order to force the remote entity to carry out the resynchronization (see § 6.14).

#### 9.2.3.2 *Expedited data*

The transport entities shall use either of the network normal data or the network expedited variants of the expedited data transfer procedure (see § 6.11) if their use has been selected during connection establishment (see Note 1).

The sending transport entity shall not allocate the same ED-TPDU-NR to successive ED TPDU's (see Notes 2 and 3).

When acknowledging an ED TPDU by sending an EA TPDU the transport entity shall put into the YR-EDTU-NR parameter of the EA TPDU the value received in the ED-TPDU-NR parameter of the ED TPDU.

*Note 1* – The negotiation of the variant of expedited data transfer procedure to be used over the transport connection has been designed such that if the initiator proposes the use of the network normal data variant (i.e., the mandatory implementation option), the responder has to accept use of this option and if the initiator proposes use of the network expedited variant, the responder is entitled to select use of the network normal data variant.

*Note 2* – This numbering enables the receiving transport entity to discard repeated ED TPDU's when resynchronization (see § 6.14) has taken place.

*Note 3* – No other significance is attached to the ED-TPDU-NR parameter. It is suggested, but not essential, that the values used be consecutive modulo 128.

### 9.2.4 *Release*

The transport entities shall use the explicit variant of the release procedure (see § 6.7).

## **10 Specification for Class 2: multiplexing class**

### 10.1 *Functions of Class 2*

Class 2 provides transport connections with or without individual flow control; no error detection or error recovery is provided.

If the network connection resets or disconnects, the transport connection is terminated without the transport release procedure and the TS-user is informed.

When explicit flow control is used, a credit mechanism is defined allowing the receiver to inform the sender of the exact amount of data he is willing to receive and expedited data transfer is available.

## 10.2 *Procedures for Class 2*

### 10.2.1 *Procedures applicable at all times*

The transport entities shall use the following procedures:

- a) association of TPDU's with transport connections (see § 6.9);
- b) TPDU transfer (see § 6.2);
- c) treatment of protocol errors (see § 6.22);
- d) concatenation and separation (see § 6.4);
- e) error release (see § 6.8).

Additionally the transport entities may use the following procedure:

- f) multiplexing and demultiplexing (see § 6.15).

### 10.2.2 *Connection establishment*

The transport entities shall use the following procedures:

- a) assignment to network connection (see § 6.1); then
- b) connection establishment (see § 6.5) and, if applicable, connection refusal (see § 6.6).

### 10.2.3 *Data transfer when non-use of explicit flow control has been selected*

If this option has been selected as a result of the connection establishment, the transport entities shall use the segmenting procedure (see § 6.3).

The TPDU-NR field of DT TPDU's is not significant and may take any value.

*Note* – Expedited data transfer is not applicable (see § 6.5).

### 10.2.4 *Data transfer when use of explicit flow control has been selected*

#### 10.2.4.1 *General*

The sending transport entity shall use the following procedures:

- a) segmenting (see § 6.3); then
- b) DT TPDU numbering (see § 6.10);

The receiving transport entity shall use the following procedures:

- c) DT TPDU numbering (see § 6.10); if a DT TPDU is received which is out of sequence it shall be treated as a protocol error; then
- d) reassembling (see § 6.3).

The variant of the DT TPDU numbering which is used by both transport entities shall be that which was agreed at connection establishment.

#### 10.2.4.2 *Flow control*

The transport entities shall send an initial credit (which may be zero) in the CDT field of the CR or CC TPDU. This credit represents the initial value of the upper window allocated to the peer entity.

The transport entity that receives the CR or the CC TPDU shall consider its lower window edge as zero, and its upper window edge as the value of the CDT field in the received TPDU.

In order to authorize the transmission of DT TPDU's by its peer, a transport entity may transmit an AK TPDU at any time, subject to the following constraints:

- a) the YR-TU-NR parameter shall be at most one greater than the TPDU-NR parameter of the last received DT TPDU or shall be zero if no DT TPDU has been received;
- b) if an AK TPDU has previously been sent, the value of the YR-TU-NR parameter shall not be lower than that in the previously sent AK TPDU;

- c) the sum of the YR-TU-NR and CDT parameters shall not be less than the upper window edge allocated to the remote entity (see Note 1).

A transport entity which receives an AK TPDU shall consider the YR-TU-NR parameter as its new lower window edge, and the sum of YR-TU-NR and CDT as its new upper window edge. If either of these have been reduced or if the lower window edge has become more than one greater than the TPDU-NR of the last transmitted DT TPDU, this shall be treated as a protocol error (see § 6.22).

A transport entity shall not send a DT TPDU with a TPDU-NR outside of the transmit window (see Notes 2 and 3).

*Note 1* – This means that credit reduction is not applicable.

*Note 2* – This means that a transport entity is required to stop sending if the TPDU-NR parameter of the next DT TPDU which would be sent would be the upper window edge. Sending of DT TPDU may be resumed if an AK TPDU is received which increases the upper window edge.

*Note 3* – The rate at which a transport entity progresses the upper window edge allocated to its peer entity constrains the throughput attainable on the transport connection.

#### 10.2.4.3 Expedited data

The transport entities shall follow the network normal data variant of the expedited data transfer procedure in § 6.11 if its use has been agreed during connection establishment. ED and EA TPDU's are not subject to the flow control procedures in § 10.2.4.2. The ED-TPDU-NR and YR-EDTU-NR parameters of ED and EA TPDU's respectively are not significant and may take any value.

#### 10.2.5 Release

The transport entities shall use the explicit variant of the release procedure in § 6.7.

## 11 Specification for Class 3: error recovery and multiplexing class

### 11.1 Functions of Class 3

This class provides the functionality of Class 2 (with use of explicit flow control) plus the ability to recover after a failure signalled by the Network Layer without involving the TS-user.

The mechanisms used to achieve this functionality also allow the implementation of more flexible flow control.

### 11.2 Procedures for Class 3

#### 11.2.1 Procedures applicable at all times

The transport entities shall use the following procedures:

- a) association of TPDU's with transport connections (see § 6.9);
- b) TPDU transfer (see § 6.2) and retention until acknowledgment of TPDU's (AK variant only) (see § 6.13);
- c) treatment of protocol errors (see § 6.22);
- d) concatenation and separation (see § 6.4);
- e) reassignment after failure (see § 6.12), together with resynchronization (see § 6.14);
- f) frozen references (see § 6.18).

Additionally, the transport entities may use the following procedure:

- g) multiplexing and demultiplexing (see § 6.15).

#### 11.2.2 Connection establishment

The transport entity shall use the following procedures:

- a) assignment to network connections (see § 6.1); then
- b) connection establishment (see § 6.5) and, if appropriate, connection refusal (see § 6.6).

### 11.2.3 *Data transfer*

#### 11.2.3.1 *General*

The sending transport entity shall use the following procedures:

- a) segmenting (see § 6.3); then
- b) DT TPDU numbering (see § 6.10); after receipt of an RJ TPDU (see § 11.2.3.2) the next DT TPDU to be sent may have a value which is not the previous value of TPDU-NR plus one.

The receiving transport entity shall use the following procedures:

- c) DT TPDU numbering (see § 6.10); the TPDU-NR parameter of each received DT TPDU shall be treated as a protocol error if it exceeds the greatest such value received in a previous DT TPDU by more than one (see Note); then
- d) reassembling (see § 6.3); duplicated TPDU's shall be eliminated before reassembling is performed.

*Note* – The use of RJ TPDU's (see § 11.2.3.2) can lead to retransmission and reduction of credit. Thus the receipt of a DT TPDU which is a duplicate, or which is greater than or equal to the upper window edge allocated to the peer entity, is possible and is therefore not treated as a protocol error.

#### 11.2.3.2 *Use of RJ TPDU*

A transport entity may send an RJ TPDU at any time in order to invite retransmission or to reduce the upper window edge allocated to the peer entity (see Note 1).

When an RJ TPDU is sent, the following constraints shall be respected:

- a) the YR-TU-NR parameter shall be at most one greater than the greatest such value received in a previous DT TPDU, or shall be zero if no DT TPDU has yet been received (see Note 2);
- b) if an AK or RJ TPDU has previously been sent, the YR-TU-NR parameter shall not be lower than that in the previously sent AK or RJ TPDU.

When a transport entity receives an RJ TPDU (see Note 3):

- c) the next DT TPDU to be transmitted, or retransmitted, shall be that for which the value of the TPDU-NR parameter is equal to the value of the YR-TU-NR parameter of the RJ TPDU;
- d) the sum of the values of the YR-TU-NR and CDT parameters of the RJ TPDU becomes the new upper window edge (see Note 4).

*Note 1* – An RJ TPDU can also be sent as part of the resynchronization (see § 6.14) and reassignment after failure (see § 6.12) procedures.

*Note 2* – It is suggested that the YR-TU-NR parameter be equal to the TPDU-NR parameter of the next expected DT TPDU.

*Note 3* – These rules are a subset of those specified for when an RJ TPDU is received during resynchronization (see § 6.14) and reassignment after failure (see § 6.12).

*Note 4* – This means that RJ TPDU can be used to reduce the upper window edge allocated to the peer entity (credit reduction).

#### 11.2.3.3 *Flow control*

The procedures shall be as defined in § 10.2.4.2, except that:

- a) a credit reduction may lead to the reception of a DT TPDU with a TPDU-NR parameter whose value is not but would have been less than the upper window edge allocated to the remote entity prior to the credit reduction. This shall not be treated as a protocol error;
- b) receipt of an AK TPDU which sets the lower window edge more than one greater than the TPDU-NR of the last transmitted DT TPDU shall not be treated as a protocol error, provided that all acknowledged DT TPDU's have been previously transmitted (see Notes 1 and 2).

*Note 1* – This can only occur during retransmission following receipt of an RJ TPDU.

*Note 2* – The transport entity may either continue retransmission as before or retransmit only those DT TPDU's not acknowledged by the AK TPDU. In either case, copies of the acknowledged DT TPDU's need not be retained further.

#### 11.2.3.4 *Expedited data*

The transport entities shall follow the network normal data variant of expedited data transfer procedure in § 6.11 if its use has been agreed during connection establishment.

The sending transport entity shall not allocate the same ED-TPDU-NR to successive ED TPDU.

The receiving transport entity shall transmit an EA TPDU with the same value in its YR-EDTU-NR-parameter. If, and only if, this number is different from that of the previously received ED TPDU, shall it generate a T-EXPEDITED DATA indication to convey the data to the TS-user (see Note 2).

*Note 1* – No other significance is attached to the ED-TPDU-NR parameter. It is suggested, but not essential, that the values be consecutive modulo  $2n$ , where  $n$  is the number of bits of the parameter.

*Note 2* – This procedure ensures that the TS-user does not receive data corresponding to the same ED TPDU more than once.

#### 11.2.4 *Release*

The transport entities shall use the explicit variant of the release procedure (see § 6.7).

## 12 **Specification for Class 4: error detection and recovery class**

### 12.1 *Functions of Class 4*

Class 4 provides the functionality of Class 3, plus the ability to detect and recover from lost, duplicated or out of sequence TPDU without involving the TS-user.

Class 4 detects signalled and unsignalled network failures (i.e. resets or disconnects or inactivity) and recovers from these failures by using timeout mechanisms.

This detection of errors is made by extended user of the sequence numbering of Classes 2 and 3, by timeout mechanisms, and by additional procedures.

This class additionally detects and recovers from damaged TPDU by using a checksum mechanism. The use of the checksum mechanism must be available but its use or its non-use is subject to negotiation. Furthermore, this class provides additional resilience against network failure and increased throughput capability by allowing a transport connection to make use of multiple network connections.

### 12.2 *Procedures for Class 4*

#### 12.2.1 *Procedures available at all times*

##### 12.2.1.1 *Timers used at all times*

This sub-clause defines timers that apply at all times in Class 4. These timers are listed in Table 7/X.224.

TABLE 7/X.224

## Timer parameters related to the operation of Class 4

Symbol	Name	Definition
$M_{LR}$	NSDU lifetime local-to-remote	A time bound for the maximum time which may elapse between the transmission of an NSDU by a local transport entity and the receipt of any copy of it by a remote peer entity.
$M_{RL}$	NSDU lifetime remote-to-local	A time bound for the maximum time which may elapse between the transmission of an NSDU by a remote transport entity and the receipt of any copy of it by a local peer entity.
$E_{LR}$	Expected maximum transit delay local-to-remote	A time bound for the maximum delay suffered by all but a small proportion of NSDUs transferred from the local transport entity to a remote peer entity.
$E_{RL}$	Expected maximum transit delay remote-to-local	A time bound for the maximum delay suffered by all but a small proportion of NSDUs transferred from a remote transport entity to the local peer entity.
$A_L$	Local acknowledge time	A time bound for the maximum time which can elapse between the receipt of a TPDU by the local transport entity from the network layer and the transmission of the corresponding acknowledgement.
$A_R$	Remote acknowledge time	As $A_L$ , but for the remote entity.
$T_1$	Local retransmission time	A time bound for the maximum time the local transport entity will wait for acknowledgement before retransmitting a TPDU.
$R$	Persistence time	A time bound for the maximum time that the local transport entity will continue to transmit a TPDU that requires acknowledgement.
$N$	Maximum number of transmissions	A bound for the number of times which the local transport entity will continue to transmit a TPDU that requires acknowledgement.
$L$	Time bound on references and sequence numbers	A time bound for the maximum time between the transmission of a TPDU and the receipt of any acknowledgement relating to it.
$I$	Inactivity time	A time bound for the time after which a transport entity will, if it does not receive a TPDU, initiate the release procedure to terminate the transport connection. <i>Note</i> – This parameter is required for protection against unsignalled failures.
$W$	Window time	A time bound for the maximum time a transport entity will wait before retransmitting up-to-date window information.

This Recommendation does not define specific values for the timers, and the derivations described in this subclause are not mandatory. The values should be chosen so that the required quality of service can be provided, given the known characteristics of the network.

Timers that apply only to specific procedures are defined under the appropriate procedure.

#### 12.2.1.1.1 NSDU lifetimes ( $M_{LR}$ , $M_{RL}$ )

The network layer is assumed to provide, as an aspect of its quality of service, for a bound on the maximum lifetime of NSDUs in the network. This value may be different in each direction of transfer through a network between two transport entities. The values, for both directions of transfer, are assumed to be known by the transport entities.

The maximum NSDU lifetime local-to-remote ( $M_{LR}$ ) is the maximum time which may elapse between the transmission of an NSDU from the local transport entity to the network layer and the receipt of any copy of the NSDU from the network layer at the remote transport entity.

The maximum NSDU lifetime remote-to-local ( $M_{RL}$ ) is the maximum time which may elapse between the transmission of an NSDU from the remote transport entity to the network layer and receipt of any copy of the NSDU from the network layer at the local transport entity.

#### 12.2.1.1.2 *Expected maximum transit delay ( $E_{LR}, E_{RL}$ )*

The network layer is assumed to provide, as an aspect of its quality of service, an expected maximum transit delay for NSDUs in the network. This value may be different in each direction of transfer through a network between two transport entities. The values, for both directions of transfer, are assumed to be known by the transport entities.

The expected maximum transit delay local-to-remote ( $E_{RL}$ ) is the maximum delay suffered by all but a small proportion of NSDUs transferred through the network from the local transport entity to the remote transport entity.

The expected maximum transit delay remote-to-local ( $E_{LR}$ ) is the maximum delay suffered by all but a small proportion of NSDUs transferred through the network from the remote transport entity to the local transport entity.

#### 12.2.1.1.3 *Acknowledge time ( $A_R, A_L$ )*

Any transport entity is assumed to provide a bound for the maximum time which can elapse between its receipt of a TPDU from the Network Layer and its transmission of the corresponding response. This value is referred to as  $A_L$ . The corresponding time given by the remote transport entity is referred to as  $A_R$ .

#### 12.2.1.1.4 *Local retransmission time ( $T1$ )*

The local transport entity is assumed to maintain a bound on the time it will wait for an acknowledgment before retransmitting the TPDU.

Its value is given by:

$$T1 = E_{LR} + E_{RL} + A_R + X$$

where:

$E_{LR}$  = expected maximum transit delay local-to-remote,

$E_{RL}$  = expected maximum transit delay remote-to-local,

$A_R$  = remote acknowledge time, and

$X$  = local processing time for a TPDU.

*Note* – During connection establishment the value of  $A_R$  is not known. In this case a suitable bound for  $T1$  may be established either by estimating (or having “a priori” knowledge of)  $A_R$  or by applying a suitable algorithm to the TC establishment delay QOS parameter.

#### 12.2.1.1.5 *Persistence time ( $R$ )*

The local transport entity is assumed to provide a bound for the maximum time for which it may continue to retransmit a TPDU requiring positive acknowledgement and which is not outside the current transmit window, even after credit reduction. This value is referred to as  $R$ .

This value is clearly related to the time elapsed between retransmission,  $T1$ , and the maximum number of retransmissions,  $N$ . It is not less than  $T1 \times (N - 1) + x$ , where  $x$  is a small quantity to allow for additional internal delays, the granularity of the mechanism used to implement  $T1$  and so on. Because  $R$  is a bound, the exact value of  $x$  is unimportant as long as it is bounded and the value of a bound is known.

#### 12.2.1.1.6 *Time bound on references and sequence numbers ( $L$ )*

A time bound for the maximum time between the decision to transmit a TPDU and the receipt of any acknowledgement relating to it ( $L$ ) is given by:

$$L = M_{LR} + M_{RL} + R + A_R$$

where:

$M_{LR}$  = NSDU lifetime local-to-remote,

$M_{RL}$  = NSDU lifetime remote-to-local,

$A_R$  = remote acknowledge time, and

$R$  = persistence time.

It is necessary to wait for a period L before re-using any reference or sequence number, to avoid confusion in case a TPDU referring to it may be duplicated or delayed.

*Note 1* – In practice, the value of L may be unacceptably large. It may also be only a statistical figure at a certain confidence level. A smaller value may therefore be used where this still allows the required quality of service to be provided.

*Note 2* – The relationships between the times discussed above are illustrated in Figures 3 and 4/X.224.

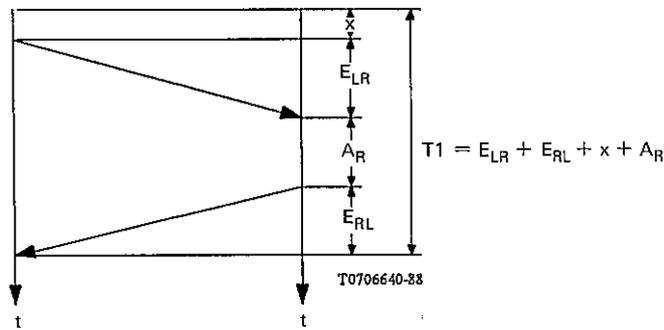


FIGURE 3/X.224

**The interrelationship of times for the average case in class 4**

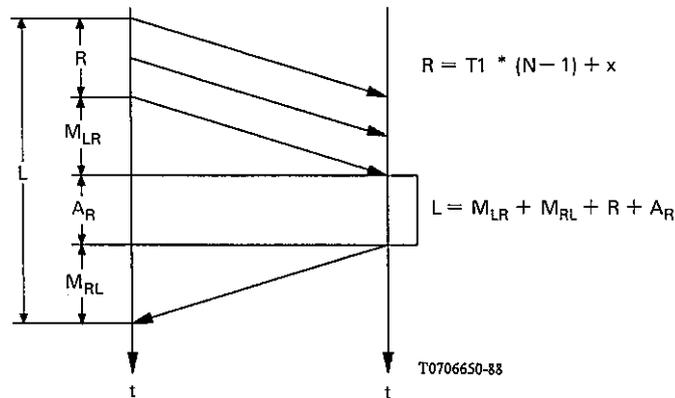


FIGURE 4/X.224

**The interrelationship of times for the maximum delay in class 4**

### 12.2.1.2 General procedures

The transport entity shall use the following procedures:

- a) TPDU transfer (see § 6.2);
- b) association of TPDU with transport connections (see § 6.9);
- c) treatment of protocol errors (see § 6.22);
- d) checksum (see § 6.17);
- e) splitting and recombining (see § 6.23);
- f) multiplexing and demultiplexing (see § 6.15);

- g) retention until acknowledgement of TPDU (see § 6.13);
- h) frozen reference (see § 6.18);
- i) retransmission procedures; when a transport entity has some outstanding TPDU that require acknowledgement, it will check that no T1 interval elapses without the arrival of a TPDU that acknowledges at least one of the outstanding TPDU.

If the timer expires, except if the TPDU to be retransmitted is a DT TPDU and it is outside the transmit window due to credit reduction, the first TPDU is retransmitted and the timer is restarted. After N transmissions (i.e. N-1 retransmissions), it is assumed that useful two-way communication is no longer possible and the release procedure is used, and the TS-user is informed.

*Note 1* – This procedure may be implemented by different means. For example:

- a) one interval is associated with each TPDU. If the timer expires, the associated TPDU will be retransmitted, and the timer T1 will be restarted for all subsequent DT TPDU; or
- b) one interval is associated with each transport connection:
  - 1) if the transport entity transmits a TPDU requiring acknowledgement, it starts timer T1;
  - 2) if the transport entity receives a TPDU that acknowledges one of the TPDU to be acknowledged, it restarts timer T1 unless the received TPDU is an AK TPDU which explicitly closes the transmit window;
  - 3) if the transport entity receives a TPDU that acknowledges the last TPDU to be acknowledged, it stops timer T1.

For a decision whether the retransmission timer T1 is maintained on a per TPDU or on a per transport connection basis, throughput considerations have to be taken into account.

*Note 2* – For DT TPDU, it is a local choice to retransmit either only the first DT TPDU or all TPDU waiting for an acknowledgement up to the upper window edge.

*Note 3* – It is suggested that after N transmissions of a DT TPDU, the transport entity waits  $T1 + W + M_{RL}$  to provide a higher possibility of receiving an acknowledgement before entering the release phase.

For other TPDU types which may be retransmitted, it is suggested that after N transmissions, the transport entity waits  $T1 + M_{RL}$  to provide a higher possibility of receiving the expected reply.

## 12.2.2 Procedures for connection establishment

### 12.2.2.1 Timers used in connection establishment

There are no timers specific to connection establishment.

### 12.2.2.2 General procedures

The transport entities shall use the following procedures:

- a) when a network connection to which the transport connection is assigned is released (N DISind received):
  - 1) if a CC TPDU is awaited the initiator shall perform a new assignment according to QOS and retransmission procedure (i.e., no more than  $N \times T1$  keeping sending CR TPDU);
  - 2) if there is at least one other network connection to which the transport connection is assigned both initiator and acceptor may either perform a new assignment or continue operation using one of the remaining network connections;
  - 3) if the transport connection becomes unassigned the acceptor may either perform new assignment or wait (there is no risk of deadlock since either T1 or I is running), the initiator shall perform a new assignment (except in the closing state);
- b) connection establishment (see § 6.5) and, if appropriate, connection refusal (see § 6.6) together with the additional procedures:
  - 1) a connection is not considered established until the successful completion of a 3-way TPDU exchange. The sender of a CR TPDU must respond to the corresponding CC TPDU by immediately sending a DT, ED, DR or AK TPDU;

- 2) as a result of duplication or retransmission, a CR TPDU may be received specifying a source referene which is already in use with the sending transport entity. If the receiving transport entity is in the data transfer phase, having completed the 3-way TPDU exchange procedure, or is waiting for the T-CONNECT response from the TS-user, the receiving transport entity shall discard such a TPDU. Otherwise, a CC TPDU shall be transmitted;
- 3) as a result of duplication or retransmission, a CC TPDU may be received specifying a paired reference which is already in use. The receiving transport entity shall only acknowledge the duplicate CC TPDU according to the procedure in § 12.2.2.2, b) 1);
- 4) a CC TPDU may be received specifying a reference which is in the frozen state. The response to such a TPDU shall be a DR TPDU;
- 5) the retransmission procedures (see § 12.2.1.2) are used for both the CR TPDU and CC TPDU.

*Note* – After receiving a CR TPDU, it is recomended that the transport entity enforce a time limit upon the Transport Service user so that its late acceptance of the transport connection will not cause a delayed CC TPDU to be sent.

### 12.2.3 *Procedures for data transfer*

#### 12.2.3.1 *Timers used in data transfer*

The data transfer procedures use two additional timers.

##### 12.2.3.1.1 *Inactivity time (I)*

To protect against unsignalled breaks in the network connection, or failure of the peer transport entity, (half-open connections), each transport entity maintains an inactivity time interval.

*Note* – A suitable value for I is given by  $2 \times [N \times \text{maximum of (T1, W)}]$  unless local needs indicate another more appropriate value.

##### 12.2.3.1.2 *Window time (W)*

A transport entity maintains a timer interval to ensure that there is a bound on the maximum interval between window updates.

#### 12.2.3.2 *General procedures for data transfer*

The transport entities shall use the following procedures:

- a) inactivity control (see § 6.21);
- b) expedited data (see § 6.11);
- c) explicit flow control (see § 6.16).

The sending transport entity shall use the following procedures in the following order:

- d) segmenting (see § 6.3);
- e) DT TPDU numbering (see § 6.10).

The receiving transport entity shall use the following procedures in the following order:

- f) DT TPDU numbering (see § 6.10);
- g) resequencing (see § 6.20);
- h) reassembling (see § 6.3).

#### 12.2.3.3 *Inactivity control*

If the interval of the inactivity timer I expires without receipt of some TPDU, the transport entity shall initiate the release procedures. To prevent expiration of the remote transport entity's inactivity timer when no data is being sent, the local transport entity must send AK TPDUs at suitable intervals in the absence of data, having regard to the probability of TPDU loss. The window synchronization procedures (see § 12.2.3.8) ensure that this requirement is met.

*Note* – It is likely that the release procedures initiated due to inactivity timer expiration will fail, as such expiration indicates probable failure of the supporting network connection or of the remote transport entity.

#### 12.2.3.4 Expedited data

The transport entities shall follow the network normal data variant of the expedited data transfer procedures (see § 6.11), if the use of transport expedited data transfer service option has been agreed during connection establishment.

The ED TPDU shall have a TPDU-NR which is allocated from a separate sequence space from that of the DT TPDU. A transport entity shall allocate the sequence number zero to the ET-TPDU-NR of the first ED TPDU which it transmits for a transport connection. For subsequent ED TPDU sent on the same transport connection, the transport entity shall allocate a sequence number one greater than the previous one.

Modulo  $2^7$  arithmetic shall be used when normal formats have been selected and modulo  $2^{31}$  arithmetic shall be used when extended formats have been selected.

The receiving transport entity shall transmit an EA TPDU with the same sequence number in its YR-EDTU-NR parameter. If this number is one greater than in the previously received in-sequence ED TPDU, the receiving transport entity shall transfer the data in the ED TPDU to the TS-user.

If a transport entity does not receive an EA TPDU in acknowledgement to an ET TPDU, it shall follow the retransmission procedures (see Note and § 12.2.1.2).

The sender of an ED-TPDU shall not send any new DT TPDU created from a T-DATA request subsequent to the T-EXPEDITED DATA request, until it receives the EA TPDU.

*Note* – This procedure ensures that ED TPDU are delivered to the TS-user in sequence and that the TS-user does not receive data corresponding to the same ED TPDU more than once. Also, it guarantees the arrival of the ED TPDU before any data subsequently sent by the TS-user.

#### 12.2.3.5 Resequencing

The receiving transport entity shall deliver all DT TPDU to the TS-user in the order specified by the TPDU-NR parameter.

DT TPDU received out of sequence but within the transmit window shall not be delivered to the TS-user until in-sequence TPDU have also been received. DT TPDU received out-of-sequence and outside the transmit window shall be discarded, but may result in transmission of an AK TPDU with up-to-date window information (see § 12.2.3.8).

Duplicate TPDU can be detected because the sequence number matches that of previously received TPDU. Sequence numbers shall not be reused for the period L after their previous use. Otherwise, a new, valid TPDU could be confused with a duplicated TPDU which had previously been received and acknowledged.

Duplicated DT TPDU shall be acknowledged, since the duplicated TPDU may be the result of a retransmission resulting from the loss of an AK TPDU.

The data contained in a duplicated DT TPDU shall be discarded.

#### 12.2.3.6 Explicit flow control

The transport entities shall send an initial credit (which may take the value 0) in the CDT parameter of the CR TPDU or CC TPDU. This credit represents the initial value of the upper window edge of the peer entity.

The transport entity which receives the CR TPDU or CC TPDU shall consider its lower window edge as zero and its upper window edge as the value in the CDT parameter in the received TPDU.

In order to authorize the transmission of DT TPDU by its peer, a transport entity may transmit an AK TPDU at any time.

The sequence number of an AK TPDU shall not exceed the sequence number of the next expected DT TPDU, i.e., shall not be greater than the highest sequence number of a received DT TPDU, plus one.

A transport entity may send a duplicate AK TPDU containing the same sequence number, CDT, and subsequence number field at any time.

A transport entity may increase or decrease the upper window edge at any time.

A transport entity which receives an AK TPDU shall consider the value of the YR-TU-NR parameter as its new lower window edge if it is greater than any previously received in a YR-TU-NR parameter, and the sum of YR-TU-NR and CDT as its new upper window edge subject to the procedures for sequencing AK TPDU (see § 12.2.3.8). A transport entity shall not transmit or retransmit a DT TPDU with a sequence number outside the transmit window.

### 12.2.3.7 Sequencing of received AK TPDU

To allow a receiving transport entity to properly sequence a series of AK TPDU that all contain the same sequence number and thereby use the correct CDT value, AK TPDU may contain a sub-sequence parameter. For the purpose of determining the correct sequence of AK TPDU, the absence of the sub-sequence parameter shall be equivalent to the value of the parameter set to zero.

An AK TPDU is defined to be in sequence if:

- a) the sequence number is greater than in any previously received AK TPDU, or
- b) the sequence number is equal to the highest in any previously received AK TPDU, and the sub-sequence parameter is greater than in any previously received AK TPDU having the same value of YR-TU-NR field, or
- c) the sequence number and sub-sequence parameter are both equal to the highest in any previously received AK TPDU, and the CDT parameter is greater than or equal to that in any previously received AK TPDU having the same YR-TU-NR parameter.

When the receiving transport entity recognizes an out of sequence AK TPDU, it shall discard it.

### 12.2.3.8 Procedures for transmission of AK TPDU

#### 12.2.3.8.1 Transmission of AK TPDU

An in-sequence DT TPDU shall be acknowledged within time  $A_L$ , by the transmission of AK TPDU whose "YR-TU-NR" field is set to at least the sequence number of the received DT TPDU plus one.

An AK TPDU shall be transmitted containing up-to-date window information if:

- a) a DT TPDU is received whose sequence number is lower than the lower window edge, but greater than or equal to the lower window edge minus the maximum credit value ever given for this transport connection, or
- b) a DT TPDU is received whose sequence number is above the current upper window but following credit reduction is within the upper window edge which has been granted and then withdrawn.

*Note 1* – A simpler implementation may send an AK TPDU upon receipt of any DT TPDU outside the transmit window.

*Note 2* – The procedure a) is required so that loss of an AK TPDU is correctly recovered, i.e., when the sender of the DT TPDU retransmits it following nonreceipt of an acknowledgement.

*Note 3* – The procedure b) is required due to the possibility of loss of the AK TPDU indicating the upper window edge reduction, which could otherwise cause incorrect termination of the transport connection.

A transport entity shall not allow an interval  $W$  to pass without the transmission of an AK TPDU. If the transport entity is not using the procedure following setting CDT to zero (see § 12.2.3.8.3) or reduction of the upper window edge (see § 12.2.3.8.4), and does not have to acknowledge receipt of any DT TPDU, then it shall achieve this by retransmission of the most recent AK TPDU, with up-to-date window information.

*Note* – The use of the procedures defined in §§ 12.2.3.8.3 and 12.2.3.8.4 are optional for any transport entity. The protocol operates correctly either with or without these procedures which are defined to enhance the efficiency of its operation.

#### 12.2.3.8.2 Sequence control for transmission of AK TPDU

To allow the receiving transport entity to process AK TPDU in the correct sequence, as described in § 12.2.3.7, the sub-sequence parameter shall be included following reduction of CDT. If the value of the sub-sequence number to be transmitted is zero, then the parameter should be omitted.

The value of the sub-sequence parameter, if used, shall be zero (either explicitly or by absence of the parameter) if the sequence number is greater than the parameter in previous AK TPDU sent by the transport entity.

If the sequence number is the same as the previous AK TPDU sent and the CDT parameter is equal to or greater than the CDT parameter in the previous AK TPDU sent, then the sub-sequence parameter, if used, shall be equal to that in the previously sent AK TPDU.

If the sequence number is the same as the previous AK TPDU sent and the CDT parameter is less than the value of the CDT parameter in the previous AK TPDU sent, then the sub-sequence parameter, if used, shall be one greater than the value in the previous AK TPDU.

*Note* – If a transport entity never reduces credit then it does not need to use the sub-sequence parameter.

#### 12.2.3.8.3 *Retransmission of AK TPDUs after CDT set to zero*

Due to the possibility of loss of AK TPDUs, the upper window edge as perceived by the transport entity transmitting an AK TPDU may differ from that perceived by the intended recipient. To avoid the possibility of extra delay, the retransmission procedure (see § 12.2.1.2) should be followed for an AK TPDU, if it opens the transmit window which has previously been closed by sending an AK TPDU with CDT field set to zero.

The retransmission procedure, if used, terminates and the procedure in § 12.2.3.8.1 is used when:

- a) AK TPDU is received containing the flow control confirmation parameter, whose lower window edge and sub-sequence fields are equal to the sequence number and sub-sequence number in the retained AK TPDU and whose credit field is not zero;
- b) an AK TPDU is transmitted with a sequence number higher than that in the retained AK TPDU, due to reception of a DT TPDU whose sequence number is equal to the lower window edge;
- c) N transmissions of the retained AK TPDU have taken place. In this case, the transport entity shall continue to transmit the AK TPDU at an interval of W.

An AK TPDU which is subject to the retransmission procedure shall not contain the flow control confirmation parameter. If it is required to transmit this parameter concurrently, an additional AK TPDU shall be transmitted having the same values in the YR-TU-NR, sub-sequence (if applicable) and CDT parameters.

#### 12.2.3.8.4 *Retransmission procedures following reduction of the upper window edge*

This subsection specifies the procedure for retransmission of AK TPDUs after a transport entity has reduced the upper window edge (see § 12.2.3.6) or for an AK TPDU with the credit field set to zero. This procedure is used until the lower window edge exceeds the highest value of the upper window edge ever transmitted (i.e. the value existing at the time of credit reduction, unless a higher value is retained from a previous credit reduction).

The retransmission procedure should be followed for any AK TPDU which increases the upper window edge, unless it is known that the remote transport entity has an open window. This is known if:

- a flow control confirmation (FCC) parameter has been received, corresponding to an AK TPDU transmitted following the most recent credit reduction, and
- this FCC parameter conveys an upper window edge value (i.e., the sum of the lower window edge and credit fields) which is greater than the lower window edge of the transmitted AK TPDU.

This retransmission procedure for any particular AK TPDU shall terminate when:

- a) an AK TPDU is received containing the flow control confirmation parameter, whose lower window edge and sub-sequence fields are equal to the lower window edge (YR-TU-NR) and sub-sequence number in the retained AK TPDU; or
- b) N transmission of the retained AK TPDU have taken place. In this case, the transport entity shall continue to transmit the AK TPDU as an interval of W.

An AK TPDU which is subject to the retransmission procedure shall not contain the flow control confirmation parameter. If it is required to transmit this parameter concurrently, an additional AK TPDU shall be transmitted having the same values in the sequence, sub-sequence (if applicable) and credit fields.

*Note* – Retransmission of AK TPDUs is normally not necessary, except following explicit closing of the window (i.e., transmission of an AK TPDU with CDT parameter set to zero). If data is available to be transmitted, the retransmission procedure for DT TPDUs will ensure that an AK TPDU is received granting further credit where this is available. Following credit reduction, this may no longer be so, because retransmission may be inhibited by the credit reduction. The rules described in this clause avoid extra delay.

The rules for determining whether to apply the retransmission procedure to an AK TPDU may be expressed alternatively as follows:

- LWE = lower window edge
- UWE = upper window edge
- KUWE = lower bound on upper window edge held by remote transport entity.

The retransmission procedure is used whenever:

$$(UWE > LWE) \text{ and } (KUWE = LWE)$$

i.e. when the window is opened and it is not known definitely that the remote transport entity is aware of this.

KUWE is maintained as follows. When credit is reduced, KUWE is set to LWE. Subsequently, it is increased only upon receipt of a valid flow control confirmation (i.e. one which matches the retained lower window edge and sub-sequence). In the case, KUWE is set to the implied upper window edge of the flow control confirmation, i.e. the sum of its lower window edge and credit fields. By this means, it can be ensured that KUWE is always less than or equal to the actual upper window edge in use by the transmitter of DT TPDU.

#### 12.2.3.9 Use of flow control confirmation parameter

At any time, an AK TPDU may be transmitted containing a flow control confirmation parameter. The lower window edge, the sub-sequence and the credit fields shall be set to the same values as the corresponding parameters in the most recently received in-sequence AK TPDU.

An AK TPDU containing a flow control confirmation parameter should be transmitted whenever:

- a) a duplicate AK TPDU is received, with the value of YR-TU-NR, CDT, and sub-sequence fields equal to the most recently received AK TPDU, but not itself containing the flow control confirmation parameter;
- b) an AK TPDU is received which increases the upper window edge but not the lower window edge, and the upper window edge was formerly equal to the lower edge; or
- c) an AK TPDU is received which increases the upper window edge but not the lower window edge, and the lower window edge is lower than the highest value of the upper window edge ever received and subsequently reduced (i.e. following credit reduction).

#### 12.2.4 Procedures for release

##### 12.2.4.1 Timers used for release

There are no timers used only for release.

##### 12.2.4.2 General procedures for release

The transport entity shall use the explicit variant of normal release (see § 6.7).

## 13 Structure and encoding of TPDU

### 13.1 Validity

Table 8/X.224 specifies those TPDU which are valid for each class and the code for each TPDU.

### 13.2 Structure

All the transport protocol data units (TPDUs) shall contain an integral number of octets. The octets in a TPDU are numbered starting from 1 and increasing in the order they are put into an NSDU. The bits in an octet are numbered from 1 to 8, where bit 1 is the low-order bit.

When consecutive octets are used to represent a binary number or a binary coded decimal number (one digit per octet), the lower octet number has the most significant value.

*Note 1* – The numbering of bits within an octet is a convention local to this Recommendation.

*Note 2* – The use of the terms “high order” and “low order” is common to this Recommendation and to adjacent layer Recommendations.

*Note 3* – The use of the above conventions does not affect the order of the bit transmission on a serial communications link.

*Note 4* – As described in § 6.2.3, both transport entities respect these bit and octet ordering conventions, thus allowing communication to take place.

TABLE 8/X.224

## TPDU codes

	Validity within classes					See	Code
	0	1	2	3	4		
CR: Connection Request	x	x	x	x	x	§ 13.3	1110 xxxx
CC: Connection Confirm	x	x	x	x	x	§ 13.4	1101 xxxx
DR: Disconnect Request	x	x	x	x	x	§ 13.5	1000 0000
DC: Disconnect Confirm		x	x	x	x	§ 13.6	1100 0000
DT: Data	x	x	x	x	x	§ 13.7	1111 0000
ED: Expedited Data		x	NF	x	x	§ 13.8	0001 0000
AK: Data Acknowledgement		NRC	NF	x	x	§ 13.9	0110 zzzz
EA: Expedited Data Acknowledgement		x	NF	x	x	§ 13.10	0010 0000
RJ: Reject		x		x		§ 13.11	0101 zzzz
ER: TPDU Error	x	x	x	x	x	§ 13.12	0111 0000
PI: Transport Protocol Id.						Annex B	0000 0001
Not available (see Note)						–	0000 0000
						–	0011 0000
						–	1001 xxxx
						–	1010 xxxx

xxxx (bits 4 to 1): used to signal the CDT in classes 2,3,4; set to 0000 in classes 0 and 1.

zzzz (bits 4 to 1): used to signal the CDT in classes 2,3,4; set to 1111 in class 1.

NF: not available when the non explicit flow control option is selected.

NRC: not available when the receipt confirmation option is selected.

Note – These codes are already in use in related protocols defined by standards organisations other than CCITT/ISO.

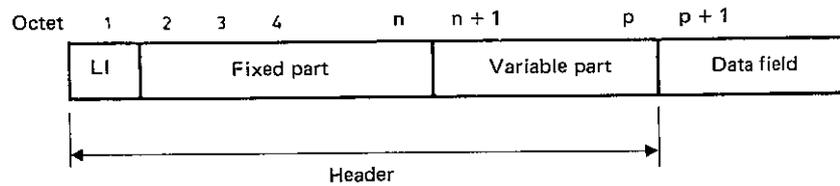
Note 5 – When the encoding of a TPDU is represented using a diagram in this clause, the following representation is used:

- a) octets are shown with the lowest numbered octet to the left; higher numbered octets being further to the right;
- b) within an octet, bits are shown with bit 8 to the left and bit 1 to the right.

TPDUs shall contain, in the following order:

- a) the header, comprising:
  - 1) the length indicator (LI) field,
  - 2) the fixed part,
  - 3) the variable part, if present;
- b) the data field, if present.

This structure is illustrated as follows:



### 13.2.1 Length indicator field

This field is contained in the first octet of the TPDU. The length is indicated by a binary number, with a maximum value of 254, (1111 1110). The length indicated shall be the header length in octets including parameters, but excluding the length indicator field and user data, if any. The value 255 (1111 1111) is reserved for possible extensions.

If the length indicated exceeds, or is equal to, the size of the NS user data which is present, this is a protocol error.

### 13.2.2 Fixed part

#### 13.2.2.1 General

The fixed part contains frequently occurring parameters including the code of the TPDU. The length and the structure of the fixed part are defined by the TPDU code and in certain cases by the protocol class and the formats in use (normal or extended).

If any of the parameters of the fixed part have an invalid value, or if the fixed part cannot be contained within the header (as defined by LI), this is a protocol error.

*Note* – In general the TPDU code defines unambiguously the fixed part. However, different variants may exist for the same TPDU code (see normal and extended formats).

#### 13.2.2.2 TPDU code

This field contains the TPDU code and is contained in octet 2 of the header. It is used to define the structure of the remaining header. This field is a full octet except in the following cases:

- 1110 xxxx Connection request
- 1101 xxxx Connection confirm
- 0101 xxxx Reject
- 0110 xxxx Data acknowledgement

where xxxx (bits 4-1) is used to signal the CDT.

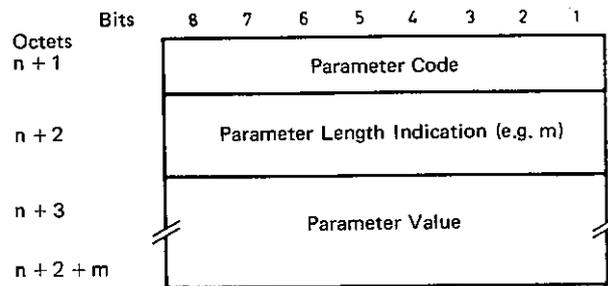
Only those codes defined in § 13.1 are valid.

### 13.2.3 Variable part

The variable part is used to define less frequently used parameters. If the variable part is present, it shall contain one or more parameters.

*Note* – The number of parameters that may be contained in the variable part is indicated by the length of the variable part which is LI minus the length of the fixed part.

Each parameter contained within the variable part is structured as follows:



- The parameter code field is coded in binary.

*Note* – Without extensions, it provides a maximum number of 255 different parameters. However, as noted below, bits 8 and 7 cannot take every possible value, so the practical maximum number of different parameters is less. Parameter code 1111 1111 is reserved for possible extensions of the parameter code.

- The parameter length indication indicates the length, in octets, of the parameter value field.

*Note* – The length is indicated by a binary number,  $m$ , with a theoretical maximum value of 255. The practical maximum value of  $m$  is lower. In the case of a single parameter contained within the variable part, two octets are required for the parameter code and the parameter length indication itself. Thus, the value of  $m$  is limited to 248. For larger fixed parts of the header and for each succeeding parameter, the maximum value of  $m$  decreases.

- The parameter value field contains the value of the parameter identified in the parameter code field.

- No parameter codes use bits 8 and 7 with the value 00.

- The parameters defined in the variable part may be in any order. If any parameter is duplicated, then the later value shall be used. A parameter not defined in this Recommendation shall be treated as a protocol error in any received TPDU except a CR TPDU. In a CR TPDU, it shall be ignored. If the responding transport entity selects a class for which a parameter of the CR TPDU is not defined, it may ignore this parameter, except the class and option and alternative protocol class parameters which shall always be interpreted. A parameter defined in this Recommendation, but having an invalid value, shall be treated as a protocol error in any received TPDU except a CR TPDU. In a CR TPDU, it shall be treated as a protocol error if it is the class and option parameter or the alternative class parameter or the additional option selection parameter; otherwise, it shall be either ignored or treated as a protocol error.

#### 13.2.3.1 Checksum parameter (Class 4 only)

All TPDU types may contain a 16-bit checksum parameter in their variable part. This parameter shall be present in a CR TPDU and shall be present in all other TPDUs except when the non-use of checksum option is selected. It is encoded as follows:

Parameter Code:	1100 0011
Parameter Length:	2
Parameter Value:	Result of checksum algorithm. This algorithm is specified in § 6.17.

#### 13.2.4 Data field

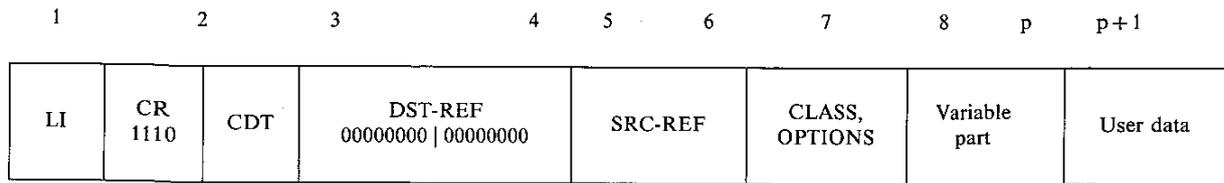
This field contains transparent user data. Restrictions on its size are noted for each TPDU.

#### 13.3 Connection request (CR) TPDU

The length of the CR TPDU shall not exceed 128 octets.

13.3.1 *Structure*

The structure of the CR TPDU is as follows:



13.3.2 *LI*

See § 13.2.1.

13.3.3 *Fixed part (octets 2 to 7)*

The structure of this part shall contain:

- a) CR: Connection request code: 1110 Bits 8-5 of octet 2.
- b) CDT: Initial credit allocation (set to 0000 in Classes 0 and 1 when specified as preferred class). Bits 4-1 of octet 2.
- c) DST-REF: Set to zero.
- d) SRC-REF: Reference selected by the transport entity initiating the CR TPDU to identify the requested transport connection.
- e) CLASS, OPTIONS: Bits 8-5 of octet 7 defines the preferred transport protocol class to be operated over the requested transport connection. This field shall take one of the following values:  
 0000 Class 0  
 0001 Class 1  
 0010 Class 2  
 0011 Class 3  
 0100 Class 4.

The CR contains the first choice of class in the fixed part. Second and subsequent choices are listed in the variable part if required.

Bits 4-1 of octet 7 define options to be used on the requested transport connection as follows:

Bit	Option
4	0 always
3	0 always
2	= 0 use of normal formats in all classes = 1 use of extended formats in Classes 2, 3, 4
1	= 0 use of explicit flow control in Class 2 = 1 no use of explicit flow control in Class 2

Bits related to options particular to a class are not meaningful if that class is not proposed and may take any value.

*Note 1* – The connection establishment procedure (see § 6.5) does not permit a given CR TPDU to request use of transport expedited data transfer service (additional option parameter) and no use of explicit flow control in Class 2 (bit 1 = 1).

*Note 2* – Bits 4 to 1 are always zero in Class 0 and have no meaning.

### 13.3.4 Variable part (octets 8 to p)

The following parameters are permitted in the variable part:

a) *Transport Service Access Point identifier (TSAP-ID)*

Parameter code:                1100 0001 for the identifier of the calling TSAP;  
                                      1100 0010 for the identifier of the called TSAP.  
Parameter length:                not defined in this Recommendation.  
Parameter value:                 identifier of the calling or called TSAP respectively.  
If a TSAP-ID is given in the request, it may be returned in the confirmation.

b) *TPDU size*

This parameter defines the proposed maximum TPDU size (in octets, including the header) to be used over the requested transport connection. The coding of this parameter is:

Parameter code:                1100 0000  
Parameter length:                1 octet  
Parameter value:  
0000 1101                8192 octets (not allowed in Class 0)  
0000 1100                4096 octets (not allowed in Class 0)  
0000 1011                2048 octets  
0000 1010                1024 octets  
0000 1001                512 octets  
0000 1000                256 octets  
0000 0111                128 octets  
Default value is 0000 0111 (128 octets).

c) *Version number* (not used if Class 0 is the preferred class)

Parameter code:                1100 0100  
Parameter length:                1 octet  
Parameter value:                0000 0001  
Default value is 0000 0001 (not used in Class 0).

d) *Protection parameters* (not used if Class 0 is the preferred class)

This parameter is user defined.  
Parameter code:                1100 0101  
Parameter length:                user defined  
Parameter value:                user defined.

e) *Checksum* (used only if Class 4 is the preferred class) (see § 13.2.3.1)

This parameter shall always be present in a CR TPDU requesting Class 4, even if the checksum selection parameter is used to request non-use of the checksum facility.

f) *Additional option selection* (not used if Class 0 is the preferred class)

This parameter defines the selection to be made as to whether or not additional options are to be used.  
Parameter code:                1100 0110  
Parameter length:                1

Parameter value:

Bit	Option
4	= 1 Use of network expedited in Class 1 = 0 Non use of network expedited in Class 1
3	= 1 Use of receipt confirmation in Class 1 = 0 Use of explicit AK variant in Class 1
2	= 0 16-bit checksum defined in § 6.17 is to be used in Class 4 = 1 16-bit checksum defined in § 6.17 is not to be used in Class 4
1	= 1 Use of transport expedited data transfer service = 0 No use of transport expedited data transfer service

Default value is 0000 0001.

Bits 8 to 5 shall be set to zero when sending the TPDU and ignored upon receipt.

Bits related to options particular to a class are not meaningful if that class is not proposed and may take any value.

g) *Alternative protocol class(es)* (not used if Class 0 is the preferred class)

Parameter code: 1100 0111

Parameter length: n

Parameter value: encoded as a sequence of single octets. Each octet is encoded as for octet 7 but with bits 4-1 set to zero (i.e. not alternative option selections permitted).

h) *Acknowledge time* (used only if Class 4 is the preferred class)

This parameter conveys the maximum acknowledge time  $A_L$  to the remote transport entity. It is an indication only, and is not subject to negotiation (see § 12.2.1.1.3).

Parameter code: 1000 0101

Parameter length: 2

Parameter value:  $n$ , a binary number, where  $n$  is the maximum acknowledge time, expressed in milliseconds.

i) *Throughput* (not used if Class 0 is the preferred class)

Parameter code: 1000 1001

Parameter length: 12 or 24

Parameter value:

1st 12 octets: maximum throughput, as follows:

1st 3 octets: target value, calling-called user direction;

2nd 3 octets: min. acceptable, calling-called user direction;

3rd 3 octets: target value, called-calling user direction;

4th 3 octets: min. acceptable, called-calling user direction.

2nd 12 octets (optional): average throughput, as follows:

5th 3 octets: target value, calling-called user direction;

6th 3 octets: min. acceptable, calling-called user direction;

7th 3 octets: target value, called-calling user direction;

8th 3 octets: min. acceptable, called-calling user direction.

Where the average throughput is omitted, it is considered to have the same value as the maximum throughput.

Values are expressed in octets per second.

j) *Residual error rate* (not used if Class 0 is the preferred class)

Parameter code: 1000 0110  
 Parameter length: 3  
 Parameter value:  
     1st octet: target value, power of 10  
     2nd octet: min. acceptable, power of 10  
     3rd octet: TSDU size of interest, expressed as a power of 2.

k) *Priority* (not used if Class 0 is the preferred class)

Parameter code: 1000 0111  
 Parameter length: 2  
 Parameter value: integer (0 is the highest priority).

l) *Transit delay* (not used if Class 0 is the preferred class)

Parameter code: 1000 1000  
 Parameter length: 8  
 Parameter value:  
     1st 2 octets: target value, calling-called user direction;  
     2nd 2 octets: max. acceptable, calling-called user direction;  
     3rd 2 octets: target value, called-calling user direction;  
     4th 2 octets: max. acceptable, called-calling user direction.

Values are expressed in milliseconds, and are based upon a TSDU size of 128 octets.

m) *Reassignment time* (not used if Class 0, 2 or 4 is the preferred class)

This parameter conveys the Time to Try Reassignment/Resynchronization (TTR) which will be used when following the procedure for reassignment after failure (see § 6.12).

Parameter code: 1000 1011  
 Parameter length: 2  
 Parameter value:  $n$ , a binary number, where  $n$  is the TTR value expressed in seconds.

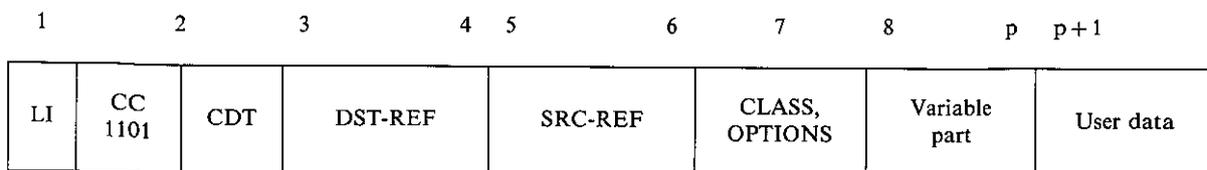
### 13.3.5 *User data (octets p+1 to the end)*

No user data are permitted in Class 0, and are optional in the other classes. Where permitted, it may not exceed 32 octets.

## 13.4 *Connection confirm (CC) TPDU*

### 13.4.1 *Structure*

The structure of the CC TPDU is as follows:



### 13.4.2 *LI*

See § 13.2.1.

13.4.3 *Fixed part (octets 2 to 7)*

The fixed part shall contain:

- a) CC: Connection confirm code: 1101. Bits 8-5 of octet 2.
- b) CDT: Initial credit allocation (set to 0000 in Classes 0 and 1). Bits 4-1 of octet 2.
- c) DST-REF: Reference identifying the requested transport connection at the remote transport entity.
- d) SRC-REF: Reference selected by the transport entity initiating the CC TPDU to identify the confirmed transport connection.
- e) CLASS, OPTIONS: Defines the selected transport protocol class and options to be operated over the accepted transport connection according to the negotiation rules specified in § 6.5.

13.4.4 *Variable part (octet 8 to p)*

The parameters are defined in § 13.3.4 and are subject to the constraints stated in § 6.5 (connection establishment). Parameters ruled out by selection of an alternative class and option shall not be present.

13.4.5 *User data (octets p+1 to the end)*

No user data are permitted in Class 0, and are optional in the other classes. Where permitted, it may not exceed 32 octets. The user data are subject to the constraints of the negotiation rules of § 6.5.

13.5 *Disconnect request (DR) TPDU*

13.5.1 *Structure*

The structure of the DR TPDU is as follows:

	1	2	3	4	5	6	7	8	p	p+1
LI	DR 1000 0000		DST-REF	SRC-REF	REASON	Variable part		User data		

13.5.2 *LI*

See § 13.2.1.

13.5.3 *Fixed part (octets 2 to 7)*

The fixed part shall contain:

- a) DR: Disconnect request code: 1000 0000.
- b) DST-REF: Reference identifying the transport connection at the remote transport entity.
- c) SRC-REF: Reference identifying the transport connection at the transport entity initiating the TPDU. Value zero when reference is unassigned.
- d) REASON: Defines the reason for disconnecting the transport connection. This field shall take one of the following values:

The following values can be used for Classes 1 to 4:

- 1) 128 + 0 – Normal disconnect initiated by the session entity;
- 2) 128 + 1 – Remote transport entity congestion at connect request time;
- 3) \* 128 + 2 – Connection negotiation failed (i.e. proposed class(es) not supported);
- 4) 128 + 3 – Duplicate source reference detected for the same pair of NSAPs;
- 5) 128 + 4 – Mismatched references;

- 6) 128 + 5 – Protocol error;
- 7) 128 + 6 – Not used;
- 8) 128 + 7 – Reference overflow;
- 9) 128 + 8 – Connection request refused on this network connection;
- 10) 128 + 9 – Not used;
- 11) 128 + 10 – Header or parameter length invalid;

The following values can be used for all classes:

- 12) 0 – Reason not specified;
- 13) 1 – Congestion at TSAP;
- 14) \* 2 – Session entity not attached to TSAP;
- 15) \* 3 – Address unknown.

*Note* – Reasons marked with “\*” may be reported to the TS-User as “persistent”, other reasons as “transient”.

#### 13.5.4 Variable part (octets 8 to p)

The variable part may contain:

- a) *Additional information* related to the clearing of the connection.

Parameter code: 1110 0000

Parameter length: Any value provided that the length of the DR TPDU does not exceed the maximum agreed TPDU size or 128 when the DR TPDU is used during the connection refusal procedure.

Parameter value: Additional information. The content of this field is user defined.

- b) *Checksum*: shall be present if the condition in § 13.2.3.1 applies.

#### 13.5.5 User data (octets p+1 to the end)

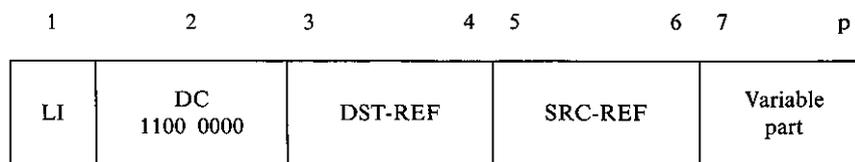
This field shall not exceed 64 octets and is used to carry TS-user data. The successful transfer of this data is not guaranteed by the transport protocol. When a DR TPDU is used in Class 0, it shall not contain this field.

#### 13.6 Disconnect confirm (DC) TPDU

This TPDU shall not be used in Class 0.

##### 13.6.1 Structure

The structure of the DC TPDU shall be as follows:



##### 13.6.2 LI

See § 13.2.1.

##### 13.6.3 Fixed part (octets 2 to 6)

The fixed part shall contain:

- a) DC: Disconnect confirm code: 1100 0000.
- b) DST-REF: See § 13.4.3.
- c) SRC-REF: See § 13.4.3.

13.6.4 *Variable part*

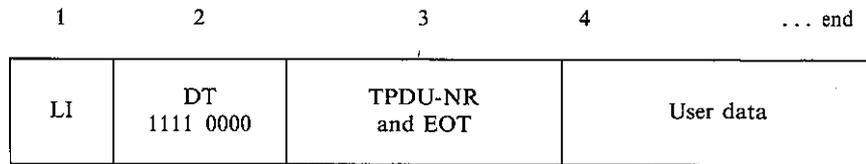
The variable part shall contain the checksum parameter if the condition in § 13.2.3.1 applies.

13.7 *Data (DT) TPDU*

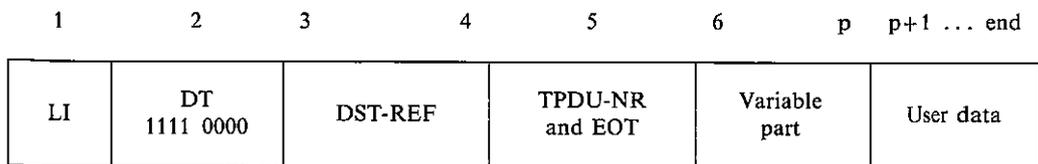
13.7.1 *Structure*

Depending on the class and the option, the DT TPDU shall have one of the following structures:

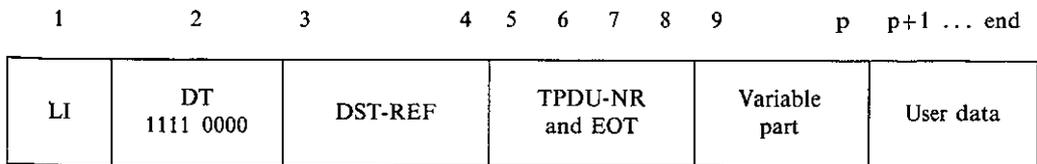
a) *Normal format for Classes 0 and 1*



b) *Normal format for Classes 2, 3 and 4*



c) *Extended Format for use in Classes 2, 3 and 4 when selected during connection establishment*



13.7.2 *LI*

See 13.2.1.

13.7.3 *Fixed part*

The fixed part shall contain:

- a) DT: Data transfer code: 1111 0000.
- b) DST-REF: See § 13.4.3.
- c) EOT: When set to ONE, indicates that the current DT TPDU is the last data unit of a complete DT TPDU sequence (end of TSDU). EOT is bit 8 of octet 3 in Classes 0 and 1 and bit 8 octet 5 for normal formats for Classes 2, 3 and 4.
- d) TPDU-NR: TPDU Send Sequence Number (zero in Class 0). May take any value in Class 2 without explicit flow control. TPDU-NR is bits 7-1 of octet 3 for Classes 0 and 1, bits 7-1 of octet 5 for normal formats in Classes 2, 3 and 4 and bits 7-1 of octet 5 together with octets 6, 7 and 8 for extended formats.

*Note* – Depending on the class, the fixed part of the DT TPDU uses the following octets:

- Classes 0 and 1: Octets 2 to 3.
- Classes 2, 3, 4 normal format: Octets 2 to 5.
- Classes 2, 3, 4 extended format: Octets 2 to 8.

13.7.4 *Variable part*

The variable part shall contain the checksum parameter if the condition in § 13.2.3.1 applies.

13.7.5 *User data field*

This field contains data of TSDU being transmitted.

*Note* – The length of this field is limited to the negotiated TPDU size for this transport connection minus 3 octets in Classes 0 and 1 and minus 5 octets (normal header format) or 8 octets (extended header format) in other classes. The variable part, if present, may further reduce the size of the user data field.

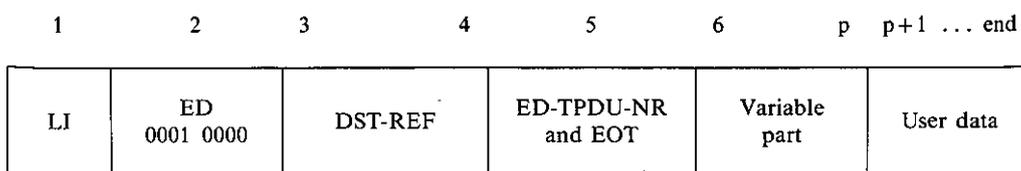
13.8 *Expedited data (ED) TPDU*

The ED TPDU shall not be used in Class 0 or in Class 2 when the “no explicit flow control” option is selected, or when the expedited data transfer service has not been selected for the connection.

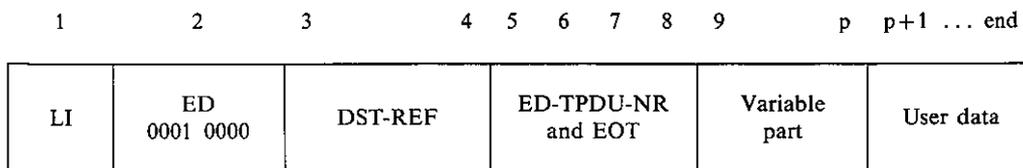
13.8.1 *Structure*

Depending on the format negotiated at connection establishment, the ED TPDU shall have one of the following structures:

a) *Normal format (Classes, 1, 2, 3, 4)*



b) *Extended format (for use in Classes 2, 3, 4 when selected during connection establishment)*



13.8.2 *LI*

See § 13.2.1.

13.8.3 *Fixed part*

The fixed part shall contain:

- a) ED: Expedited data code: 0001 0000.
- b) DST-REF: See § 13.4.3.
- c) ED-TPDU-NR: Expedited TPDU identification number. ED-TPDU-NR is used in Classes 1, 3 and 4 and may take any value in Class 2. Bits 7-1 of octet 5 for normal formats and bits 7-1 of octet 5 together with octets 6, 7 and 8 for extended formats.
- d) EOT: end of TSDU always set to 1 (bit 8 of octet 5).

*Note* – Depending on the format, the fixed part shall be either octets 2 to 5 or 2 to 8.

13.8.4 *Variable part*

The variable part shall contain the checksum parameter if the condition in § 13.2.3.1 applies.

13.8.5 *User data field*

This field contains an expedited TSDU (1 to 16 octets).

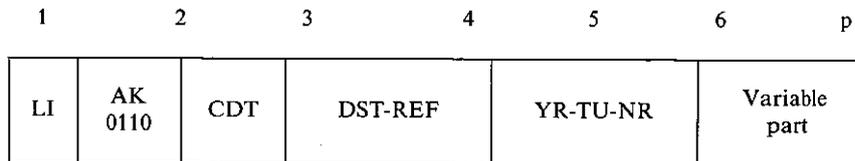
### 13.9 Data acknowledgement (AK) TPDU

This TPDU shall not be used for Class 0; for Class 2 when the “no explicit flow control” option is selected; for Class 1 when the network receipt confirmation option is selected.

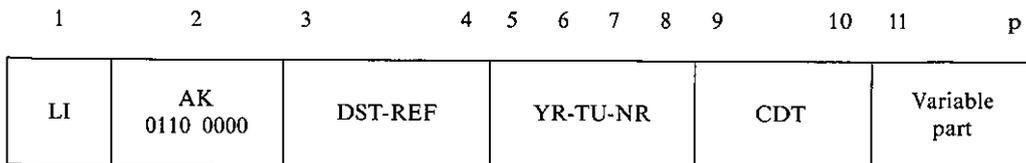
#### 13.9.1 Structure

Depending on the class and option agreed, the AK TPDU shall have one of the following structures:

a) Normal format (Classes 1, 2, 3, 4)



b) Extended format (for use in Classes 2, 3, 4 when selected during connection establishment)



#### 13.9.2 LI

See § 13.2.1.

#### 13.9.3 Fixed part

The fixed part shall contain (in octets 2 to 5 when normal format is used, 2 to 10 otherwise) the following parameters:

- a) AK: Acknowledgement code: 0110.
- b) CDT: Credit value (set to 1111 in Class 1) bits 4-1 of octet 2 for normal formats and octets 9 and 10 for extended formats.
- c) DST-REF: See § 13.4.3.
- d) YR-TU-NR: Sequence number indicating the next expected DT TPDU number. For normal formats, bits 7-1 of octet 5; bit 8 of octet 5 is not significant and shall take the value 0. For extended formats, bits 7-1 of octet 5 together with octets 6, 7 and 8; bit 8 octet 5 is not significant and shall take the value 0.

#### 13.9.4 Variable part

The variable part contains the following parameters:

- a) *Checksum*: shall be present if the condition in § 13.2.3.1 applies.
- b) *Sub-sequence number* (when optionally used under the conditions defined in Class 4).

This parameter is used to ensure that AK TPDUs are processed in correct sequence. If it is absent, this is equivalent to transmitting the parameter with a value of zero.

Parameter code: 1000 1010.

Parameter length: 2.

Parameter value: 16-bit sub-sequence number.

c) *Flow control confirmation (when optionally used under the conditions defined in Class 4)*

This parameter contains a copy of the information received in an AK TPDU, to allow the transmitter of the AK TPDU to be certain of the state of the receiving transport entity (see § 12.2.3.9).

Parameter code: 1000 1100.

Parameter length: 8.

Parameter value: defined as follows:

1) Lower window edge (32 bits)

Bit 8 of octet 1 of the parameter value field is set to zero, the remainder contain the YR-TU-NR value of the received AK TPDU. When normal format has been selected, only the least significant seven bits (bits 1 to 7 of octet 4 of the parameter value field) of this field are significant.

2) Your sub-sequence (16 bits).

Contains the value of the sub-sequence parameter of the received AK TPDU, or zero if this parameter was not present.

3) Your credit (16 bits).

Contains the value of the CDT field of the received AK TPDU. When normal format has been selected, only the least significant four bits (bits 1 to 4 of octet 2 of the parameter value field) of this field are significant.

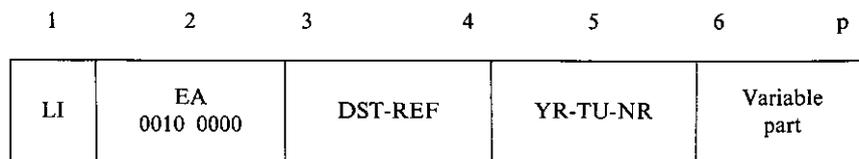
13.10 *Expedited data acknowledgment (EA)*

The EA TPDU shall not be used for Class 0, or for Class 2 when the “no explicit flow control” option is selected, or when the expedited data transfer service has not been selected for the connection.

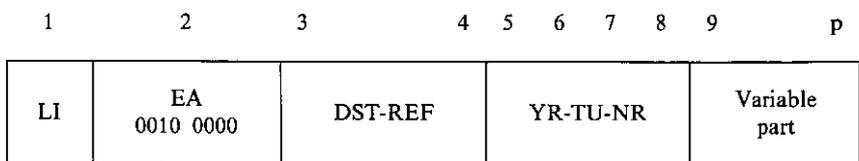
13.10.1 *Structure*

Depending on the option (normal or extended format) the TPDU structure shall be:

a) *Normal format (Classes 1, 2, 3, 4)*



b) *Extended format (for use in Classes 2, 3, 4 if selected during connection establishment)*



13.10.2 *LI*

See § 13.2.1.

13.10.3 *Fixed part*

The fixed part shall contain (in octets 2 to 5 when normal format is used, in octets 2 to 8 otherwise):

- a) EA: Expedited acknowledgement code: 0010 0000.
- b) DST-REF: See § 13.4.3.
- c) YR-EDTU-NR: Identification of the ED TPDU being acknowledged. May take any value in Class 2. For normal formats bits 7-1 of octet 5; bit 8 of octet 5 is not significant and shall take the value 0. For extended formats, bits 7-1 of octet 5 together with octets 6, 7 and 8; bit 8 of octet 5 is not significant and shall take the value 0.

13.10.4 *Variable part*

The variable part shall contain the checksum parameter if the condition in § 13.2.3.1 applies.

13.11 *Reject (RJ) TPDU*

The RJ TPDU shall not be used in Classes 0, 2 and 4.

13.11.1 *Structure*

The RJ TPDU shall have one of the following formats:

a) *Normal format (Classes 1 and 3)*

1	2	3	4	5
LI	RJ 0101	CDT	DST-REF	YR-TU-NR

b) *Extended format (for use in Class 3 if selected during connection establishment)*

1	2	3	4	5	6	7	8	9	10
LI	RJ 0101 0000	DST-REF	YR-TU-NR	CDT					

13.11.2 *LI*

See § 13.2.1.

13.11.3 *Fixed part*

The fixed part shall contain (in octets 2 to 5 when normal format is used, in octets 2 to 10 otherwise):

- a) RJ: Reject code: 0101. Bits 8-5 of octet 2.
- b) CDT: Credit value (set to 1111 in Class 1). Bits 4-1 of octet 2 for normal formats and octets 9 and 10 for extended formats.
- c) DST-REF: See § 13.4.3.
- d) YR-TU-NR: Sequence number indicating the next expected TPDU from which retransmission should occur. For normal formats, bits 7-1 of octet 5; bit 8 of octet 5 is not significant and shall take the value 0. For extended formats, bits 7-1 of octet 5 together with octets 6, 7 and 8; bit 8 of octet 5 is significant and shall take the value 0.

13.11.4 *Variable part*

There is no variable part for this TPDU type.

13.12 *TPDU error (ER) TPDU*

13.12.1 *Structure*

1	2	3	4	5	6	p
LI	ER 0111 0000	DST-REF	Reject cause	Variable part		

### 13.12.2 *LI*

See § 13.2.1.

### 13.12.3 *Fixed part*

The fixed part shall contain:

- a) ER: TPDU error code: 0111 0000.
- b) DST-REF: See § 13.4.3.
- c) Reject cause:
  - 0000 0000 Reason not specified.
  - 0000 0001 Invalid parameter code.
  - 0000 0010 Invalid TPDU type.
  - 0000 0011 Invalid parameter value.

### 13.12.4 *Variable part*

The variable part may contain the following parameters:

- a) *Invalid TPDU*
  - Parameter code: 1100 0001
  - Parameter length: Number of octets of the value field;
  - Parameter value: Contains the bit pattern of the rejected TPDU header up to and including the octet which caused the rejection. This parameter is mandatory in Class 0.
- b) *Checksum*: shall be present if the condition in § 13.2.3.1 applies.

## 14 **Conformance**

14.1 A system claiming to implement the procedures specified in this Recommendation shall comply with the requirements in §§ 14.2-14.4.

14.2 The system shall implement Class 0.

14.3 If the system implements Class 3 or Class 4, it shall also implement Class 2.

14.4 For each class which the system claims to implement, the system shall be capable of:

- a) initiating CR TPDU's or responding to CR TPDU's with CC TPDU's or both. Where a CR TPDU proposing Class 2, 3 or 4 is initiated, Class 0 shall be explicitly indicated as an alternative class except if there is already one (or several) transport connection(s) assigned to the network connection (multiplexing being possible);
- b) responding to any other TPDU and operating network service in accordance with the procedures for the class;
- c) operating all the procedures for the class listed as mandatory in Table 9/X.224;
- d) operating those procedures for the class listed as optional in Table 9/X.224 for which conformance is claimed;
- e) handing all TPDU's of lengths up to the lesser value of:
  - 1) the maximum length for the class [see § 13.3.4 b)];
  - 2) the maximum for which conformance is claimed (see Note 2).

*Note 1* – The procedures or classes 0-4 are specified in §§ 8-12 respectively. The procedures refer to the elements of procedures specified in § 6.

*Note 2* – The requirement in § 14.4.e indicates that TPDU sizes of 128 octets are always implemented.

14.5 Claims of conformance shall state:

- a) which class or classes of protocol are implemented;
- b) whether the system is capable of initiating or responding to CR TPDU's or both;
- c) which of the procedures listed as optional in Table 9/X.224 are implemented;
- d) for each class, the maximum size of TPDU implemented; the value shall be chosen from the following list and all values in the list which are less than this maximum shall be implemented [see § 13.3.4 b)]:  
128, 256, 512, 1024, 2048, 4096 or 8192 octets.

TABLE 9/X.224

Provision of options

Procedure	Class 0	Class 1	Class 2	Class 3	Class 4
TPDU with checksum	NA	NA	NA	NA	M
TPDU without checksum	M	M	M	M	O
Expedited data transfer	NA	M	M	M	M
No expedited data transfer	M	M	M	M	M
Flow control in Class 2	NA	NA	M	NA	NA
No flow control in Class 2	NA	NA	O	NA	NA
Normal formats	M	M	M	M	M
Extended formats	NA	NA	O	O	O
Use of receipt confirmation in Class 1	NA	O	NA	NA	NA
No use of receipt confirmation in Class 1	NA	M	NA	NA	NA
Use of network expedited in Class 1	NA	O	NA	NA	NA
No use of network expedited in Class 1	NA	M	NA	NA	NA

NA: not applicable

M: mandatory

O: optional

## ANNEX A

(to Recommendation X.224)

### State tables

This Annex provides a more precise description of the protocol.

The state tables also define the mapping between service and protocol events that TS-users can expect.

This Annex describes the transport protocol in terms of state tables. The state tables show the state of a transport connection, the events that occur in the protocol, the actions taken, and the resultant state.

The state tables only describe the operation of a single transport connection. They do not necessarily describe all possible combinations of sequences of events at the transport and network service boundaries, nor do they describe the exact mapping between TPDU's and NSDU's.

#### A.1 *Conventions*

A.1.1 The incoming events are represented in the state tables by their abbreviated name as defined in Table A-1/X.224;

A.1.2 The states are represented in the tables by their abbreviated name as defined in Table A-2/X.224;

A.1.3 The intersection of each state and event which is invalid is left blank. The action to be taken in this case is one of the following:

- a) for an event related to the transport service (i.e. coming from the TS-user), take no action;
- b) for an event related to a received TPDU, follow the procedure for treatment of protocol errors (see § 6.22) if the state of the supporting network connection makes it possible;
- c) for an event falling into neither of the above categories (including those which are impossible by the definition of the behaviour of the transport entity or NS-provider) take no action.

A.1.4 At each intersection of state and event which is valid the state tables specify an action which may include one of the following:

- a) one action constituted of a list of any number of outgoing events (none, one, or more) given by their abbreviated name defined in Table A-3/X.224 followed by the abbreviated name of the resultant state (see Table A-2/X.224);
- b) conditional actions separated by a semi-colon (;). Each conditional action contains a predicate followed by a colon (:) and by an action as defined in § A.1.4 a). The predicates are boolean expressions given by their abbreviated name and defined in the clauses related to the state tables of each class. Only the action corresponding to the predicate which is true is to be taken.

A.1.5 The state tables also include:

- a) informal comments giving explanatory material;
- b) references to notes using the following notation: (note number);
- c) references to other actions defined in separate tables using the following notation: (action number).

#### A.2 *General*

Table A-1/X.224 specifies the names and abbreviated names of the incoming events, classified as TS-user events, NS-provider events or TPDU events.

Table A-2/X.224 specifies the names and abbreviated names of the states.

Table A-3/X.224 specifies the names and abbreviated names of the outgoing events classified as TS-provider events, NS-user events or TPDU events.

TABLE A-1/X.224

## Incoming events

Abbreviated name	Category	Name
TCONreq	TS-user	T-CONNECT request primitive
TCONresp	TS-user	T-CONNECT response primitive
TDTreq	TS-user	T-DATA request primitive
TEXreq	TS-user	T-EXPEDITED DATA request primitive
TDISreq	TS-user	T-DISCONNECT request primitive
NDISind	NS-provider	N-DISCONNECT indication primitive
NCONconf	NS-provider	N-CONNECT confirm primitive
NRSTind	NS-provider	N-RESET indication primitive
CR	TPDU	Connection request TPDU
CC	TPDU	Connection confirm TPDU
DR	TPDU	Disconnect request TPDU
DC	TPDU	Disconnect confirm TPDU
AK	TPDU	Data acknowledgement TPDU
EA	TPDU	Expedited data acknowledgement TPDU
DT	TPDU	Data TPDU
ED	TPDU	Expedited data TPDU
ER	TPDU	TPDU error TPDU
RJ	TPDU	Rejet TPDU

TABLE A-2/X.224

## States

Abbreviated name	Name
WFNC	Wait for network connection
WFCC	Wait for the CC TPDU
WBCL	Wait before releasing (wait for CC TPDU before sending the DR TPDU)
OPEN	Transport connection is open
CLOSING	Release in progress
WFTRESP	Wait for T-CONNECT response
CLOSED	Transport connection is closed
WFNC-R	Wait for network connection and reassignment in progress
WFCC-R	Wait for CC TPDU and reassignment in progress
WBCL-R	Wait before releasing and reassignment in progress
OPEN-R	Open and reassignment in progress
OPEN-WR	Open and wait for reassignment
CLOSING-R	Release in progress and reassignment in progress
CLOSING-WR	Release in progress and wait for reassignment
WFTRESP-WR	Wait for T-CONNECT response and wait for reassignment
WBCL-WR	Wait before releasing and wait for reassignment
WBOC	Wait before open complete (CC is unacknowledged)
WBOC-WR	Wait before open complete and wait for reassignment
CLOSING BOC	Wait before open complete and release in progress
CLOSING BOC-WR	Idem and wait for reassignment
AKWAIT	Wait for acknowledgement of CC TPDU
REFWAIT	Waiting for frozen reference time

TABLE A-3/X.224

## Outgoing events

Abbreviated name	Category	Name
TCONind	TS-provider	T-CONNECT indication primitive
TCONconf	TS-provider	T-CONNECT confirm primitive
TDTind	TS-provider	T-DATA indication primitive
TEXind	TS-provider	T-EXPEDITED DATA indication primitive
TDISind	TS-provider	T-DISCONNECT indication primitive
NDISreq	NS-user	N-DISCONNECT request primitive
NRSTresp	NS-user	N-RESET response primitive
NCONreq	NS-user	N-CONNECT request primitive
CR	TPDU	Connection request TPDU
CC	TPDU	Connection confirm TPDU
DR	TPDU	Disconnect request TPDU
DC	TPDU	Disconnect confirm TPDU
AK	TPDU	Data acknowledgement TPDU
EA	TPDU	Expedited data acknowledgement TPDU
DT	TPDU	Data TPDU
ED	TPDU	Expedited data TPDU
ER	TPDU	TPDU error TPDU
RJ	TPDU	Reject TPDU

A.3 *State tables for Classes 0 and 2*

This section provides a more precise description of a transport entity for a transport connection of Class 0 or Class 2.

The description uses predicates defined in Table A-4/X.224, and specific actions defined in Table A-5/X.224.

The description does not include a complete specification of the data transfer procedures but makes reference to the specification of the classes (see §§ 8 and 10). Table A-6/X.224 gives the state automata for Classes 0 and 2.

TABLE A-4/X.224  
**Predicates for classes 0 and 2**

Name	Description
P0	T-CONNECT request unacceptable
P1	Unacceptable CR TPDU
P2	No network connection available
P3	Network connection available and open
P4	Network connection available and open in progress
P5	Class is class 0 (class selected in CC)
P6	Unacceptable CC
P7	Class is class 2
P8	Acceptable CC
P9	Class 4 CR

TABLE A-5/X.224  
**Specific actions for classes 0 and 2**

Name	Description
[1]	If the network connection is not used by another transport connection assigned to it, it may be released.
[2]	See § 6.22 (receipt of an ER TPDU)
[3]	See data transfer procedures of the class
[4]	See expedited data transfer procedures of the class
[5]	An N-RESET response has to be issued once for the network connection if the network connection has not been released. In class 0, an N-DISCONNECT request has to be issued.

TABLE A-6/X.224

State table for classes 0 and 2 (part 1 of 2)

STATE EVENT	WFNC	WFCC	WBCL (Class 2 only)	OPEN	CLOSING (Class 2 only)	WFTRESP	CLOSED
TCONreq							P0: TDISind CLOSED; P2: NCONreq WFNC; P3: CR WFCC; P4: WFNC
TCONresp						CC OPEN	
TDTreq				[3] OPEN			
TEXreq	DOES NOT EXIST IN CLASS 0						
				[4] OPEN			
TDISreq	[1] CLOSED	not P7: NDISreq CLOSED P7: WBCL		P5: NDISreq CLOSED; P7: DR CLOSING		DR CLOSED	
NCONconf	CR WFCC						
NRSTind		TDISind [1] [5] CLOSED	[1] [5] CLOSED	TDISind [1] [5] CLOSED	[1] [5] CLOSED	TDISind [1] [5] CLOSED	
NDISind	TDISind CLOSED	TDISind CLOSED	CLOSED	TDISind CLOSED	CLOSED	TDISind CLOSED	
CR				P9: OPEN	P9: CLOSING	P9: WFTRESP	P1: DR (1) CLOSED; not P1: TCONind WFTRESP
DR		TDISind [1] CLOSED	[1] CLOSED	P5: (2); P7: DC TDISind CLOSED	[1] CLOSED		(4) CLOSED;
							DC CLOSED

TABLE A-6/X.224

State table for classes 0 and 2 (Part 2 of 2)

STATE EVENT	WFNC	WFCC	WBCL (Class 2 only)	OPEN	CLOSING (Class 2 only)	WFTRESP	CLOSED
DC	DOES NOT EXIST IN CLASS 0 (2)						
					P7: [1] CLOSED		CLOSED
CC		P8: TCONconf OPEN; P6 and P5: TDISind NDISreq CLOSED; P6 and P7: TDISind DR CLOSING	P5: (3) NDISreq CLOSED; P7: DR CLOSING				DR CLOSED
AK	DOES NOT EXIST IN CLASS 0 (2)						
				[3] OPEN	CLOSING		CLOSED
EA	DOES NOT EXIST IN CLASS 0 (2)						
				[4] OPEN	CLOSING		CLOSED
ED	DOES NOT EXIST IN CLASS 0 (2)						
				[4] OPEN	CLOSING		CLOSED
DT				[3] OPEN	CLOSING		CLOSED
ER		TDISind [1] CLOSED	[1] CLOSED	[2]	[2]		CLOSED

Note 1 – An ER TPDU shall be sent in certain cases (see § 6.6).

Note 2 – If received, it shall be processed as a protocol error (see § 6.22).

Note 3 – A CR with Class 2 has been sent and a CC Class 0 is received.

Note 4 – If DC is not available (i.e. Class 0 only implemented), or SRC-REF is zero.

A.4 *State tables for Classes 1 and 3*

This section provides a more precise description of a transport entity for a transport connection of Class 1 or Class 3.

The description uses the predicates defined in Table A-7/X.224.

Specific actions are defined in Table A-8/X.224 and specific additional notes are given in Table A-9/X.224.

The description does not include a complete specification of the data transfer procedures but makes reference to the specification of the classes (see §§ 9 and 11). Table A-10/X.224 gives the state automata for Classes 1 and 3.

TABLE A-7/X.224

**Predicates for classes 1 and 3**

Name	Description
P0	T-CONNECT request unacceptable
P1	No available network connection can be used for assignment or reassignment
P2	A network connection can be used for assignment or reassignment; the network connection opening is in progress
P3	A network connection can be used for assignment or reassignment; the network connection is open
P4	TTR timer has previously run out
P5	Local choice
P6	Initiator of the transport connection
P7	Unacceptable CR TPDU
P8	TWR is running
P9	Class 4 CR
P10	Class selected in CC is class 0 or 2

TABLE A-8/X.224

## Specific actions for classes 1 and 3

Name	Description
[1]	The network connection can be disconnected if not used by any transport connection assigned to it.
[2]	Process TDT request or TEX request which have been stored when waiting for reassignment (if any). If an RJ TDPDU has been received, enable also data TPDU transmission (if any). If an ED TPDU was received, handle according to procedures for class if not a duplicate.
[3]	Network connection can be disconnected if not used by any transport connection and was locally opened.
[4]	Start TWR timer if not already running. Disable sending DT TPDUs until an RJ TPDU is received (see Note 3).
[5]	Stop TWR timer.
[6]	Issue an N-RESET response if not already done.
[7]	See data transfer procedure for the class.
[8]	Start TTR timer if not already running.
[9]	Stop TTR timer if running or remove information that TTR timer has run out (see Notes 1 and 2).
[10]	Store information that TTR timer has run out (see Note 1).
[11]	Store request.
[12]	See state table appropriate to class selected in CC.

*Note 1* – The information is used by predicate P4.

*Note 2* – This action is not performed if the transport entity is the responder or if neither reassignment nor resynchronization is in progress.

*Note 3* – The method of disabling transmission of DT-TPDUs is a local matter. In Class 3 for example, it may be effected by setting credit to zero. In Class 1, this may be effected by the setting of a boolean indicator.

TABLE A-9/X.224

## Specific notes for classes 1 and 3

Name	Description
(1)	Any TPDU except DR and CC having an unknown destination reference.
(2)	CC TPDU having an unknown destination reference or a mismatched source reference.
(3)	CR TPDU which is not duplicated but rejected. If CR TPDU is duplicated, ignore it.
(4)	Or send any DT or ED TPDU waiting for transmission or use N-DATA ACKNOWLEDGE request if available and selected (class 1 only).
(5)	Same as for (9) and issue a T-DISCONNECT indication.
(6)	If the resultant state is CLOSED, the reference shall be frozen except in the cases described in § 6.18.
(7)	An ER TPDU shall be sent in certain cases (see § 6.6).
(8)	Receipt of a DC TPDU is a protocol error since DC cannot be used for reassignment. It is suggested to stop the TWR timer [5] and to consider the transport connection as released (CLOSED state).
(9)	Receipt of one of these TPDU's in this state is a protocol error. It is suggested to stop the TWR timer [5], send a DR TPDU and enter the CLOSING state.
(10)	Or a DR with mismatched source reference has been received.

TABLE A-10/X.224

State table for classes 1 and 3 (part 1 of 3)  
(connection – responder side)

STATE EVENT	CLOSED	WFTRESP	WFTRESP- WR	WBCL- WR	WBOC	WBOC- WR	CLOSING BOC	CLOSING BOC-WR
TDISreq		DR CLOSED (6)	WBCL- WR		DR CLOSING BOC	CLOSING BOC- WR		
TCONreq		P10: [12]; not P10: CC WBOC	WBOC-WR					
NRSTind		[4] [6] WFTRESP- WR	[6] WFTRESP- WR	[6] WBCL- WR	[4] [6] WBOC- WR	[6] WBOC- WR	[4] [6] CLOSING BOC-WR	[6] CLOSING BOC-WR
NDISind		[4] WFTRESP- WR	WFTRESP- WR	WBCL- WR	[4] WBOC- WR	WBOC- WR	[4] CLOSING BOC-WR	CLOSING BOC-WR
CR	P7: DR (3.7) CLOSED (6); not P7: TCONind WFTRESP	P9: WFTRESP	[5] WFTRESP	[5] DR CLOSED (6)	P9: WBOC	[5] CC WBOC	P9: BOC	[5] DR CLOSED (6)
DR	DC CLOSED				TDISind DC CLOSED (6)	DC [5] TDISind CLOSED (6)	CLOSED (6)	[5] DC CLOSED (6)
RJ or ED	CLOSED				OPEN [7]	[5] [2] RJ OPEN	CLOSING	[5] DR CLOSING
DC	CLOSED						CLOSED (6)	(8)
ER					TDISind DR CLOSING BOC		CLOSED (6)	
First TPDU other than CR, ER, DR, DC, ED or RJ	CLOSED				OPEN [7]		CLOSING	(9)
TWR Time-out			TDISind CLOSED (6)	CLOSED (6)		TDISind CLOSED (6)		CLOSED (6)
TDTreq					[7] WBOC	[11] WBOC-WR		
TEXreq					[7] WBOC	[11] WBOC-WR		

TABLE A-10/X.224

State table for classes 1 and 3 (part 2 of 3)  
(connection – initiator side)

STATE EVENT	CLOSED	WFNC	WFNC-R	WFCC	WFCC-R	WBCL	WBCL-R
TCONreq	P0: TDISind CLOSED; not P0 and P1: NCONreq WFNC; not P0 and P2: WFNC; not P0 and P3: CR WFCC						
NCONconf		CR WFCC	CR WFCC		CR WFCC		CR WBCL
NRSTind				P4: TDISind [1] [6] CLOSED (6); not P4: CR [6] [8] WFCC		P4: [6] CLOSED [1]; not P4: CR [6] [8] WBCL	
NDISind		P1: NCONreq [2] WFNC-R [8]; P2: [5]; WFNC-R; P3: CR [8] WFCC	P1: NCONreq WFNC-R; P2: WFNC-R; P3: CR WFCC	P4: TDISind CLOSED (6); (not P4) and P1: [8] NCONreq WFCC-R; (not P4) and P2: [8] WFCC-R; (not P4) and P3: [8] CR WFCC	[2] WFCC-R	P4 or P5: [1] [9] CLOSED (6); (not P4 or P5) and P1: [8] NCONreq WBCL-R; (not P4 or P5) and P2: [8] WBCL-R; (not P4 or P5) and P3: [8] CR WBCL	P5: [9] CLOSED (6); (not P5) and P1: NCONreq WBCL-R; (not P5) and P2: WBCL-R (not P5) and P3: CR WBCL
TDISreq		[1] CLOSED (6)	[1] CLOSED (6) [9]	WBCL	P5: [1] [9] CLOSED (6); not P5: WBCL-R		
DR	(10) DC CLOSED			TDISind [1] [9] CLOSED (6)		[1] [9] CLOSED (6)	
CC	DR CLOSED			P10: [12]; not P10: TCONconf AK (4) [9] OPEN		P10: [12]; not P10: DR [9] CLOSING	
ER				TDISind [1] [9] CLOSED (6)		[1] [9] CLOSED (6)	
(1)	CLOSED						
(2)	DR CLOSED						
Time-out TTR			TDISind [1] [9] CLOSED (6)	[10]	TDISind [1] [9] CLOSED (6)	[10]	[1] CLOSED (6)

TABLE A-10/X.224

State table for classes 1 and 3 (part 3 of 3)  
(open and closing states)

STATE EVENT	OPEN	OPEN-R	OPEN-WR	CLOSING	CLOSING-R	CLOSING- WR
NCONconf		RJ [2] OPEN			DR CLOSING	
TDISreq	P8: CLOSING; not P8: DR CLOSING	CLOSING-R	CLOSING- WR			
NRSTind	P6 and P4: [6] TDISind [3] CLOSED (6); P6 and not P4: [6] [2] [8] RJ OPEN; not P6: [4, 6] OPEN			P6 and P4: [6] [3] CLOSED (6) P6 and not P4: [6] [8] DR CLOSING; not P6: [4, 6] CLOSING		
NDISind	P6 and P4: TDISind CLOSED (6); (P6 and not P4) and P1: [8] NCONreq OPEN-R; (P6 and not P4) and P2: [8] [2] [8] OPEN-R; (P6 and not P4) and P3: RJ OPEN not P6: [4] OPEN-WR	P1: NCONreq OPEN-R; P2: OPEN-R; P3: [2] RJ OPEN		P6 and (P4 or P5) CLOSED (6); P6 and not (P4 or P5) and P1: [8] NCONreq CLOSING-R; P6 and not (P4 or P5) and P2: [8] CLOSING-R; P6 and not (P4 or P5) and P3: [8] DR CLOSING; not P6: [4] CLOSING-WR	P5: CLOSED (6); (not P5) and P1: NCONreq CLOSING-R; (not P5) and P2: CLOSING-R; (not P5) and P3: DR CLOSING	
RJ	P8: [5] [2] RJ OPEN; not P8: [7] [9] OPEN		RJ [5] [2] OPEN	P8: [5] DR CLOSING; not P8: [9] CLOSING		DR [5] CLOSING
Time-out TWR	TDISind CLOSED (6)		TDISind CLOSED (6)	CLOSED (6)		CLOSED (6)
DR	P8: TDISind DC (6) [5] CLOSED; not P8: TDISind DC (6) [9] CLOSED		[5] DC TDISind CLOSED (6)	P8: [5] DC (6) CLOSED; not P8: [3] [9] (6) CLOSED		[5] DC CLOSED (6)
DC				P8: (8); not P8: [3] [9] CLOSED (6)		(8)
ER	TDISind DR CLOSING		TDISind DR CLOSING	CLOSED (6)		CLOSED (6)
DT, AK or EA TPDU	[7] OPEN		(5)	CLOSING		(9)
Time-out TTR	[10]	TDISind [1] CLOSED (6)		[10]	[1] CLOSED (6)	
TDTreq	P8: [11] OPEN; not P8: [7] OPEN	[11] OPEN-R	[11] OPEN-WR			
TEXreq	P8: [11] OPEN; not P8: [7]	[11] OPEN-R	[11] OPEN-R			

## A.5 State tables for Class 4

This section provides a more precise description of a Class 4 transport connection. Tables A-11, A-12, A-13 give the predicates, actions and notes for Class 4 respectively. Table A-14/X.224 is the state table for Class 4 transport connection.

### *Assumptions and notations:*

- a) the state of every network connection is known as being open or opening (i.e., a NCONreq has been issued and the NCONCconf is awaited);
- b) for each transport connection the transport entity maintains the set of network connections to which the transport is assigned. A network connection in this set is either in open or opening state;
- c) when a N-CONNECT confirmation, N-RESET indication or N-DISCONNECT indication is received this event is associated with the transport connection if the network connection belongs to the set;
- d) when an N-DISCONNECT is received the network connection becomes unexisting and is therefore withdrawn from the set. When a NCONCconf is received the state of the nc becomes "open";

*Note* – This is not shown by explicit action in the state table. Conversely adding an nc to a set and setting the state of an nc to "opening" is shown by explicit action;

- e) when the state goes back to CLOSED or REFWAIT state, it is assumed that all timers are stopped (if running), the count is set to zero and the set becomes empty;
- f) when a PDU is received the network connection on which it has been received is assumed to be known;
- g) the variable "current-nc" is used to designate either the nc on which a TPDU has been received or the nc which has been chosen for a new assignment (either an existing one or a new one which is created);
- h) we also assume the following variables:
  - 1) local-ref: the reference (local) of the TC is chosen when sending the CR or when accepting a CR,
  - 2) remote-ref: the reference of remote entity is initially set to zero and initialized when processing the CC except if the CC is ignored,
  - 3) SRC-REF, DST-REF: designates the corresponding field of the received TPDU,
  - 4) src-ref, dst-ref: designates the corresponding field of the sent TPDU,
  - 5) count: designates the number of times a TPDU has been sent (retransmission);
- i) the data transfer phase is not completely described in the state table but refers to the main text;
- j) a "spontaneous" event called "decision of making a new assignment" has been introduced. It may occur at any time provided:
  - P1 and P2 are true (see predicate table) and the remote-ref is not zero (i.e., a CR TPDU has been received or a CC TPDU has been received and processed);
- k) when an N-RESET indication is received, an N-RESET response is issued.

TABLE A-11/X.224

## Predicates for class 4

Name	Description
P0	T-CONNECT request is acceptable.
P1	An assignment can be done to a suitable Network Connection (either open or opening).
P2	It is possible to open a new Network Connection.
P3	Local choice.
P4	A CR TPDU has never been sent.
P5	The transport entity is the initiator and the set of Network Connections is now empty (i.e., a new assignment shall be done) or a new assignment is decided as a local choice.
P6	Local choice not to perform a new assignment if the set of Network Connections is empty (for closing state only).
P7	Count = maximum.
P8	Acceptable CR TPDU.
P9	Acceptable class 4 CC TPDU.
P10	Unacceptable classe 4 CC TPDU.
P11	CC TPDU not specifying class 4.

TABLE A-12/X.224  
Specific actions for class 4

Name	Description
[0]	Set reference timer.
[1]	Count = count + 1.
[2]	Count = 0.
[3]	Set retransmission timer.
[4]	Stop retransmission timer if running.
[5]	Set window timer.
[6]	Stop window timer if running.
[7]	Set inactivity timer.
[8]	Stop inactivity timer.
[9]	Set initial credit for sending according to the received CR/CC TPDU.
[10]	Set initial credit for controlling reception according to the sent CR/CC TPDU.
[11]	Send the CR TPDU if there is a Network Connection in the open state in the set.
[12]	Add the current Network Connection to the set, if not already included.
[13]	The current Network Connection is now in the opening state.
[14]	Send the CC TPDU if a Network Connection in the open state is in the set.
[15]	Send the DR TPDU if a Network Connection in the open state is in the set. This DR TPDU is sent with SRC-REF = local-ref and DST-REF = remote-ref (may be ZERO).
[16]	Send the DR TPDU if a Network Connection in the open state is in the set. The DR TPDU is sent with SRC-REF = 0 and DST-REF = remote-ref.
[17]	Send a TPDU according to data transfer procedure.
[18]	See state table if the class specified in the CC TPDU (refer to data transfer).
[19]	See state table for the class (refer to release procedure); send a DR TPDU if the class is not 0, otherwise issue an N-DISCONNECT request.
[20]	Store request an exercise flow control to the user.
[21]	Send a DR TPDU with SRC-REF field set to zero.
[22]	Send a DC TPDU DC except if the SRC-REF field of the received DR TPDU is equal to zero.

TABLE A-13/X.224

## Specific notes for class 4

Name	Description
(1)	Not possible as not set of Network Connection is associated with this Transport Connection.
(2)	It is also possible to remain in the same state (T1 is still running) until: <ul style="list-style-type: none"> <li>- a CC TPDU is received which performs a new assignment,</li> <li>- a new assignment is tried (spontaneous event),</li> <li>- T1 runs out and the count is equal to the maximal value.</li> </ul>
(3)	No new assignment was possible: if the set is empty, the Transport Entity waits until a new assignment is received, or can be locally performed (spontaneous event).
(4)	It is also possible to perform a new assignment (this may be done in triggering the event «new Network Connection assignment»).
(5)	Not a duplicated CR TPDU. If CR TPDU is duplicated, ignore it.
(6)	Since a new Network Connection is now assigned, it is recommended that the appropriate TPDU be sent on this Network Connection (if open) in order to make the remote entity aware of this assignment. It is also possible to allow the normal retransmission procedures to pause for the TPDU to be sent; however, the first TPDU available for sending should be sent on the new Network Connection.
(7)	As a local choice it is also possible to apply the following: [0], TDISind, REFWAIT.
(8)	Association to this Transport Connection is done regardless of the SRC-REF field. If SRC-REF is not zero, a DC TPDU is sent back.
(9)	At least an AK TPDU shall be sent if the Transport Entity is the initiator in order to ensure that the responder will complete its three-way handshake.
(10)	If association has been made, and DST-REF is zero, then the DC TPDU contains a SRC-REF field set to zero.
(11)	If the CLOSING state has been entered coming from WFCC state, the remote-ref is zero – the SRC-REF field of the CC TPDU is ignored (i.e, if the DR TPDU is retransmitted, it will be with DST-REF field set to zero).
(12)	If the CLOSING state has been entered, coming from WFCC state, the remote-ref (which is zero) shall be set with SRC-REF in order to comply with the release procedure of the negotiated class.
(13)	The DR TPDU may be either repeated immediately or when T1 will run out.
(14)	If the set is empty, this event may be used as a criteria for triggering the event «new Network Connection assignment».
(15)	Previously stored T-DATA or T-EXPEDITED-DATA requests are ready for processing according to data transfer procedures.
(16)	See data transfer procedures.
(17)	When an N-RESET indication is received, an N-RESET response to be issued once independent of the state automata.

TABLE A-14/X.224

Class 4 connection/disconnection

STATE EVENT	REFWAIT	CLOSED	WFCC	WBCL	OPEN	WFTRESP	AKWAIT	CLOSING
TCONreq		not P0: TDISind CLOSED; P0 & P1: [12, 1, 3, 10, 11] WFCC; P0 & not P1 & P2: [13, 12, 1, 3, 10] NCONreq WFCC; P0 & not P1 & not P2: TDISind CLOSED						
TCONresp						[3, 2, 1, 10, 14] AKWAIT		
TDISreq			P4: CLOSED; not P4 & P3 WBCL; not P4 & not P3 [4, 3, 2, 1, 15] CLOSING		[6, 8, 4, 3, 2, 1, 15] CLOSING	[16] CLOSED	[4, 3, 2, 1, 15] CLOSING	
NDISind	(1)	(1)	P1: [12, 4] WFCC; (not P1) & P2: [13, 12] NCONreq WFCC; not P1 & not P2: [0] (2) TDISind REFWAIT	P3: [0] REFWAIT; not P3 & P1: [12, 11] WBCL; not P3 & not P1 & P2: [13, 12] NCONreq WBCL; not P3 & not P1 & not P2): [0] REFWAIT	P5 & P1 [12, 17] (6) OPEN; P5 & not P1 & P2: [13, 12] NCONreq OPEN; P5 & not P1 & not P2: OPEN (3); not P5: OPEN	WFTRESP (4)	P5 & P1: [12, 14] (6) AKWAIT; P5 & not P1 & P2: [13, 12] NCONreq AKWAIT; P5 & not P1 & not P2: AKWAIT (3); not P5: AKWAIT	P6: [0] REFWAIT; not P6 & P5 & P1: [12, 15] CLOSING (6); not P6 & P5 & not P1 & P2: [13, 12] NCONreq CLOSING; not P6 & P5 & AKWAIT (3); not P5: AKWAIT CLOSING

TABLE A-14/X.224 (continued)

STATE EVENT	REFWAIT	CLOSED	WFCC	WBCL	OPEN	WFTRESP	AKWAIT	CLOSING
NRSTind			(17)	(17)	(17)	(17)	(17)	(17)
TDTreq TEXreq					(16) OPEN		[20] AKWAIT	
NCONconf	(1)	(1)	CR WFCC (6)	CR WBCL (6)	[17] OPEN (16)	WFTRESP	CC AKWAIT (6)	[15] CLOSING (6)
New Network Connection Assignment					P1: [12, 17] OPEN (6); not P1 & P2: [13, 12] NCONreq OPEN	P1: [12] WFTRESP; not P1 & P2: [13, 12] NCONreq WFTRESP	P1: [12, 14] (6) AKWAIT; not P1 & P2: [13, 12] NCONreq AKWAIT	P1: [12, 15] (6) CLOSING; not P1 & P2: [13, 12] NCONreq CLOSING
Retrans-t			P7 & P3: [0] TDisind REFWAIT: P7 & not P3: [3, 2, 1, 15] TDisind CLOSING (14); not P7: [1, 3, 11] WFCC	P7 & P3: [0] REFWAIT: P7 & not P3: [3, 2, 1, 15] CLOSING (14); not P7: [1, 3, 11] WBCL	P7: [6, 8, 3, 2, 1, 15] TDisind CLOSING (14); not P7: (16) (14) OPEN		P7: [3, 2, 1, 15] TDisind (14) CLOSING; not P7: [1, 3, 14] (14) AKWAIT	P7: [0] REFWAIT; not P7: [1, 3, 15] (14) CLOSING
I-t					[6, 4, 3, 2, 1, 15] TDisind CLOSING (7)			
Ref-t	CLOSED							
CR		not P8: [21] CLOSED (5); P8: [9, 12] TCONind WFTRESP (5)			[12, 8, 7] OPEN	[12] WFTREST	[12, 14] AKWAIT	[12] CLOSING (13)

TABLE A-14/X.X.224 (end)

STATE EVENT	REFWAIT	CLOSED	WFCC	WBCL	OPEN	WFTRESP	AKWAIT	CLOSING
CC	DR REFWAIT	DR CLOSED	P9: [12, 9, 2, 4, 5, 7, 17] TCONconf (9) OPEN; P10: [12, 4, 3, 2, 1, 15] TDISind CLOSING; P11: [18]	P11: [19] non P11: [12, 2, 4, 3, 1, 15] CLOSING	[12, 17, 8, 7] (9) OPEN			P11: [19] (12); not P11: [12] CLOSING (11)
ER	REFWAIT	CLOSED	[0] TDisind REFWAIT	[0] REFWAIT	[12, 6, 8, 4, 3, 2, 1, 15] TDisind CLOSING		[12, 4, 3, 2, 1, 15] TDisind CLOSING	[0] REFWAIT
DR	[22] REFWAIT	[22] CLOSED	(8) TDisind [0] REFWAIT	(8) [0] REFWAIT	DC (10) [0] TDisind REFWAIT	DC (10) TDisind CLOSED	DC (10) [0] TDisind REFWAIT	[0] REFWAIT
DC	REFWAIT	CLOSED						[0] REFWAIT
EA	REFWAIT	CLOSED			[12, 8, 7] OPEN (16)			[12] CLOSING (13)
DT/AK/ED	REFWAIT	CLOSED			[12, 8, 7] OPEN (16)		[12, 7] OPEN (15) (16)	[12] CLOSING (13)

## ANNEX B

(to Recommendation X.224)

### Transport protocol identification

#### B.1 Introduction

The transport protocol identifier is used to signal, from the initiator of the network connection to the acceptor, what transport protocol is to be used on that network connection. The procedure uses a PI TPDU, whose encoding is defined in § B.3 below. There are two ways that the PI TPDU may be conveyed, as described in § B.2.

This case of transport protocol identification only applies to the use of a network connection for a single transport protocol (such as that described in the remainder of this Recommendation). The general case, whereby the network connection can be used for several different transport protocols, whether sequentially or concurrently is for further study.

*Note 1* – No negotiation of PI is currently provided for. It is for further study what mechanisms and TPDU's would be needed to support such negotiation in the future.

*Note 2* – It is for further study how to achieve compatibility of this transport protocol identification with existing terminals.

#### B.2 Conveyance of PI TPDU

The PI TPDU may be conveyed either as NS-user data of the N-CONNECT request (and indication) or as N-DATA, as described in the following two subsections.

##### B.2.1 N-CONNECT NS-user data available

When a transport entity opens a new network connection for the purpose of assignment (or reassignment) the transport entity shall use the NS-user data parameter of the N-CONNECT request primitive, as follows: either

- a) a PI TPDU shall be placed as the first, or only, TPDU in the NS-user data field. This TPDU explicitly indicates which protocol is to be used over the network connection; or
- b) no PI TPDU is conveyed with the N-CONNECT – in this case the default is considered to apply, namely that the transport protocol specified in the remainder of this Recommendation is to be used.

*Note 1* – Teletex and Group 4 facsimile terminals operating over the PSPDN will send (and thus may expect to receive) a protocol identifier of a single octet with the value "02", as specified in Reference 4. This must be taken into account when interworking with Teletex or Group 4 facsimile terminals is required.

*Note 2* – It is for further study whether N-DISCONNECT NS-user data, if provided, may be used, for example in negotiation of protocol identification.

##### B.2.2 No N-CONNECT NS-user data available

For an interim period, NS-user data may not be available in the N-CONNECT request under some circumstances. Under these circumstances, the PI TPDU shall be conveyed using the N-DATA request, and shall be concatenated with the CR TPDU which requests opening of the transport connection, or the first TPDU sent on that network connection performing assignment or reassignment.

When no PI TPDU is conveyed, the default transport protocol, i.e. the protocol defined in the remainder of this Recommendation, will apply.

B.3 *Encoding of the PI TPDU*

B.3.1 *Structure*

The PI TPDU shall have the following format:

1	2	3	4	5	p
LI	PI 0000 0001	PRT-ID	SHARE	Variable part	

B.3.2 *LI*

See § 13.2.1.

B.3.3 *Fixed part*

The fixed part shall contain (in octets 2 to 4):

- a) PI: Protocol identification code: 0000 0001.
- b) PRT-ID: Protocol Id
  - 00000000 reserved
  - 00000001 OSI transport protocol (remainder of this Recommendation)
  - 00000010-01111111 reserved for other OSI protocols
  - 10000000-11111111 reserved for private use.
- c) SHARE: Sharing of network connection
  - 00000000 no sharing
  - 00000001 sharing allowed (for further study)
  - 00000010-11111111 reserved.

B.3.4 *Variable part*

One optional parameter is defined – however its use is for further study:

*protocol sharing*

- Parameter code: 1101 1111
- Parameter length: number of protocol ids
- Parameter value: list of protocol ids, one per octet

This parameter is not permitted unless SHARE has the value 00000001.

## APPENDIX I

(to Recommendation X.224)

### Checksum algorithms

#### I.1 Symbols

The following symbols are used:

$C0$  } variables used in the algorithms  
 $C1$  }

$i$  number (i.e., position) of an octet within the TPDU (see § 13.2)

$n$  number (i.e., position) of the first octet of the checksum parameter

$L$  length of the complete TPDU

$X$  value of the first octet of the checksum parameter

$Y$  value of the second octet of the checksum parameter.

#### I.2 Arithmetic conventions

Addition is performed in one of the two following modes:

a) modulo 255 arithmetic;

b) one's complement arithmetic in which if any of the variables has the value minus zero (i.e. 255) it shall be regarded as though it was plus zero (i.e. 0).

#### I.3 Algorithm for generating checksum parameters

I.3.1 Set up the complete TPDU with the value of the checksum parameter field set to zero.

I.3.2 Initialize  $C0$  and  $C1$  to zero.

I.3.3 Process each octet sequentially from  $i = 1$  to  $L$  by

a) adding the value of the octet to  $C0$ ; then

b) adding the value of  $C0$  to  $C1$ .

I.3.4 Calculate  $X$  and  $Y$  such that

$$X = -C1 + (L - n) \cdot C0$$

$$Y = C1 - (L - n + 1) \cdot C0.$$

I.3.5 Place the values  $X$  and  $Y$  in octets  $n$  and  $(n + 1)$  respectively.

*Note* – This algorithm calculates:

$$C1 = \sum_{i=1}^L (L - i + 1)a_i$$

which is equal to zero, if the formulas in § 6.17.3 are followed, since:

$$\sum_{i=1}^L (L - i + 1)a_i = (L + 1) \sum_{i=1}^L a_i - \sum_{i=1}^L ia_i = 0$$

#### I.4 Algorithm for checking checksum parameters

I.4.1 Initialize  $C0$  and  $C1$  to zero.

I.4.2 Process each octet of the TPDU sequentially from  $i = 1$  to  $L$  by

a) adding the value of the octet to  $C0$ ; then

b) adding the value of  $C0$  to  $C1$ .

I.4.3 If, when all the octets have been processed, either or both of  $C0$  and  $C1$  does not have the value zero, the checksum formulas in § 6.17 have not been satisfied.

*Note* – The nature of the algorithm is such that it is not necessary to compare explicitly the stored checksum bytes.

## APPENDIX II

(to Recommendation X.224)

### Differences between Recommendation X.224 and ISO 8073 (1986)

Recommendation X.224 and ISO 8073 (1986) are technically aligned except for the differences listed below:

#### II.1 *Defect reports*

In collaboration, ISO and CCITT have agreed to corrections resulting from defect reports. Recommendation X.224 includes the necessary corrections from reports 6, 11, 32, 33, 35, 65, 66, 67, 69, 71, 72, 73, 74, 77 (item 2), 80, 81, 82, 84, 85, 88, 89, 90, 91, 92, 93, 94, 95, 103, 106, and 107 which are not included in ISO 8073 (see Notes 1 and 2). An update of ISO 8073 has not yet been published (see Note 3).

*Note 1* – Defect reports 1, 2, 3, 4, 5, 7, 9, 10, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29 (items 1-7), 34, 36, 38, 39 (item 2), 40, 41, 42, 43, 45, 51, 52, 56, 57, 58, 59, 60, and 61 have had the necessary changes applied to Recommendation X.224 and ISO 8073.

*Note 2* – Defect report 18 does not apply to Recommendation X.224 or ISO 8073 and in addition defect reports 8, 24, 29 (item 8), 39 (item 1), 46, 48, 55, 62, 64 (item 2), 68, 76 (item 2), 77 (item 1), 79, 86, 87, 97, 99, and 104 have been rejected. No further changes will be made as a result of these defect reports.

*Note 3* – ISO plans to issue a new edition of ISO 8073 shortly. This new edition of ISO 8073 is intended to be aligned with this Recommendation with respect to this difference.

#### II.2 *Conformance*

Recommendation X.224 in § 14, requires all systems to implement Class 0. ISO 8073, in Clause 14, requires all systems to implement either Class 0 or Class 2.

#### II.3 *Class negotiation*

An additional restriction regarding the classes of transport protocol that are proposed in a CR TPDU is defined in item (a) in § 14.4 of Recommendation X.224. No such restriction is contained in ISO 8073.

#### II.4 *Precedence*

ISO 8073, in its Annex A (state tables), includes the statement: «In the event of a discrepancy between the description in these tables and that contained in the text, the text takes precedence.» Recommendation X.224 does not contain this statement.

#### II.5 *Assignment to network connections*

In § 6.1.3 of Recommendation X.224 it is stated that the responder to the transport connect request becomes aware of the assignment when it receives particular TPDU's. In Clause 6.1.3 of ISO 8073 it is the non-owner of the network connection that becomes aware of the assignment upon receipt of these same TPDU's.

#### II.6 *Transport protocol identification*

The material contained in Annex B of Recommendation X.224 is not present in ISO 8073.

#### II.7 *Checksum algorithms*

The material contained in Appendix I to Recommendation X.224 is contained in Annex B to ISO 8073.

#### II.8 The material in this appendix is not present in ISO 8073 (see Note 3 above.)



## ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
<b>Series X</b>	<b>Data networks and open system communications</b>
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems