SESSION PROTOCOL SPECIFICATION FOR OPEN

SYSTEMS INTERCONNECTION FOR CCITT APPLICATIONS[1]

(*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.215 specifies the Session Service Definition for Open Systems Interconnection for CCITT Applications;

(c) that Recommendation T.62 defines the Control Procedures for Teletex and Group 4 facsimile services,

unanimously declares

that this Recommendation defines the Session Protocol of Open Systems Interconnection for CCITT Applications as given in the Scope and Field of Application.

CONTENTS

0 *Introduction*

1 *Scope and field of application*

2 *References*

3 *Definitions*

4 *Symbols and abbreviations*

4.1 Data units

4.2 SPDU fields

4.3 Timer variables

4.4 Miscellaneous

4.5 Local variables

5 *Overview of the session protocol*

5.1 Model of the session layer

5.2 Services provided by the session layer

5.3 Services assumed from the transport layer

5.4 Functions of the session layer

5.5 Functional units

5.6 Tokens

5.7 Negotiation

5.8 Local variables

6 *Use of the transport service*

6.1 Assignment of a session connection to the transport connection

---

[1] Recommendation X.225 is technically aligned with ISO 8327 [Information Processing Systems - Open Systems Interconnection - Basic Connection oriented session protocol specification] which includes corrections resulting from ISO defect reports numbered 8326/6, 8326/7, 8326/20, 8327/1, 8327/3, 8327/4 through 8327/10, 8327/12, 8327/17, 8327/18, 8327/19, 8327/26, 8327/27, 8327/30, 8327/34 and 8327/35 and Addendum 2 to incorporate unlimited user data, with the exception of the differences noted in Appendix I.

6.2 Reuse of the transport connection
6.3 Use of transport normal data
6.4 Use of transport expedited data
6.5 Flow control
6.6 Transport disconnection
7 *Elements of procedure related to SPDUs*
7.1 CONNECT SPDU
7.2 OVERFLOW ACCEPT SPDU
7.3 CONNECT DATA OVERFLOW SPDU
7.4 ACCEPT SPDU
7.5 REFUSE SPDU
7.6 FINISH SPDU
7.7 DISCONNECT SPDU
7.8 NOT FINISHED SPDU
7.9 ABORT SPDU
7.10 ABORT ACCEPT SPDU
7.11 DATA TRANSFER SPDU
7.12 EXPEDITED SPDU
7.13 TYPED DATA SPDU
7.14 CAPABILITY DATA SPDU
7.15 CAPABILITY DATA ACK SPDU
7.16 GIVE TOKENS SPDU
7.17 PLEASE TOKENS SPDU
7.18 GIVE TOKENS CONFIRM SPDU
7.19 GIVE TOKENS ACK SPDU
7.20 MINOR SYNC POINT SPDU
7.21 MINOR SYNC ACK SPDU
7.22 MAJOR SYNC POINT SPDU
7.23 MAJOR SYNC ACK SPDU
7.24 RESYNCHRONIZE SPDU
7.25 RESYNCHRONIZE ACK SPDU
7.26 PREPARE SPDU
7.27 EXCEPTION REPORT SPDU
7.28 EXCEPTION DATA SPDU
7.29 ACTIVITY START SPDU
7.30 ACTIVITY RESUME SPDU
7.31 ACTIVITY INTERRUPT SPDU
7.32 ACTIVITY INTERRUPT ACK SPDU
7.33 ACTIVITY DISCARD SPDU

7.34 ACTIVITY DISCARD ACK SPDU

7.35 ACTIVITY END SPDU

7.36 ACTIVITY END ACK SPDU

7.37 Additional elements of procedure for segmented SSDUs

8 *Structure and encoding of SPDUs*

8.1 TSDU structure

8.2 SPDU structure

8.3 SPDU identifiers and associated parameter fields

8.4 Additional encoding rules for segmented SSDUs

9 *Conformance to this standard*

Annex A - State tables

Annex B - Relationship to CCITT Recommendation T.62 encoding

Annex C - PGIs and PIs reserved for use by Recommendation T.62

Annex D - Compatibility between Protocol Version 1 and Protocol Version 2

Appendix I - Difference between Recommendation X.225 and ISO International Standard 8327

0 Introduction

The Session Protocol Standard 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 Session Protocol Standard is positioned with respect to other related Recommendations by the layers defined in the Reference Model for Open Systems Interconnection (Recommendation X.200). It is most closely related to and lies within the field of application of the Session Service Definition (Recommendation X.215). It also uses and references the Transport Service Definition (Recommendation X.214), whose provisions it assumes in order to accomplish the aims of the session protocol. The interrelationship of these Recommendations is depicted in Figure 1/X.225.

FIGURE 1/X.225 - T0706630-88

This Recommendation specifies a single protocol with a common encoding.

It is intended that the session protocol should be general enough to cater for the total range of session service users without restricting future extensions.

The protocol is structured so that subsets of protocol can be defined.

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 session 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 a part of an agreement for the admittance of systems into the open systems environment;

d) as a refinement to 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.

The variations and options available within this Recommendation are essential to enable a session service to be provided for a wide variety of applications. 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 of with statements of the intended purpose of provision or use.

This Recommendation contains the following annexes and appendix:

a) Annex A - State Tables.

b) Annex B - Relationship to CCITT Recommendation T.62 encoding.

c) Annex C - PGIs and PIs reserved for use by Recommendation T.62.

d) Annex D - Compatibility between Protocol Version 1 and Protocol Version 2.

e) Appendix I - Difference between Recommendation X.225 and ISO International Standard 8327.

1 Scope and field of application

1.1 This Recommendation specifies:

a) procedures for a single protocol for the transfer of data and control information from one session entity to a peer session entity;

b) the means of selecting the functional units to be used by the session entities;

c) the structure and encoding of the session 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 session entities through the exchange of session protocol data units;

b) the interactions between a session entity and the session service user in the same system through the exchange of session service primitives;

c) the interactions between a session entity and the transport service provider through the exchange of transport service primitives.

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

1.4 This Recommendation also specifies conformance requirements for systems implementing these procedures. It does not contain tests which can be used to demonstrate this conformance.

2 References

Recommendation X.200 - Reference Model of Open Systems Interconnection for CCITT applications. (See also ISO 7498-1)

Recommendation X.214 - Transport Service Definition for Open Systems Interconnection for CCITT applications. (See also ISO 8072)

Recommendation X.215 - Session Service Definition for Open Systems Interconnection for CCITT applications. (See also ISO 8326 and ISO 8326 Addendum 2)

Recommendation T.62 - CCITT Recommendation T.62 - Control Procedures for the Teletex and Group 4 Facsimile Services.

ISO 7498-3 - Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 3: Naming and Addressing[2].

Note - CCITT Recommendation T.62 is not essential for the application of this Recommendation, but is included in the list of references as it has been referred to, for information, in relation to interworking with the CCITT Telematic services (see Annexes B and C).

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 Recommendation X.200, and makes use of the following terms defined in that Recommendation:

a) expedited session service data unit;

b) session connection;

c) session layer;

d) session protocol data unit;

e) session service;

f) session service access point;

g) session service data unit;

h) transport layer;

i) transport connection;

j) transport service;

k) transport service access point;

l) concatenation;

m) segmenting;

n) session selection (defined in ISO 7498-3).

3.2 This Recommendation is also based on concepts developed in Recommendation X.215 and makes use of the following terms defined in that Recommendation:

a) token;

b) calling SS-user;

c) called SS-user;

d) sending SS-user;

e) receiving SS-user;

f) requesting SS-user;

g) accepting SS-user;

h) requestor;

i) acceptor.

Note - The following terms used in this Recommendation are used in relation to tokens and are explained in Recommendation X.215:

a) assigned;

b) not assigned;

c) available;

---

[2] At present at the stage of draft; publication anticipated in due course.

d) not available.

3.3 For the purposes of this Recommendation, the following definitions apply:

3.3.1 **Session Protocol Machine; SPM**

An abstract machine that carries out the procedures specified in this protocol.

Note - A session entity is comprised of one or more SPMs.

3.3.2 **session service user; SS-user**

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

3.3.3 **transport service provider; TS-provider**

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

3.3.4 **local matter**

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

3.3.5 **initiator**

An SPM that initiates a CONNECT SPDU.

3.3.6 **responder**

An SPM with whom an initiator wishes to establish a session connection.

Note - Initiator and responder are defined with respect to a single session connection.

3.3.7 **sending SPM**

An SPM that sends a given SPDU.

3.3.8 **receiving SPM**

An SPM that receives a given SPDU.

3.3.9 **owner (of a token)**

The SPM to whom a token is assigned.

3.3.10 **proposed parameter**

The value for a parameter proposed by an SPM in a CONNECT SPDU or an ACCEPT SPDU that it wishes to use on the session connection.

3.3.11 **negotiation**

The process by which two SPMs agree on a common set of functional units and protocol values and on the initial setting of available tokens.

3.3.12 **selected parameter**

The value for a parameter that has been chosen for use on the session connection.

3.3.13 **valid SPDU**

An SPDU which complies with the requirements of this Recommendation with respect to structure and encoding.

3.3.14 **invalid SPDU**

An SPDU which does not comply with the requirements of this Recommendation with respect to structure and encoding.

3.3.15 **protocol error**

Use of an SPDU that does not comply with the procedures agreed for the session connection.

3.3.16 **transparent (data)**

SS-user data which is transferred intact between SPMs and which is unavailable for use by the SPMs.

3.3.17 **SPDU identifier (SI)**

Heading information that identifies the SPDU concerned.

3.3.18 **length indicator (LI)**

An indicator that represents the length of an associated parameter field.

3.3.19 **parameter field**

A group of one or more octets used to represent a particular set of information.

3.3.20 **parameter identifier (PI)**

An identifier, defined in this Recommendation, that indicates the type of information contained in its associated parameter field.

3.3.21 **PI unit**

An element of an SPDU that contains a PI field together with its associated LI field and parameter field.

3.3.22 **parameter group identifier (PGI)**

An identifier, defined in this Recommendation, that indicates the type of information contained in its associated parameter field. The associated parameter field may consist of a set of PI units.

3.3.23 **PGI unit**

An element of an SPDU that contains a PGI field together with its associated LI field and parameter field.

3.3.24 **parameter value (PV)**

Information that represents the value of the parameter identified by either a PI or PGI.

3.3.25 **local variable**

A local variable within the SPM which is used as a means of clarifying the effects of certain actions and clarifying the conditions under which certain actions are permitted.

4 Symbols and abbreviations

4.1 *Data units*

SPDU Session Protocol Data Unit

SSDU Session Service Data Unit

TSDU Transport Service Data Unit

4.2 *SPDU fields*

SI SPDU Identifier (see ¤ 3.3.17)

LI Length Indicator (see ¤ 3.3.18)

PI Parameter Identifier (see ¤ 3.3.20)

PGI Parameter Group Identifier (see ¤ 3.3.22)

PV Parameter Value (see ¤ 3.3.24)

4.3 *Timer variables*

TIM Disconnection and abort timer

4.4 *Miscellaneous*

SPM Session Protocol Machine (see ¤ 3.3.1)

SS Session Service

SSAP Session service access point

TSAP Transport service access point

4.5 *Local variables*

Vact See ¤ 5.8.1

Vnextact See ¤ 5.8.2

V(A) See ¤ 5.8.3

V(M) See ¤ 5.8.4

V(R) See ¤ 5.8.5

Vsc See ¤ 5.8.6

5 **Overview of the** session protocol

5.1 Model of the session layer

The SPM (see Note) within the session layer communicates with the SS-user through an SSAP by means of the service primitives as defined by the session service definition in Recommendation X.215. Service primitives will cause or be the result of session protocol data unit exchanges between the peer SPMs using a transport connection. These protocol exchanges are effected using the services of the transport layer as defined by the transport service definition in Recommendation X.214 through two TSAPs.

Session connection endpoints are identified in end systems by an internal, implementation dependent mechanism, so that the SS-user and the SPM can refer to each session connection.

The model of the Session Layer is illustrated in Figure 2/X.225.

Note - A session entity is comprised of one or more SPMs.

FIGURE 2/X.225 - CCITT - 79940

5.2 Services provided by the session layer

The protocol specified in this Recommendation supports the session service defined in Recommendation X.215. Information is transferred to and from the SS-user using the session service primitives listed in Table 1/X.225. Table 1/X.225 also defines the SPDUs associated with each of the service primitives.

TABLE 1/X.225

Session service primitives

Service	Primitives	Associated SPDUs
Session Connection	S-CONNECT request	CONNECT SPDU
	S-CONNECT indication	CONNECT SPDU
	S-CONNECT (accept) response	ACCEPT SPDU
	S-CONNECT (accept) confirm	ACCEPT SPDU
	S-CONNECT (reject) response	REFUSE SPDU
	S-CONNECT (reject) confirm	REFUSE SPDU
Normal Data Transfer	S-DATA request	DATA TRANSFER SPDU
	S-DATA indication	DATA TRANSFER SPDU
Expedited Data Transfer	S-EXPEDITED-DATA request	EXPEDITED DATA SPDU
	S-EXPEDITED-DATA indication	EXPEDITED DATA SPDU
Typed Data Transfer	S-TYPED-DATA request	TYPED DATA SPDU
	S-TYPED-DATA indication	TYPED DATA SPDU
Capability Data Exchange	S-CAPABILITY-DATA request	CAPABILITY DATA SPDU
	S-CAPABILITY-DATA indication	CAPABILITY DATA SPDU
	S-CAPABILITY-DATA response	CAPABILITY DATA ACK SPDU
	S-CAPABILITY-DATA confirm	CAPABILITY DATA ACK SPDU
Give Tokens	S-TOKEN-GIVE request	GIVE TOKENS SPDU
	S-TOKEN-GIVE indication	GIVE TOKENS SPDU
Please Tokens	S-TOKEN-PLEASE request	PLEASE TOKENS SPDU
	S-TOKEN-PLEASE indication	PLEASE TOKENS SPDU
Give Control	S-CONTROL-GIVE request	GIVE TOKENS CONFIRM SPDU
	S-CONTROL-GIVE indication	GIVE TOKENS CONFIRM SPDU

TABLE 1/X.225 *(cont.)*

Service	Primitives	Associated SPDUs
Minor Synchronization Point	S-SYNC-MINOR request	MINOR SYNC POINT SPDU
	S-SYNC-MINOR indication	MINOR SYNC POINT SPDU
	S-SYNC-MINOR response	MINOR SYNC ACK SPDU
	S-SYNC-MINOR confirm	MINOR SYNC ACK SPDU
Major Synchronization Point	S-SYNC-MAJOR request	MAJOR SYNC POINT SPDU
	S-SYNC-MAJOR indication	MAJOR SYNC POINT SPDU
	S-SYNC-MAJOR response	MAJOR SYNC ACK SPDU
	S-SYNC-MAJOR confirm	MAJOR SYNC ACK SPDU
Resynchronize	S-RESYNCHRONIZE request	RESYNCHRONIZE SPDU
	S-RESYNCHRONIZE indication	RESYNCHRONIZE SPDU
	S-RESYNCHRONIZE response	RESYNCHRONIZE ACK SPDU
	S-RESYNCHRONIZE confirm	RESYNCHRONIZE ACK SPDU
P-Exception Report	S-P-EXCEPTION-REPORT indication	EXCEPTION REPORT SPDU
U-Exception Reporting	S-U-EXCEPTION-REPORT request	EXCEPTION DATA SPDU
	S-U-EXCEPTION-REPORT indication	EXCEPTION DATA SPDU
Activity Start	S-ACTIVITY-START request	ACTIVITY START SPDU
	S-ACTIVITY-START indication	ACTIVITY START SPDU
Activity Resume	S-ACTIVITY-RESUME request	ACTIVITY RESUME SPDU
	S-ACTIVITY-RESUME indication	ACTIVITY RESUME SPDU

TABLE 1/X.225 *(end)*

Service	Primitives	Associated SPDUs
Activity Interrupt	S-ACTIVITY-INTERRUPT request	ACTIVITY INTERRUPT SPDU
	S-ACTIVITY-INTERRUPT indication	ACTIVITY INTERRUPT SPDU
	S-ACTIVITY-INTERRUPT response	ACTIVITY INTERRUPT ACK SPDU
	S-ACTIVITY-INTERRUPT confirm	ACTIVITY INTERRUPT ACK SPDU
Activity Discard	S-ACTIVITY-DISCARD request	ACTIVITY DISCARD SPDU
	S-ACTIVITY-DISCARD indication	ACTIVITY DISCARD SPDU
	S-ACTIVITY-DISCARD response	ACTIVITY DISCARD ACK SPDU
	S-ACTIVITY-DISCARD confirm	ACTIVITY DISCARD ACK SPDU
Activity End	S-ACTIVITY-END request	ACTIVITY END SPDU
	S-ACTIVITY-END indication	ACTIVITY END SPDU
	S-ACTIVITY-END response	ACTIVITY END ACK SPDU
	S-ACTIVITY-END confirm	ACTIVITY END ACK SPDU
Orderly Release	S-RELEASE request	FINISH SPDU
	S-RELEASE indication	FINISH SPDU
	S-RELEASE (accept) response	DISCONNECT SPDU
	S-RELEASE (accept) confirm	DISCONNECT SPDU
	S-RELEASE (reject) response	NOT FINISHED SPDU
	S-RELEASE (reject) confirm	NOT FINISHED SPDU
U-Abort	S-U-ABORT request	ABORT SPDU
	S-U-ABORT indication	ABORT SPDU
P-Abort	S-P-ABORT indication	ABORT SPDU

5.3 Services assumed from the transport layer

The protocol specified in this Recommendation assumes the use of the connection-oriented transport service defined in Recommendation X.214.

Information is transferred to and from the TS provider in the transport service primitives listed in Table 2/X.225.

TABLE 2/X.225

Transport service primitives

Primitive	X/Y	Parameters
T-CONNECT request indication	X	Called Address, Calling Address, Expedited Data option, Quality of Service, TS User-Data
T-CONNECT response confirm	X	Quality of Service, Responding Address, Expedited Data option, TS User-Data
T-DATA request indication	X	TS-User-Data
T-EXPEDITED-DATA request indication	Y	TS User-Data
T-DISCONNECT request	X	TS User-Data
T-DISCONNECT indication	X	Disconnect reason, TS User-Data

X: The session protocol assumes that this service is always available.

Y: The session protocol assumes that this service is provided by the transport layer when requested by the SPM during the session connection establishment phase.

5.4 Functions of the session layer

5.4.1 *Overview of functions*

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

The functions in the session layer are concerned with dialogue management, data flow synchronization, and data flow resynchronization. They are described below; the descriptions are grouped into those concerned with the connection establishment phase, the data transfer phase, and the release phase.

5.4.2 *Connection establishment phase*

The purpose of the connection establishment phase is to establish a session connection between two SS-users, and:

a) to map session addresses onto transport addresses;

b) to select transport quality of service parameters needed (see ¤ 6.1.4);

c) to negotiate session parameters (see ¤¤ 7.1, 7.2 and 7.4);

d) to transfer session selectors (see ¤¤ 7.1 and 7.4) if required;

e) to distinguish between session connections (see ¤¤ 7.1 and 7.4);

f) to transfer transparent user data (see ¤¤ 7.1, 7.2, 7.3 and 7.4);

g) to select a protocol version (see Note).

Note - This Recommendation specifies the following protocol versions:

i) Protocol Version 1 which imposes restrictions on the length of the user data field;

ii) Protocol Version 2 which imposes no explicit restrictions on the length of the user data field.

Annex D identifies the compatibility between Protocol Version 1 and Protocol Version 2.

5.4.3 *Data transfer phase*

The purpose of the data transfer phase is to transport SSDUs between two SS-users connected by a session connection. This purpose is achieved by means of transmission of SPDUs and by the following functions, each of which may or may not be used, depending on the functional units selected in the session connection establishment phase. These concepts are defined in Recommendation X.215:

a) *Normal Data Transfer* (see ¤ 7.11), which may involve segmenting of SSDUs into SPDUs and reassembly by the destination SPM; and concatenation and separation of certain SPDUs. There are two modes of operation:

1) Half-duplex, when the right to send data is restricted to the owner of the data token;

2) Duplex, when there is no restriction on the right to send data.

b) *Token management* (see ¤¤ 7.16 to 7.19), to enable the SS-users to request and transfer tokens which control the exclusive right to exercise certain functions (see Table 5/X.225).

c) *Exception Reporting* (see ¤¤ 7.27 and 7.28), to enable the SS-provider or the SS-user to report exception conditions that are less severe than those requiring abort.

d) *Typed Data Transfer* (see ¤ 7.13), to enable transfer of information which is not subject to assignment of the data token.

e) *Minor Synchronization Point* (see ¤¤ 7.20 and 7.21), to enable the SS-users to define minor synchronization points in the normal data flow. These minor synchronization points may optionally be confirmed, but have no implications on the data flow. Minor synchronization points are identified by synchronization point serial numbers. The serial number is incremented by one on each occasion that a minor synchronization point is placed in the data flow, and each time a minor synchronization point is received, such that both SS-users have the same serial numbers for the same synchronization point.

f) *Major Synchronization Point* (see ¤¤ 7.22 and 7.23 and e) above), to enable the SS-users to define major synchronization points in the normal data flow. These major synchronization points are required to be confirmed before the requesting SS-user is permitted to send any subsequent data on either the normal flow or the expedited flow and as such clearly separate the dialogue units.

g) *Resynchronize* (see ¤¤ 7.24 and 7.25), a function that allows a session connection to be set or reset to a defined synchronization point and reassign the tokens.

h) *Expedited Data Transfer* (see ¤ 7.12), a function used to convey a limited amount of user data with special handling. Such data may bypass normal data en route, but will be delivered prior to any data subsequently sent on the transport normal flow or the transport expedited flow.

i) *Activity Management* (see ¤¤ 7.29 to 7.36) provides a means explicitly to start, end, resume, interrupt or discard an activity. This provides a way:

1) to identify the entered activity and commence synchronization point serial numbering;

2) to identify the continued activity and reset the synchronization point serial number in case of resumption.

j) *Capability Data Exchange* (see ¤¤ 7.14 and 7.15), to provide a confirmed transfer of a limited amount of user data.

5.4.4 *Connection release phase*

The purpose of the release phase is to provide disconnection of the session connection, by using the following functions:

a) orderly release (negotiated and non-negotiated);

b) abort (provider and user initiated);

c) transfer of transparent user data.

5.5 Functional units

Functional units are logical groupings of related elements of procedure defined by this Recommendation for the purpose of:

a) negotiation for use during session connection establishment;

b) specification of conformance requirements.

The SPDUs associated with elements of procedure for each functional unit are specified in Table 3/X.225.

Tokens are associated with functional units (see ¤ 5.6).

5.5.1 Kernel functional unit

The kernel functional unit supports the basic protocol elements of procedure required to establish a session connection, transfer normal data and release the session connection.

5.5.2 *Negotiated release functional unit*

The negotiated release functional unit supports the negotiated release service which enables the SS-users to negotiate the orderly release of the session connection. If this functional unit has been selected, an attempt to release the session connection may be refused by the accepting SS-user.

5.5.3 *Half-duplex functional unit*

The half-duplex functional unit is used to control the right to send data. It is not valid to select both this functional unit and the duplex functional unit for use on the same session connection.

5.5.4 *Duplex functional unit*

The duplex functional unit is used when the right to send data is not controlled. It is not valid to select both this functional unit and the half-duplex functional unit for use on the same session connection.

5.5.5 *Expedited data functional unit*

The expedited data functional unit supports the expedited data service and allows the transfer of a limited amount of SS-user data.

The services supported by this functional unit can only be requested when the transport expedited flow is available to this session connection.

5.5.6 *Typed data functional unit*

The typed data functional unit enables the SS-users to transfer data in a manner which is not subject to the control imposed by the availability of the data token.

TABLE 3/X.225

Functional units

Functional unit	SPDU code	SPDU name	Reference
Kernel	CN	CONNECT (Note 1)	7.1
	OA	OVERFLOW ACCEPT (Note 2)	7.2
	CDO	CONNECT DATA OVERFLOW (Note 2)	7.3
	AC	ACCEPT (Note 1)	7.4
	RF	REFUSE (Note 1)	7.5
	FN	FINISH	7.6
	DN	DISCONNECT	7.7
	AB	ABORT	7.9
	AA	ABORT ACCEPT (Note 3)	7.10
	DT	DATA TRANSFER	7.11
Negotiated release	NF	NOT FINISHED	7.8
	GT	GIVE TOKENS (Note 5)	7.16
	PT	PLEASE TOKENS (Note 5)	7.17
Half-duplex	GT	GIVE TOKENS (Note 4)	7.16
	PT	PLEASE TOKENS (Note 4)	7.17
Duplex		No additional associated SPDUs	
Expedited data	EX	EXPEDITED DATA	7.12
Typed data	TD	TYPED DATA	7.13
Capability data exchange	CD	CAPABILITY DATA	7.14
	CDA	CAPABILITY DATA ACK	7.15
Minor synchronize	MIP	MINOR SYNC POINT	7.20
	MIA	MINOR SYNC ACK	7.21
	GT	GIVE TOKENS (Note 6)	7.16
	PT	PLEASE TOKENS (Note 6)	7.17
Major synchronize	MAP	MAJOR SYNC POINT	7.22
	MAA	MAJOR SYNC ACK	7.23
	PR	PREPARE (Note 7)	7.26
	GT	GIVE TOKENS (Note 8)	7.16
	PT	PLEASE TOKENS (Note 8)	7.17
Resynchronize	RS	RESYNCHRONIZE	7.24
	RA	RESYNCHRONIZE ACK	7.25
	PR	PREPARE (Note 7)	7.26
Exceptions	ER	EXCEPTION REPORT	7.27
	ED	EXCEPTION DATA	7.28

TABLE 3/X.225 *(cont.)*

Functional unit	SPDU code	SPDU name	Reference
Activity management	AS	ACTIVITY START	7.29
	AR	ACTIVITY RESUME	7.30
	AI	ACTIVITY INTERRUPT	7.31
	AIA	ACTIVITY INTERRUPT ACK	7.32
	AD	ACTIVITY DISCARD	7.33
	ADA	ACTIVITY DISCARD ACK	7.34
	AE	ACTIVITY END	7.35
	AEA	ACTIVITY END ACK	7.36
	PR	PREPARE (Note 7)	7.26
	GT	GIVE TOKENS (Note 8)	7.16
	PT	PLEASE TOKENS (Note 8)	7.17
	GTC	GIVE TOKENS CONFIRM (Note 9)	7.18
	GTA	GIVE TOKENS ACK (Note 9)	7.19

Note 1 - An implementation (see ¤ 9) is required to be able to:

a) send a CONNECT SPDU and receive an ACCEPT SPDU or a REFUSE SPDU, or

b) receive a CONNECT SPDU and send an ACCEPT SPDU or a REFUSE SPDU, or

c) send and receive both.

Note 2 - These SPDUs are only used when the SSDU passed in the S-CONNECT request is segmented [see ¤ 6.3.5 b)].

Note 3 - Reception and correct action is mandatory; transmission is optional if the transport connection is not to be reused (see ¤ 7.10.2).

Note 4 - Used to manage the data token.

Note 5 - Used to manage the release token.

Note 6 - Used to manage the synchronize-minor token.

Note 7 - PREPARE SPDU is mandatory if the transport expedited flow is available to this session connection, otherwise it is not used (see ¤ 6.4).

Note 8 - Used to manage the major/activity token.

Note 9 - Used only on session connections on which activity management has been selected, for giving all available tokens, when no activity is in progress.

5.5.7 *Capability data exchange functional unit*

The capability data functional unit supports the capability data exchange service, which allows a confirmed transfer of SS-user data when the activity management functional unit has been selected, but when no activity is in progress.

5.5.8 *Minor synchronize functional unit*

The minor synchronize functional unit supports the minor synchronization service which enables the SS-user to request that the SPM places minor synchronization points in the normal data flow. These minor synchronization points are identified by serial numbers.

5.5.9 *Major synchronize functional unit*

The major synchronize functional unit supports the major synchronize service which enables the SS-user to request that the SPM places major synchronization points in the normal data flow. These major synchronization before ane identified by serial numbers, and clearly separate the data flow before after the major synchronization point.

5.5.10 *Resynchronize functional unit*

The resynchronize functional unit supports the resynchronize service which enables the SS-users to modify the synchronization point serial number and reassign the tokens.

5.5.11 *Exceptions functional unit*

The exceptions functional unit allows both the SPM and the SS-users to report detected errors, rather than aborting the session connection.

This functional unit can only be selected when the half-duplex functional unit has been selected.

5.5.12 *Activity management functional unit*

The activity management functional unit supports the activity management services which allow the SS-users to manage synchronized logical pieces of work.

5.6 Tokens

Table 4/X.225 specifies those functional units that have tokens associated with them.

The SPM may only send an SPDU listed in Table 5/X.225 (and accept the associated service primitive) subject to the availability and assignment of tokens defined in that table.

TABLE 4/X.225

Tokens associated with functional units

Functional unit	Token
Negotiated release	release token
Half-duplex	data token
Minor synchronize	synchronize-minor token
Major synchronize	major/activity token
Activity management	major/activity token

TABLE 5/X.225

Token restrictions

SPDUs	data token	synchronize-minor token	major/activity token	release token
FINISH SPDU	2	2	2	2
NOT FINISHED SPDU	nr	nr	nr	0
DATA TRANSFER SPDU (Half-duplex)	1	nr	nr	nr
DATA TRANSFER SPDU (Duplex)	3	nr	nr	nr
CAPABILITY DATA SPDU	2	2	1	nr
GIVE TOKEN SPDU:				
Data token	1	nr	nr	nr
synchronize-minor token	nr	1	nr	nr
major/activity token	nr	nr	1	nr
release token	nr	nr	nr	1
PLEASE TOKEN SPDU:				
data token	0	nr	nr	nr
synchronize-minor token	nr	0	nr	nr
major/activity token	nr	nr	0	nr
release token	nr	nr	nr	0
GIVE TOKENS CONFIRM SPDU	2	2	1	2
MINOR SYNC POINT SPDU	2	1	nr	nr
MAJOR SYNC POINT SPDU	2	2	1	nr
EXCEPTION REPORT SPDU	0	nr	nr	nr
EXCEPTION DATA SPDU	0	nr	nr	nr
ACTIVITY START SPDU	2	2	1	nr
ACTIVITY RESUME SPDU	2	2	1	nr
ACTIVITY INTERRUPT SPDU	nr	nr	1	nr
ACTIVITY DISCARD SPDU	nr	nr	1	nr
ACTIVITY DISCARD SPDU	2	2	1	nr

0: Token available and not assigned to the SS-user who initiated the associated service primitive

1: Token available and assigned to the SS-user who initiated the associated service primitive

2: Token not available or token assigned to the SS-user who initiated the associated service primitive

3: Token not available

nr: No restriction

5.7 Negotiation

Negotiation takes place between both SPMs during session connection establishment according to the following rules.

5.7.1 *Negotiation of functional units*

Each SPM proposes use or non-use of each functional unit, except for the kernel functional unit, based on requirements from the SS-users. The functional unit is selected only if both the initiator and the responder propose use of the functional unit.

The capability data exchange functional unit can only be proposed if the activity management functional unit is also proposed.

The exceptions functional unit can only be proposed if the half-duplex functional unit is also proposed.

5.7.2 *Negotiation of initial token settings*

When the initiator proposes use of a functional unit that requires a token, it also proposes the initial token setting:

a) initiator's side;

b) responder's side;

c) called SS-user's choice.

If use of the functional unit is selected, the token is set to the side proposed by the initiator. If the initiator proposed Òcalled SS-user choiceÓ, the responder's proposed token setting is selected.

5.7.3 *Negotiation of initial serial number*

When the initiator proposes any of the minor synchronize, major synchronize or resynchronize functional units but does not propose the activity management functional unit, it also proposes an initial serial number.

When the initiator proposes any of the minor synchronize, major synchronize or resynchronize functional units and also proposes the activity management functional unit, it may also propose an initial serial number.

In all other cases, the initiator does not propose an initial serial number.

When the responder proposes any of the minor synchronize, major synchronize or resynchronize functional units but does not propose the activity management functional unit, it also proposes an initial serial number, which is the first serial number to be used.

In all other cases, the responder does not propose an initial serial number.

5.7.4 *Negotiation of version number*

Each SPM indicates all the appropriate versions of the protocol that it is capable of supporting.

5.7.5 *Negotiation of maximum TSDU size*

Each SPM proposes a maximum TSDU size that the initiator is permitted to send. The lesser of the two numbers is used. A zero value is interpreted to mean unlimited TSDU size. If either SPM proposes zero, the initiator may not send segmented data or typed data SSDUs.

Each SPM also proposes a maximum TSDU size that the responder is permitted to send. The lesser of the two numbers is used. A zero value is interpreted to mean unlimited TSDU size. If either SPM proposes zero, the responder may not send segmented data or typed data SSDUs on the session connection.

5.8 *Local variables*

This Recommendation uses local variables as a means of clarifying the effect of certain actions and clarifying the conditions under which certain actions are valid.

5.8.1 *Vact*

Vact is used by the SPM to determine if an activity is in progress when the activity management functional unit has been selected:

Vact = true: an activity is in progress;

Vact = false: no activity is in progress.

5.8.2 *Vnextact*

Vnextact is used by the SPM when the activity management functional unit has been selected:

Vnextact = true: a MAJOR SYNC POINT SPDU has been sent or received;

Vnextact = false: an ACTIVITY END SPDU has been sent or received.

5.8.3 *V(A)*

V(A) is used by the SPM and is the lowest serial number to which a synchronization point confirmation is expected. No confirmation is expected when V(A) = V(M).

5.8.4 *V(M)*

V(M) is used by the SPM and is the next serial number to be used.

5.8.5 *V(R)*

V(R) is used by the SPM and is the lowest serial number to which resynchronization restart is permitted.

5.8.6 *Vsc*

Vsc is used by the SPM to determine whether or not the SS-user has the right to send minor synchronization point responses. Vsc has the following values:

Vsc = true: the SS-user has the right to issue minor synchronization point responses when V(A) is less than V(M);

Vsc = false: the SS-user does not have the right to issue minor synchronization point responses.

Note - The manipulation of V(A), V(M), V(R) and Vsc and the circumstances under which they are updated are specified in ¤ 7 and are summarized in a Table A-4/X.225 in Annex A.

6 Use of the transport service

This section defines the way that the transport service primitives are used by the SPM.

6.1 *Assignment of a session connection to the transport connection*

6.1.1 *Purpose*

Assignment of a session connection to a transport connection.

6.1.2 Transport service primitives

The procedure uses the following transport service primitives:

T-CONNECT request

T-CONNECT indication

T-CONNECT response

T-CONNECT confirm

T-DISCONNECT request

T-DISCONNECT indication

6.1.3 *SPDUs used*

No SPDUs are used during assignment to a transport connection.

6.1.4 *Description*

A session connection is assigned to an existing transport connection suitable for reuse, or a new transport connection is created for the purpose. This assignment is based on the quality of service (see Recommendation X.215) requested by the SS-user in the S-CONNECT request.

If a transport connection is established with the transport expedited data option, the transport expedited data flow is available for the duration of the transport connection. Use of transport expedited data is specified in ¤ 6.4.

The transport expedited flow is requested by the SPM when the T-CONNECT request is issued if:

a) the SS-user requested the expedited data functional unit, or

b) the SS-user requested an extended control QOS for the session connection.

Only the initiator of the transport connection is permitted to issue the CONNECT SPDU.

When a session connection is terminated, the underlying transport connection is also terminated, unless reuse of the transport connection has been agreed.

Use of the TS-user data parameter in T-CONNECT request, indication, response and confirm is reserved for future use. When T-CONNECT request or a T-CONNECT response is issued, this parameter is empty. When a T-CONNECT indication or T-CONNECT confirm is received, this parameter is ignored.

6.2 Reuse of the transport connection

6.2.1 *Purpose*

To allow the transport connection to be retained for reuse by another session connection.

6.2.2 *Transport service primitives*

The procedure uses the following transport service primitives:

T-DATA request

T-DATA indication

6.2.3 *SPDUs used*

The following SPDUs are related to reuse of the transport connection:

REFUSE SPDU (see ¤ 7.5);

FINISH SPDU (see ¤ 7.6);

DISCONNECT SPDU (see ¤ 7.7);

ABORT SPDU (see ¤ 7.9);

ABORT ACCEPT SPDU (see ¤ 7.10).

6.2.4 *Description*

When a session connection is refused, or has been successfully connected and subsequently disconnected, by abort or orderly release, the supporting transport connection may be either disconnected or reused.

The transport connection may be kept for reuse provided that the transport expedited flow is not available, and either:

a) the SPM which established the transport connection requests retention of the transport connection by parameter in an ABORT SPDU or a FINISH SPDU, or

b) the SPM which established the transport connection receives a REFUSE SPDU or an ABORT SPDU which indicates by parameter that the transport connection is to be retained.

To avoid contention for a retained transport connection, only the transport connection initiator may reuse the transport connection by sending a CONNECT SPDU to establish a new session connection.

6.3 Use of transport normal data

6.3.1 *Purpose*

To convey SPDUs in user data fields of transport service normal data primitives.

6.3.2 *Transport service primitives*

The procedure uses the following transport service primitives:

T-DATA request

T-DATA indication

6.3.3 *SPDUs used*

The following SPDUs are sent on the transport normal flow:

CONNECT SPDU (see ¤ 7.1);

OVERFLOW ACCEPT SPDU (see ¤ 7.2);

CONNECT DATA OVERFLOW SPDU (see ¤ 7.3);

ACCEPT SPDU (see ¤ 7.4);

REFUSE SPDU (see ¤ 7.5);

FINISH SPDU (see ¤ 7.6);

DISCONNECT SPDU (see ¤ 7.7);

NOT FINISHED SPDU (see ¤ 7.8);

DATA TRANSFER SPDU (see ¤ 7.11);

TYPED DATA SPDU (see ¤ 7.13);

CAPABILITY DATA SPDU (see ¤ 7.14);

CAPABILITY DATA ACK SPDU (see ¤ 7.15);

GIVE TOKENS SPDU (see (¤ 7.16);

PLEASE TOKENS SPDU (see ¤ 7.17);

GIVE TOKENS CONFIRM SPDU (see ¤ 7.18);

GIVE TOKENS ACK SPDU (see ¤ 7.19);

MINOR SYNC POINT SPDU (see ¤ 7.20);

MINOR SYNC ACK SPDU (see ¤ 7.21);

MAJOR SYNC POINT SPDU (see ¤ 7.22);

MAJOR SYNC ACK SPDU (see ¤ 7.23);

RESYNCHRONIZE SPDU (see ¤ 7.24);

RESYNCHRONIZE ACK SPDU (see ¤ 7.25);

EXCEPTION REPORT SPDU (see ¤ 7.27);

EXCEPTION DATA SPDU (see ¤ 7.28);

ACTIVITY START SPDU (see ¤ 7.29);

ACTIVITY RESUME SPDU (see ¤ 7.30);

ACTIVITY INTERRUPT SPDU (see ¤ 7.31);

ACTIVITY INTERRUPT ACK SPDU (see ¤ 7.32);

ACTIVITY DISCARD SPDU (see ¤ 7.33);

ACTIVITY DISCARD ACK SPDU (see ¤ 7.34);

ACTIVITY END SPDU (see ¤ 7.35);

ACTIVITY END ACK SPDU (see ¤ 7.36).

If the SS-user data exceeds 9 octets or if the transport extended flow is not available, the following additional SPDUs are sent on the transport normal flow:

ABORT SPDU (see ¤ 7.9);

ABORT ACCEPT SPDU (see ¤ 7.10).

6.3.4 *Transfer of SPDUs*

The SPDUs listed in ¤ 6.3.3 are transferred using the transport normal data transfer service.

6.3.5 *Segmenting*

Segmenting of SSDUs takes place under the following circumstances:

a) when a maximum TSDU size has been selected, in which case a data SSDU or a typed data SSDU may be mapped on to more than one SPDU;

b) when protocol version 2 is proposed or selected and either

i) the SPDU size would exceed the maximum TSDU size, or

ii) the SPDU size would exceed 65539 octets for an SPDU to be sent on the transport normal flow or 16 octets for an SPDU to be sent on the transport expedited flow

in which case SSDUs other than data SSDUs, typed data SSDUs and expedited data SSDUs are mapped into more than one SPDU.

In all other cases, each SSDU is mapped one-to-one onto an SPDU.

Note - Implementors should note that when segmenting is selected:

a) the control information of each SPDU indicates whether or not it contains the first or last segment of the SSDU;

b) the size of the segments of the SSDU is constrained by the maximum TSDU size selected for that direction of transfer.

6.3.6 *Maximum TSDU size*

When a maximum TSDU size has been selected, the SPDU size may not exceed the maximum TSDU size selected for that direction of transfer and a sequence of concatenated SPDUs may not exceed the maximum TSDU size selected for that direction of transfer.

6.3.7 Concatenation

Each SPDU is defined in Table 6/X.225 as belonging to one of the following categories:

a) *Category 0 SPDUs* which may be mapped one-to-one onto a TSDU or may be concatenated with one or more Category 2 SPDUs;

b) *Cateogry 1 SPDUs* which are always mapped one-to-one onto a TSDU;

c) *Category 2 SPDUs* which are never mapped one-to-one onto a TSDU.

Basic concatenations of a category 0 SPDU with a single category 2 SPDU, defined as valid and in the order indicated in Table 7/X.225, may always be mapped onto a single TSDU.

If the receiving SPM has indicated that it can accept extended concatenation, the sending SPM may map a category 0 SPDU with one or more category 2 SPDUs (as specified in Table 8/X.225) onto a single TSDU in the case where this concatenated sequence does not fit into a single TSDU, extended concatenation cannot be applied.

The valid mappings of SPDUs onto TSDUs are illustrated in Figure 3/X.225.

Any other concatenation of SPDUs is defined as invalid.

6.3.7.1 *Processing order of concatenated SPDUs*

On receipt of SPDUs that have been concatenated using basic concatenation, the category 2 SPDUs are processed before the category 0 SPDU.

On receipt, SPDUs that have been concatenated using extended concatenation are processed in the following order:

a) ACTIVITY START SPDU or

ACTIVITY RESUME SPDU;

b) DATA TRANSFER SPDU;

c) MINOR SYNC POINT SPDU or

MINOR SYNC ACK SPDU or

MAJOR SYNC POINT SPDU or

MAJOR SYNC ACK SPDU or

ACTIVITY END SPDU or

ACTIVITY END ACK SPDU;

d) GIVE TOKENS SPDU or

PLEASE TOKENS SPDU.

TABLE 6/X.225

Category 0, 1 and 2 SPDUs

Category 0 SPDUs	Category 1 SPDUs	Category 2 SPDUs
GIVE TOKENS SPDU	CONNECT SPDU	DATA TRANSFER SPDU
PLEASE TOKENS SPDU	ACCEPT SPDU	
		MINOR SYNC POINT SPDU
	REFUSE SPDU	MINOR SYNC POINT SPDU
	FINISH SPDU	
	DISCONNECT SPDU	MAJOR SYNC POINT SPDU
	NOT FINISHED SPDU	MAJOR SYNC ACK SPDU
	ABORT SPDU	
	ABORT ACCEPT SPDU	RESYNCHRONIZE SPDU
		RESYNCHRONIZE ACK SPDU
	GIVE TOKENS CONFIRM SPDU	
	GIVE TOKENS ACK SPDU	ACTIVITY START SPDU
		ACTIVITY RESUME SPDU
	EXPEDITED SPDU	ACTIVITY DISCARD SPDU
	PREPARE SPDU	ACTIVITY DISCARD ACK SPDU
	TYPED DATA SPDU	ACTIVITY INTERRUPT SPDU
		ACTIVITY INTERRUPT ACK SPDU
		ACTIVITY END SPDU
	OVERFLOW ACCEPT SPDU	ACTIVITY END ACK SPDU
	CONNECT DATA OVERFLOW SPDU	CAPABILITY DATA SPDU
		CAPABILITY DATA ACK SPDU
		EXCEPTION REPORT SPDU
		EXCEPTION DATA SPDU

TABLE 7/X.225

Valid basic concatenation of SPDUs

First SPDU	Second SPDU
GIVE TOKENS SPDU	DATA TRANSFER SPDU
GIVE TOKENS SPDU PLEASE TOKENS SPDU	MINOR SYNC POINT SPDU MINOR SYNC ACK SPDU
GIVE TOKENS SPDU PLEASE TOKENS SPDU	MAJOR SYNC POINT SPDU MAJOR SYNC ACK SPDU
GIVE TOKENS SPDU [a)] PLEASE TOKENS SPDU	RESYNCHRONIZE SPDU RESYNCHRONIZE ACK SPDU
GIVE TOKENS SPDU GIVE TOKENS SPDU	ACTIVITY START SPDU ACTIVITY RESUME SPDU
GIVE TOKENS SPDU [a)] PLEASE TOKENS SPDU	ACTIVITY DISCARD SPDU ACTIVITY DISCARD ACK SPDU
GIVE TOKENS SPDU [a)] PLEASE TOKENS SPDU	ACTIVITY INTERRUPT SPDU ACTIVITY INTERRUPT ACK SPDU
GIVE TOKENS SPDU PLEASE TOKENS SPDU	ACTIVITY END SPDU ACTIVITY END ACK SPDU
GIVE TOKENS SPDU [a)] PLEASE TOKENS SPDU	CAPABILITY DATA SPDU CAPABILITY DATA ACK SPDU
PLEASE TOKENS SPDU PLEASE TOKENS SPDU	EXCEPTION REPORT SPDU EXCEPTION DATA SPDU

a) Indicates that the Token Item parameter is not present in the GIVE TOKENS SPDU. In all other cases, the Token Item parameter may or may not be present.

In all cases, the Token Item parameter may only be present in the first SPDU if the second SPDU contains either a complete SSDU, or the last segmented SSDU.

Basic concatenation of a PLEASE TOKENS SPDU or a GIVE TOKENS SPDU with a second SPDU is only permitted when the user data parameter is not present in the PLEASE TOKENS SPDU or the GIVE TOKENS SPDU.

TABLE 8/X.225

Valid extended concatenation of SPDUs

First SPDU	Second SPDU	Third SPDU	Fourth SPDU	Status
GIVE TOKENS SPDU	MINOR SYNC ACK SPDU			
GIVE TOKENS SPDU	MAJOR SYNC ACK SPDU			
GIVE TOKENS SPDU	ACTIVITY END ACK SPDU			
GIVE TOKENS SPDU	ACTIVITY START SPDU	MINOR SYNC POINT SPDU		
GIVE TOKENS SPDU	ACTIVITY RESUME SPDU	MINOR SYNC POINT SPDU		
GIVE TOKENS SPDU [a)]	ACTIVITY START SPDU	ACTIVITY END SPDU		
GIVE TOKENS SPDU [a)]	ACTIVITY RESUME SPDU	ACTIVITY END SPDU		
GIVE TOKENS SPDU [a)]	ACTIVITY START SPDU	MAJOR SYNC POINT SPDU		
GIVE TOKENS SPDU [a)]	ACTIVITY RESUME SPDU	MAJOR SYNC POINT SPDU		
GIVE TOKENS SPDU	MINOR SYNC POINT SPDU	DATA TRANSFER SPDU		CL
GIVE TOKENS SPDU	MINOR SYNC ACK SPDU	DATA TRANSFER SPDU		CL
GIVE TOKENS SPDU	MAJOR SYNC POINT SPDU	DATA TRANSFER SPDU		CL
GIVE TOKENS SPDU	MAJOR SYNC ACK SPDU	DATA TRANSFER SPDU		CL
GIVE TOKENS SPDU	ACTIVITY START SPDU	DATA TRANSFER SPDU		CF
GIVE TOKENS SPDU	ACTIVITY RESUME SPDU	DATA TRANSFER SPDU		CF
GIVE TOKENS SPDU	ACTIVITY END SPDU	DATA TRANSFER SPDU		CL
GIVE TOKENS SPDU	ACTIVITY END ACK SPDU	DATA TRANSFER SPDU		CL

TABLE 8/X.225 *(cont.)*

First SPDU	Second SPDU	Third SPDU	Fourth SPDU	Status
GIVE TOKENS SPDU	ACTIVITY START SPDU	MINOR SYNC POINT SPDU	DATA TRANSFER SPDU	C
GIVE TOKENS SPDU	ACTIVITY RESUME SPDU	MINOR SYNC POINT SPDU	DATA TRANSFER SPDU	C
GIVE TOKENS SPDU [a)]	ACTIVITY START SPDU	ACTIVITY END SPDU	DATA TRANSFER SPDU	C
GIVE TOKENS SPDU [a)]	ACTIVITY RESUME SPDU	ACTIVITY END SPDU	DATA TRANSFER SPDU	C
GIVE TOKENS SPDU [a)]	ACTIVITY START SPDU	MAJOR SYNC POINT SPDU	DATA TRANSFER SPDU	C
GIVE TOKENS SPDU [a)]	ACTIVITY RESUME SPDU	MAJOR SYNC POINT SPDU	DATA TRANSFER SPDU	C

a) Indicates that the Token Item parameter is not present in the GIVE TOKENS SPDU.

Status:

CL: The DATA TRANSFER SPDU contains a complete SSDU or the last segment of an SSDU.

CF: The DATA TRANSFER SPDU contains a complete SSDU or the first segment of an SSDU. In the latter case, the Token Item parameter is not present in the GIVE TOKENS SPDU.

C: The DATA TRANSFER SPDU contains a complete SSDU.

Figure 3/X.225, p.15

6.4 Use of transport expedited data

6.4.1 *Purpose*

To convey SPDUs on a separate transport flow.

6.4.2 *Transport service primitives*

The procedure uses the following transport service primitives:

T-EXPEDITED-DATA request

T-EXPEDITED-DATA indication

6.4.3 *SPDUs used*

The following SPDUs are sent on the transport expedited flow when it is available:

ABORT SPDU (see ¤ 7.9);

ABORT ACCEPT SPDU (see ¤ 7.10);

EXPEDITED DATA SPDU (see ¤ 7.12);

PREPARE SPDU (see ¤ 7.26).

6.4.4 *Description*

The SPDUs listed in ¤ 6.4.3 are sent on the transport expedited flow if it is selected, and may be used to bypass any flow control restrictions or congestion on the transport normal flow. SPDUs sent on the transport expedited flow may be delivered to the accepting SS-user earlier than SSDUs submitted previously by the sending SS-user and sent on the transport normal flow, but no later than subsequently submitted SSDUs.

When the transport expedited flow is not available:

a) EXPEDITED DATA SPDUs are not sent;

b) ABORT and ABORT ACCEPT SPDUs are sent on the transport normal flow;

c) PREPARE SPDUs are not sent.

6.5 Flow control

There is no peer flow control in the session layer. To prevent the SS-users from being overloaded with data, the receiving SPM may apply back pressure across the transport connection, using the transport flow control. The decision on when or how back pressure is applied is a local matter.

6.6 Transport disconnection

6.6.1 *Purpose*

To release a transport connection.

6.6.2 *Transport service primitives*

The procedure uses the following transport service primitives:

T-DISCONNECT request

T-DISCONNECT indication

6.6.3 *SPDUs used*

No SPDUs are used.

6.6.4 *Description*

After the session connection has been released or aborted and the transport connection is not to be reused, the transport connection is disconnected.

When a T-DISCONNECT indication is received, as a result of an error detected by the transport service provider, the SPM issues an S-P-ABORT indication to the local SS-user.

When issuing a T-DISCONNECT request, the SPM may optionally use the T-DISCONNECT user data field to indicate the reason for the transport disconnection to the remote SPM. The reason code consists of one octet with the following values:

a) 0 - session protocol error for which an ABORT SPDU could not be sent;

b) 1 - normal transport disconnection when the transport connection is not to be reused;

c) 2 - normal transport disconnection when the transport connection was to be reused, but reuse is not possible for local reasons.

The use of the Disconnect Reason parameter in T-DISCONNECT indication is a local matter.

7 Elements of procedure related to SPDUs

This section defines valid sequences of operation of the protocol.

A more precise definition of procedures is contained in Annex A which incoporates all the checks to determine the validity of a particular event at a particular point in time. In case of arbitration or dispute, Annex A takes precedence over this section.

The elements of procedure specified in ¤¤ 7.4-7.8, 7.14-7.18, 7.20-7.23 and 7.28-7.36 do not consider the case where an SSDU is segmented. (The circumstances under which an SSDU may be segmented are specified in ¤ 6.3.5). Additional elements of procedure for segmented SSDUs are specified in ¤ 7.37.

7.1 *CONNECT SPDU*

The CONNECT SPDU is transmitted by the initiator of the transport connection on a previously assigned transport connection in order to initiate a session connection.

7.1.1 *Content of CONNECT SPDU*

The CONNECT SPDU contains:

a) Connection Identifier parameter group, which is supplied by the calling SS-user, to enable the SS-users to identify this specific session connection. This parameter group has no effect on the SPM. It contains:

1) Calling SS-user Reference parameter;

2) Common Reference parameter;

3) Additional Reference Information parameter.

b) Connect/Accept Item parameter group containing:

1) Protocol Options parameter which enables the initiator to indicate its ability to receive extended concatenated SPDUs;

2) TSDU maximum size parameter which, if present and not zero, indicates that the initiator's proposed values for the maximum TSDU sizes for each direction of transfer (see ¤¤ 5.7.5 and 6.3.5). If this parameter is not present or is zero, the TSDU size is not limited.

3) Version Number parameter to identify all versions of this protocol which are supported and are suitable for this session connection.

Note - Protocol version 1 is not suitable if there are more than 512 octets of SS-user data in this SPDU.

4) Initial Serial Number parameter which is proposed by the calling SS-user in the case where the activity management functional unit is not proposed and any of the minor synchronize, major synchronize or resynchronize functional units are proposed. As an SS-user option, an Initial Serial Number parameter may be proposed even if the activity management functional unit is proposed provided that any of the minor synchronize, major synchronize or resynchronize functional units are also proposed;

5) Token Setting Item parameter supplied by the calling SS-user, which proposes the initial token positions for each token available on this connection, as derived from the functional units proposed in the Session User Requirements parameter (see Table 4/X.225). The initial token positions can be specified to be on the initiator's side or on the acceptor's side or the initiator can specify that the decision is to be made by the called SS-user.

c) Session User Requirements parameter containing a list of the functional units proposed by the calling SS-user. At least one of the half-duplex and the duplex functional untis shall be proposed. The SPM is required to provide the associated protocol functions.

d) Calling Session Selector and Called Session Selector parameters corresponding to the calling SS-user and the called SS-user may be present and are derived from session addresses provided by the calling SS-user.

e) Either a User Data parameter which allows a limited (512 or less octets) amount of transparent user data to be passed from the calling SS-user to the called SS-user or

an Extended User Data parameter which allows between 513 and 10 240 octets of transparent user data to be passed from the calling SS-user to the called SS-user. This parameter shall not be present if protocol version 1 is proposed.

Only one of these two parameters may be used on the CONNECT SPDU.

f) Data Overflow parameter which shall be present if and only if there is more than 10 240 octets of SS-user data and which indicates to the responder that there is more SS-user data to follow. The first 10240 octets of SS-user data are sent in the Extended User Data parameter. This parameter shall not be present if protocol version 1 is proposed.

7.1.2 *Sending the CONNECT SPDU*

An S-CONNECT request results in the assignment of a transport connection. When the transport connection is established, a CONNECT SPDU is sent on the transport normal flow. If the Data Overflow parameter was not present in the CONNECT SPDU, the SPM waits until it receives an ACCEPT SPDU or a REFUSE SPDU. If the Data Overflow parameter was present in the CONNECT SPDU, the SPM waits until it receives an OVERFLOW ACCEPT or a REFUSE SPDU.

7.1.3 *Receiving the CONNECT SPDU*

A valid incoming CONNECT SPDU which is acceptable to the receiving SPM and does not contain the Data Overflow parameter results in an S-CONNECT indication to an SS-user, according to the Called Session Selector parameter of the CONNECT SPDU. The SPM then waits for an S-CONNECT response from the called SS-user.

A valid incoming CONNECT SPDU which is acceptable to the receiving SPM, contains the Data Overflow parameter and provided that protocol version 2 is to be selected, results in the SPM sending an OVERFLOW ACCEPT SPDU; the SPM then waits until it receives a CONNECT DATA OVERFLOW SPDU. Otherwise the SPM sends a REFUSE SPDU (see ¤ 7.5).

7.2 OVERFLOW ACCEPT SPDU

The OVERFLOW ACCEPT SPDU is used by the SPM to request the remainder of the S-CONNECT request SS-user data.

7.2.1 *Content of the OVERFLOW ACCEPT SPDU*

The OVERFLOW ACCEPT SPDU contains:

a) TSDU Maximum Size parameter which, if present and not zero indicates that segmenting has been proposed by the responder (see ¤ 6.3.5). The responder proposes alternative values for the maximum TSDU sizes for each direction of transfer (see ¤ 5.7.5). These values may be larger than, smaller than or equal to the values supplied by the initiator in the CONNECT SPDU. The smaller value for each direction of transfer is used for the maximum TSDU size for that direction of transfer;

b) Version Number parameter indicating that at least protcol version 2 is supported.

7.2.2 *Sending the OVERFLOW ACCEPT SPDU*

A valid incoming CONNECT SPDU which contains the Data Overflow parameter results in the SPM sending an OVERFLOW ACCEPT SPDU. The SPM then waits until it receives a CONNECT DATA OVERFLOW SPDU.

7.2.3 *Receiving the OVERFLOW ACCEPT SPDU*

A valid incoming OVERFLOW ACCEPT SPDU results in the SPM sending one or more CONNECT DATA OVERFLOW SPDUs. When the last CONNECT DATA OVERFLOW SPDU has been sent, the SPM waits until it receives an ACCEPT SPDU or a REFUSE SPDU.

7.3 CONNECT DATA OVERFLOW SPDU

The CONNECT DATA OVERFLOW SPDU is used by the initiator to send subsequent segments of the User Data associated with an S-CONNECT request.

7.3.1 *Content of the CONNECT DATA OVERFLOW SPDU*

The CONNECT DATA OVERFLOW SPDU contains:

a) an Enclosure Item parameter to indicate whether the SPDU is the middle or the end of the SSDU;

b) a User data parameter which allows a maximum of 65528 octets of transparent user data to be transferred.

7.3.2 *Sending the CONNECT DATA OVERFLOW SPDU*

A valid incoming OVERFLOW ACCEPT SPDU results in the SPM sending one or more CONNECT DATA OVERFLOW SPDUs. These SPDUs will be sent as an ordered sequence with the appropriate value for the Enclosure Item parameter until the complete SSDU has been transferred.

7.3.3 *Receiving the CONNECT DATA OVERFLOW SPDU*

A valid incoming CONNECT DATA OVERFLOW SPDU with Enclosure Item parameter indicating Òend of SSDUÓ results in an S-CONNECT indication to pass the entire SSDU to the SS-user, according to the Called Session selector parameter of the CONNECT SPDU. The SPM then waits for an S-CONNECT response from the called SS-user.

If the Enclosure Item parameter in a valid incoming CONNECT DATA OVERFLOW SPDU indicates Ònot end of SSDUÓ, the SPM waits for a subsequent valid CONNECT DATA OVERFLOW SPDU.

7.4 *ACCEPT SPDU*

An SPM receiving a CONNECT SPDU with the Data Overflow parameter absent may accept a proposal to establish a session connection by transferring an ACCEPT SPDU (after receiving an S-CONNECT response primitive) to the initiator, on the same transport connection.

An SPM which has previously issued an OVERFLOW ACCEPT SPDU in response to a CONNECT SPDU with the Data Overflow parameter present and which subsequently receives the sequence of CONNECT DATA OVERFLOW SPDUs which complete the segmented SSDU may accept a proposal to establish a session connection by trasferring an ACCEPT SPDU (after receiving an S-CONNECT response primitive) to the initiator, on the same transport connection.

7.4.1 *Content of ACCEPT SPDU*

The ACCEPT SPDU contains:

a) Connection Identifier parameter group, which is supplied by the called SS-user, to enable the SS-users to identify this specific session connection. This parameter group has no effect on the SPM. It contains:

1) Called SS-user Reference parameter;

2) Common Reference parameter;

3) Additional Reference Information parameter.

b) Connect/Accept Item parameter group containing:

1) Protocol Options parameter which allows the responder to indicate its ability to receive extended concatenated SPDUs;

2) TSDU maximum size parameter which, if present and not zero, indicates the responder's proposed values for the maximum TSDU sizes for each direction of transfer (see ¤¤ 5.7.6 and 6.3.5). These values may be larger than, smaller than or equal to the values supplied by the initiator in the CONNECT SPDU. The smaller value is used for the maximum TSDU size for each direction of transfer. If an OVERFLOW ACCEPT SPDU has previously been sent on this session connection then:

i) if the TSDU Maximum Size parameter was present in the OVERFLOW ACCEPT SPDU, then it shall also be present in the ACCEPT SPDU, with the same values as were given in the OVERFLOW ACCEPT SPDU.

ii) if the TSDU Maximum Size parameter was not present in the OVERFLOW ACCEPT SPDU, then it shall not be present in the ACCEPT SPDU;

3) Version Number parameter to identify all versions of this protocol which are supported and are suitable for this session connection. If an OVERFLOW ACCEPT SPDU has been previously sent on this session connection then the Version Number parameter shall be present in the ACCEPT SPDU, with the same value as was given in the OVERFLOW ACCEPT SPDU. The highest version number indicated by both initiator and responder is used;

Note - Protocol version 1 is not suitable if there are more than 512 octets of User Data in this SPDU.

4) Initial Serial Number parameter which is present if the activity management functional unit is not selected and any of the minor synchronize, major synchronize or resynchronize functional units are selected regardless of whether or not the activity management functional unit is proposed. The called SS-user proposes the value, which is the value of the first serial number to be used;

5) Token Setting Item parameter supplied by the called SS-user, which indicates the initial token positions for each token available on this session connection, as derived from the selected functional units. A token is only available if any functional unit which requires that token has been selected for use on this session connection (see Table 4/X.225), regardless of the settings of the Token Setting Item parameter in the CONNECT SPDU (see ¤ 7.1.1 b) 5)). If a token-controlled functional unit has been selected, then in the case where the calling SS-user has indicated that the initial assignment of the related token is the called SS-user's choice, this parameter contains a value chosen by the called SS-user. Otherwise, the values indicated by the calling SS-user in the CONNECT SPDU are selected and must be returned.

c) Token Item parameter which allows the called SS-user to request tokens which have been assigned to the calling SS-user in the CONNECT SPDU.

d) Session User Requirements parameter which contains a list indicating the functional units proposed by the called SS-user and can be supported by the responder. The functional units selected for use on this session connection are the intersection of this set and the set proposed in the CONNECT SPDU (i.e. only those functional units indicated in both the CONNECT SPDU and the ACCEPT SPDU are selected). If both the half-duplex functional unit and the duplex functional unit were indicated in the CONNECT SPDU, then the ACCEPT SPDU must propose which one is to be available. If only one of these functional units was indicated in the CONNECT SPDU, then the ACCEPT SPDU must indicate that the same functional unit is to be used (or the connection attempt must be rejected).

e) Calling Session Selector parameter corresponding to the calling SS-user may be present, in which case it will have the same value as in the CONNECT SPDU. Responding Session Selector parameter corresponding to the responding Session Address provided by the responding SS-user.

f) User Data parameter which allows a limited amount of transparent user data to be passed from the called SS-user to the calling SS-user.

7.4.2 *Sending the ACCEPT SPDU*

An S-CONNECT (accept) response results in an ACCEPT SPDU. This SPDU is sent on the transport normal flow. After this successful connection, the SPM enters the data transfer phase and can receive any service request or SPDU that is allowed by the selected functional units and current token positions. If any of the minor synchronize, major synchronize or resynchronize functional units are selected but the activity management functional unit is not selected, the SPM sets V(A) and V(M) to the Initial Serial Number proposed by the called SS-user, which is the serial number to be used for the first synchronization point. V(R) is set to zero. Vsc is set false. If the activity management functional unit has been selected, Vact is set false.

7.4.3 *Receiving the ACCEPT SPDU*

A valid incoming ACCEPT SPDU results in an S-CONNECT (accept) confirm. After this successful connection, the SPM enters the data transfer phase and can receive any service request or SPDU that is allowed by the available functional units and current token positions. If any of the minor synchronize, major synchronize or resynchronize functional units are selected but the activity management functional unit is not selected, the SPM sets V(A) and V(M) to the Initial Serial Number contained in the ACCEPT SPDU, which is the serial number to be used for the first synchronization point. V(R) is set to zero. Vsc is set false. If the activity management functional unit has been selected, Vact is set false.

If the called SS-user has requested any tokens in the Token Item parameter of the ACCEPT SPDU (see ¤ 7.4.1 c)), an S-PLEASE-TOKEN indication is also generated.

7.5 *REFUSE SPDU*

A REFUSE SPDU is used by the responder to reject an attempt to establish a session connection.

7.5.1 *Content of REFUSE SPDU*

The REFUSE SPDU contains:

a) Connection Identifier parameter group, which is supplied by the called SS-user, to enable the SS-users to identify this specific session connection. This parameter group has no effect on the SPM. It contains:

1) Called SS-user Reference parameter;

2) Common Reference parameter;

3) Additional Reference Information parameter.

b) Transport Disconnect parameter which indicates whether or not the transport connection is to be kept.

c) Session User Requirements parameter which contains a list of the functional units supported by the sending SPM, and required by the called SS-user.

d) Version Number parameter to identify which versions of this protocol have been implemented by the sending SPM.

e) Reason Code parameter giving the reason for refusal of the attempt to establish a session connection, together with a limited amount of transparent user data.

7.5.2 *Sending the REFUSE SPDU*

An S-CONNECT (reject) response results in a REFUSE SPDU. This SPDU is sent on the transport normal flow. No session connection is established. If the Transport Disconnect parameter indicates that the transport connection can be reused, the SPM waits for a CONNECT SPDU. Otherwise, the SPM starts the timer, TIM, and waits for a T-DISCONNECT indication. If the timer expires before receipt of a T-DISCONNECT indication, the SPM requests transport disconnection with a T-DISCONNECT request. The timer is cancelled on receipt of a T-DISCONNECT indication.

Note - The value of TIM is a local implementation dependent matter, related to quality of service.

7.5.3 *Receiving the REFUSE SPDU*

A valid incoming REFUSE SPDU results in an S-CONNECT (reject) confirm. No session connection is established. If the Transport Disconnect parameter indicates that retention of the transport connection has been requested by the called SPM, and this is acceptable to the calling SPM, the SPM waits for an S-CONNECT request. Otherwise, the SPM releases the transport connection, by making a T-DISCONNECT request.

7.6 *FINISH SPDU*

Orderly release is initiated by transfer of a FINISH SPDU, which may be transferred during the data transfer phase. It requests as a response either:

a) a DISCONNECT SPDU to complete the release of the session connection, or

b) a NOT FINISHED SPDU to refuse the release of the session connection if the release token is available.

The FINISH SPDU is transferred in sequence with any normal data being transferred. The right to issue a FINISH SPDU is restricted to the owner of all available tokens.

7.6.1 *Content of FINISH SPDU*

The FINISH SPDU contains:

a) Transport Disconnect parameter which indicates whether or not the transport connection is to be kept, subject to the restrictions specified in ¤ 6.2.4;

b) User Data parameter which allows a limited amount of transparent user data to be transferred.

7.6.2 *Sending the FINISH SPDU*

An S-RELEASE request results in a FINISH SPDU. This SPDU is sent on the transport normal flow. After transferring a FINISH SPDU, the SPM may not send any further SPDUs (except ABORT SPDU or, in the case of collision of FINISH SPDUs, a DISCONNECT SPDU) unless a NOT FINISHED SPDU or a RESYNCHRONIZE SPDU is received, after which the data transfer phase may be resumed. Receipt of a DISCONNECT SPDU signals completion of orderly session release.

7.6.3 *Receiving the FINISH SPDU*

A valid incoming FINISH SPDU results in an S-RELEASE indication. The user data is passed to the SS-user. The SPM waits for an S-RELEASE response.

7.7 DISCONNECT SPDU

After receipt of a FINISH SPDU, a DISCONNECT SPDU may be transferred. Receipt of a DISCONNECT SPDU after transferring a FINISH SPDU, signals the orderly release of the session connection. The DISCONNECT SPDU is transferred in sequence with any normal data being transferred.

7.7.1 *Content of DISCONNECT SPDU*

The DISCONNECT SPDU contains a User Data parameter which allows a limited amount of transparent user data to be transferred.

7.7.2 *Sending the DISCONNECT SPDU*

An S-RELEASE (accept) response results in a DISCONNECT SPDU. This SPDU is sent on the transport normal flow. The session connection ceases to exist.

If the FINISH SPDU indicated that the transport connection is to be kept for reuse, and this is acceptable, the SPM waits for a CONNECT SPDU. Otherwise, the SPM starts the timer, TIM, and waits for a T-DISCONNECT indication. If the timer expires before receipt of a T-DISCONNECT indication, the SPM requests transport disconnection with a T-DISCONNECT request. The timer is cancelled on receipt of a T-DISCONNECT indication.

Note - The value of TIM is a local implementation dependent matter, related to quality of service.

7.7.3 *Receiving the DISCONNECT SPDU*

A valid incoming DISCONNECT SPDU results in an S-RELEASE (accept) confirm. The session connection ceases to exist.

If the transport connection is to be kept for reuse (see ¤ 6.2.4), the SPM waits for a suitable S-CONNECT request. Otherwise, a T-DISCONNECT request is issued.

Note 1 - In the case of collision of FINISH SPDU and ABORT SPDU (see ¤ 7.9), the ABORT SPDU takes preference and thus the indication in the FINISH SPDU to keep or release the transport connection is ignored.

Note 2 - In the case of collision of FINISH SPDUs (data token and release token not available) the transport connection cannot be reused. The SPM receiving the DISCONNECT SPDU issues a T-DISCONNECT request.

7.8 *NOT FINISHED SPDU*

After receipt of a FINISH SPDU, a NOT FINISHED SPDU may be transferred subject to the token restrictions specified in Table 5/X.225. No confirmation is sought.

7.8.1 *Content of NOT FINISHED SPDU*

The NOT FINISHED SPDU contains a User Data parameter which allows a limited amount of transparent user data to be transferred.

7.8.2 *Sending the NOT FINISHED SPDU*

An S-RELEASE (reject) response results in a NOT FINISHED SPDU. This SPDU is sent on the transport normal flow. The SPM remains in the data transfer phase and can receive any service request or SPDU that is allowed by the available functional units and current token positions.

7.8.3 *Receiving the NOT FINISHED SPDU*

A valid incoming NOT FINISHED SPDU results in an S-RELEASE (reject) confirm. The SPM remains in the data transfer phase and can receive any service request or SPDU that is allowed by the available functional units and current token positions.

7.9 *ABORT SPDU*

The ABORT SPDU is used to reject a session connection establishment attempt, or to cause abnormal release of a session connection at any time. This SPDU is also used by an SPM to release the session connection when a protocol error is detected. The ABORT SPDU may or may not request that the transport connection be released by the receiving SPM. Use of the ABORT SPDU may result in loss of data.

7.9.1 *Content of ABORT SPDU*

7.9.1.1 If there is no SSDU or segmenting of the SSDU is not required (see ¤ 6.3.5), the ABORT SPDU contains:

a) a Transport Disconnect parameter which indicates whether or not the transport connection is to be kept;

b) a Reflect Parameter Values parameter which, if present, allows implementation defined information to be transferred;

c) a User Data parameter which, if present, allows a limited amount of transparent user data to be transferred.

7.9.1.2 If the SSDU is to be segmented, the first ABORT SPDU contains:

a) a Transport Disconnect parameter which indicates whether or not the transport connection is to be kept;

b) an Enclosure Item parameter which indicates that this SPDU is the beginning of the SSDU and not the end of the SSDU;

c) a User Data parameter which allows a limited amount of transparent user data to be transferred.

The second and any subsequent ABORT SPDUs in the sequence of ABORT SPDUs transmitting the SSDU contain:

d) an Enclosure Item parameter to indicate whether the SPDU is the middle or end of the SSDU;

e) a User Data parameter which allows a limited amount of transparent user data to be transferred.

7.9.2 *Sending the ABORT SPDU*

An S-U-ABORT request or the detection of a protocol error in any state of the SPM results in either a single ABORT SPDU or, if the SSDU provided in the S-U-ABORT request is to be segmented (see ¤ 6.3.5), a sequence of ABORT SPDUs, which shall not be interrupted.

If the SS-user data does not exceed 9 octets, the ABORT SPDU is sent on the transport expedited flow, if it is available to this session connection. If the transport expedited flow is not available to this session connection this SPDU is sent on the transport normal flow.

If the SS-user data exceeds 9 octets, the SPDU or sequence of SPDUs are sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (ABORT) SPDU is sent on the transport expedited flow simultaneously or earlier than the first or only ABORT SPDU. The SPM starts the timer, TIM, and waits for an ABORT ACCEPT SPDU or a T-DISCONNECT indication. Any other SPDUs are discarded. If the timer expires before receipt of an ABORT ACCEPT SPDU or a T-DISCONNECT indication, the SPM shall request transport disconnection with a T-DISCONNECT request. On receipt of a T-DISCONNECT indication, the timer is cancelled.

Note - The value of TIM is a local implementation dependent matter, related to quality of service.

7.9.3 *Receiving the ABORT SPDU*

A valid incoming ABORT SPDU, without an Enclosure Item parameter, or an Enclosure Item parameter indicating Òend of SSDUÓ results in an S-U-ABORT indication or an S-P-ABORT indication, depending on whether the abort is user generated or provider generated. The session connection ceases to exist. If the Transport Disconnect parameter in the received ABORT SPDU indicates that the transport connection is to be kept for reuse and this is acceptable to the receiving SPM, an ABORT ACCEPT SPDU is sent. If the Transport Disconnect parameter in the received ABORT SPDU indicates that the transport connection is not to be kept for reuse or reuse of the transport connection is not acceptable to the receiving SPM, the receiving SPM either:

a) releases the transport connection, or

b) sends an ABORT ACCEPT SPDU (see ¤ 7.10).

Receiving an ABORT SPDU sent in response to a CONNECT SPDU results in:

a) a T-DISCONNECT request, unless retention of the transport connection has been requested in the ABORT SPDU, in which case the ABORT SPDU is acknowledged with an ABORT ACCEPT SPDU (see ¤ 7.10); and

b) an S-P-ABORT indication or an S-U-ABORT indication to the SS-user.

7.10 *ABORT ACCEPT SPDU*

The ABORT ACCEPT SPDU is used to return a confirmation to the ABORT SPDU.

7.10.1 *Content of ABORT ACCEPT SPDU*

The ABORT ACCEPT SPDU contains no parameters.

7.10.2 *Sending the ABORT ACCEPT SPDU*

A valid incoming ABORT SPDU results in sending an ABORT ACCEPT SPDU, when the transport connection can be reused, i.e. when:

a) the transport expedited service is not available to this session connection, and

b) retention of the transport connection has been requested in the ABORT SPDU and it is acceptable to reuse the transport connection.

The SPM, as a local implementation decision, may send an ABORT ACCEPT SPDU in response to an ABORT SPDU, even if the transport connection is not to be kept.

This SPDU is sent on the transport expedited flow, if it is available to this session connection. Otherwise, this SPDU is sent on the transport normal flow. The session connection ceases to exist.

7.10.3 *Receiving the ABORT ACCEPT SPDU*

A valid incoming ABORT ACCEPT SPDU results in resetting the timer, TIM, and:

a) releasing the transport connection, if release of the transport connection was requested in the previously sent ABORT SPDU;

b) if retention of the transport connection was requested, the transport connection is now available for reuse by a new session connection, if this SPM was the initiator of the transport connection (see ¤ 6.1).

The session connection ceases to exist.

7.11 *DATA TRANSFER SPDU*

Normal data is transferred by use of the DATA TRANSFER SPDU. If the extended concatenation option was selected during connection establishment, certain concatenations of the DATA TRANSFER SPDU with other SPDUs is allowed (see ¤ 6.3.7).

The right to issue a DATA TRANSFER SPDU is subject to the token restrictions specified in Table 5/X.225.

7.11.1 *Content of DATA TRANSFER SPDU*

The DATA TRANSFER SPDU contains:

a) Enclosure Item parameter to indicate the beginning and end of SSDU when segmenting has been selected. When segmenting has been selected, the Enclosure Item parameter is always present and indicates whether the SPDU is the beginning, middle or end of the SSDU. When segmenting has not been selected, the Enclosure Item parameter is not present;

b) User Information Field to transfer transparent user data whose maximum size is unlimited when segmenting has not been selected and whose maximum size is limited by the maximum TSDU size when segmenting has been selected.

7.11.2 *Sending the DATA TRANSFER SPDU*

An S-DATA request results in a DATA TRANSFER SPDU unless segmenting has been selected, in which case an ordered sequence of DATA TRANSFER SPDUs will be sent with the appropriate value for the Enclosure Item parameter until the complete SSDU has been transferred.

The concatenation of any segment of an SSDU with any other SPDU will not result in a TSDU larger than the selected maximum TSDU size for that direction of transfer. However, there is no requirement that the resulting TSDU should be of the maximum size for that direction of transfer. All DATA TRANSFER SPDUs, except the last DATA TRANSFER SPDU in a sequence greater than one, must have user information. DATA TRANSFER SPDUs are sent on the transport normal flow.

Sending a segmented SSDU shall be interrupted when the SPM which is sending the segmented SSDU sends or receives one of:

RESYNCHRONIZE SPDU

EXCEPTION REPORT SPDU

EXCEPTION DATA SPDU

ACTIVITY INTERRUPT SPDU

ACTIVITY DISCARD SPDU

ABORT SPDU

PREPARE (RESYNCHRONIZE) SPDU

PREPARE (ABORT) SPDU

or receives a T-DISCONNECT indication. This will have a destructive effect on the entire SSDU. The SPM is not required to send the remainder of the ordered sequence of SPDUs which comprise the segmented SSDU (but may do so if it wishes).

7.11.3 *Receiving the DATA TRANSFER SPDU*

A valid incoming DATA TRANSFER SPDU results in an S-DATA indication unless segmenting has been selected. In this case, a valid incoming DATA TRANSFER SPDU, which indicates end of SSDU, results in an S-DATA indication to pass the entire SSDU to the SS-user.

Where segmenting has been selected and an incomplete segmented SSDU is outstanding, the receipt of:

RESYNCHRONIZE SPDU

EXCEPTION REPORT SPDU

EXCEPTION DATA SPDU

ACTIVITY INTERRUPT SPDU

ACTIVITY DISCARD SPDU

ABORT SPDU

PREPARE (RESYNCHRONIZE) SPDU

has a destructive effect on the entire SSDU (i.e. the SPDUs which have already been received are discarded, the remaining SPDUs will not be received).

Receiving a segmented SSDU shall be interrupted when the SPM which is receiving the segmented SSDU sends or receives one of:

RESYNCHRONIZE SPDU

EXCEPTION REPORT SPDU

EXCEPTION DATA SPDU

ACTIVITY INTERRUPT SPDU

ACTIVITY DISCARD SPDU

ABORT SPDU

PREPARE (RESYNCHRONIZE) SPDU

PREPARE (ABORT) SPDU

or receives a T-DISCONNECT indication. This will have a destructive effect on the entire SSDU (i.e. the SPDUs comprising part of the segmented SSDU which have already been received are discarded, and any SPDUs comprising part of the segmented SSDU which are received subsequently are discarded).

The receipt of any other SPDUs is a protocol error.

7.12 *EXPEDITED SPDU*

The EXPEDITED SPDU is used to transfer expedited SSDUs.

The right to send expedited data is not associated with any tokens. When this functional unit is selected, both SS-users may send expedited data. An EXPEDITED SSDU may be delivered to the receiving SS-user prior to other SSDUs previously transferred on the transport normal flow; it may not be delivered to the receiving SS-user later than any SSDUs transferred after it.

Expedited SSDUs are delivered to the receiving SS-user in the same sequence in which they were issued by the sending SS-user.

7.12.1 *Content of EXPEDITED SPDU*

The EXPEDITED SPDU contains a User Information Field which allows a limited amount of transparent user data to be transferred.

7.12.2 *Sending the EXPEDITED SPDU*

An S-EXPEDITED-DATA request results in an EXPEDITED SPDU being sent. This SPDU is sent on the transport expedited flow.

7.12.3 *Receiving the EXPEDITED SPDU*

A valid incoming EXPEDITED SPDU results in an S-EXPEDITED-DATA indication.

7.13 *TYPED DATA SPDU*

The TYPED DATA SPDU enables the SS-users to transmit transparent user data, irrespective of the availability or assignment of the data token. In all other respects, the same constraints apply as for normal data (see ¤ 7.11). The same rules for segmenting also apply.

7.13.1 *Content of TYPED DATA SPDU*

The TYPED DATA SPDU contains:

a) Enclosure Item parameter to indicate the beginning and end of SSDU when segmenting has been selected. When segmenting has been selected, the Enclosure Item parameter is always present and indicates whether the SPDU is the beginning, middle or end of the SSDU. When segmenting has not been selected, the Enclosure Item parameter is not present;

b) User Information Field to transfer transparent user data whose maximum size is unlimited when segmenting has not been selected and whose maximum size is limited by the maximum TSDU size when segmenting has been selected.

7.13.2 *Sending the TYPED DATA SPDU*

An S-TYPED-DATA request results in the transfer of a TYPED DATA SPDU unless segmenting has been selected, in which case an ordered sequence of TYPED DATA SPDUs will be sent with the appropriate value for the Enclosure Item parameter until the complete SSDU has been transferred. Each SPDU is mapped onto one TSDU and will not be larger than the selected maximum TSDU size for that direction of transfer. However, there is no requirement that the resulting TSDU should be of the maximum size for that direction of transfer. All TYPED DATA SPDUs, except the last TYPED DATA SPDU in a sequence greater than one, must have user information. TYPED DATA SPDUs are sent on the transport normal flow. When segmenting has been selected the rules governing the sending or receipt of SPDUs other than TYPED DATA SPDUs, while sending a segmented TYPED DATA SSDU are the same as for the DATA TRANSFER SPDU (see ¤ 7.11.2).

7.13.3 *Receiving the TYPED DATA SPDU*

A valid incoming TYPED DATA SPDU results in an S-TYPED-DATA indication, unless segmenting has been selected. In this case, a valid incoming TYPED DATA SPDU which indicates end of SSDU results in an S-TYPED-DATA indication to pass the entire SSDU to the SS-user. The current state of the SPM is not changed.

When segmenting has been selected the rules governing the sending or receipt of SPDUs other than TYPED DATA SPDUs while receiving a segmented TYPED DATA SSDU are the same as for the DATA TRANSFER SPDU (see ¤ 7.11.3).

7.14 *CAPABILITY DATA SPDU*

The CAPABILITY DATA SPDU is used to transfer a limited amount of transparent user data outside activities (i.e. when the activity management functional unit has been selected and Vact is false). The right to send this SPDU is restricted to the side having the right to start the next activity (i.e. the activity management functional unit has been selected and Vact is false and subject to the token restrictions specified in Table 5/X.225).

7.14.1 *Content of CAPABILITY DATA SPDU*

The CAPABILITY DATA SPDU contains a User Data parameter which allows a limited amount of transparent user data to be transferred.

7.14.2 *Sending the CAPABILITY DATA SPDU*

An S-CAPABILITY-DATA request results in a CAPABILITY DATA SPDU being sent. This SPDU is sent on the transport normal flow. The SS-user is not permitted to issue a further S-CAPABILITY-DATA request until this CAPABILITY DATA SPDU is acknowledged.

7.14.3 *Receiving the CAPABILITY DATA SPDU*

A valid incoming CAPABILITY DATA SPDU results in an S-CAPABILITY-DATA indication to the SS-user.

7.15 *CAPABILITY DATA ACK SPDU*

The CAPABILITY DATA ACK SPDU is used to complete the capability data exchange.

7.15.1 *Content of CAPABILITY DATA ACK SPDU*

The CAPABILITY DATA ACK SPDU contains a User Data parameter which allows a limited amount of transparent user data to be transferred.

7.15.2 *Sending the CAPABILITY DATA ACK SPDU*

The SS-user generates an S-CAPABILITY-DATA response which results in a CAPABILITY DATA ACK SPDU. This SPDU is sent on the transport normal flow.

7.15.3 *Receiving the CAPABILITY DATA ACK SPDU*

A valid incoming CAPABILITY DATA ACK SPDU results in an S-CAPABILITY-DATA confirm. This allows the SS-user to issue a further S-CAPABILITY-DATA request.

7.16 *GIVE TOKENS SPDU*

The GIVE TOKENS SPDU is used:

a) to introduce a concatenated sequence of SPDUs; and/or

b) to cause assignment of currently owned tokens to be changed.

If the GIVE TOKENS SPDU does not contain any parameter fields, it is used to indicate concatenation without assignment of tokens and, in this case, the sending and receiving procedures do not apply.

7.16.1 *Content of GIVE TOKENS SPDU*

The GIVE TOKENS SPDU contains:

a) a Token Item parameter which indicates which tokens are being transferred from the sending SS-user to the receiving SS-user.

b) a User Data parameter which allows a limited amount of transparent user data to be transferred. This parameter shall not be present if protocol version 1 is selected.

7.16.2 *Sending the GIVE TOKENS SPDU*

An S-TOKEN-GIVE request results in a GIVE TOKENS SPDU. This SPDU is sent on the transport normal flow.

7.16.3 *Receiving the GIVE TOKENS SPDU*

A valid incoming GIVE TOKENS SPDU results in an S-TOKEN-GIVE indication.

7.17 *PLEASE TOKENS SPDU*

The PLEASE TOKENS SPDU is used:

a) to introduce a concatenated sequence of SPDUs; and/or

b) to request that the token assignments be changed to permit the requestor to be authorized to perform a function associated with the requested tokens.

If the PLEASE TOKENS SPDU does not contain any parameter fields, it is used to indicate concatenation without requesting tokens and, in this case, the sending and receiving procedures do not apply.

7.17.1 *Content of PLEASE TOKENS SPDU*

The PLEASE TOKENS SPDU contains:

a) Token Item parameter which indicates which tokens are being requested by the sending SS-user;

b) User Data parameter which allows a limited amount of transparent user data to be transferred.

7.17.2 *Sending the PLEASE TOKENS SPDU*

An S-TOKEN-PLEASE request results in a PLEASE TOKENS SPDU. This SPDU is sent on the transport normal flow.

7.17.3 *Receiving the PLEASE TOKENS SPDU*

A valid incoming PLEASE TOKENS SPDU results in an S-TOKEN-PLEASE indication. Receiving a PLEASE TOKENS SPDU for tokens which are not currently assigned to the accepting SS-user is not a protocol error.

7.18 *GIVE TOKENS CONFIRM SPDU*

The GIVE TOKENS CONFIRM SPDU is used as a result of an S-CONTROL-GIVE request to cause assignment of all of the currently assigned tokens to be changed when Vact is false. Receipt of the GIVE TOKENS CONFIRM SPDU by the receiving SPM is acknowledged by the GIVE TOKENS ACK SPDU.

7.18.1 *Content of GIVE TOKENS CONFIRM SPDU*

The GIVE TOKENS CONFIRM SPDU contains a User Data parameter which allows a limited amount of transparent user data to be transferred. This parameter shall not be present if protocol version 1 is selected.

7.18.2 *Sending the GIVE TOKENS CONFIRM SPDU*

An S-CONTROL-GIVE request when Vact is false results in a GIVE TOKENS CONFIRM SPDU. The SPM then waits for a GIVE TOKENS ACK SPDU before permitting further SPDUs, associated with the available tokens, to be sent or received. SPDUs not associated with tokens (e.g. TYPED DATA SPDU) may be sent or received as normal. This SPDU is sent on the transport normal flow.

7.18.3 *Receiving the GIVE TOKENS CONFIRM SPDU*

A valid incoming GIVE TOKENS CONFIRM SPDU results in an S-CONTROL-GIVE indication, followed by a GIVE TOKENS ACK SPDU.

7.19 *GIVE TOKENS ACK SPDU*

The GIVE TOKENS ACK SPDU is used to acknowledge receipt of a GIVE TOKENS CONFIRM SPDU. The GIVE TOKENS ACK SPDU can only be sent when Vact is false.

7.19.1 *Content of GIVE TOKENS ACK SPDU*

The GIVE TOKENS ACK SPDU contains no parameters.

7.19.2 *Sending the GIVE TOKENS ACK SPDU*

A valid incoming GIVE TOKENS CONFIRM SPDU results in a GIVE TOKENS ACK SPDU (see also ¤ 7.18.3). The SPM may now transmit SPDUs associated with the token controlled functional units. This SPDU is sent on the transport normal flow.

7.19.3 *Receiving the GIVE TOKENS ACK SPDU*

After receiving a valid incoming GIVE TOKENS ACK SPDU, the SPM is now prepared to receive any SPDUs associated with the token controlled functional units.

7.20 *MINOR SYNC POINT SPDU*

The MINOR SYNC POINT SPDU is used to define a minor synchronization point. A confirmation may be returned by the receiver but is not required by the SPM (see ¤ 7.21). All acknowledgement rules are defined by the SS-users. In particular, whether confirmation is requested or not is transparent to the SPM. The right to issue a MINOR SYNC POINT SPDU is subject to the token restrictions specified in Table 5/X.225.

7.20.1 *Content of MINOR SYNC POINT SPDU*

The MINOR SYNC POINT SPDU contains:

a) Sync Type Item parameter which is used to indicate if an explicit confirmation is required (see ¤ 7.21);

b) Serial Number parameter which indicates the serial number of this minor synchronization point, and is set by the SPM to the current value of V(M);

c) User Data parameter which allows a limited amount of transparent user data to be transferred.

7.20.2 *Sending the MINOR SYNC POINT SPDU*

An S-SYNC-MINOR request results in a MINOR SYNC POINT SPDU. This SPDU is sent on the transport normal flow. If Vsc is true, V(A) is set equal to V(M) and Vsc is set false. V(M) is incremented by one.

7.20.3 *Receiving the MINOR SYNC POINT SPDU*

A valid incoming MINOR SYNC POINT SPDU results in an S-SYNC-MINOR indication. If Vsc is false, V(A) is set equal to V(M) and Vsc is set true. V(M) is incremented by one.

7.21 *MINOR SYNC ACK SPDU*

The MINOR SYNC ACK SPDU is used to return a confirmation to minor synchronization points. The SPM sends a MINOR SYNC ACK SPDU for each S-SYNC-MINOR response.

7.21.1 *Content of MINOR SYNC ACK SPDU*

The MINOR SYNC ACK SPDU contains:

a) Serial Number parameter, provided by the SS-user which indicates the serial number of the minor synchronization point which is being confirmed;

b) User Data parameter which allows a limited amount of transparent user data to be transferred.

7.21.2 *Sending the MINOR SYNC ACK SPDU*

An S-SYNC-MINOR response (with Vsc true and serial number greater than or equal to V(A) and less than V(M)) results in sending a MINOR SYNC ACK SPDU. This SPDU is sent on the transport normal flow. The SPM sets V(A) equal to the serial number plus one.

7.21.3 *Receiving the MINOR SYNC ACK SPDU*

A valid incoming MINOR SYNC ACK SPDU (with Vsc false and received serial number greater than or equal to V(A) and less than V(M)) results in an S-SYNC-MINOR confirm. The SPM sets V(A) equal to the received serial number plus one.

7.22 *MAJOR SYNC POINT SPDU*

The MAJOR SYNC POINT SPDU is used to define a major synchronization point. A confirmation has to be received before more data can be sent on the normal and expedited flows. The right to issue a MAJOR SYNC POINT SPDU is subject to the token restrictions specified in Table 5/X.225.

7.22.1 *Content of MAJOR SYNC POINT SPDU*

The MAJOR SYNC POINT SPDU contains:

a) Sync Type Item parameter which is only present when indicating that this major synchronization point is not the end of the current activity;

b) Serial Number parameter which indicates the serial number of this major synchronization point, and is set by the SPM to the current value of V(M);

c) User Data parameter which allows a limited amount of transparent user data to be transferred.

7.22.2 *Sending the MAJOR SYNC POINT SPDU*

An S-SYNC-MAJOR request results in a MAJOR SYNC POINT SPDU. This SPDU is sent on the transport normal flow. If Vsc is true, V(A) is set equal to V(M) and Vsc is set false. V(M) is incremented by one. If the activity management functional unit has been selected, Vnextact is set true. If the transport expedited flow is available to this session connection, the SPM waits for a PREPARE (MAJOR SYNC ACK) SPDU, followed by a MAJOR SYNC ACK SPDU. Otherwise, just a MAJOR SYNC ACK is expected. Any other SPDUs received prior to the MAJOR SYNC ACK SPDU will result in the appropriate service indications being given to the SS-user.

7.22.3 *Receiving the MAJOR SYNC POINT SPDU*

A valid incoming MAJOR SYNC POINT SPDU (with received serial number equal to V(M)) results in an S-SYNC-MAJOR indication. If Vsc is false, V(A) is set equal to V(M). V(M) is incremented by one. If the activity management functional unit has been selected, Vnextact is set true.

7.23 *MAJOR SYNC ACK SPDU*

The MAJOR SYNC ACK SPDU is used to return a confirmation to a major synchronization point.

7.23.1 *Content of MAJOR SYNC ACK SPDU*

The MAJOR SYNC ACK SPDU contains:

a) Serial Number parameter which indicates the serial number of the major synchronization point which is being confirmed (which is equal to V(M) minus one);

b) User Data parameter which allows a limited amount of transparent user data to be transferred.

7.23.2 *Sending the MAJOR SYNC ACK SPDU*

An S-SYNC-MAJOR response results in a MAJOR SYNC ACK SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (MAJOR SYNC ACK) SPDU is sent simultaneously, or earlier, on the transport expedited flow. V(A) and V(R) are set equal to V(M). If the activity management functional unit has been selected, Vact is set to Vnextact.

7.23.3 *Receiving the MAJOR SYNC ACK SPDU*

A valid incoming MAJOR SYNC ACK SPDU (with received serial number equal to V(M) minus one) results in an S-SYNC-MAJOR confirm.

If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (MAJOR SYNC ACK) SPDU on the transport expedited flow, followed by

b) MAJOR SYNC ACK SPDU on the transport normal flow.

V(A) and V(R) are set equal to V(M). If the activity management functional unit has been selected, Vact is set to Vnextact.

7.24 *RESYNCHRONIZE SPDU*

The RESYNCHRONIZE SPDU is used to provide the SS-users with a selective means to resynchronize the exchange of data to a synchronization point and to reposition the tokens to an agreed side. Use of this procedure may result in loss of data.

This SPDU can also be used to ÒpurgeÓ the session connection, since that is a particular case of resynchronization. The following options are provided:

a) abandon;

b) set;

c) restart.

Since the resynchronization protocol provides a repositioning of the tokens a particular use of it is the destructive way to get the tokens.

When used with activity management, the RESYNCHRONIZE SPDU can only be sent when Vact is true.

7.24.1 *Content of RESYNCHRONIZE SPDU*

The RESYNCHRONIZE SPDU contains:

a) Token Setting Item which indicates the requestor's proposed token positions for all available tokens;

b) Resync Type Item parameter which indicates the resynchronize option (abandon, set or restart);

c) Serial Number parameter which indicates the serial number to which resyncrhonization is being requested. The serial number is supplied by the SS-user if the resynchronize option is set or restart. If the resynchronize option is abandon, the serial number is set to the value of V(M) of the sending SPM;

d) User Data parameter which allows a limited amount of transparent user data to be transferred.

7.24.2 *Sending the RESYNCHRONIZE SPDU*

An S-RESYNCHRONIZE request (with serial number greater than or equal to V(R) and less than or equal to V(M) if the resynchronize option is restart) results in a RESYNCHRONIZE SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (RESYNCHRONIZE) SPDU is sent simultaneously, or earlier, on the transport expedited flow.

The SPM goes into a state where all the incoming SPDUs are discarded except PREPARE (RESYNCHRONIZE), RESYNCHRONIZE, PREPARE (RESYNCHRONIZE ACK), RESYNCHRONIZE ACK, ACTIVITY DISCARD, ACTIVITY INTERRUPT and ABORT SPDUs.

If a RESYNCHRONIZE, PREPARE (RESYNCHRONIZE), ACTIVITY INTERRUPT or ACTIVITY DISCARD SPDU is received when the SPM is in this state, a resynchronization contention situation has occurred and is dealt with as specified in ¤ 7.24.4.

7.24.3 *Receiving the RESYNCHRONIZE SPDU*

Except when a resynchronization contention situation has occurred, a valid incoming RESYNCHRONIZE SPDU (with received serial number greater than or equal to V(R) if the resynchronize option is restart) results in an S-RESYNCHRONIZE indication. If the resynchronize option is abandon, this indication contains a serial number which is equal to V(M) or the received serial number, whichever is higher; V(M) is set to this value. If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (RESYNCHRONIZE) SPDU on the transport expedited flow, followed by

b) RESYNCHRONIZE SPDU on the transport normal data flow.

When the PREPARE (RESYNCHRONIZE) SPDU is received, all subsequently received SPDUs, except ABORT SPDU, are discarded until the RESYNCHRONIZE SPDU is received on the transport normal flow.

The SPM now waits for an S-RESYNCHRONIZE response.

If a resynchronization contention situation has occurred, only the contention loser (see ¤ 7.24.4) passes an S-RESYNCRONIZE indication to the SS-user.

7.24.4 *Resynchronization contention*

The contention between two RESYNCHRONIZE, ACTIVITY INTERRUPT, or ACTIVITY DISCARD SPDUs is resolved according to Table 9/X.225. The table defines the contention winner whose SPDU is taken into account; the other SPDU is discarded.

If an incoming RESYNCHRONIZE SPDU is not acceptable, the receiving SS-user may issue another if it prevails over the original proposal according to the decision rules.

7.25 *RESYNCHRONIZE ACK SPDU*

The RESYNCHRONIZE ACK SPDU is used to notify the sender of a RESYNCHRONIZE SPDU of the completion of resynchronization.

TABLE 9/X.225

Contention winner

Incoming SPDU from SPM B / Outgoing SPDU from SPM A	RESYNC. (abandon)	RESYNC. (set)	RESYNC. (restart)	ACTIVITY INTERRUPT	ACTIVITY DISCARD
RESYNC. (abandon)	Initiator	SPM A	SPM A	SPM B	SPM B
RESYNC. (set)	SPM B	Initiator	SPM A	SPM B	SPM B
RESYNC. (restart)	SPM B	SPM B	SPM with lower serial number or if equal then initiator	SPM B	SPM B
ACTIVITY INTERRUPT	SPM A	SPM A	SPM A	See note	See note
ACTIVITY DISCARD	SPM A	SPM A	SPM A	See note	See note

Note - Collision is not possible in these cases because only the owner of the major/activity token is permitted to send ACTIVITY DISCARD SPDU or ACTIVITY INTERRUPT SPDU.

7.25.1 *Content of RESYNCHRONIZE ACK SPDU*

The RESYNCHRONIZE ACK SPDU contains:

a) Token Setting Item parameter which indicates the selected token positions.

b) Serial Number parameter which indicates the first serial number to be used in the resynchronized flow. This parameter is set according to the Resync Type Item parameter in the received RESYNCHRONIZE SPDU:

1) for the restart option, to the serial number in the received RESYNCHRONIZE SPDU;

2) for the set option, to the serial number in the S-RESYNCHRONIZE response;

3) for the abandon option, to V(M).

c) User Data parameter which allows a limited amount of transparent user data to be transferred.

7.25.2 *Sending the RESYNCHRONIZE ACK SPDU*

An S-RESYNCHRONIZE response results in a RESYNCHRONIZE ACK SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (RESYNCHRONIZE ACK) SPDU is sent simultaneously, or earlier, on the transport expedited flow.

The tokens are set to the values proposed by the requestor. If the requestor has indicated Òaccepting SS-user's choiceÓ for a token, then the acceptor's proposed value for that token is used. The selected token settings are returned in the Token Setting Item of the RESYNCHRONIZE ACK SPDU.

V(A) and V(M) are set to the serial number contained in the RESYNCHRONIZE ACK SPDU.

V(R) is unchanged if the Resync Type Item parameter in the received RESYNCHRONIZE SPDU indicated the restart option. Otherwise, V(R) is set to zero.

7.25.3 *Receiving the RESYNCHRONIZE ACK SPDU*

A valid incoming RESYNCHRONIZE ACK SPDU results in an S-RESYNCHRONIZE confirm. If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (RESYNCHRONIZE ACK) SPDU on the transport expedited flow, followed by

b) RESYNCHRONIZE ACK on the transport normal flow.

The tokens are set to the positions specified in the RESYNCHRONIZE ACK SPDU.

V(A) and V(M) are set to the serial number contained in the RESYNCHRONIZE ACK SPDU.

V(R) is unchanged if the Resync Type Item parameter in the transmitted RESYNCHRONIZE SPDU indicated the restart option. Otherwise, V(R) is set to zero.

7.26 *PREPARE SPDU*

The PREPARE SPDU is only used when the transport expedited flow is available to this session connection. It notifies the imminent arrival of certain SPDUs and indicates to the receiving SPM that SPDUs received on the transport normal flow may be discarded under certain circumstances.

7.26.1 *Content of PREPARE SPDU*

The PREPARE SPDU contains a Prepare Type parameter which indicates which SPDU should be expected on the transport normal flow.

7.26.2 *Sending the PREPARE SPDU*

The PREPARE SPDU is sent before the associated SPDUs specified in Table 10/X.225 when the transport expedited flow is available to this session connection. Table 10/X.225 also specifies the value of the Prepare Type parameter.

TABLE 10/X.225

SPDUs associated with the PREPARE SPDU

Associated SPDU	Prepare Type
RESYNCHRONIZE SPDU	RESYNCHRONIZE
RESYNCHRONIZE ACK SPDU	RESYNCHRONIZE ACK
MAJOR SYNC ACK SPDU	MAJOR SYNC ACK
ACTIVITY INTERRUPT SPDU	RESYNCHRONIZE
ACTIVITY INTERRUPT ACK SPDU	RESYNCHRONIZE ACK
ACTIVITY DISCARD SPDU	RESYNCHRONIZE
ACTIVITY DISCARD ACK SPDU	RESYNCHRONIZE ACK
ACTIVITY END ACK SPDU	MAJOR SYNC ACK
ABORT SPDU	ABORT

The PREPARE SPDU is sent on the transport expedited flow (its associated SPDU being sent on the transport normal flow). The SPM goes to a state which is determined by the initial request.

7.26.3 *Receiving the PREPARE SPDU*

A valid incoming PREPARE SPDU results in the SPM entering a state where it is waiting for the associated SPDU on the transport normal flow. If the Prepare Type parameter indicates MAJOR SYNC ACK, any SPDUs received on the transport normal flow are processed normally. Otherwise, SPDUs received on the transport normal flow before the indicated SPDU are discarded. In any case, if an EXPEDITED DATA SPDU is received after a PREPARE SPDU, but before the associated SPDU on the transport normal flow, the S-EXPEDITED-DATA indication is not passed to the SS-user until the associated SPDU has been received and processed.

7.27 *EXCEPTION REPORT SPDU*

The EXCEPTION REPORT SPDU is used to report that a protocol error has been detected within the SPM. It can only be sent in the data transfer phase and subject to the token restrictions specified in Table 5/X.225.

7.27.1 *Content of EXCEPTION REPORT SPDU*

The EXCEPTION REPORT SPDU contains a Reflect Parameter Values parameter which is used to indicate a field of arbitrary length, which contains the bit pattern of the SPDU received with a protocol error, up to and including the detected error.

7.27.2 *Sending the EXCEPTION REPORT SPDU*

On detection of a protocol error, for example an SPDU received at an unexpected time, or an invalid SPDU, the SPM may generate an EXCEPTION REPORT SPDU. This SPDU is sent on the transport normal flow. At the same time an S-P-EXCEPTION-REPORT indication will be generated. The SPM enters an error state which is only left when any of the following SPDUs, or their associated local service requests, are received:

ACTIVITY DISCARD;

ACTIVITY INTERRUPT;

RESYNCHRONIZE;

ABORT;

GIVE TOKENS (with the data token);

PREPARE (RESYNCHRONIZE).

Any other SPDUs received will be discarded. However, V(A) and V(M) will be updated appropriately if valid MINOR SYNC POINT SPDUs or MAJOR SYNC POINT SPDUs are received.

7.27.3 *Receiving the EXCEPTION REPORT SPDU*

When an incoming EXCEPTION REPORT SPDU is received, an S-P-EXCEPTION-REPORT indication is given and the SPM enters an error state.

The SPM leaves the error state when any of the following SPDUs, or their associated local service requests, are received:

ACTIVITY DISCARD;

ACTIVITY INTERRUPT;

RESYNCHRONIZE;

ABORT;

GIVE TOKENS (with the data token);

PREPARE (RESYNCHRONIZE).

Note - This action is dependent on the receipt of the EXCEPTION REPORT SPDU, not on examination of its parameter value. This enables the procedure to be followed in cases where the implementation cannot deal with an SPDU length greater than the minimum specified in ¤ 8.3.27.3.

7.28 *EXCEPTION DATA SPDU*

The EXCEPTION DATA SPDU is used to put the SPM into an error state.

It can only be sent subject to the token restrictions specified in Table 5/X.225 and:

a) when the activity management functional unit has been selected and an activity is in progress, or

b) the activity management functional unit has not been selected.

7.28.1 *Content of EXCEPTION DATA SPDU*

The EXCEPTION DATA SPDU contains:

a) Reason Code parameter which indicates the reason for sending the EXCEPTION DATA SPDU;

b) User Data parameter which allows a limited amount of transparent user data to be transferred.

7.28.2 *Sending the EXCEPTION DATA SPDU*

An S-U-EXCEPTION-REPORT request results in the SPM sending an EXCEPTION DATA SPDU on the transport normal flow. The SPM enters an error state. The error state will be left when an S-U-ABORT request or a T-DISCONNECT indication is received or when any of the following SPDUs are received:

ACTIVITY DISCARD;

ACTIVITY INTERRUPT;

ABORT;

RESYNCHRONIZE;

GIVE TOKENS (with the data token);

PREPARE (RESYNCHRONIZE).

Any other SPDUs received will be discarded. However, V(A) and V(M) will be updated appropriately if MINOR SYNC POINT SPDUs or MAJOR SYNC POINT SPDUs are received.

7.28.3 *Receiving the EXCEPTION DATA SPDU*

A valid incoming EXCEPTION DATA SPDU results in an S-U-EXCEPTION-REPORT indication. The SPM enters an error state, unless the data token is not assigned to this SPM, in which case the SPM state is unchanged.

The SPM leaves the error state when any of the following service primitives are invoked by the SS-user:

S-U-ABORT request;

S-RESYNCHRONIZE request;

S-ACTIVITY-DISCARD request;

S-ACTIVITY-INTERRUPT request;

S-TOKEN-GIVE request (with the data token).

7.29 *ACTIVITY START SPDU*

The ACTIVITY START SPDU is used to notify the beginning of an activity. The right to issue an ACTIVITY START SPDU is subject to the token restrictions specified in Table 5/X.225.

7.29.1 *Content of ACTIVITY START SPDU*

The ACTIVITY START SPDU contains:

a) Activity Identifier parameter which allows the SS-users to identify the activity being started;

b) User Data parameter which allows a limited amount of transparent user data to be transferred.

7.29.2 *Sending the ACTIVITY START SPDU*

An S-ACTIVITY-START request (when Vact is false) results in an ACTIVITY START SPDU. V(A), V(M) and V(R) are set to one. Vact is set true. This SPDU is sent on the transport normal flow.

7.29.3 *Receiving the ACTIVITY START SPDU*

A valid incoming ACTIVITY START SPDU (when Vact is false) results in an S-ACTIVITY-START indication. V(A), V(M) and V(R) are set to one. Vact is set true.

7.30 *ACTIVITY RESUME SPDU*

The ACTIVITY RESUME SPDU is used to notify the resumption of a previously interrupted activity. The right to issue an ACTIVITY RESUME SPDU is subject to the token restrictions specified in Table 5/X.225.

7.30.1 *Content of ACTIVITY RESUME SPDU*

The ACTIVITY RESUME SPDU contains:

a) Linking Information parameter group which contains:

1) Called SS-user Reference parameter;

2) Calling SS-user Reference parameter;

3) Common Reference parameter;

4) Additional Reference Information parameter;

5) Old Activity Identifier which enables the SS-users to identify the old activity which is being resumed;

6) Serial Number parameter which indicates the first serial number to be used minus one.

b) New Activity Identifier parameter which allows the SS-users to assign a new identifier to the activity being resumed.

c) User Data parameter which allows a limited amount of transparent user data to be transferred.

7.30.2 *Sending the ACTIVITY RESUME SPDU*

An S-ACTIVITY-RESUME request (when Vact is false) results in an ACTIVITY RESUME SPDU. V(A) and V(M) are set to the serial number provided by the SS-user plus one. V(R) is set to one. Vact is set true. This SPDU is sent on the transport normal flow.

7.30.3 *Receiving the ACTIVITY RESUME SPDU*

A valid incoming ACTIVITY RESUME SPDU (when Vact is false) results in an S-ACTIVITY-RESUME indication. V(A) and V(M) are set to the received serial number plus one. V(R) is set to one. Vact is set true.

7.31 *ACTIVITY INTERRUPT SPDU*

The ACTIVITY INTERRUPT SPDU is used to notify the interruption of an ongoing activity. The right to issue an ACTIVITY INTERRUPT SPDU is subject to the token restrictions specified in Table 5/X.225. Use of this procedure may result in loss of data.

7.31.1 *Content of ACTIVITY INTERRUPT SPDU*

The ACTIVITY INTERRUPT SPDU contains:

a) a Reason Code parameter which indicates the reason for sending the ACTIVITY INTERRUPT SPDU;

b) a User Data parameter which allows a limited amount of transparent user data to be transferred. This parameter shall not be present if protocol version 1 is selected.

7.31.2 *Sending the ACTIVITY INTERRUPT SPDU*

An S-ACTIVITY-INTERRUPT request results in an ACTIVITY INTERRUPT SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (RESYNCHRONIZE) SPDU is sent simultaneously, or earlier, on the transport expedited flow. The SPM goes into a state where all incoming SPDUs are discarded except PREPARE (RESYNCHRONIZE ACK), ACTIVITY INTERUPT ACK and ABORT SPDUs.

7.31.3 *Receiving the ACTIVITY INTERRUPT SPDU*

A valid incoming ACTIVITY INTERRUPT SPDU results in an S-ACTIVITY-INTERRUPT indication. If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (RESYNCHRONIZE) SPDU (see ¤ 7.24) on the transport expedited flow, followed by

b) ACTIVITY INTERRUPT SPDU on the transport normal flow.

The SPM now waits for an S-ACTIVITY-INTERRUPT response.

7.32 *ACTIVITY INTERRUPT ACK SPDU*

The ACTIVITY INTERRUPT ACK SPDU is used to notify the sender of an ACTIVITY INTERRUPT SPDU of the completion of the interruption of the ongoing activity. On completion, all available tokens are assigned to the sender of the ACTIVITY INTERRUPT SPDU.

7.32.1 *Content of ACTIVITY INTERRUPT ACK SPDU*

The ACTIVITY INTERRUPT ACK SPDU contains a User Data parameter which allows a limited amount of transparent user data to be transfered. This parameter shall not be present if protocol version 1 is selected.

7.32.2 *Sending the ACTIVITY INTERRUPT ACK SPDU*

An S-ACTIVITY-INTERRUPT response results in an ACTIVITY INTERRUPT ACK SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (RESYNCHRONIZE ACK) SPDU is sent simultaneously, or earlier, on the transport expedited flow. Vact is set false when the ACTIVITY INTERRUPT ACK SPDU has been sent.

7.32.3 *Receiving the ACTIVITY INTERRUPT ACK SPDU*

A valid incoming ACTIVITY INTERRUPT ACK SPDU results in an S-ACTIVITY-INTERRUPT confirm. If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (RESYNCHRONIZE ACK) SPDU (see ¤ 7.25) on the transport expedited flow, followed by

b) ACTIVITY INTERRUPT ACK SPDU on the transport normal flow.

Vact is set false when the ACTIVITY INTERRUPT ACK SPDU has been received.

7.33 *ACTIVITY DISCARD SPDU*

The ACTIVITY DISCARD SPDU is used to notify the cancellation of an ongoing activity. The right to issue an ACTIVITY DISCARD SPDU is subject to the token restrictions specified in Table 5/X.225. Use of this procedure may result in the loss of data.

7.33.1 *Content of ACTIVITY DISCARD SPDU*

The ACTIVITY DISCARD SPDU contains:

a) a Reason Code parameter which indicates the reason for sending the ACTIVITY DISCARD SPDU,

b) a User Data parameter which allows a limited amount of transparent user data to be transferred. This parameter shall not be present if protocol version 1 is selected.

7.33.2 *Sending the ACTIVITY DISCARD SPDU*

An S-ACTIVITY-DISCARD request results in an ACTIVITY DISCARD SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE

(RESYNCHRONIZE) SPDU is sent simultaneously, or earlier, on the transport expedited flow. The SPM goes into a state where all the incoming SPDUs are discarded except PREPARE (RESYNCHRONIZE ACK), ACTIVITY DISCARD ACK and ABORT SPDUs.

7.33.3 *Receiving the ACTIVITY DISCARD SPDU*

A valid incoming ACTIVITY DISCARD SPDU results in an S-ACTIVITY-DISCARD indication. If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (RESYNCHRONIZE) SPDU (see ¤ 7.24) on the transport expedited flow, followed by

b) ACTIVITY DISCARD SPDU on the transport normal flow.

The SPM now waits for an S-ACTIVITY-DISCARD response.

7.34 *ACTIVITY DISCARD ACK SPDU*

The ACTIVITY DISCARD ACK SPDU is used to notify the sender of an ACTIVITY DISCARD SPDU of the completion of the cancellation of the ongoing activity. On completion, all available tokens are assigned to the sender of the ACTIVITY DISCARD SPDU.

7.34.1 *Content of ACTIVITY DISCARD ACK SPDU*

The ACTIVITY DISCARD ACK SPDU contains a User Data parameter which allows a limited amount of transparent user data to be transfered. This parameter shall not be present if protocol version 1 is selected.

7.34.2 *Sending the ACTIVITY DISCARD ACK SPDU*

An S-ACTIVITY-DISCARD response results in an ACTIVITY DISCARD ACK SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (RESYNCHRONIZE ACK) SPDU is sent simultaneously, or earlier, on the transport expedited flow. Vact is set false when the ACTIVITY DISCARD ACK SPDU has been sent.

7.34.3 *Receiving the ACTIVITY DISCARD ACK SPDU*

A valid incoming ACTIVITY DISCARD ACK SPDU results in an S-ACTIVITY-DISCARD confirm. If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (RESYNCHRONIZE ACK) SPDU (see ¤ 7.25) on the transport expedited flow, followed by

b) ACTIVITY DISCARD ACK SPDU on the transport normal flow.

Vact is set false when the ACTIVITY DISCARD ACK SPDU has been received.

7.35 *ACTIVITY END SPDU*

The ACTIVITY END SPDU is used to define an implicit major synchronization point at the end of an activity. A confirmation has to be received before more data can be sent on the normal and expedited flows. The right to issue an ACTIVITY END SPDU is subject to the token restrictions specified in Table 5/X.225.

An ACTIVITY END SPDU can only be validly sent when Vact is true.

7.35.1 *Content of ACTIVITY END SPDU*

The ACTIVITY END SPDU contains:

a) Serial Number parameter which indicates the serial number of this major synchronization point, and is set by the SPM to the current value of V(M);

b) User Data parameter which allows a limited amount of transparent user data to be transferred.

7.35.2 *Sending the ACTIVITY END SPDU*

An S-ACITIVITY-END request (when Vact is true) results in an ACTIVITY END SPDU. This SPDU is sent on the transport normal flow. If Vsc is true, V(A) is set equal to V(M) and Vsc is set false. V(M) is incremented by one. Vnextact is set false. If the transport expedited flow is available to this session connection, the SPM waits for a PREPARE (MAJOR SYNC ACK) SPDU, followed by an ACTIVITY END ACK SPDU. Otherwise, just an ACTIVITY END ACK SPDU is expected. Any other SPDUs received prior to the ACTIVITY END ACK SPDU will result in the appropriate service indications being given to the SS-user.

7.35.3 *Receiving the ACTIVITY END SPDU*

A valid incoming ACTIVITY END SPDU (when Vact is true, and the received serial number equals V(M)) results in an S-ACTIVITY-END indication. If Vsc is false, V(A) is set equal to V(M). V(M) is incremented by one. Vnextact is set false.

7.36 *ACTIVITY END ACK SPDU*

The ACTIVITY END ACK SPDU is used to return a confirmation to an ACTIVITY END SPDU.

7.36.1 *Content of ACTIVITY END ACK SPDU*

The ACTIVITY END ACK SPDU contains:

a) Serial Number parameter which indicates the serial number of the major synchronization point which is being confirmed (which is equal to V(M) minus one);

b) User Data parameter which allows a limited amount of transparent user data to be transferred.

7.36.2 *Sending the ACTIVITY END ACK SPDU*

An S-ACTIVITY-END response results in an ACTIVITY END ACK SPDU. This SPDU is sent on the transport normal flow. If the transport expedited flow is available to this session connection, a PREPARE (MAJOR SYNC ACK) SPDU is sent simultaneously, or earlier, on the transport expedited flow. V(A) and V(R) are set equal to V(M). Vact is set to Vnextact.

7.36.3 *Receiving the ACTIVITY END ACK SPDU*

A valid incoming ACTIVITY END ACK SPDU (with Vsc false and received serial number equal to V(M) minus one) results in an S-ACTIVITY-END confirm.

If the transport expedited flow is available to this session connection, two successive SPDUs will be received:

a) PREPARE (MAJOR SYNC ACK) SPDU on the transport expedited flow, followed by

b) ACTIVITY END ACK SPDU on the transport normal flow.

V(A) and V(R) are set equal to V(M). Vact is set to Vnextact.

7.37 *Additional elements of procedure for* segmented SSDUs

The following SPDUs may contain segments of the associated SSDU:

ACCEPT SPDU

REFUSE SPDU

FINISH SPDU

DISCONNECT SPDU

NOT FINISHED SPDU

CAPABILITY DATA SPDU

CAPABILITY DATA ACK SPDU

GIVE TOKENS SPDU

PLEASE TOKENS SPDU

GIVE TOKENS CONFIRM SPDU

MINOR SYNC POINT SPDU

MINOR SYNC ACK SPDU

MAJOR SYNC POINT SPDU

MAJOR SYNC ACK SPDU

RESYNCHRONIZE SPDU

RESYNCHRONIZE ACK SPDU

EXCEPTION DATA SPDU

ACTIVITY START SPDU

ACTIVITY RESUME SPDU

ACTIVITY INTERRUPT SPDU

ACTIVITY INTERRUPT ACK SPDU

ACTIVITY DISCARD SPDU

ACTIVITY DISCARD ACK SPDU

ACTIVITY END SPDU

ACTIVITY END ACK SPDU

These SPDUs are subject to the following additional procedures.

7.37.1 *Content of the SPDU*

Where an SSDU is segmented, the first SPDU contains all the parameters which would have been present in the SPDU if the SSDU had not been segmented, together with an Enclosure Item parameter, which indicates beginning of SSDU and not end of SSDU, and at least one octet of User Data. The last SPDU of the SSDU contains the Enclosure Item parameter which indicates not beginning of SSDU and end of SSDU and may or may not contain User Data. Intermediate SPDUs of the SSDU, if present, contain the Enclosure Item parameter which indicates not beginning of SSDU and not end of SSDU and at least one octet of User Data.

7.37.2 *Sending the SPDU*

The sending procedures for SPDUs where these additional elements of procedure apply are extended in the following way:

a) where the SPM sends an SPDU, it shall send an ordered sequence of SPDUs which together comprise the complete SSDU;

b) sending this ordered sequence of SPDUs shall be interrupted when the SPM sends an ABORT SPDU or a PREPARE (ABORT) SPDU (for example, as a result of an S-U-ABORT request or a detected protocol error) or when the SPM receives an ABORT SPDU, a PREPARE (ABORT) SPDU or a T-DISCONNECT indication. In this case, the SPM shall stop sending the ordered sequence of SPDUs and shall take the appropriate defined actions;

Note - the ordered sequence of SPDUs sent so far will not comprise a complete SSDU. The Enclosure Item parameter will not have been sent with a value which indicates end of SSDU.

c) as a local matter, sending this ordered sequence of SPDUs may be interrupted when the SPM receives an SPDU which will cause the ordered sequence of SPDUs to be discarded by the remote SPM. In this situation, the SPM which is sending the ordered sequence of SPDUs is not required to send the remainder of the ordered sequence.

Note - This situation will occur if the received destructive SPDU was sent by the remote SPM before the first SPDU of the ordered sequence of SPDUs was received by the remote SPM, or if the remote SPM took the local implementation decision indicated in ¤ 7.37.3 d).

7.37.3 *Receiving the SPDU*

The receiving procedures for SPDUs where these additional elements of procedure apply are extended in the following way:

a) where the SPM receives an SPDU, it shall receive an ordered sequence of SPDUs which together comprise the complete SSDU;

b) receiving this ordered sequence of SPDUs shall be interrupted when the SPM receives an ABORT SPDU, a PREPARE (ABORT) SPDU or a T-DISCONNECT indication. This shall have a destructive effect on the entire segmented SSDU (i.e. the SPDUs which have already been received are discarded). The SPM shall take the appropriate defined actions;

c) the SPM may send an ABORT SPDU or a PREPARE (ABORT) SPDU (for example, as a result of an S-U-ABORT request or a detected protocol error) while receiving an ordered sequence of SPDUs. This shall have a destructive effect on the entire segmented SSDU being received (i.e. SPDUs comprising part of the segmented SSDU which have already been received are discarded and any SPDUs comprising part of the segmented SSDU which are received subsequently are discarded). The SPM shall take the appropriate defined actions;

d) as a local matter, while receiving this ordered sequence of SPDUs, the SPM may send any other appropriate SPDU which will have a destructive effect on the entire SSDU being received (i.e. SPDUs comprising part of the segmented SSDU which have already been received will be discarded and any SPDUs comprising part of the segmented SSDU which are received subsequently will be discarded).

Note - The conditions and effect stated here are as though the segmented SSDU had been sent in a single SPDU and the SPDU causing the destructive effect had been sent before that SPDU is received.

8 Structure and encoding of SPDUs

8.1 *TSDU structure*

Each TSDU consists of one or more SPDUs complying with the requirements for concatenation (see ¤ 6.3.7).

Each SPDU within a TSDU consists of one or more octets that are numbered sequentially starting from 1.

Each octet within an SPDU consists of eight bits numbered 8 to 1 where 1 is the low ordered bit.

The sequence of octets within an SPDU and the sequence of bits within an octet are defined for each SPDU in ¤ 8.3, with the additional convention that where the text refers to bits within a two-octet field and the bits are numbered 16 to 1, then 1 is the low order bit and the octet containing bits 16 to 9 precedes the octet containing bits 8 to 1 in the SPDU.

Within each TSDU:

a) the sequential ordering of SPDUs is maintained;

b) the ordering of the octets is maintained in the same order as in the SPDU;

c) the ordering of bits within each TSDU is maintained in the same order as in the SPDU (i.e. the low order bit is mapped onto the low order bit and the high order bit is mapped onto the high order bit).

Note 1 - The TSDU structure is illustrated in Figure 4/X.225. The integrity of this structure is maintained over a transport connection. This Recommendation does not define the way in which the TSDU is transmitted.

Note 2 - When the structure of an SPDU is illustrated in this Recommendation, the following convention is used:

a) octets are shown with the lowest numbered octet to the left, higher numbered octets being shown further to the right;

b) within an octet, bits are shown with bit 8 to the left and bit 1 to the right.

Figure 4/X.225, p.

8.2 *SPDU structure*

This subsection specifies the general structure of SPDUs in terms of their constituent fields. This structure is illustrated in Figure 5/X.225.

Codings and structural requirements specific to particular SPDUs are specified in ¤ 8.3.

Examples of valid SPDU structure are illustrated in Figure 6/X.225.

8.2.1 *SPDUs*

SPDUs shall contain, in the following order:

a) *the SI field* that identifies the type of SPDU (see Note);

b) *the LI field* that indicates the length of the associated parameter field defined in ¤ 8.2.1 c);

c) *the parameter field* which, if present, consists of the PGI units (see ¤ 8.2.2) and/or PI units (see ¤ 8.2.3) defined for the SPDU;

d) *the user information field*, if defined for the SPDU and if present.

Note - The SI field encompasses both the CI field and the RI field defined in Recommendation T.62. The protocol specified in this Recommendation does not require a distinction to be made between these two fields.

8.2.2 *PGI units*

PGI units shall contain, in the following order:

a) *the PGI field* that identifies the parameter group;

b) *the LI field* that indicates the length of the associated parameter field defined in ¤ 8.2.2 c);

c) *the parameter field* which, if present, consists of either:

1) a single parameter value (see Note); or

2) one or more PI units (see ¤ 8.2.3).

Note - A PGI unit with one parameter is structurally equivalent to a PI unit, but the distinction has been retained in order to maintain compatibility with Recommendation T.62.

Figure 5/X.225, p.

Figure 6/X.225, p.

8.2.3 *PI units*

PI units shall contain, in the following order:

a) the *PI field* that identifies the parameter;

b) the *LI field* that indicates the length of the associated parameter field defined in ¤ 8.2.3 c);

c) the *parameter field* which, if present, consists of the parameter value.

8.2.4 *Identifier fields*

The SI field shall comprise one octet. The value of the SI field, specified as a decimal number in ¤ 8.3, shall be encoded as a binary number.

The PGI and PI fields shall each comprise one octet and shall contain a PGI or PI code respectively. The PGI and PI codes are expressed as decimal numbers in the tables in ¤ 8.3 and shall be encoded as a binary number.

8.2.5 *Length indicator field*

The value of the LI field is expressed as a binary number representing the length, in octets, of the associated parameter field (see Note). A value of zero indicates that the associated parameter field is absent.

LI fields indicating lengths within the range 0-254 shall comprise one octet.

LI fields indicating lengths within the range 255-65 535 shall comprise three octets. The first octet shall be coded 1111 1111 and the second and third octets shall contain the length of the associated parameter field with the high order bits in the first of these two octets.

Note - The value of the LI field does not include either itself or any subsequent user information.

8.2.6 *Parameter fields*

PGI units and PI units defined as mandatory in the tables in ¤ 8.3 shall contain a parameter field of one or more octets.

Any PGI unit or PI unit defined as non-mandatory in the tables in ¤ 8.3 may be omitted if it is not required for conveying information (i.e. a parameter value). If a PGI unit or a PI unit contains an LI field with the value zero, the associated parameter field is absent (see Note) and the value of the parameter field shall be considered as its default value.

Note - It is recommended that if a non-mandatory parameter is absent, the associated PGI (or PI) and LI fields should not be included in the SPDU.

PGI units and PI units within the same nesting level shall be ordered in increasing value of their PGI and PI codes.

PGI or PI units containing:

a) a PGI or PI code listed in Annex C,

b) a PGI or PI code not listed in ¤ 8.3 or in Annex C,

are defined as valid.

Note - See ¤ A.4.3 for actions to be taken by the SPM on receipt of SPDUs containing these PGI or PI units.

8.2.7 *Parameter values*

Bits within a parameter field which are indicated as reserved shall have those bits set to zero in the SPDU.

Note - See ¤ A.4.3 for actions to be taken by the SPM on receipt of SPDUs containing such bits.

8.2.8 *User information fields*

Segments of a segmented SSDU shall be contained in the User Information Fields of SPDUs such that the order of the segments is maintained. An SSDU which is not segmented shall be contained in the User Information Field of a single SPDU. The order of the octets and the order of the bits in the SSDU shall be maintained in the SPDUs.

8.3 *SPDU identifiers and associated parameter fields*

The SPDUs specified in the remainder of this subsection do not, with certain exceptions, consider the case where an SSDU is segmented. When protocol version 2 is selected, most SSDUs may be segmented. (The circumstances under which an SSDU may be segmented are specified in ¤ 6.3.5.) The additional encoding requirements when an SSDU is segmented are specified in ¤ 8.4.

8.3.1 *CONNECT (CN) SPDU*

8.3.1.1 The SI field shall contain the value 13.

8.3.1.2 The parameter fields shall be as specified in Table 11/X.225.

8.3.1.3 The Calling SS-user Reference PV field shall be as defined by the calling SS-user.

8.3.1.4 The Common Reference PV field shall be as defined by the calling SS-user.

8.3.1.5 The Additional Reference Information PV field shall be as defined by the calling SS-user.

8.3.1.6 If the Connect/Accept Item is absent, the default values defined for the enclosed PI units shall apply.

8.3.1.7 The Protocol Options PV field shall indicate whether or not the initiator is able to receive extended concatenated SPDUs (see ¤ 6.3.7). The encoding for this field shall be:

a) bit 1 = 1 : able to receive extended concatenated SPDUs;

b) bit 1 = 0 : not able to receive extended concatenated SPDUs.

Bits 2-8 are reserved.

If the Protocol Options PI unit or PV field is absent, SPDUs with extended concatenation cannot be received.

8.3.1.8 The TSDU Maximum Size PV shall be present if a TSDU Minimum Size PV is proposed:

a) the first two octets of the PV field shall contain the proposed maximum TSDU size, expressed in octets, in the direction from the initiator to the responder, encoded as a binary number, where the first of the two octets is the high order part of the number;

b) the second two octets of the PV field shall contain the proposed maximum TSDU size, expressed in octets, in the direction from the responder to the initiator, encoded as a binary number, where the first of the two octets is the high order part of the number.

If this parameter is not present, the TSDU Maximum Size is not limited over the session connection. If either pair of octets has the value zero, the TSDU size is not limited in the direction of transfer associated with that pair of octets.

8.3.1.9 The bits in the Version Number PV field shall indicate which protocol versions are proposed for use over the session connection:

a) bit 1 protocol version 1;

b) bit 1 protocol version 2.

Bits 3-8 are reserved.

The encoding for each bit shall be:

c) 0 : use of the protocol version not proposed;

d) 1 : use of the protocol version proposed.

If this PI unit or PV field is absent, the default shall be protocol version 1.

8.3.1.10 The Initial Serial Number PV field shall be present if the activity management functional unit is not proposed and any of the minor synchronize, major synchronize or resynchronize functional units are proposed. As an SS-user option, an Initial Serial Number PV field may be present if the activity management functional unit is proposed provided that any of the minor synchronize, major synchronize or resynchronize functional units are also proposed.

TABLE 11/X.225

Parameters of the CONNECT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
Connection identifier	nm	1	Calling SS-user reference	nm	10	64 octets maximum	¤ 7.1.1 a) 1) ¤ 8.3.1.3
			Common reference	nm	11	64 octets maximum	¤ 7.1.1 a) 2) ¤ 8.3.1.4
			Additional reference information	nm	12	4 octets maximum	¤ 7.1.1 a) 3) ¤ 8.3.1.5
			Protocol options	m	19	1 octet	¤ 7.1.1 b) 1) ¤ 8.3.1.7
Connect/Accept item (see ¤ 8.3.1.6)	nm	5	TSDU maximum size	nm	21	4 octets	¤ 7.1.1 b) 2) ¤ 8.3.1.8
			Version number	m	22	1 octet	¤ 7.1.1 b) 3) ¤ 8.3.1.9
			Initial serial number	nm	23	6 octets maximum	¤ 7.1.1 b) 4) ¤ 8.3.1.10
			Token setting item	nm	26	1 octet	¤ 7.1.1 b) 5) ¤ 8.3.1.11
			Session user requirements	nm	20	2 octets	¤ 7.1.1 c) ¤ 8.3.1.12
			Calling session selector	nm	51	16 octets maximum	¤ 7.1.1 d) ¤ 8.3.1.13
			Called session selector	nm	52	16 octets maximum	¤ 7.1.1 d) ¤ 8.3.1.14
User data	nm	193				512 octets maximum	¤ 7.1.1 e) ¤ 8.3.1.15
			Data overflow	nm	60	1 octet	¤ 7.1.1 f) ¤ 8.3.1.17
Extended user data	nm	194				10 240 octets maximum	¤ 7.1.1 g) ¤ 8.3.1.16

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

Each digit of the serial number is encoded as an octet, as follows:

a) 0 : 0011 0000;

b) 1 : 0011 0001;

c) 2 : 0011 0010;

d) 3 : 0011 0011;

e) 4 : 0011 0100;

f) 5 : 0011 0101;

g) 6 : 0011 0110;

h) 7 : 0011 0111;

i) 8 : 0011 1000;

j) 9 : 0011 1001.

Serial number can range from 0 to 999 999. The most significant digit is encoded first in the PV field. Leading zeros may be omitted.

8.3.1.11 The Token Setting Item PV field, if present, shall indicate the initial position of the tokens. The bits of the Token Setting Item PV field are defined as bit pairs:

a) bits 8, 7 release token;

b) bits 6, 5 major/activity token;

c) bits 4, 3 synchronize-minor token;

d) bits 2, 1 data token.

The encoding for each bit pair shall be:

e) 00 : initiator's side;

f) 01 : responder's side;

g) 10 : called SS user's choice;

h) 11 : reserved.

The values are relevant only if the appropriate functional units are requested in the Session User Requirements parameter. If no functional unit requiring a token has been requested, this parameter need not be present.

If this PI unit or PV field is absent, the default shall be that all tokens whose availability is proposed in the Session User Requirements parameter are assigned to the calling SS-user.

8.3.1.12 The bits in the Session User Requirements PV field shall indicate the functional units proposed by the calling SS-user, for use over this session connection:

a) bit 1 half-duplex functional unit;

b) bit 2 duplex functional unit;

c) bit 3 expedited data functional unit;

d) bit 4 minor synchronize functional unit;

e) bit 5 major synchronize functional unit;

f) bit 6 resynchronize functional unit;

g) bit 7 activity management functional unit;

h) bit 8 negotiated release functional unit;

i) bit 9 capability data functional unit;

j) bit 10 exceptions functional unit;

k) bit 11 typed data functional unit.

Bits 12-16 are reserved.

When this parameter is present, at least one of the half-duplex and the duplex functional units shall be proposed.

The encoding for each bit shall be:

l) 0 : use of functional unit not proposed;

m) 1 : use of functional unit proposed.

When this parameter is absent, the default shall be as though bits 1, 4, 7, 9 and 10 are set to one and the remaining bits are set to zero.

8.3.1.13 The Calling Session Selector, if present, shall be derived from the Calling Session Address supplied by the calling SS-user. When this parameter is absent, the default shall be as though this parameter was set to a null value.

8.3.1.14 The Called Session Selector, if present, shall be derived from the Called Session Address supplied by the calling SS-user. When this parameter is absent, the default shall be as though this parameter was set to a null value.

8.3.1.15 The User Data PV field, if present, shall contain user data supplied by the calling SS-user.

8.3.1.16 The Extended User Data parameter, if present, shall contain user data supplied by the calling SS-user. This parameter shall be present if the Data Overflow parameter is present. This parameter shall not be present if protocol version 1 is proposed.

Only one of the User Data and Extended User Data parameters may be present (see ¤ 7.1.1).

8.3.1.17 The Data Overflow parameter, if present, shall indicate that there is more than 10 240 octets of user data to be transformed. This parameter shall not be present if protocol version 1 is proposed.

The encoding of this field shall be:

bit 1 = 1 more than 10 240 octets of user data;

bit 1 shall never be set equal to 0.

Bits 2-8 are reserved.

If the Data Overflow PI unit or PV field is absent, there are not more than 10 240 octets of user data.

8.3.2 *OVERFLOW ACCEPT (OA) SPDU*

8.3.2.1 The SI field shall contain the value 16.

8.3.2.2 The parameter fields shall be as specified in Table 12/X.225.

TABLE 12/X.225

Parameters of the OVERFLOW ACCEPT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			TSDU maximum size	nm	21	4 octets	¤ 7.2.1 a) ¤ 8.3.2.3
			Version number	m	22	1 octet	¤ 7.2.1 b) ¤ 8.3.2.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.2.3 The TSDU Maximum Size parameter shall be present if a TSDU maximum size is proposed by the receiver. The encoding and default for this field is defined in ¤ 8.3.1.8.

8.3.2.4 In the Version Number PV field bit 2 shall have the value 1 indicating that protocol version 2 is proposed (and selected) for use over this session connection. Bit 1 shall have the value 0 indicating that protocol version 1 is not proposed.

Bits 3-8 are reserved.

8.3.3 *CONNECT DATA OVERFLOW (CDO) SPDU*

8.3.3.1 The SI field shall contain the value 15.

8.3.3.2 The parameter fields shall be as specified in Table 13/X.225.

TABLE 13/X.225

Parameters of the CONNECT DATA OVERFLOW

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure item	m	25	1 octet	¤ 7.3.1 a) ¤ 8.3.3.3
User data	nm	193				65 528 octets maximum	¤ 7.3.1 b) ¤ 8.3.3.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.3.3 The Enclosure Item PV field shall indicate whether or not this SPDU is the end of the SSDU.

The encoding for this field shall be:

a) bit 1 = 0 not beginning of SSDU;

b) bit 2 = 1 end of SSDU;

bit 2 = 0 not end of SSDU.

Bits 3-8 are reserved.

8.3.3.4 The User Data field, if present, shall contain a segment of the associated SSDU. The User Data field shall be present if the Enclosure Item has bit 2 = 0.

8.3.4 *ACCEPT (AC) SPDU*

8.3.4.1 The SI field shall contain the value 14.

8.3.4.2 The parameter fields shall be as specified in Table 14/X.225.

8.3.4.3 The Called SS-user Reference PV field shall be as defined by the called SS-user.

8.3.4.4 The Common Reference PV field shall be as defined by the called SS-user.

8.3.4.5 The Additional Reference Information PV field shall be as defined by the called SS-user.

8.3.4.6 If the Connect/Accept Item is absent, the default values defined for the enclosed PI units shall apply.

8.3.4.7 The Protocol Options PV field shall indicate whether or not the responder is able to receive extended concatenated SPDUs (see ¤ 6.3.7). The encoding and default for this field is defined in ¤ 8.3.1.7.

8.3.4.8 The TSDU Maximum Size parameter shall be present if a TSDU maximum size is proposed by the receiver. The encoding and default for this field is defined in ¤ 8.3.1.8. If an OVERFLOW ACCEPT SPDU has been sent previously, the TSDU maximum size parameter shall have the same value as was indicated on the OVERFLOW ACCEPT SPDU.

TABLE 14/X.225

Parameters of the ACCEPT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
Connection identifier	nm	1	Called SS-user reference	nm	9	64 octets maximum	¤ 7.4.1 a) 1) ¤ 8.3.4.3
			Common reference	nm	11	64 octets maximum	¤ 7.4.1 a) 2) ¤ 8.3.4.4
			Additional reference information	nm	12	4 octets maximum	¤ 7.4.1 a) 3) ¤ 8.3.4.5
Connect/Accept item (see ¤ 8.3.4.6)	nm	5	Protocol options	m	19	1 octet	¤ 7.4.1 b) 1) ¤ 8.3.4.7
			TSDU maximum size	nm	21	4 octets	¤ 7.4.1 b) 2) ¤ 8.3.4.8
			Version number	m	22	1 octet	¤ 7.4.1 b) 3) ¤ 8.3.4.9
			Initial serial number	nm	23	6 octets maximum	¤ 7.4.1 b) 4) ¤ 8.3.4.10
			Token setting item	nm	26	1 octet	¤ 7.4.1 b) 5) ¤ 8.3.4.11
			Token item	nm	16	1 octet	¤ 7.4.1 c) ¤ 8.3.4.12
			Session user requirements	nm	20	2 octets	¤ 7.4.1 d) ¤ 8.3.4.13
			Calling session selector	nm	51	16 octets maximum	¤ 7.4.1 e) ¤ 8.3.4.14
			Responding session selector	nm	52	16 octets maximum	¤ 7.4.1 e) ¤ 8.3.4.15
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.4.17
User data	nm	193				See reference	¤ 7.4.1 f) ¤ 8.3.4.16

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.4.9 The Version Number PV field shall have the value and encoding specified in ¤ 8.3.1.9.

If an OVERFLOW ACCEPT SPDU has been sent previously on this session connection then the Version Number parameter shall have the same value as was indicated in the OVERFLOW ACCEPT SPDU.

8.3.4.10 The Initial Serial Number PV field shall only be present if the activity management functional unit is not selected and if any of the following functional units are selected:

a) minor synchronize functional unit;

b) major synchronize functional unit;

c) resynchronize functional unit.

The encoding for the Initial Serial Number PV field is defined in ¤ 8.3.1.10.

8.3.4.11 The Token Setting Item PV field indicates the initial token settings for each token available on this session connection. The bits and encoding are defined in ¤ 8.3.1.11. In the case where the initial assignment of the related token was indicated as the called SS-user's choice (in the Token Setting Item PV field of the associated CONNECT SPDU), the field shall contain the value chosen by the called SS-user. Otherwise, the values set in the CONNECT SPDU must be returned. The value Òcalled SS-user's choiceÓ is not a permitted value in the ACCEPT SPDU. The values are relevant only if the appropriate functional units are requested in the Session User Requirements parameter. If no functional unit requiring a token has been requested, this parameter need not be present.

8.3.4.12 The Token Item PV field, if present, shall indicate which tokens are requested by the called SS-user:

a) bit 7 = 1 release token;

b) bit 5 = 1 major/activity token;

c) bit 3 = 1 synchronize-minor token;

d) bit 1 = 1 data token.

Bits 2, 4, 6 and 8 are reserved.

Bits corresponding to tokens which are not available are ignored.

8.3.4.13 The bits in the Session User Requirements PV field shall indicate the functional units proposed by the called SS-user, for use over this session connection. This PV field shall not have both bit 1 set (half-duplex functional unit) and bit 2 set (duplex functional unit), but the chosen bit must have been set in the CONNECT SPDU. The encoding and default value is defined in ¤ 8.3.1.12.

8.3.4.14 The Calling Session Selector, if present, shall be the same value as in the CONNECT SPDU.

8.3.4.15 The Responding Session Selector, if present, shall be derived from the Responding Session Address supplied by the responding SS-user. When this parameter is absent, the default shall be as though this parameter is set to a null value.

8.3.4.16 The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets. If the Enclosure Item parameter is present, the User Data parameter is mandatory.

8.3.4.17 The Enclosure Item parameter, if present, shall indicate that the SPDU is the beginning, but not end of the SSDU. This parameter shall not be present if protocol version 1 is selected. The encoding for this field shall be:

a) bit 1 = 1 beginning of SSDU;

b) bit 2 = 0 not end of SSDU.

Bits 3-8 are reserved.

See ¤ 8.4.2 for encoding subsequent SPDUs in the sequence.

8.3.5 *REFUSE (RF) SPDU*

8.3.5.1 The SI field shall contain the value 12.

8.3.5.2 The parameter fields shall be as specified in Table 15/X.225.

TABLE 15/X.225

Parameters of the REFUSE SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
Connection identifier	nm	1	Called SS-user reference	nm	9	64 octets maximum	¤ 7.5.1 a) 1) ¤ 8.3.53
			Common reference	nm	11	64 octets maximum	¤ 7.5.1 a) 2) ¤ 8.3.5.4
			Additional reference information	nm	12	4 octets maximum	¤ 7.5.1 a) 3) ¤ 8.3.5.5
			Transport disconnect	nm	17	1 octet	¤ 7.5.1 b) ¤ 8.3.5.6
			Session user requirements	nm	20	2 octets	¤ 7.5.1 c) ¤ 8.3.5.7
			Version number	nm	22	1 octet	¤ 7.5.1 d) ¤ 8.3.5.8
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.5.10
			Reason code	nm	50	See reference	¤ 7.5.1 e) ¤ 8.3.5.9

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.5.3 The Called SS-user Reference PV field shall be as defined by the called SS-user.

8.3.5.4 The Common Reference PV field shall be as defined by the called SS-user.

8.3.5.5 The Additional Reference Information PV field shall be as defined by the called SS-user.

8.3.5.6 The Transport Disconnect PV field shall indicate whether or not the transport connection is to be kept. The encoding for this field shall be:

a) bit 1 = 0 transport connection is kept;

b) bit 1 = 1 transport connection is released.

Bits 2-8 are reserved.

If this parameter is absent, the transport connection is released.

8.3.5.7 The Session User Requirements PV field shall only be present if the Reason Code is 2 and shall indicate the functional units required by the called SS-user and supported by the responder. The encoding shall be the same as in the CONNECT SPDU (see ¤ 8.3.1.12).

8.3.5.8 The Version Number PV field shall have the value and encoding specified in ¤ 8.3.1.9. If an OVERFLOW ACCEPT SPDU has been sent previously on this session connection, then the Version Number parameter shall have the same value as was indicated in the OVERFLOW ACCEPT SPDU.

8.3.5.9 The Reason Code PV field shall contain a reason code in the first octet. Depending on the value of this first octet, additional octets may be used. The following values are defined for the first octet:

a) 0 : rejection by the called SS user; reason not specified;

b) 1 : rejection by called SS-user due to temporary congestion;

c) 2 : rejection by called SS-user. The following octets may be used for user data up to a length such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets;

d) * 128 + 1 : session selector unknown;

e) * 128 + 2 : SS-user not attached to SSAP;

f) 128 + 3 : SPM congestion at connect time;

g) * 128 + 4 : proposed protocol versions not supported;

h) 128 + 5 : rejection by the SPM; reason not specified;

i) 128 + 6 : rejection by the SPM; implementation restriction stated in the PICS.

Note - Reasons marked with an asterisk (*) may be reported to the SS-user as persistent, others reported as transient.

All other values are reserved.

The Session User Requirements parameter may only be present if the value of the reason code is 2. If the reason code has the value 2 and the Session User Requirements parameter is not present, the default value shall be assumed (see ¤ 8.3.1.12).

If the Enclosure Item parameter is present, the Reason Code parameter is mandatory and shall be followed by octets of user data.

8.3.5.10 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.6 *FINISH (FN) SPDU*

8.3.6.1 The SI field shall contain the value 9.

8.3.6.2 The parameter fields shall be as specified in Table 16/X.225.

TABLE 16/X.225

Parameters of the FINISH SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Transport disconnect	nm	17	1 octet	¤ 7.6.1 a) ¤ 8.3.6.3
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.6.5
User data	nm	193				See reference	¤ 7.6.1 b) ¤ 8.3.6.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.6.3 The Transport Disconnect PV field shall indicate whether or not the transport connection is to be kept. The encoding for this field shall be:

a) bit 1 = 0 transport connection is kept;

b) bit 1 = 1 transport connection is released.

Bits 2-8 are reserved.

If this parameter is absent, the transport connection shall be released.

8.3.6.4 The User Data PV field, if present, shall contain user data supplied by the SS-user.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.6.5 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.7 *DISCONNECT (DN) SPDU*

8.3.7.1 The SI field shall contain the value 10.

8.3.7.2 The parameter field shall be as specified in Table 17/X.225.

TABLE 17/X.225

Parameters of the DISCONNECT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.7.4
User data	nm	193				See reference	¤ 7.7.1 ¤ 8.3.7.3

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.7.3 The User Data PV field, if present, shall contain user data supplied by the SS-user.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.7.4 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.8 *NOT FINISHED (NF) SPDU*

8.3.8.1 The SI field shall contain the value 8.

8.3.8.2 The parameter field shall be as specified in Table 18/X.225.

TABLE 18/X.225

Parameters of the NOT FINISHED SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.8.4
User data	nm	193				See reference	¤ 7.8.1 ¤ 8.3.8.3

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.8.3 The User Data PV field, if present, shall contain user data supplied by the SS-user. The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.8.4 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.9 *ABORT (AB) SPDU*

8.3.9.1 The SI field shall contain the value 25.

8.3.9.2 The parameter fields shall be as specified in Table 19/X.225.

TABLE 19/X.225

Parameters of the ABORT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Transport disconnect	m	17	1 octet	¤ 7.9.1 a) ¤ 8.3.9.3
			Reflect parameter values	nm	49	9 octets maximum	¤ 7.9.1 b) ¤ 8.3.9.4
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.9.6
User data	nm	193				See reference	¤ 7.9.1 c) ¤ 8.3.9.5

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.9.3 The Transport Disconnect PV field shall indicate whether or not the transport connection is to be kept, together with an optional reason code. The encoding for this field shall be:

a) bit 1 = 0 transport connection is kept;

b) bit 1 = 1 transport connection is released;

c) bit 2 = 1 user abort (see ¤ 8.3.9.5);

d) bit 3 = 1 protocol error (see ¤ 8.3.9.4);

e) bit 4 = 1 no reason;

f) bit 5 = 1 implementation restriction stated in the PICS.

Bits 6-8 are reserved.

8.3.9.4 The Reflect Parameter Values PV field shall only be present if the Transport Disconnect PV field indicates protocol error and shall contain an implementation defined value and semantics.

8.3.9.5 The User Data PV field shall only be present if the Transport Disconnect PV field indicates user abort and shall contain user data supplied by the SS-user.

If this SPDU is to be sent on the transport expedited flow, the length of the User Data parameter is limited to 9 octets and the Enclosure Item shall not be present. If the SPDU is to be sent on the transport normal flow, the length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.9.6 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.10 *ABORT ACCEPT (AA) SPDU*

8.3.10.1 The SI field shall contain the value 26.

8.3.10.2 There is no parameter field associated with this SPDU.

8.3.11 *DATA TRANSFER (DT) SPDU*

8.3.11.1 The SI field shall contain the value 1.

8.3.11.2 The parameter field shall be as specified in Table 20/X.225.

TABLE 20/X.225

Parameters of the DATA TRANSFER SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure item	nm	25	1 octet	¤ 7.11.1 a) ¤ 8.3.10.3
User information field						unlimited	¤ 7.11.1 b) ¤ 8.3.10.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.11.3 The Enclosure Item PV field, if present, shall indicate whether or not this SPDU is the beginning or end of the SSDU. This field shall be present if segmenting has been selected. This field shall not be present if segmenting has not been selected. The encoding for this field shall be:

a) bit 1 = 1 beginning of SSDU;

bit 1 = 0 not beginning of SSDU;

b) bit 2 = 1 end of SSDU;

bit 2 = 0 not end of SSDU.

Bits 3-8 are reserved.

If this field is not present, the default shall be as though bit 1 = 1 and bit 2 = 1 (i.e. beginning and end of SSDU).

8.3.11.4 The User Information Field, if present, shall contain user data supplied by the SS-user. The User Information Field shall be present if the Enclosure Item is not present, or has bit 2 = 0.

8.3.12 *EXPEDITED (EX) SPDU*

8.3.12.1 The SI field shall contain the value 5.

8.3.12.2 This SPDU contains only a User Information Field as specified in Table 21/X.225.

TABLE 21/X.225

Parameters of the EXPEDITED SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
User information field						14 octets maximum	¤ 7.12.1 ¤ 8.3.12.3

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.12.3 The User Information Field shall contain user data supplied by the SS-user.

8.3.13 *TYPED DATA (TD) SPDU*

8.3.13.1 The SI field shall contain the value 33.

8.3.13.2 The parameter field shall be as specified in Table 22/X.225.

TABLE 22/X.225

Parameters of the TYPED DATA SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure item	nm	25	1 octet	¤ 7.13.1 a) ¤ 8.3.13.3
User information field						unlimited	¤ 7.13.1 b) ¤ 8.3.13.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.13.3 The Enclosure Item PV field, if present, shall indicate whether or not this SPDU is the beginning or end of the SSDU. This field shall be present if segmenting has been selected. This field shall not be present if segmenting has not been selected. The encoding for this field shall be:

a) bit 1 = 1 beginning of SSDU;

bit 1 = 0 not beginning of SSDU;

b) bit 2 = 1 end of SSDU;

bit 2 = 0 not end of SSDU.

Bits 3-8 are reserved.

If this field is not present, the default shall be as though bit 1 = 1 and bit 2 = 1 (i.e. beginning and end of SSDU).

8.3.13.4 The User Information Field, if present, shall contain user data supplied by the SS-user. The User Information Field shall be present if the Enclosure Item is not present, or has bit 2 = 0.

8.3.14 *CAPABILITY DATA (CD) SPDU*

8.3.14.1 The SI field shall contain the value 61.

8.3.14.2 The parameter field shall be as specified in Table 23/X.225.

TABLE 23/X.225

Parameters of the CAPABILITY DATA SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.14.4
User data	nm	193				See reference	¤ 7.14.1 ¤ 8.3.14.3

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.14.3 The User Data PV field, if present, shall contain user data supplied by the SS-user.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.14.4 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.15 *CAPABILITY DATA ACK (CDA) SPDU*

8.3.15.1 The SI field shall contain the value 62.

8.3.15.2 The parameter field shall be as specified in Table 24/X.225.

TABLE 24/X.225

Parameters of the CAPABILITY DATA ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.15.4
User data	nm	193				See reference	¤ 7.15.1 ¤ 8.3.15.3

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.15.3 The User Data PV field, if present, shall contain user data supplied by the SS-user.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.15.4 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.16 *GIVE TOKENS (GT) SPDU*

8.3.16.1 The SI field shall contain the value 1.

8.3.16.2 The parameter field shall be as specified in Table 25/X.225.

TABLE 25/X.225

Parameters of the GIVE TOKENS SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Token item	nm	16	1 octet	¤ 7.16.1 ¤ 8.3.16.3
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.16.5
User data	nm	193				See reference	¤ 7.16.1 b)

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.16.3 The Token Item PV field, if present, shall indicate which tokens are being given by the sending SS-user:

a) bit 7 = 1 release token;

b) bit 5 = 1 major/activity token;

c) bit 3 = 1 synchronize-minor token;

d) bit 1 = 1 data token.

Bits 2, 4, 6 and 8 are reserved.

Bits corresponding to tokens which are not available are ignored.

If this PV field is present, at least one bit corresponding to an available token shall be set to one.

8.3.16.4 This SPDU may be used without the Token Item PI unit when concatenated with Category 2 SPDUs according to Tables 7/X.225 and 8/X.225. With some concatenations (see Tables 7/X.225 and 8/X.225) the Token Item PI unit must be absent.

8.3.16.5 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.16.6 The User Data PV field, if present, shall contain user data supplied by the SS-user. This PGI unit shall only be present if the Token Item PI unit is present and shall not be present if protocol version 1 is selected. The length of the User Data parameter is limited such that the total length (including SI and LI) does not exceed 65 539 octets.

8.3.17 *PLEASE TOKENS (PT) SPDU*

8.3.17.1 The SI field shall contain the value 2.

8.3.17.2 The parameter fields shall be as specified in Table 26/X.225.

TABLE 26/X.225

Parameters of the PLEASE TOKENS SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Token item	nm	16	1 octet	¤ 7.17.1. a) ¤ 8.3.17.3
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.17.6
User data	nm	193				See reference	¤ 7.17.1 b) ¤ 8.3.17.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.17.3 The Token Item PV field, if present, shall indicate which tokens are being requested by the sending SS-user:

a) bit 7 = 1 release token;

b) bit 5 = 1 major/activity token;

c) bit 3 = 1 synchronize-minor token;

d) bit 1 = 1 data token.

Bits 2, 4, 6 and 8 are reserved.

Bits corresponding to tokens which are not available are ignored.

If this PV field is present, at least one bit corresponding to an available token shall be set to one.

8.3.17.4 The User Data PV field, if present, shall contain user data supplied by the SS-user. This PGI unit shall only be present if the Token Item PI unit is present.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.17.5 This SPDU may be used without the Token Item PI unit and the User Data PGI unit when concatenated with Category 2 SPDUs according to Tables 7/X.225 and 8/X.225. In this case, the SPDU does not achieve any Please Token function.

8.3.17.6 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.18 *GIVE TOKENS CONFIRM (GTC) SPDU*

8.3.18.1 The SI field shall contain the value 21.

8.3.18.2 The parameter fields shall be as specified in Table 27/X.225.

TABLE 27/X.225

Parameters of the GIVE TOKENS CONFIRM SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.18.3
User data	nm	193				See reference	¤ 7.18.1 ¤ 8.3.18.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.18.3 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.18.4 The User Data PV field, if present, shall contain user data supplied by the SS-user. This PGI unit shall not be present if protocol version 1 is selected. The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.19 *GIVE TOKENS ACK (GTA) SPDU*

8.3.19.1 The SI field shall contain the value 22.

8.3.19.2 There is no parameter field associated with this SPDU.

8.3.20 *MINOR SYNC POINT (MIP) SPDU*

8.3.20.1 The SI field shall contain the value 49.

8.3.20.2 The parameter fields shall be as specified in Table 28/X.225.

TABLE 28/X.225

Parameters of the MINOR SYNC POINT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Sync type item	nm	15	1 octet	¤ 7.20.1 a) ¤ 8.3.20.3
			Serial number	m	42	6 octets maximum	¤ 7.20.1 b) ¤ 8.3.20.4
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.20.6
User data	nm	193				See reference	¤ 7.20.1 c) ¤ 8.3.20.5

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.20.3 The Sync Type Item PV field, if present, shall indicate that an explicit confirmation is not required:

bit 1 = 1 explicit confirmation not required.

Bits 2-8 are reserved.

This parameter field shall be absent if an explicit confirmation is required.

8.3.20.4 The Serial Number PV field shall be coded as specified in ¤ 8.3.1.10.

8.3.20.5 The User Data PV field, if present, shall contain user data supplied by the SS-user.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.20.6 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.21 *MINOR SYNC (MIA) SPDU*

8.3.21.1 The SI field shall contain the value 50.

8.3.21.2 The parameter fields shall be as specified in Table 29/X.225.

TABLE 29/X.225

Parameters of the MINOR SYNC ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Serial number	m	42	6 octets maximum	¤ 7.21.1 a) ¤ 8.3.21.3
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.21.5
			User data	nm	46	See reference	¤ 7.21.1 b) ¤ 8.3.21.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.21.3 The Serial Number PV field shall be coded as specified in ¤ 8.3.1.10.

8.3.21.4 The User Data PV field, if present, shall contain user data supplied by the SS-user.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.21.5 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.22 *MAJOR SYNC POINT (MAP) SPDU*

8.3.22.1 The SI field shall contain the value 41. This is the same value as the SI field for the ACTIVITY END SPDU (see ¤ 8.3.35).

8.3.22.2 The parameter fields shall be as specified in Table 30/X.225.

TABLE 30/X.225

Parameters of the MAJOR SYNC POINT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Sync type item	m	15	1 octet	¤ 7.22.1 a) ¤ 8.3.22.3
			Serial number	m	42	6 octets maximum	¤ 7.20.1 b) ¤ 8.3.22.4
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.22.6
User data	nm	193				See reference	¤ 7.20.1 c) ¤ 8.3.22.5

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.22.3 The Sync Type Item PV field shall indicate that this is not the end of an activity:

bit 1 = 1 major synchronization point without end of activity.

Bits 2-8 are reserved.

8.3.22.4 The Serial Number PV field shall be coded as specified in ¤ 8.3.1.10.

8.3.22.5 The User Data PV field, if present, shall contain user data supplied by the SS-user.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.22.6 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.23 *MAJOR SYNC ACK (MAA) SPDU*

8.3.23.1 The SI field shall contain the value 42.

8.3.23.2 The parameter fields shall be as specified in Table 31/X.225.

TABLE 31/X.225

Parameters of the MAJOR SYNC ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Serial number	m	42	6 octets maximum	¤ 7.23.1 a) ¤ 8.3.23.3
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.23.5
User data	nm	193				See reference	¤ 7.23.1 b) ¤ 8.3.23.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.23.3 The Serial Number PV field shall be coded as specified in ¤ 8.3.1.10.

8.3.23.4 The User Data PV field, if present, shall contain user data supplied by the SS-user.

Note - This SPDU is identical to the ACTIVITY END ACK SPDU (see ¤ 8.3.36).

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.23.5 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.24 *RESYNCHRONIZE (RS) SPDU*

8.3.24.1 The SI field shall contain the value 53.

8.3.24.2 The parameter fields shall be as specified in Table 32/X.245.

8.3.24.3 The Token Setting Item PV field indicates the requesting SS-user's proposed settings for each available token. The bits of the Token Setting Item PV field are defined as bit pairs:

a) bits 8, 7 release token;

b) bits 6, 5 major/activity token;

c) bits 4, 3 synchronize-minor token;

d) bits 2, 1 data token.

The encoding for each bit pair shall be:

e) 00 : requestor's side;

f) 01 : acceptor's side;

g) 10 : accepting SS-user's choice;

h) 11 : reserved.

The values are relevant only if the appropriate token is available. If no token is available, this parameter need not be present.

TABLE 32/X.225

Parameters of the RESYNCHRONIZE SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Token setting item	nm	26	1 octet	¤ 7.24.1 a) ¤ 8.3.24.3
			Resync type	m	27	1 octet	¤ 7.24.1 b) ¤ 8.3.24.3
			Serial number	m	42	6 octets maximum	¤ 7.24.1 c) ¤ 8.3.24.4
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.24.7
User data	nm	193				See reference	¤ 7.24.1 d) ¤ 8.3.24.6

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.24.4 The Resync Type PV field indicates the resynchronize type which is required:

a) 0 : resynchronize restart;

b) 1 : resynchronize abandon;

c) 2 : resynchronize set.

All other values are reserved.

8.3.24.5 The Serial Number PV field shall be encoded as specified in ¤ 8.3.1.10.

8.3.24.6 The User Data PV field, if present, shall contain user data supplied by the SS-user.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.24.7 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.25 *RESYNCHRONIZE ACK (RA) SPDU*

8.3.25.1 The SI field shall contain the value 34.

8.3.25.2 The parameter fields shall be as specified in Table 33/X.245.

TABLE 33/X.225

Parameters of the RESYNCHRONIZE ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Token setting item	nm	26	1 octet	¤ 7.25.1 a) ¤ 8.3.25.3
			Serial number	m	42	6 octets maximum	¤ 7.25.1 b) ¤ 8.3.25.4
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.25.6
User data	nm	193				See reference	¤ 7.25.1 c) ¤ 8.3.25.5

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.25.3 The Token Setting Item PV field indicates token settings for each token available on the session connection. The bits and encoding are defined in ¤ 8.3.24.3. For the case where the requesting SS-user has indicated that the assignment is the accepting SS-user's choice, the field shall contain the values chosen by the accepting SS-user. Otherwise, the values in the RESYNCHRONIZE SPDU must be returned.

This parameter need not be present if no tokens are available.

8.3.25.4 The Serial Number PV field shall be coded as specified in ¤ 8.3.1.10.

8.3.25.5 The User Data PV field, if present, shall contain user data supplied by the SS-user.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.25.6 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.26 *PREPARE (PR) SPDU*

8.3.26.1 The SI field shall contain the value 7.

8.3.26.2 The parameter field shall be as specified in Table 34/X.225.

TABLE 34/X.225

Parameters of the PREPARE SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Prepare type	m	24	1 octet	¤ 7.26.1 ¤ 8.3.26.3

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.26.3 The Prepare Type PV field indicates which SPDU should be expected on the transport normal flow. The value for this field shall be:

a) 1 Prepare for MAJOR SYNC ACK SPDU;

b) 2 Prepare for RESYNCHRONIZE SPDU;

c) 3 Prepare for RESYNCHRONIZE ACK SPDU;

d) 4 Prepare for ABORT SPDU.

All other values are reserved and shall not be used.

8.3.27 *EXCEPTION REPORT (ER) SPDU*

8.3.27.1 The SI field shall contain the value 0.

8.3.27.2 The parameter field shall be as specified in Table 35/X.225.

TABLE 35/X.225

Parameters of the EXCEPTION REPORT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Reflect parameter values	m	49	65 531 octets maximum	¤ 7.27.1 ¤ 8.3.27.3

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.27.3 The Reflect Parameter Values PV field shall contain the bit pattern of the SPDU in error, up to and including the detected error, to a maximum of *n* octets,

where 1024 £ *n* £ 65 531

Note - Not all implementations may be able to deal with field lengths greater than 1024. It is recommended that, whenever possible, the Reflect Parameters PV field should contain the bit pattern of the SPDU in error up to and including the detected error.

8.3.28 *EXCEPTION DATA (ED) SPDU*

8.3.28.1 The SI field shall contain the value 48.

8.3.28.2 The parameter fields shall be as specified in Table 36/X.225.

8.3.28.3 The Reason Code PV field shall contain one of the following values:

a) 0 No specific reason stated;

b) 1 Temporarily unable to continue;

c) 2 Reserved;

d) 3 User sequence error;

e) 4 Reserved;

f) 5 Local SS-user error;

g) 6 Unrecoverable procedural error;

h) 128 Demand data token.

All other values are reserved and shall not be used.

TABLE 36/X.225

Parameters of the EXCEPTION DATA SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Reason code	m	50	1 octet	¤ 7.28.1 a) ¤ 8.3.25.3
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.28.5
User data	nm	193				See reference	¤ 7.28.1 b) ¤ 8.3.28.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.28.4 The User Data PV field, if present, shall contain user data supplied by the SS-user.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.28.5 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.29 *ACTIVITY START (AS) SPDU*

8.3.29.1 The SI field shall contain the value 45.

8.3.29.2 The parameter fields shall be as specified in Table 37/X.225.

TABLE 37/X.225

Parameters of the ACTIVITY START SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Activity identifier	m	41	6 octets maximum	¤ 7.29.1 a) ¤ 8.3.29.3
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.29.5
User data	nm	193				See reference	¤ 7.29.1 b) ¤ 8.3.29.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.29.3 The Activity Identifier PV field shall be as defined by the sending SS-user.

8.3.29.4 The User Data PV field, if present, shall contain user data supplied by the SS-user.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.29.5 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.30 *ACTIVITY RESUME (AR) SPDU*

8.3.30.1 The SI field shall contain the value 29.

8.3.30.2 The parameter fields shall be as specified in Table 38/X.225.

8.3.30.3 The Called SS-user Reference PV field shall be as defined by the sending SS-user.

8.3.30.4 The Calling SS-user Reference PV field shall be as defined by the sending SS-user.

8.3.30.5 The Common Reference PV field shall be as defined by the sending SS-user.

8.3.30.6 The Additional Reference Information PV field shall be as defined by the sending SS-user.

8.3.30.7 The Old Activity Identifier PV field shall be as defined by the sending SS-user.

TABLE 38/X.225

Parameters of the ACTIVITY RESUME SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
Linking information	m	33	Called SS-user reference	nm	9	64 octets maximum	¤ 7.30.1 a) 1) ¤ 8.3.30.3
			Calling SS-user reference	nm	10	64 octets maximum	¤ 7.30.1 a) 2) ¤ 8.3.30.4
			Common reference	nm	11	64 octets maximum	¤ 7.30.1 a) 3) ¤ 8.3.30.5
			Additional reference information	nm	12	4 octets maximum	¤ 7.30.1 a) 4) ¤ 8.3.30.6
			Old activity identifier	m	41	6 octets maximum	¤ 7.30.1 a) 5) ¤ 8.3.30.7
			Serial number	m	42	6 octets maximum	¤ 7.30.1 a) 6) ¤ 8.3.30.8
			New activity identifier	m	41	6 octets maximum	¤ 7.30.1 b) ¤ 8.3.30.9
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.30.11
User data	nm	193				See reference	¤ 7.30.1 c) ¤ 8.3.30.10

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.30.8 The Serial Number PV field shall be coded as specified in ¤ 8.3.1.10.

8.3.30.9 The New Activity Identifier PV field shall be as defined by the sending SS-user.

8.3.30.10 The User Data PV field, if present, shall contain user data supplied by the SS-user.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.30.11 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.31 *ACTIVITY INTERRUPT (AI) SDPU*

8.3.31.1 The SI field shall contain the value 25.

8.3.31.2 The parameter fields shall be as specified in Table 39/X.225.

TABLE 39/X.225

Parameters of the ACTIVITY INTERRUPT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Reason code	nm	50	1 octet	¤ 7.31.1 ¤ 8.3.31.3
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.31.4
User data	nm	193				See reference	¤ 7.31.1 b) ¤ 8.3.31.5

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.31.3 The Reason Code PV field shall contain one of the following values:

a) 0 No specific reason stated;

b) 1 Temporarily unable to continue;

c) 2 Reserved;

d) 3 User sequence error;

e) 4 Reserved;

f) 5 Local SS-user error;

g) 6 Unrecoverable procedural error;

h) 128 Demand data token.

All other values are reserved and shall not be used.

8.3.31.4 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.31.5 The User Data PV field, if present, shall contain user data supplied by the SS-user. This PGI unit shall not be present if protocol version 1 is selected.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.32 *ACTIVITY INTERRUPT ACK (AIA) SPDU*

8.3.32.1 The SI field shall contain the value 26.

8.3.32.2 The parameter fields shall be as specified in Table 40/X.225.

TABLE 40/X.225

Parameters of the ACTIVITY INTERRUPT ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.32.3
User data	nm	193				See reference	¤ 7.32.1 ¤ 8.3.32.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.32.3 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.32.4 The User Data PV field, if present, shall contain user data supplied by the SS-user. This PGI unit shall not be present if protocol version 1 is selected.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.33 *ACTIVITY DISCARD (AD) SPDU*

8.3.33.1 The SI field shall contain the value 57.

8.3.33.2 The parameter fields shall be as specified in Table 41/X.225.

TABLE 41/X.225

Parameters of the ACTIVITY DISCARD SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Reason code	nm	50	1 octet	¤ 7.33.1 ¤ 8.3.33.3
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.33.4
User data	nm	193				See reference	¤ 7.33.1 b) ¤ 8.3.33.5

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.33.3 The Reason Code PV field shall contain one of the following values:

a) 0 No specific reason stated;

b) 1 Temporarily unable to continue;

c) 2 Reserved;

d) 3 User sequence error;

e) 4 Reserved;

f) 5 Local SS-user error;

g) 6 Unrecoverable procedural error;

h) 128 Demand data token.

All other values are reserved and shall not be used.

8.3.33.4 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.33.5 The User Data PV field, if present, shall contain user data supplied by the SS-user. This PGI unit shall not be present if protocol version 1 is selected.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.34 *ACTIVITY DISCARD ACK (ADA) SPDU*

8.3.34.1 The SI field shall contain the value 58.

8.3.34.2 The parameter fields shall be as specified in Table 42/X.225.

TABLE 42/X.225

Parameters of the ACTIVITY DISCARD ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.34.3
User data	nm	193				See reference	¤ 7.34.1 ¤ 8.3.34.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.34.3 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.34.4 The User Data PV field, if present, shall contain user data supplied by the SS-user. This PGI unit shall not be present if protocol version 1 is selected.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.35 *ACTIVITY END (AE) SPDU*

8.3.35.1 The SI field shall contain the value 41. This is the same value as the SI field for the MAJOR SYNC POINT SPDU (see ¤ 8.3.22).

8.3.35.2 The parameter fields shall be as specified in Table 43/X.225.

TABLE 43/X.225

Parameters of the ACTIVITY END SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Serial number	m	42	6 octets maximum	¤ 7.36.1 a) ¤ 8.3.35.3
			Enclosure item	nm	25	1 octet	¤ 7.37.1 ¤ 8.3.35.5
User data	nm	193				See reference	¤ 7.36.1 b) ¤ 8.3.35.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.3.35.3 The Serial Number PV field shall be coded as specified in ¤ 8.3.1.10.

8.3.35.4 The User Data PV field, if present, shall contain user data supplied by the SS-user.

The length of the User Data parameter is limited such that the total length (including SI and LI) of the SPDU does not exceed 65 539 octets.

8.3.35.5 The Enclosure Item parameter, if present, shall be encoded as specified in ¤ 8.3.4.17. This parameter shall not be present if protocol version 1 is selected.

8.3.36 *ACTIVITY END ACK (AEA) SPDU*

The ACTIVITY END ACK SPDU is identical to the MAJOR SYNC ACK SPDU (see ¤ 8.3.23).

8.4 *Additional encoding rules for segmented SSDUs*

ACCEPT SPDU

REFUSE SPDU

FINISH SPDU

DISCONNECT SPDU

NOT FINISHED SPDU

ABORT SPDU

CAPABILITY DATA SPDU

CAPABILITY DATA ACK SPDU

GIVE TOKENS SPDU

PLEASE TOKENS SPDU

GIVE TOKENS CONFIRM SPDU

MINOR SYNC POINT SPDU

MINOR SYNC POINT ACK SPDU

MAJOR SYNC POINT SPDU

MAJOR SYNC POINT ACK SPDU

RESYNCHRONIZE SPDU

RESYNCHRONIZE ACK SPDU

EXCEPTION DATA SPDU

ACTIVITY START SPDU

ACTIVITY RESUME SPDU

ACTIVITY INTERRUPT SPDU

ACTIVITY INTERRUPT ACK SPDU

ACTIVITY DISCARD SPDU

ACTIVITY DISCARD ACK SPDU

ACTIVITY END SPDU

ACTIVITY END ACK SPDU

8.4.1 *First SPDU in sequence*

The first SPDU in the sequence shall be as specified in ¤ 8.3.

8.4.2 *Subsequent SPDUs in sequence*

8.4.2.1 For all SPDUs except the REFUSE SPDU and the MINOR SYNC ACK SPDU, the encoding shall be as follows:

8.4.2.1.1 The SI field shall have the same value as the SI field value of the initial SPDU of the sequence.

8.4.2.1.2 The parameter fields shall be as specified in Table 44/X.225.

8.4.2.1.3 The Enclosure Item PV field shall indicate whether or not this SPDU is the end of the SSDU. The encoding shall be:

a) bit 1 = 0 not beginning of SSDU;

b) bit 2 = 1 end of SSDU;

bit 2 = 0 not end of SSDU.

Bits 3-8 are reserved.

8.4.2.1.4 The User Data field, if present, shall contain a segment of the associated SSDU. The User Data field shall be present if the Enclosure Item has bit 2 = 0.

TABLE 44/X.225

Parameters of subsequent SPDUs when segmenting is required

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure item	m	25	1 octet	¤ 7.37.1 ¤ 8.4.2.1.3
User data	nm	193				65528 octets maximum	¤ 7.37.1 ¤ 8.4.2.1.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.4.2.2 For the REFUSE SPDU the encoding shall be as follows:

8.4.2.2.1 The SI field shall have the same value as the SI field value of the initial SPDU of the sequence.

8.4.2.2.2 The parameter fields shall be as specified in Table 45/X.225.

8.4.2.2.3 The Enclosure Item PV field shall indicate whether or not this SPDU is the end of the SSDU. The encoding shall be:

a) bit 1 = 0 not beginning of SSDU;

b) bit 2 = 1 end of SSDU;

bit 2 = 0 not end of SSDU.

Bits 3-8 are reserved.

8.4.2.2.4 The Reason Code field, if present, shall contain a segment of the associated SSDU. The Reason Code field shall be present if the Enclosure Item has bit 2 = 0.

TABLE 45/X.225

Parameters of subsequent REFUSE SPDUs when segmenting is required

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure item	m	25	1 octet	¤ 7.37.1 ¤ 8.4.2.2.3
Reason code	nm				50	65 528 octets maximum	¤ 7.37.1 ¤ 8.4.2.2.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

8.4.2.3 For the MINOR SYNC ACK SPDU the encoding shall be as follows:

8.4.2.3.1 The SI field shall have the same value as the SI field value of the initial SPDU of the sequence.

8.4.2.3.2 The parameter fields shall be as specified in Table 46/X.225.

8.4.2.3.3 The Enclosure Item PV field shall indicate whether or not this SPDU is the end of the SSDU. The encoding shall be:

a) bit 1 = 0 not beginning of SSDU;

b) bit 2 = 1 end of SSDU;

bit 2 = 0 not end of SSDU.

Bits 3-8 are reserved.

8.4.2.3.4 The User Data field, if present, shall contain a segment of the associated SSDU. The User Data field shall be present if the Enclosure Item has bit 2 = 0.

TABLE 46/X.225

Parameters of subsequent MINOR SYNC ACK SPDUs when segmenting is required

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure item	m	25	1 octet	¤ 7.37.1 ¤ 8.4.2.3.3
			User data	nm	46	65 528 octets maximum	¤ 7.37.1 ¤ 8.4.2.3.4

m: mandatory.

nm: not mandatory (see ¤ 8.2.6).

9 Conformance to this standard

9.1 A system claiming conformance to this Recommendation shall exhibit external behaviour consistent with having implemented an SPM for the Kernel functional unit together with either or both of the Half-duplex and the Duplex functional units.

9.2 The system may exhibit external behaviour consistent with containing an implementation of any other functional unit provided that:

a) if the Capability Data functional unit is implemented, the Activity Management functional unit shall also be implemented; and

b) if the Exceptions functional unit is implemented, the Half-duplex functional unit shall also be implemented.

9.3 For all protocol versions which are claimed to be implemented, the system shall be capable of:

a) initiating a session connection (by sending a CONNECT SPDU) or responding to a CONNECT SPDU (according to the procedures specified in section 7), or both;

b) following all the remaining procedures in the Kernel functional unit; and

c) following all the procedures for each functional unit that the system claims to implement,

where following the procedures specified in b) and c) shall mean:

d) accepting all correct sequences of SPDUs received from peer equipment, and responding with correct SPDU sequences, for the defined states of a session connection;

e) responding correctly to all incorrect sequences of SPDUs received for a defined state of a session connection.

9.4 Claims of conformance shall state:

a) which functional units are implemented;

b) whether or not extended concatenation is implemented;

c) whether or not segmenting is implemented and, if segmenting is implemented, the maximum size of TSDU which the system is capable of handling;

d) whether or not the use of transport expedited service is implemented;

e) which protocol versions are implemented.

9.5 The implementor shall provide a Protocol Implementation Conformance Statement (PICS).

Note - In particular any limit imposed by an implementation on the number of octets of SS-user data which can be passed in a single session primitive shall be stated in the PICS.

ANNEX A

(to Recommendation X.225)

State tables

A.1 *General*

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

These state tables do not constitute a formal definition of the session protocol; they are included to provide a more precise specification of the elements of procedure described in ¤ 7.

Table A-1/X.225 specifies the abbreviated name, category and name of each incoming event. The categories are SS-user event, TS-provider event, timer event and valid SPDU event.

Table A-2/X.225 specifies the abbreviated name and name of each state.

Table A-3/X.225 specifies the abbreviated name, category and name of each outgoing event. The categories are SS-provider event, TS-user event and SPDU event.

Table A-4/X.225 summarizes the operations on the variables (VA), V(M), V(R) and Vsc.

Table A-5/X.225 specifies the specific actions.

Table A-6/X.225 specifies the predicates.

Tables A-7/X.225 to A-15/X.225 specify the state tables.

A.2 *Notation for state tables*

A.2.1 Incoming events, states and outgoing events are represented by their abbreviated names.

A.2.2 Specific actions are represented by the notation [*n*], where *n* is the number of the specific action in Table A-5/X.225.

A.2.3 Notes are represented by the notation (*n*), where *n* is the number of the note at the foot of Table A-6/X.225.

A.2.4 Predicates are represented by the notation p*n*, where *n* is the number of the predicate in Table A-6/X.225.

A.2.5 Boolean operators are represented by the following notation:

& AND

" NOT

OR OR

A.3 *Conventions for entries in state tables*

A.3.1 The intersection of each state and incoming event which is invalid is left blank.

A.3.2 The intersection of each state and incoming event which is valid contains entries which are either:

a) an *action list* which:

1) may contain outgoing events and/or specific actions;

2) always contains the resultant state;

or

b) one or more *conditional action lists*, each consisting of:

1) a predicate expression comprising predicates and Boolean operators;

2) an action list (as in ¤ A.3.2 a)).

Note - The action lists and conditional action lists use the notation in ¤ A.2.

A.3.3 The intersection of each state and incoming event which is not logically possible for the SPM is indicated by // in the top lefthand corner of the intersection.

Note - Such entries are a consequence of the tabular presentation of the state tables.

A.4 *Actions to be taken by the SPM*

The state tables define the action to be taken by the SPM.

A.4.1 *Invalid intersections*

If the intersection of the state and an incoming event is invalid, one of the following actions shall be taken.

A.4.1.1 If the incoming event comes from the SS-user, any action taken by the SPM is a local matter.

A.4.1.2 If the incoming event is related to a received SPDU and if the state of the transport connection makes it possible, the SPM shall either:

a) take the following actions:

1) issue an S-P-ABORT indication;

2) send an ABORT SPDU;

3) start the timer, TIM;

4) wait for a T-DISCONNECT indication or an ABORT ACCEPT SPDU (STA 16); or

b) if the following conditions hold:

1) the data token is available but not assigned to the SPM; and

2) - the activity management functional unit has not been selected; or

- the activity management functional unit has been selected and an activity is in progress; or

- the activity management functional unit has been selected and the SPM is in STA 22; and

3) the exceptions functional unit has been selected; and

4) the session connection is in the data transfer phase (i.e. states 4A, 4B, 5A, 5B, 5C, 6, 10A, 10B, 11A, 11B, 11C, 15A, 15B, 15C, 19, 20, 22, 713);

take the following actions:

5) send an EXCEPTION REPORT SPDU;

6) issue an S-P-EXCEPTION-REPORT indication;

7) enter STA 20 and wait for a recovery request or SPDU.

Note: - It should be noted that sending an EXCEPTION REPORT SPDU may lead to an SPM deadlock. It is therefore advised to send the ABORT SPDU rather than the EXCEPTION REPORT SPDU, especially in the case of protocol errors.

A.4.1.3 If the incoming event falls into neither of the above categories (including those which are impossible by the definition of the behaviour of the SPM or TS-provider) any action taken by the SPM is a local matter.

A.4.2 *Valid intersections*

If the intersection of the state and incoming event is valid, one of the following actions shall be taken.

A.4.2.1 If the intersection contains an action list, the SPM shall take the specific actions in the order specified in the state table.

A.4.2.2 If the intersection contains one or more conditional action lists, for each predicate expression that is true the SPM shall take the specific actions in the order given in the action list associated with the predicate expression. If none of the predicate expressions are true, the SPM shall take one of the actions defined in ¤ A.4.1.

A.4.2.3 *Procedures for segmented SSDUs*

The state tables do not take account of segmented SSDUs. When an outgoing SSDU is to be segmented or an incoming SSDU is segmented, the procedures defined in ¤ 7.37 apply to the outgoing event at the appropriate intersection of the state tables (that part of the action which transmits the SPDU).

A.4.3 *Receipt of SPDUs*

A.4.3.1 *Valid SPDUs*

The SPM shall process valid SPDUs as specified in Tables A-7/X.225 to A-15/X.225.

A.4.3.1.1 *Rules of extensibility*

This Recommendation does not specify the action to be taken in response to a PGI unit containing a PGI code listed in Annex C, or to a PI unit containing a PI code listed in Annex C.

An SPM receiving an SPDU containing a valid SI field but containing a PGI unit whose PGI code is not specified in ¤ 8.3 or in Annex C, shall ignore that PGI unit (see notes).

An SPM reciving an SPDU containing a valid SI field but containing a PI unit whose PI code is not specified in ¤ 8.3 or in Annex C, shall ignore that PI unit (see notes).

The SPM shall ignore any bits within a parameter field which are specified as reserved in ¤ 8.3.

Note 1 - The received SPDU is processed as through the unknown PGI and/or PI units were not present in the SPDU.

Note 2 - The provisions permit communication with systems operating other versions of this protocol.

A.4.3.1.2 *User Data length restrictions*

If an SPM receives an SPDU, or an ordered sequence of SPDUs which together comprise a single SSDU, which contains more SS-user data than the SPM is prepared to accept (and as stated in the PICS), it shall take the actions defined in either ¤ A.4.1.2 a) or ¤ A.4.1.2 b).

A.4.3.2 *Invalid SPDUs*

If an invalid SPDU is received, the SPM shall:

a) take the actions defined in ¤ A.4.1.2 a); or

b) take the actions defined in ¤ A.4.1.2 b); or

c) take any other action that does not violate the procedures specified in this Recommendation; or

d) take no action.

A.5 *Definitions of sets and variables*

The following sets and variables are specified in this Recommendation.

A.5.1 *Functional units*

The set of all functional units specified in this Recommendation is defined as:

fu-dom = {FD, HD, EXCEP, TD, NR, SY, MA, RESYN, EX, ACT, CD}

where

FD = Duplex functional unit

HD = Half-duplex functional unit

EXCEP = Exceptions functional unit

TD = Typed data functional unit

NR = Negotiated release functional unit

SY = Minor synchronize functional unit

MA = Major synchronize functional unit

RESYN = Resynchronize functional unit

EX = Expedited data functional unit

ACT = Activity management functional unit

CD = Capability data functional unit

A Boolean function FU is defined over fu-dom as follows:

for f in fu-dom

FU(f) = true: if and only if the functional unit f has been selected during the session connection establishment phase.

The value is set when the ACCEPT SPDU is sent or received.

A.5.2 Tokens

The set of all tokens specified in this Recommendation is defined as:

tk-dom = {mi, ma, tr, dk}

where

mi = synchronize-minor token

ma = major/activity token

tr = release token

dk = data token

The following Boolean functions are defined over tk-dom:

a) AV(t), for t in tk-dom, is a function which defines the availability of the corresponding token and has the following values:

AV(mi) = FU(SY)

AV(dk) = FU(HD)

AV(tr) = FU(NR)

AV(ma) = FU(MA) or FU(ACT)

b) OWNED(t), for t in tk-dom, is a function which defines the assignment of the corresponding token and is defined as:

OWNED(t) = true: if token assigned to the SPM

OWNED(t) = false: if token not assigned to the SPM

OWNED(t) is not defined if AV(t) = false. OWNED(t) is set when:

1) the ACCEPT SPDU is sent or received; or

2) the RESYNCHRONIZE ACK SPDU is sent or received; or

3) the GIVE TOKENS SPDU is sent or received; or

4) the GIVE TOKENS CONFIRM SPDU is sent or received;

5) the ACTIVITY INTERRUPT ACK SPDU is sent or received;

6) the ACTIVITY DISCARD ACK SPDU is sent or received.

c) I(t), for t in tk-dom, is a function which, when true, indicates that the SPM has *I*nitiating rights for the behaviour controlled by the token. This applies even if the corresponding token is not available:

I(t) = ”AV(t) OR OWNED(t)

d) A(t), for t in tk-dom, is a function which, when true, indicates that the SPM has *A*ccepting rights for the behaviour controlled by the token. This applies even if the corresponding token is not available:

A(t) = ”AV(t) OR ”OWNED(t)

e) II(t), for t in tk-dom, is a function which, when true, indicates that the SPM has *I*nitiating rights as I(t), but this applies to the case when the behaviour may only be initiated if the corresponding token is available and owned:

II(t) = AV(t) AND OWNED(t)

f) AA(t), for t in tk-dom, is a function which, when true, indicates that the SPM has *A*ccepting rights as A(t), but only if the corresponding token is available, but not owned:

AA(t) = AV(t) AND ”OWNED(t)

A.5.3 *SET of tokens*

The following subsets of tk-dom are defined:

RT = {tokens requested in the input event}

GT = {tokens given in the input event}

For the purpose of the following function definitions, two further set are defined:

F = {AV, OWNED, I, A, II, AA} (the set of functions defined in ¤ A.5.2);

S = the set of subsets of tk-dom.

The following functions are defined over F and S:

a) ALL(f, s) for f in F and s in S:

ALL(f, s) = true: all of the f(t) for t in s are true or s is empty.

For example:

ALL(A, tk-dom) = true: none of the available tokens are owned (e.g. on receipt of a FINISH SPDU).

b) ANY(f, s), for f in F and s in S:

ANY(f, s) = true: any f(t) = true for t in s when s is not empty.

For example:

ANY(II, tk-dom) = true: at least one of the available tokens is owned.

A.5.4 *Variables*

A.5.4.1 *TEXP*

TEXP is a Boolean variable having the following values:

TEXP = true: use of transport expedited service is selected for use on this session connection.

TEXP = false: use of transport expedited service is not selected for use on this session connection.

A.5.4.2 *Vact*

Vact is a Boolean variable having the following values when the activity management functional unit has been selected (FU(ACT) = true):

Vact = true: an activity is in progress;

Vact = false: no activity is in progress;

Vact has no defined value if FU(ACT) = false.

Vact is set as follows:

a) Vact is set false during the connection establishment phase, if the activity management functional unit has been selected (FU(ACT) = true). Otherwise, Vact is not set;

b) Vact is set true when the ACTIVITY START SPDU or ACTIVITY RESUME SPDU is sent or received (only possible when FU(ACT) = true);

c) Vact is set false when the ACTIVITY DISCARD ACK SPDU or ACTIVITY INTERRUPT ACK SPDU is sent or received;

d) Vact is set to Vnextact when a MAJOR SYNC ACK SPDU or an ACTIVITY END ACK SPDU is sent or received.

A.5.4.3 *Vnextact*

Vnextact is a Boolean variable which is used when the activity management functional unit has been selected (FU(ACT) = true). It is used to indicate the next value of Vact when a MAJOR SYNC ACK SPDU or an ACTIVITY END ACK SPDU is sent or received. Vnextact is set when a MAJOR SYNC POINT SPDU or an ACTIVITY END SPDU is sent or received:

a) Vnextact is set false if FU(ACT) = true and an ACTIVITY END SPDU is sent or received;

b) Vnextact is set true if FU(ACT) = true and a MAJOR SYNC POINT SPDU is sent or received.

Vnextact has not defined value if FU(ACT) = false.

A.5.4.4 *Vrsp and Vrspnb*

These variables are used to resolve resynchronization collisions.

Vrsp indicates what kind of resynchronization is currently in progress:

Vrsp = no no resynchronization in progress

Vrsp = a resynchronize abandon

Vrsp = r resynchronize restart

Vrsp = s resynchronize set

Vrsp = dsc discard activity

Vrsp = int interrupt activity

Vrspnb indicates the serial number in the case of resynchronize restart.

Vrsp and, if necessary Vrspnb, are set when a RESYNCHRONIZE SPDU, ACTIVITY INTERRUPT SPDU or an ACTIVITY DISCARD SPDU is sent or received. Vrsp is set to no when the SPM goes to STA 713.

A.5.4.5 *SPMwinner*

SPMwinner is a Boolean function which is used during resynchronization collision, that is when:

a) a RESYNCHRONIZE SPDU is received and Vrsp is not equal to no;

b) an S-RESYNCHRONIZE request is received and Vrsp is not equal to no.

The SPMwinner condition is true if the SPM (which holds the current resynchronization) wins against the new colliding event.

The SPMwinner condition is calculated as follows:

a) the next Vrsp and Vrspnb values are evaluated according to the parameters of the received event. The new calculated value for Vrsp is compared to the current Vrsp with the following ordering rule:

dsc prevails over int

int prevails over a

a prevails over s

s prevails over r

If both are equal to r, then the new calculated value for Vrspnb is compared to the current value of Vrspnb and the lower value prevails;

b) If the current value of Vrsp (and Vrspnb if necessary) prevails, then the SPMwinner condition is true (in this case, the current resynchronization prevails over the colliding one);

c) If the current value of Vrsp (and Vrspnb if necessary) does not prevail, then the SPMwinner condition is false (in this case, the colliding resynchronization prevails over the current one);

d) If the above comparison results in the equality and if the colliding event has been generated by the initiator of the session connection (either a RESYNCHRONIZE SPDU was received from the session connection initiator or a local S-RESYNCHRONIZE request was issued by the session connection initiator), then the SPMwinner condition is false.

If the SPM is winner (SPMwinner condition is true) then the current resynchronization wins against the colliding one and Vrsp and Vrspnb are not updated.

If the SPM is not winner (SPMwinner condition is false) then the colliding resynchronization is taken into account and Vrsp and Vrspnb are updated.

A.5.4.6 *Vtca*

Vtca is a Boolean variable having the following values:

Vtca = false: the SPM initiated the T-CONNECT request (transport connection initiator);

Vtca = true: the SPM received the T-CONNECT indication (transport connection acceptor).

A.5.4.7 *Vtrr*

Vtrr is a Boolean variable having the following values:

Vtrr = true: the transport connection can be reused by the SPM for another session connection;

Vtrr = false: the transport connection cannot be reused by the SPM for another session connection.

A.5.4.8 *Vcoll*

Vcoll is a Boolean variable having the following values:

Vcoll = true: a collision of FINISH SPDUs has been detected;

Vcoll = false: there has not been a collision of FINISH SPDUs.

This variable is set false during the session connection establishment phase.

A.5.4.9 *Vdnr*

Vdnr is a Boolean variable having the following values:

Vdnr = true: a DISCONNECT SPDU has been received in STA09 (following a collision of FINISH SPDUs);

Vdnr = false: no DISCONNECT SPDU has been received.

This variable is set to false during the session connection establishment phase.

A.5.4.10 *V(A)*

V(A) is used by the SPM and is the lowest serial number to which a synchronization point confirmation is expected. No confirmation is expected when V(A) = V(M).

A.5.4.11 *V(M)*

V(M) is used by the SPM and is the next serial number to be used.

A.5.4.12 *V(R)*

V(R) is used by the SPM and is the lowest serial number to which resynchronization restart is permitted.

A.5.4.13 *Vsc*

Vsc is a Boolean variable having the following values:

Vsc = true: the SS-user has the right to issue minor synchronization point responses when V(A) is less than V(M);

Vsc = false: the SS-user does not have the right to issue minor synchronization point responses.

Vsc is set false during the connection establishment phase and when a MINOR SYNC POINT SPDU is sent. Vcs is set true when a MINOR SYNC POINT SPDU is received.

Note - Table A-4/X.225 summarizes the operations on V(A), V(M), V(R) and Vsc.

TABLE A-1/X.225

Incoming Events

Abbreviated name	Category	Name and description
SACTDreq	SS-user	S-ACTIVITY-DISCARD request primitive
SACTDrsp	SS-user	S-ACTIVITY-DISCARD response primitive
SACTEreq	SS-user	S-ACTIVITY-END request primitive
SACTErsp	SS-user	S-ACTIVITY-END response primitive
SACTIreq	SS-user	S-ACTIVITY-INTERRUPT request primitive
SACTIrsp	SS-user	S-ACTIVITY-INTERRUPT response primitive
SACTRreq	SS-user	S-ACTIVITY-RESUME request primitive
SACTSreq	SS-user	S-ACTIVITY-START request primitive
SCDreq	SS-user	S-CAPABILITY-DATA request primitive
SCDrsp	SS-user	S-CAPABILITY-DATA response primitive
SCGreq	SS-user	S-CONTROL-GIVE request primitive
SCONreq	SS-user	S-CONNECT request primitive
SCONrsp+	SS-user	S-CONNECT (accept) response primitive
SCONrsp-	SS-user	S-CONNECT (reject) response primitive
SDTreq	SS-user	S-DATA request primitive
SEXreq	SS-user	S-EXPEDITED-DATA request primitive
SGTreq	SS-user	S-TOKEN-GIVE request primitive
SPTreq	SS-user	S-TOKEN-PLEASE request primitive
SRELreq	SS-user	S-RELEASE request primitive
SRELrsp+	SS-user	S-RELEASE (accept) response primitive
SRELrsp-	SS-user	S-RELEASE (reject) response primitive
SRSYNreq(a)	SS-user	S-RESYNCHRONIZE (abandon) request primitive
SRSYNreq(r)	SS-user	S-RESYNCHRONIZE (restart) request primitive
SRSYNreq(s)	SS-user	S-RESYNCHRONIZE (set) request primitive
SRSYNrsp	SS-user	S-RESYNCHRONIZE response primitive
SSYNMreq	SS-user	S-SYNC-MAJOR request primitive
SSYNMrsp	SS-user	S-SYNC-MAJOR response primitive
SSYNmreq	SS-user	S-SYNC-MINOR request primitive

TABLE A-1/X.225 *(cont.)*

Abbreviated name	Category	Name and description
SSYNmrsp	SS-user	S-SYNC-MINOR response primitive
STDreq	SS-user	S-TYPED-DATA request primitive
SUABreq	SS-user	S-U-ABORT request primitive
SUERreq	SS-user	S-U-EXCEPTION-REPORT request primitive
TCONind	TS-provider	T-CONNECT indication primitive
TCONcnf	TS-provider	T-CONNECT confirm primitive
TDISind	TS-provider	T-DISCONNECT indication primitive
TIM	Timer	Time out
AA	SPDU	ABORT ACCEPT SPDU
AB-nr	SPDU	ABORT (not reuse) SPDU
AB-r	SPDU	ABORT (reuse) SPDU
AC	SPDU	ACCEPT SPDU (see Note 1)
AD	SPDU	ACTIVITY DISCARD SPDU
ADA	SPDU	ACTIVITY DISCARD ACK SPDU
AE	SPDU	ACTIVITY END SPDU
AEA	SPDU	ACTIVITY END ACK SPDU
AI	SPDU	ACTIVITY INTERRUPT SPDU
AIA	SPDU	ACTIVITY INTERRUPT ACK SPDU
AR	SPDU	ACTIVITY RESUME SPDU
AS	SPDU	ACTIVITY START SPDU
CD	SPDU	CAPABILITY DATA SPDU
CDA	SPDU	CAPABILITY DATA ACK SPDU
CDO	SPDU	CONNECT DATA OVERFLOW SPDU
CN	SPDU	CONNECT SPDU
DN	SPDU	DISCONNECT SPDU
DT	SPDU	DATA TRANSFER SPDU
ED	SPDU	EXCEPTION DATA SPDU
ER	SPDU	EXCEPTION REPORT SPDU
EX	SPDU	EXPEDITED DATA SPDU
FN-nr	SPDU	FINISH (not reuse) SPDU
FN-r	SPDU	FINISH (reuse) SPDU
GT	SPDU	GIVE TOKENS SPDU with Token Item parameter (see Note 2)
GTA	SPDU	GIVE TOKENS ACK SPDU
GTC	SPDU	GIVE TOKENS CONFIRM SPDU
MAA	SPDU	MAJOR SYNC ACK SPDU
MAP	SPDU	MAJOR SYNC POINT SPDU
MIA	SPDU	MINOR SYNC ACK SPDU

TABLE A-1/X.225 *(end)*

Abbreviated name	Category	Name and description
MIP	SPDU	MINOR SYNC POINT SPDU
NF	SPDU	NOT FINISHED SPDU
OA	SPDU	OVERFLOW ACCEPT SPDU
PR-AB	SPDU	PREPARE (ABORT) SPDU
PR-MAA	SPDU	PREPARE (MAJOR SYNC ACK) SPDU
PR-RA	SPDU	PREPARE (RESYNCHRONIZE ACK) SPDU
PR-RS	SPDU	PREPARE (RESYNCHRONIZE) SPDU
PT	SPDU	PLEASE TOKENS SPDU with Token Item parameter (see Notes 1 and 2)
RA	SPDU	RESYNCHRONIZE ACK SPDU
RF-nr	SPDU	REFUSE (not reuse) SPDU
RF-r	SPDU	REFUSE (reuse) SPDU
RS-a	SPDU	RESYNCHRONIZE (abandon) SPDU
RS-r	SPDU	RESYNCHRONIZE (restart) SPDU
RS-s	SPDU	RESYNCHRONIZE (set) SPDU
TD	SPDU	TYPED DATA SPDU

Note 1 - If the Token Item parameter is present in the ACCEPT SPDU, both the AC event and the PT event occur.

Note 2 - GIVE TOKENS SPDU without Token Item parameter and PLEASE TOKENS SPDU without Token Item parameter are used to herald a concatenated sequence of SPDUs Concatenation of SPDUs and separation of TSDUs are not handled by the state tables.

TABLE A-2/X.225

States

Abbreviated name	Name and description
STA 01	Idle, no transport connection
STA 01A	Wait for the ABORT ACCEPT SPDU
STA 01B	Wait for T-CONNECT confirm
STA 01C	Idle, transport connected
STA 01D	Wait for the CONNECT DATA OVERFLOW SPDU
STA 02A	Wait for the ACCEPT SPDU
STA 02B	Wait for the OVERFLOW ACCEPT SPDU
STA 03	Wait for the DISCONNECT SPDU
STA 04A	Wait for the MAJOR SYNC ACK SPDU or PREPARE (MAJOR SYNC ACK) SPDU
STA 04B	Wait for the ACTIVITY END ACK SPDU or PREPARE (MAJOR SYNC ACK) SPDU
STA 05A	Wait for the RESYNCHRONIZE ACK SPDU or PREPARE (RESYNCHRONIZE ACK) SPDU
STA 05B	Wait for the ACTIVITY INTERRUPT ACK SPDU or PREPARE (RESYNCHRONIZE ACK) SPDU
STA 05C	Wait for the ACTIVITY DISCARD ACK SPDU or PREPARE (RESYNCHRONIZE ACK) SPDU
STA 06	Wait for the RESYNCHRONIZE SPDU [resynchronization collision after receiving PREPARE (RESYNCHRONIZE) SPDU]
STA 08	Wait for S-CONNECT response
STA 09	Wait for S-RELEASE response
STA 10A	Wait for S-SYNC-MAJOR response
STA 10B	Wait for S-ACTIVITY-END response
STA 11A	Wait for S-RESYNCHRONIZE response
STA 11B	Wait for S-ACTIVITY-INTERRUPT response
STA 11C	Wait for S-ACTIVITY-DISCARD response
STA 15A	After PREPARE, wait for the MAJOR SYNC ACK SPDU or the ACTIVITY END ACK SPDU
STA 15B	After PREPARE, wait for the RESYNCHRONIZE SPDU or the ACTIVITY INTERRUPT SPDU or the ACTIVITY DISCARD SPDU
STA 15C	After PREPARE, wait for the RESYNCHRONIZE ACK SPDU or the ACTIVITY INTERRUPT ACK SPDU or the ACTIVITY DISCARD ACK SPDU
STA 15D	After PREPARE, wait for the ABORT SPDU
STA 16	Wait for T-DISCONNECT indication
STA 18	Wait for the GIVE TOKENS ACK SPDU
STA 19	Wait for a recovery request or SPDU (initiator of EXCEPTION DATA SPDU)
STA 20	Wait for a recovery SPDU or request
STA 21	Wait for the CAPABILITY DATA ACK SPDU
STA 22	Wait for S-CAPABILITY-DATA response
STA 713	Data transfer state

TABLE A-3/X.225

Outgoing Events

Abbreviated name	Category	Name and description
SACTDind	SS-provider	S-ACTIVITY-DISCARD indication primitive
SACTDcnf	SS-provider	S-ACTIVITY-DISCARD confirm primitive
SACTEind	SS-provider	S-ACTIVITY-END indication primitive
SACTEcnf	SS-provider	S-ACTIVITY-END confirm primitive
SACTIind	SS-provider	S-ACTIVITY-INTERRUPT indication primitive
SACTIcnf	SS-provider	S-ACTIVITY-INTERRUPT confirm primitive
SACTRind	SS-provider	S-ACTIVITY-RESUME indication primitive
SACTSind	SS-provider	S-ACTIVITY-START indication primitive
SCDind	SS-provider	S-CAPABILITY-DATA indication primitive
SCDcnf	SS-provider	S-CAPABILITY-DATA confirm primitive
SCGind	SS-provider	S-CONTROL-GIVE indication primitive
SCONind	SS-provider	S-CONNECT indication primitive
SCONcnf+	SS-provider	S-CONNECT (accept) confirm primitive
SCONcnf-	SS-provider	S-CONNECT (reject) confirm primitive
SDTind	SS-provider	S-DATA indication primitive
SEXind	SS-provider	S-EXPEDITED-DATA indication primitive
SGTind	SS-provider	S-TOKEN-GIVE indication primitive
SPABind	SS-provider	S-P-ABORT indication primitive
SPERind	SS-provider	S-P-EXCEPTION-REPORT indication primitive
SPTind	SS-provider	S-TOKEN-PLEASE indication primitive
SRELind	SS-provider	S-RELEASE indication primitive
SRELcnf+	SS-provider	S-RELEASE (accept) confirm primitive
SRELcnf-	SS-provider	S-RELEASE (reject) confirm primitive
SRSYNind	SS-provider	S-RESYNCHRONIZE indication primitive
SRSYNcnf	SS-provider	S-RESYNCHRONIZE confirm primitive
SSYNMind	SS-provider	S-SYNC-MAJOR indication primitive
SSYNMcnf	SS-provider	S-SYNC-MAJOR confirm primitive
SSYNmind	SS-provider	S-SYNC-MINOR indication primitive
SSYNmcnf	SS-provider	S-SYNC-MINOR confirm primitive

TABLEAU A-3/X.225 *(cont.)*

Abbreviated name	Category	Name and description
STDind	SS-provider	S-TYPED-DATA indication primitive
SUABind	SS-provider	S-U-ABORT indication primitive
SUERind	SS-provider	S-U-EXCEPTION-REPORT indication primitive
TCONreq	TS-user	T-CONNECT request primitive
TCONrsp	TS-user	T-CONNECT response primitive
TDISreq	TS-user	T-DISCONNECT request primitive
AA	SPDU	ABORT ACCEPT SPDU
AB-nr	SPDU	ABORT (not reuse) SPDU
AB-r	SPDU	ABORT (reuse) SPDU
AC	SPDU	ACCEPT SPDU
AD	SPDU	ACTIVITY DISCARD SPDU
ADA	SPDU	ACTIVITY DISCARD ACK SPDU
AE	SPDU	ACTIVITY END SPDU
AEA	SPDU	ACTIVITY END ACK SPDU
AI	SPDU	ACTIVITY INTERRUPT SPDU
AIA	SPDU	ACTIVITY INTERRUPT ACK SPDU
AR	SPDU	ACTIVITY RESUME SPDU
AS	SPDU	ACTIVITY START SPDU
CD	SPDU	CAPABILITY DATA SPDU
CDA	SPDU	CAPABILITY DATA ACK SPDU
CDO	SPDU	CONNECT DATA OVERFLOW SPDU
CN	SPDU	CONNECT SPDU
DN	SPDU	DISCONNECT SPDU
DT	SPDU	DATA TRANSFERT SPDU
ED	SPDU	EXCEPTION DATA SPDU
EX	SPDU	EXPEDITED DATA SPDU
FN-nr	SPDU	FINISH (not reuse) SPDU
FN-r	SPDU	FINISH (reuse) SPDU
GT	SPDU	GIVE TOKENS SPDU
GTA	SPDU	GIVE TOKENS ACK SPDU
GTC	SPDU	GIVE TOKENS CONFIRM SPDU
MAA	SPDU	MAJOR SYNC ACK SPDU
MAP	SPDU	MAJOR SYNC POINT SPDU
MIA	SPDU	MINOR SYNC ACK SPDU
MIP	SPDU	MINOR SYNC POINT SPDU
NF	SPDU	NOT FINISHED SPDU
OA	SPDU	OVERFLOW ACCEPT SPDU
PR-MAA	SPDU	PREPARE (MAJOR SYNC ACK) SPDU
PR-AB	SPDU	PREPARE (ABORT) SPDU

TABLE A-3/X.225 *(end)*

Abbreviated name	Category	Name and description
PR-RA	SPDU	PREPARE (RESYNCHRONIZE ACK) SPDU
PR-RS	SPDU	PREPARE (RESYNCHRONIZE) SPDU
PT	SPDU	PLEASE TOKENS SPDU
RA	SPDU	RESYNCHRONIZE ACK SPDU
RF-nr	SPDU	REFUSE (not reuse) SPDU
RF-r	SPDU	REFUSE (reuse) SPDU
RS-a	SPDU	RESYNCHRONIZE (abandon) SPDU
RS-r	SPDU	RESYNCHRONIZE (restart) SPDU
RS-s	SPDU	RESYNCHRONIZE (set) SPDU
TD	SPDU	TYPED DATA SPDU

TABLE A-4/X.225

TABLE A-5/X.225

Specific actions

[1]	Set Vtca = true
[2]	Set Vtca = false
[3]	Stop timer TIM
[4]	Start timer TIM
[5]	Set V(A) = V(M) = serial number in ACCEPT SPDU
	Set V(R) = 0
	Set Vcoll = false
	Set Vrsp = no
	Set Vsc = false
	Set TEXP.
	Set FU(f) for f in fu-dom according to the intersection of Session User Requirements in the CONNECT SPDU and Session User Requirements in the ACCEPT SPDU
	If FU(ACT) = true, set Vact = false
	Set Vdnr = false
[6]	Recall the queued events until the queue is empty
[7]	Set Vtrr = true
[8]	Set Vtrr = false
[9]	Set Vtrr according to the Transport Disconnect PV field in the SPDU. As a local decision, Vtrr may always be set false
[10]	Store the event in the queue
[11]	Update the position of the tokens
[12]	Set Vact = true
[13]	Set Vnextact
[14]	Set Vact = Vnextact
[15]	Clear the queue
[16]	Update Vrsp and, if RS-r, Vrspnb
[17]	Not used
[18]	Set Vcoll = true
[19]	V(M) = maximum [V(M), received serial number]
[20]	Set Vsc = false
[21]	Set V(M) = V(M) + 1
[22]	Set V(R) = V(A) = V(M)
[23]	If Vsc = false, set V(A) = V(M). Set Vsc = true
	Set V(M) = V(M) + 1
[24]	If Vsc = true, set V(A) = V(M). Set Vsc = false
	Set V(M) = V(M) + 1
[25]	Set V(A) = serial number + 1
[26]	Set V(A) = V(M) = V(R) = 1
[27]	Set V(A) = V(M) = serial number + 1
	Set V(R) = 1
[28]	Set V(A) = V(M) = serial number
	If Vrsp = a, then set V(R) = 0
	If Vrsp = s, then set V(R) = 0
	Set Vrsp = no
[29]	Set the position of the tokens such that all available tokens are owned. Set Vact = false
	Set Vrsp = no
[30]	Set the position of the tokens such that all available tokens are not owned. Set Vact = false
	Set Vrsp = no
[31]	If Vsc = false, set V(A) = V(M)
	Set V(M) = V(M) + 1
[32]	Set Vdnr = true
[50]	Preserve user data for subsequent SCONind
[51]	If p201 send subsequent CDO SPDUs until ”p201

TABLE A-6/X.225

Predicates

p01	”Vtca
p02	local choice & ”TEXP
p03	I(dk)
p04	FU(FD) & ”Vcoll
p05	A(dk)
p06	FU(TD)
p07	FU(TD) & ”Vcoll
p08	FU(EX)
p09	FU(EX) & ”Vcoll
p10	”Vcoll
p11	II(ma)
p12	(”FU(ACT) OR Vact) & A(dk) & A(mi) & AA(ma)
p13	(”FU(ACT) OR Vact) & I(dk) & I(mi) & II(ma)
p14	(”FU(ACT) OR Vact) & A(dk) & AA(mi)
p15	(”FU(ACT) OR Vact) & I(dk) & II(mi)
p16	”TEXP
p17	(”FU(ACT) OR Vact) & FU(SY) & ”Vsc
p18	(”FU(ACT) OR Vact) & FU(SY) & Vsc
p19	serial number = V(M)
p20	serial number = V(M) - 1
p21	V(M) > serial number >= V(A)
p22	Unused
p23	FU(ACT) & ”Vnextact
p24	”SPMwinner
p25	(FU(SY) OR FU(MA)) & FU(RESYN)
p26	(”FU(ACT) OR Vact)
p27	Vrsp = no
p28	FU(RESYN)
p29	(”FU(ACT) OR Vact) & FU(RESYN)
p30	(”FU(ACT) OR Vnextact)
p31	FU(ACT) & Vnextact
p32	serial number >= V(R)
p33	V(M) >= serial number >= V(R)
p34	FU(ACT)
p35	FU(RESYN) & ”TEXP
p36	FU(RESYN) & TEXP
p37	FU(ACT) & TEXP
p38	FU(ACT) & ”TEXP
p39	Vact & II(ma)
p40	AA(ma)
p41	Vrsp = dsc
p42	Vrsp = int
p43	((Vrsp = r) & (serial number = Vrspnb)) OR ((Vrsp = a) & (serial number = V(M))) OR (Vrsp = s)
p44	(FU(ACT) & ”Vact) & A(dk) & A(mi) & A(ma)
p45	(FU(ACT) & ”Vact) & I(dk) & I(mi) & I(ma)
p46	FU(CD) & (FU(ACT) & ”Vact) & A(dk) & A(mi) & ”OWNED(ma)
p47	FU(CD) & (FU(ACT) & ”Vact) & I(dk) & I(mi) & OWNED(ma)
p48	FU(EXCEP) & FU(HD)
p49	((Vrsp = r) & (serial number = Vrspnb)) OR ((Vrsp = a) & (serial number >= V(M))) OR (Vrsp = s)
p50	FU(EXCEP) & (”FU(ACT) OR Vact) & AA(dk)
p51	FU(EXCEP) & (”FU(ACT) OR Vact) & II(dk)
p52	FU(EXCEP) & ”FU(ACT) & II(dk)
p53	ANY(AV, RT)

TABLE A-6/X.225 *(cont.)*

p54	ALL(I, GT) & ANY (AV, GT)
p55	(FU(ACT) & ”Vact) & ALL(I, tk-dom)
p56	Unused
p57	ALL(I, GT) & (dk not in GT) & ANY (AV, GT)
p58	ALL(I, GT) & (dk in GT)
p59	ALL(A, GT) & ANY (AV, GT)
p60	ALL(A, GT) & (dk not in GT) & ANY (AV, GT)
p61	ALL(A, GT) & (dk in GT)
p62	(FU(ACT) & ”Vact) & ALL(A, tk-dom)
p63	ALL(I, tk-dom) & (”FU(ACT) OR ”Vact)
p64	local choice & ”Vtca & ”TEXP
p65	ANY(AV, tk-dom)
p66	Vtrr
p67	FU(NR)
p68	ALL (A, tk-dom) & (”FU(ACT) OR ”Vact)
p69	Vcoll
p70	FU(FD)
p71	FU(ACT) & Vact & I(dk) & I(mi) & II(ma)
p72	FU(ACT) & Vact & A(dk) & A(mi) & AA(ma)
p75	(Vcoll & Vdnr) OR ”Vcoll
p76	CN SPDU is not acceptable to the SPM for transient or persistent reason (see ¤ 8.3.5.9)
p201	More user data to send
p202	End of user data
p204	More than 10 240 octets of SS-user data to be transferred

A.6 *Notes to Tables A-7/X.225 to A-15/X.225:*

Note 1 - PR is not sent if TEXP is false.

Note 2 - The serial number given in the indication is V(M).

Note 3 - SxABind means generate event SUABind if bit 2 of the Transport Disconnect PV field in the ABORT SPDU has the value ÒuserÓ. Otherwise, SxABind means generate the event SPABind.

Note 4 - PR-AB is only sent if TEXP is true and the SS-user data exceeds 9 octets (¤ 7.9.2).

TABLE A-7/X.225

TABLE A-7/X.225 (*cont*)

TABLE A-8/X.225

Data transfer state table

STATE EVENT	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA02A await AC	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA
DT	STA01A	TDISreq STA01	TDISreq STA01		p05&p10 SDTind STA03	p05 SDTind STA04A	p05 SDTind STA04B	p05 STA05A	p05 STA05B
EX	STA01A	TDISreq STA01	TDISreq STA01	[10] STA02A	p09 SEXind STA03	p08 SEXind STA04A	p08 SEXind STA04B	p08 STA05A	p08 STA05B
TD	STA01A	TDISreq STA01	TDISreq STA01		p06&p10 STDind STA03	p06 STDind STA04A	p06 STDind STA04B	p06 STA05A	p06 STA05B
SDTreq									
SEXreq									
STDreq									

TABLE A-8/X.225 *(cont.)*

Data transfer state table

STATE EVENT	STA05C await PR or ADA	STA06 await RS after coll	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp	STA15A wait after PR- MAA	STA15B wait after PR-RS
DT	p05 STA05C	p05 STA06				p05 SDTind STA15A	p05 STA15B
EX	p08 STA05C	p08 [10] STA06				p08 [10] STA15A	
TD	p06 STA05C	p06 STA06				p06 STDind STA15A	p06 STA15B
SDTreq			p04 DT STA09	p03 DT STA10A	p03 DT STA10B		p03 STA15B
SEXreq			p09 EX STA09	p08 EX STA10A	p08 EX STA10B		p08 STA15B
STDreq			p07 TD STA09	p06 TD STA10A	p06 TD STA10B		p06 STA15B

TABLE A-8/X.225 *(end)*

Data transfer state table

STATE EVENT	STA15C wait after PR-RA	STA15C wait after PR-AB	STA16 await TDISind	STA18 await GTA	STA19 await recovery (init)	STA20 await recovery	STA21 await CDA	STA713 data transfer
DT	p05 STA15C	STA15D	STA16	p70 SDTind STA18	STA19	p05 STA20	p70 SDTind STA21	p05 SDTind STA713
EX	p08 [10] STA15C		STA16	p08 SEXind STA18	p08 STA19	p08 STA20	p08 SEXind STA21	p08 SEXind STA713
TD	p06 STA15C	STA15D	STA16	p06 STDind STA18	p06 STA19	p06 STA20	p06 STDind STA21	p06 STDind STA713
SDTreq		STA15D		p70 DT STA18				p03 DT STA713
SEXreq		STA15D		p08 EX STA18				p08 EX STA713
STDreq		STA15D		p06 TD STA18				p06 TD STA713

TABLE A-9/X.225

Synchronization state table

STATE EVENT	STA01A await AA	STA01C idle CT con	STA01D await CDO	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA	STA05C await PR or ADA
MAA or AEA	STA01A	TDISreq STA01	TDISreq STA01	p16&p20 SSYNMcnf [14] [22] STA713	p16&p20 SACTEcnf [14] [22] STA713	STA05A	STA05B	STA05C
MAP	STA01A	TDISreq STA01	TDISreq STA01			p12 STA05A		
PR-MAA	STA01A	TDISreq STA01	TDISreq STA01	STA15A	STA15A	STA05A	STA05B	STA05C
SSYNMreq								
SSYNMrsp								

TABLE A-9/X.225 *(cont.)*

Synchronization state table

STATE EVENT	STA06 await RS after coll	STA10A await SSYNMrsp	STA15A wait after PR-MAA	STA15B wait after PR-RS	STA15C wait after PR-RA	STA15C wait after PR-AB
MAA or AEA	STA06		p20&”p23 SSYNMcnf [14] [22] STA713 [6] p20&p23 SACTEcnf [14] [22] STA713 [6]	STA15B	STA15C	STA15D
MAP	p12 STA06			p12 STA15B	p12 STA15C	STA15D
PR-MAA						
SSYNMreq				p13 STA15B		STA15D
SSYNMrsp		PR-MAA (1) MAA [14] [22] STA713		STA15B		STA15D

TABLE A-9/X.225 *(cont.)*

Synchronization state table

STATE EVENT	STA16 await TDISind	STA19 await recovery (init)	STA20 await recovery	STA713 data transfer
MAA or AEA	STA16		p20 STA20	
MAP	STA16	p12&p19 [31] STA19	p12&p19 [31] STA20	p12&p19 SSYNMind [13] [31] STA10A
PR-MAA	STA16			
SSYNMreq				p13 MAP [13] [24] STA04A
SSYNMrsp				

TABLE A-9/X.225 *(cont.)*

Synchronization state table

STATE EVENT	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA
AE	STA01A	TDISreq STA01	TDISreq STA01				p72 STA05A
MIA	STA01A	TDISreq STA01	TDISreq STA01	p17&p21 SSYNmcnf [25] STA03	p17&”p20&p21 SSYNmcnf [25] STA04A	p17&”p20&p21 SSYNmcnf [25] STA04B	p17 STA05A
MIP	STA01A	TDISreq STA01	TDISreq STA01				p14 STA05A
SACTEreq							
SACTErsp							
SSYNmreq							
SSYNmrsp							

TABLE A-9/X.225 *(cont.)*

Synchronization state table

STATE EVENT	STA05B await PR or AIA	STA05C await PR or ADA	STA06 await RS after coll	STA09 await SRELrsp	STA10A await SSYNMrsp
AE			p72 STA06		
MIA	p17 STA05B	p17 STA05C	p17 STA06		
MIP	p14 STA05B	p14 STA05C	p14 STA06		
SACTEreq					
SACTErsp					
SSYNmreq					
SSYNmrsp				p18&p21 MIA [25] STA09	p18&”p20&p21 MIA [25] STA10A

TABLE A-9/X.225 *(cont.)*

Synchronization state table

STATE EVENT	STA10B await SACTErsp	STA15A wait after PR-MAA	STA15B wait after PR-RS	STA15C wait after PR-RA	STA15C wait after PR-AB	STA16 await TDISind
AE			p72 STA15B	p72 STA15C	STA15D	STA16
MIA		p17&”p20&p21 SSYNmcnf [25] STA15A	p17 STA15B	p17 STA15C	STA15D	STA16
MIP			p14 STA15B	p14 STA15C	STA15D	STA16
SACTEreq			p71 STA15B		STA15D	
SACTErsp	PR-MAA (1) AEA [14] [22] STA713				STA15D	
SSYNmreq			p15 STA15B		STA15D	
SSYNmrsp	p18&”p20&p21 MIA [25] STA10B		p18&p21 STA15B		STA15D	

TABLE A-9/X.225 *(end)*

Synchronization state table

STATE EVENT	STA19 await recovery (init)	STA20 await recovery	STA713 data transfer
AE	p72&p19 [31] STA19	p72&p19 [31] STA20	p72&p19 SACTEind [13] [31] STA10B
MIA	p17&p21 [25] STA19	p17&p21 STA20	p17&p21 SSYNmcnf [25] STA713
MIP	p14&p19 [23] STA19	p14&p19 [23] STA20	p14&p19 SSYNmind [23] STA713
SACTEreq			p71 AE [13] [24] STA04B
SACTErsp			
SSYNmreq			p15 MIP [24] STA713
SSYNmrsp			p18&p21 MIA [25] STA713

TABLE A-10/X.225

Resynchronization state table

STATE EVENT	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA02A await AC	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA
PR-RA	STA01A	TDISreq STA01	TDISreq STA01				
PR-RS	STA01A	TDISreq STA01	TDISreq STA01	[10] STA02A	p10 STA15B	STA15B	STA15B
RA	STA01A	TDISreq STA01	TDISreq STA01				
RS-a	STA01A	TDISreq STA01	TDISreq STA01		p10&”p 34&p 35 [19] SRSYNind(2) [16] STA11A	p35 [19] SRSYNind(2) [16] STA11A	p35 [19] SRSYNind(2) [16] STA11A
RS-r	STA01A	TDISreq STA01	TDISreq STA01		p10&”p 34& p 35 &p32 SRSYNind [16] STA11A	p32&p35 SRSYNind [16] STA11A	p32&p35 SRSYNind [16] STA11A
RS-s	STA01A	TDISreq STA01	TDISreq STA01		p10&”p34 &p35 SRSYNind [16] STA11A	p35 SRSYNind [16] STA11A	p35 SRSYNind [16] STA11A
SRSYNreq(a)						p28 PR-RS(1) RS-a [16] STA05A	
SRSYNreq(r)							
SRSYNreq(s)						p28 PR-RS(1) RS-s [16] STA05A	
SRSYNrsp							

TABLE A-10/X.225 *(cont.)*

Resynchronization state table

STATE EVENT	STA05A await PR or RA	STA05B await PR or AIA	STA05C await PR or ADA	STA06 await RS after coll	STA09 await SRELrsp
PR-RA	STA15C	STA15C	STA15C	[10] STA06	
PR-RS	STA06	STA05B	STA05C	[10] STA06	
RA	p35&p49 SRSYNcnf [28] [11] STA713				
RS-a	”p24&p35 STA05A p24&p35 [19] SRSYNind(2) [16] STA11A	p28 STA05B	p28 STA05C	”p24 STA05A [6] p24 [19] SRSYNind(2) [16] STA11A [6]	
RS-r	”p24&p32&p35 STA05A p24&p32&p35 SRSYNind [16] STA11A	p28 STA05B	p28 STA05C	”p24&p32 STA05A [6] p24&p32 SRSYNind [16] STA11A [6]	
RS-s	”p24&p35 STA05A p24&p35 SRSYNind [16] STA11A	p28 STA05B	p28 STA05C	”p24 STA05A [6] p24 SRSYNind [16] STA11A [6]	
SRSYNreq(a)					p10&p28&”p34 PR-RS(1) RS-a [16] STA05A
SRSYNreq(r)					p10&p25&”p34&p33 PR-RS(1) RS-r [16] STA05A
SRSYNreq(s)					p10&p25&”p34 PR-RS(1) RS-s [16] STA05A
SRSYNrsp					

TABLE A-10/X.225 *(cont.)*

Resynchronization state table

STATE EVENT	STA10A await SSYNMrsp	STA10B await SACTErsp	STA11A await SRSYNrsp	STA15A wait after PR-MAA	STA15B wait after PR-RS	STA15C wait after PR-RA	STA15D wait after PR-AB
PR-RA							
PR-RS	STA15B	STA15B		[10] STA15A		[10] STA15C	
RA						p36&p49 SRSYNcnf [28] [11] STA713 [6]	STA15D
RS-a	p35 [19] SRSYNind (2) [16] STA11A				p29 [19] SRSYNind (2) [16] STA11A		STA15D
RS-r					p32&p29 SRSYNind [16] STA11A		STA15D
RS-s	p35 SRSYNind [16] STA11A				p29 SRSYNind [16] STA11A		STA15D
SRSY Nreq(a)	p28 PR-RS(1) RS-a [16] STA05A	p28 PR-RS(1) RS-a [16] STA05A	p24 PR-RS(1) RS-a [16] STA05A	p28&p30 PR-RS(1) RS-a [16] STA05A [6]	p27&p28 PR-RS(1) RS-a [16] STA06		STA15D
SRSY Nreq(r)	p25&p33 PR-RS(1) RS-r [16] STA05A	p25&p33 PR-RS(1) RS-r [16] STA05A	p24&p33 PR-RS(1) RS-r [16] STA05A		p25&p27&p33 PR-RS(1) RS-r [16] STA06		STA15D
SRSY Nreq(s)	p25 PR-RS(1) RS-s [16] STA05A	p25 PR-RS(1) RS-s [16] STA05A	p24 PR-RS(1) RS-s [16] STA05A	p28&p30 PR-RS(1) RS-s [16] STA05A [6]	p25&p27 PR-RS(1) RS-s [16] STA06		STA15D
SRSY Nrsp			p43 PR-RA(1) RA [28] [11] STA713				STA15D

TABLE A-10/X.225 *(end)*

Resynchronization state table

STATE EVENT	STA16 await TDISind	STA18 await GTA	STA19 await recovery (init)	STA20 await recovery	STA713 data transfer
PR-RA	STA16				
PR-RS	STA16	[10] STA18	STA15B	STA15B	p26 STA15B ”p26 [10] STA713
RA	STA16				
RS-a	STA16		p35 [19] SRSYNind(2) [16] STA11A	p35 [19] SRSYNind(2) [16] STA11A	p26&p35 [19] SRSYNind(2) [16] STA11A
RS-r	STA16		p32&p35 SRSYNind [16] STA11A	p32&p35 SRSYNind [16] STA11A	p32&p26&p35 SRSYNind [16] STA11A
RS-s	STA16		p35 SRSYNind [16] STA11A	p35 SRSYNind [16] STA11A	p26&p35 SRSYNind [16] STA11A
SRSYNreq(a)				p28 PR-RS(1) RS-a [16] STA05A	p29 PR-RS(1) RS-a [16] STA05A
SRSYNreq(r)				p25&p33 PR-RS(1) RS-r [16] STA05A	p25&p26&p33 PR-RS(1) RS-r [16] STA05A
SRSYNreq(s)				p25 PR-RS(1) RS-s [16] STA05A	p25&p26 PR-RS(1) RS-s [16] STA05A
SRSYNrsp					

TABLE A-11/X.225

Activity interrupt and discard state table

STATE EVENT	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA
AD	STA01A	TDISreq STA01	TDISreq STA01			p38&p40 SACTDind [16] STA11C	
ADA	STA01A	TDISreq STA01	TDISreq STA01				
AI	STA01A	TDISreq STA01	TDISreq STA01			p38&p40 SACTIind [16] STA11B	
AIA	STA01A	TDISreq STA01	TDISreq STA01				p38 SACTIcnf [29] STA713
SACTDreq				p34&p39 PR-RS(1) AD [16] STA05C	p39 PR-RS(1) AD [16] STA05C		
SACTDrsp							
SACTIreq				p34&p39 PR-RS(1) AI [16] STA05B	p39 PR-RS(1) AI [16] STA05B		
SACTIrsp							

TABLE A-11/X.225 *(cont.)*

Activity interrupt and discard state table

STATE / EVENT	STA05C await PR or ADA	STA06 await RS after coll	STA10A await SSYNMrsp	STA10B await SACTErsp	STA11A await SRSYNrsp	STA11B await SACTIrsp
AD		p37&p40 SACTDind [16] STA11C	p38&p40 SACTDind [16] STA11C	p38&p40 SACTDind [16] STA11C		
ADA	p38 SACTDcnf [29] STA713					
AI		p37&p40 SACTIind [16] STA11B	p38&p40 SACTIind [16] STA11B	p38&p40 SACTIind [16] STA11B		
AIA						
SACTDreq					p34&p39 PR-RS(1) AD [16] STA05C	
SACTDrsp						
SACTIreq					p34&p39 PR-RS(1) AI [16] STA05B	
SACTIrsp						PR-RA(1) AIA [30] STA713

TABLE A-11/X.225 *(cont.)*

Activity interrupt and discard state table

STATE EVENT	STA11C await SACTDrsp	STA15A wait after PR-MAA	STA15B wait after PR-RS	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind
AD			p37&p40 SACTDind [16] STA11C			STA16
ADA				p37&p41 SACTDcnf [29] STA713 [6]	STA15D	STA16
AI			p37&p40 SACTIind [16] STA11B		STA15D	STA16
AIA				p37&p42 SACTIcnf [29] STA713 [6]	STA15D	STA16
SACTDreq		p34&p39 PR-RS(1) AD [16] STA05C [6]	p27&p34 &p39 PR-RS(1) AD [16] STA05C		STA15D	
SACTDrsp	PR-RA(1) ADA [30] STA713				STA15D	
SACTIreq		p34&p39 PR-RS(1) AI [16] STA05B [6]	p27&p34 &p39 PR-RS(1) AI [16] STA05B		STA15D	
SACTIrsp					STA15D	

TABLE A-11/X.225 *(end)*

Activity interrupt and discard state table

STATE / EVENT	STA19 await recovery (init)	STA20 await recovery	STA713 data transfer
AD	p38&p40 SACTDind [16] STA11C	p38&p40 SACTDind [16] STA11C	p38&p40 SACTDind [16] STA11C
ADA			
AI	p38&p40 SACTIind [16] STA11B	p38&p40 SACTIind [16] STA11B	p38&p40 SACTIind [16] STA11B
AIA			
SACTDreq		p34&p11 PR-RS(1) AD [16] STA05C	p34&p39 PR-RS(1) AD [16] STA05C
SACTDrsp			
SACTIreq		p34&p11 PR-RS(1) AI [16] STA05B	p34&p39 PR-RS(1) AI [16] STA05B
SACTIrsp			

TABLE A-12/X.225

Activity start, resume and capability data state table

STATE EVENT	STA01A await AA	STA01C idle TC con	STA01A await CDO	STA15B wait after PR-RS	STA15B wait after PR-AB	STA16 await TDISind	STA21 await CDA	STA22 await SCDrsp	STA713 data transfer
AR	STA01A	TDISreq STA01	TDISreq STA01	p44 SACTRind [12] [27] STA15B	STA15D	STA16			p44 SACTRind [12] [27] STA713 [6]
AS	STA01A	TDISreq STA01	TDISreq STA01	p44 SACTSind [12] [26] STA15B	STA15D	STA16			p44 SACTSind [12] [26] STA713 [6]
CD	STA01A	TDISreq STA01	TDISreq STA01		STA15D	STA16			p46 SCDind STA22
CDA	STA01A	TDISreq STA01	TDISreq STA01		STA15D	STA16	SCDcnf STA713		
SACTRreq					STA15D				p45 AR [12] [27] STA713
SACTSreq					STA15D				p45 AS [12] [26] STA713
SCDreq					STA15D				p47 CD STA21
SCDrsp					STA15D			CDA STA713	

TABLE A-13/X.225

Token management and exceptions state table

STATE EVENT	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA
ED	STA01A	TDISreq STA01	TDISreq STA01	p52 SUERind STA20	p48&p03 SUERind STA20 p48&"p03 SUERind STA713	p48&p03 SUERind STA20 p48&"p03 SUERind STA713	p48 STA05A	p48 STA05B
ER	STA01A	TDISreq STA01	TDISreq STA01	p52 SPERind STA20	p48&p03 SPERind STA20 p48&"p03 SPERind STA713	p48&p03 SPERind STA20 p48&"p03 SPERind STA713	p48 STA05A	p48 STA05B
GT	STA01A	TDISreq STA01	TDISreq STA01		p59 SGTind [11] STA04A	p59 SGTind [11] STA04B	p59 STA05A	p59 STA05B
GTA	STA01A	TDISreq STA01	TDISreq STA01					
GTC	STA01A	TDISreq STA01	TDISreq STA01					
PT	STA01A	TDISreq STA01	TDISreq STA01	p53 SPTind STA03	p53 SPTind STA04A	p53 SPTind STA04B	p53 STA05A	p53 STA05B
SCGreq								
SGTreq					p54 GT [11] STA04A	p54 GT [11] STA04B		
SPTreq								
SUERreq								

TABLE A-13/X.225 *(cont.)*

Token management and exceptions state table

STATE / EVENT	STA05C await PR or ADA	STA06 await RS after coll	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp	STA15A wait after PR-MAA
ED	p48 STA05C	p48 STA06				
ER	p48 STA05C	p48 STA06				
GT	p59 STA05C	p59 STA06		p59 SGTind [11] STA10A	p59 SGTind [11] STA10B	p59 SGTind [11] STA15A
GTA						
GTC						
PT	p53 STA05C	p53 STA06				p53 SPTind STA15A
SCGreq						
SGTreq				p54 GT [11] STA10A	p54 GT [11] STA10B	p54 GT [11] STA15A
SPTreq			p53 PT STA09	p53 PT STA10A	p53 PT STA10B	
SUERreq			p50 ED STA19	p50 ED STA19	p50 ED STA19	

TABLE A-13/X.225 *(cont.)*

Token management and exceptions state table

STATE EVENT	STA15B wait after PR-RS	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA18 await GTA	STA19 await recovery (init)
ED		p48 STA15C	STA15D	STA16		p50 SUERind STA19
ER		p48 STA15C	STA15D	STA16		p50 SPERind STA19
GT	p59 STA15B	p59 STA15C	STA15D	STA16		p60 SGTind [11] STA19 p61 SGTind [11] STA713
GTA			STA15D	STA16	STA713 [6]	
GTC			STA15D	STA16		
PT	p53 STA15B	p53 STA15C	STA15D	STA16	p53 SPTind STA18	p53 STA19
SCGreq			STA15D			
SGTreq	p54 STA15B		STA15D			
SPTreq	p53 STA15B		STA15D			
SUERreq	p50 STA15B		STA15D			

TABLE A-13/X.225 *(end)*

Token management and exceptions state table

STATE / EVENT	STA20 await recovery	STA21 await CDA	STA22 await SCDrsp	STA713 data transfer
ED				p50 SUERind STA713 p51 SUERind STA20
ER		p48 SPERind STA20		p50 SPERind STA713 p51 SPERind STA20
GT	p60 SGTind [11] STA20 p61 SGTind [11] STA713	p59 SGTind [11] STA21		p59 SGTind [11] STA713
GTA				
GTC				p62 SCGind GTA [11] STA713
PT	p53 STA20	p53 SPTind STA21		p53 SPTind STA713
SCGreq				p55 GTC [11] STA18
SGTreq	p57 GT [11] STA20 p58 GT [11] STA713			p54 GT [11] STA713
SPTreq			p53 PT STA22	p53 PT STA713
SUERreq				p50 ED STA19

TABLE A-14/X.225

Connection release state table

STATE EVENT	STA01A await AA	STA01C idle TC con	STA01D await CDO	STA03 await DN	STA05A await PR or RA	STA06 await RS after coll	STA09 await SRELrsp
DN	STA01A	TDISreq STA01	TDISreq STA01	”p66 SRELcnf+ TDISreq STA01 p66 SRELcnf+ STA01C			p69&”p01 SRELcnf+ [32] STA09
FN-nr	STA01A	TDISreq STA01	TDISreq STA01	”p65 SRELind [8] [18] STA09	p68 STA05A	p68 STA05A	
FN-r	STA01A	TDISreq STA01	TDISreq STA01	”p65&”p01 &p16 SRELind [8] [18] STA09	p68&”p0 1&p16 STA05A		
NF	STA01A	TDISreq STA01	TDISreq STA01	p67 SRELcnf- STA713			
SRELreq							p65 FN-nr [8] [18] STA09
SRELrsp+							”p66&p75 DN [4] STA16 p66 DN STA01C p69&p01 DN STA03
SRELrsp-							p67 NF STA713

TABLE A-14/X.225 *(end)*

Connection release state table

STATE EVENT	STA15B wait after PR-RS	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA19 await recovery (init)	STA20 await recovery	STA713 data transfer
DN				STA16			
FN-nr		p68 STA15C	STA15D	STA16	p68 STA19	p68 STA20	p68 SRELind [8] STA09
FN-r		p68&”p0 1&p16 STA15C		STA16	p68&”p0 1&p16 STA19	p68&”p0 1&p16 STA20	p68&”p0 1&p16 SRELind [9] STA09
NF	p67 SRELcnf- STA15B		STA15D	STA16			
SRELreq	p63 STA15B		STA15D				p63&”p64 FN-nr [8] STA03 p63&p64 FN-r [7] STA03
SRELrsp+			STA15D				
SRELrsp-			STA15D				

TABLE A-15/X.225

Abort state table

STATE EVENT	STA01 idle no TC	STA01A await AA	STA01B await TCONcnf	STA01C idle TC con	STA01D await CDO	STA02A await AC	STA02B await OA	STA03 await DN	STA04A await PR or MAA
AA	/ /	[3] STA01C	/ /	TDISreq STA01	TDISreq STA01				
AB-nr	/ /	[3] TDISreq STA01	/ /	TDISreq STA01	TDISreq STA01	SxABind(3) TDISreq STA01	SxABind(3) TDISreq STA01	SxABind(3) TDISreq STA01	SxABind(3) TDISreq STA01
AB-r	/ /	[3] STA01C	/ /	”p02 TDISreq STA01 p02 AA STA01C	”p02 TDISreq STA01 p02 AA STA01C	”p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	”p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	”p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	”p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C
PR-AB	/ /	/ /	/ /	TDISreq STA01	/ /	STA15D	/ /	STA15D	STA15D
SUABreq			TDISreq STA01			”p02 PR- AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR- AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR- AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR- AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A
TDISind	/ /	[3] STA01	SPABind STA01	STA01	STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01
TIM	/ /	TDISreq STA01	/ /	/ /	/ /	/ /	/ /	/ /	/ /

TABLE A-15/X.225 *(cont.)*

Abort state table

STATE EVENT	STA04B await PR or AEA	STA05A await PR or RA	STA05B await PR or AIA	STA05C await PR or ADA	STA06 await RS after coll	STA08 await SCONrsp
AA						
AB-nr	SxABind(3) TDISreq STA01	SxABind(3) TDISreq STA01	SxABind(3) TDISreq STA01	SxABind(3) TDISreq STA01	SxABind(3) TDISreq STA01	SxABind(3) TDISreq STA01
AB-r	”p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	”p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	”p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C	”p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C		”p02 SxABind(3) TDISreq STA01 p02 SxABind(3) AA STA01C
PR-AB	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D
SUABreq	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR-AB(4) AB-nr [4] STA16	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A
TDISind	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01
TIM	/ /	/ /	/ /	/ /	/ /	/ /

TABLE A-15/X.225 *(cont.)*

Abort state table

STATE / EVENT	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp	STA11A await SRSYNrsp	STA11B await SACTIrsp	STA11C await SACTDrsp
AA						
AB-nr	SxABInd(3) TDISreq STA01	SxABInd(3) TDISreq STA01	SxABInd(3) TDISreq STA01	SxABInd(3) TDISreq STA01	SxABInd(3) TDISreq STA01	SxABInd(3) TDISreq STA01
AB-r	”p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	”p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	”p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	”p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	”p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	”p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C
PR-AB	STA15D	STA15D	STA15D	STA15D	STA15D	STA15D
SUABreq	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A
TDISind	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01
TIM	/ /	/ /	/ /	/ /	/ /	/ /

TABLE A-15/X.225 *(cont.)*

Abort state table

STATE EVENT	STA15A wait after PR-MAA	STA15B wait after PR-RS	STA15C wait after PR-RA	STA15D wait after PR-AB	STA16 await TDISind	STA18 await GTA	STA19 await recovery (init)
AA					[3] TDISreq STA01		
AB-nr	SxABInd(3) TDISreq STA01	SxABInd(3) TDISreq STA01	SxABInd(3) TDISreq STA01	SxABInd(3) TDISreq STA01	[3] TDISreq STA01	SxABInd(3) TDISreq STA01	SxABInd(3) TDISreq STA01
AB-r					[3] TDISreq STA01	”p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	”p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C
PR-AB	STA15D	STA15D	STA15D		[3] TDISreq STA01	STA15D	STA15D
SUABreq	”p02 PR-AB(4) AB-nr [4] STA16	”p02 PR-AB(4) AB-nr [4] STA16	”p02 PR-AB(4) AB-nr [4] STA16	[4] STA15D		”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A
TDISind	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01	[3] STA01	SPABind STA01	SPABind STA01
TIM	/ /	/ /	/ /	TDISreq STA01	TDISreq STA01	/ /	/ /

TABLE A-15/X.225 *(end)*

Abort state table

STATE / EVENT	STA20 await recovery	STA21 await CDA	STA22 await SCDrsp	STA713 data transfer
AA				
AB-nr	SxABInd(3) TDISreq STA01	SxABInd(3) TDISreq STA01	SxABInd(3) TDISreq STA01	SxABInd(3) TDISreq STA01
AB-r	”p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	”p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	”p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C	”p02 SxABInd(3) TDISreq STA01 p02 SxABInd(3) AA STA01C
PR-AB	STA15D	STA15D	STA15D	STA15D
SUABreq	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A	”p02 PR-AB(4) AB-nr [4] STA16 p02 AB-r [4] STA01A
TDISind	SPABind STA01	SPABind STA01	SPABind STA01	SPABind STA01
TIM	/ /	/ /	/ /	/ /

ANNEX B

(to Recommendation X.225)

Relationship to CCITT Recommendation T.62 encoding

This Recommendation has been designed to be compatible with Recommendation T.62.

Table B-1/X.225 shows the relationship between the Recommendation T.62 commands and responses and the SPDUs used in this Recommendation.

Table B-2/X.225 shows the relationship between the Recommendation T.62 PGI and PI parameters and the PGI and PI parameters used in this Recommendation.

Annex C lists the PGIs and PIs which are not defined in this Recommendation, but are reserved as they are used in Recommendation T.62 for parameters relevant to higher layers than the session layer. Use of these PGIs and PIs is necessary for correct operation to Recommendation T.62. An implementation of the protocol specified in this Recommendation will need to be enhanced to take account of these PGI and PI units.

TABLE B-1/X.225

Relationship between T.62 commands and responses and X.225 SPDUs

Code	T.62 name	SPDU Code	SPDU name
13	CSS	CN	CONNECT
16	xxxx	OA	OVERFLOW ACCEPT
15	xxxx	CDO	CONNECT DATA OVERFLOW
14	RSSP	AC	ACCEPT
12	RSSN	RF	REFUSE
9	CSE	FN	FINISH
10	RSEP	DN	DISCONNECT
25	CSA	AB	ABORT
26	RSAP	AA	ABORT ACCEPT
1	CSUI-CDUI	DT	DATA TRANSFER
2	RSUI	PT	PLEASE TOKENS
21	CSCC	GTC	GIVE TOKENS CONFIRM
22	RSCCP	GTA	GIVE TOKENS ACK
1	CSUI	GT	GIVE TOKENS
0	RSUI-RDGR	ER	EXCEPTION REPORT
48	RSUI-RDPBN	ED	EXCEPTION DATA
33	CSTD	TD	TYPED DATA
8	xxxx	NF	NOT FINISHED
49	CSUI-CDPB	MIP	MINOR SYNC POINT
50	RSUI-RDPBP	MIA	MINOR SYNC ACK
41	CSUI-CDE	MAP	MAJOR SYNC POINT
42	RSUI-RDEP	MAA	MAJOR SYNC ACK
7	xxxx	PR	PREPARE
53	xxxx	RS	RESYNCHRONIZE
34	xxxx	RA	RESYNCHRONIZE ACK
5	xxxx	EX	EXPEDITED DATA
45	CSUI-CDS	AS	ACTIVITY START
29	CSUI-CDC	AR	ACTIVITY RESUME
25	CSUI-CDR	AI	ACTIVITY INTERRUPT
26	RSUI-RDRP	AIA	ACTIVITY INTERRUPT ACK
57	CSUI-CDD	AD	ACTIVITY DISCARD
58	RSUI-RDDP	ADA	ACTIVITY DISCARD ACK
41	CSUI-CDE	AE	ACTIVITY END
42	RSUI-RDEP	AEA	ACTIVITY END ACK
61	CSUI-CDCL	CD	CAPABILITY DATA
62	RSUI-RDCLP	CDA	CAPABILITY DATA ACK

TABLE B-2/X.225

Relationship between Recommendation T.62 PGI/PI and Recommendation X.225 parameters

T.62 parameters	code	X.225 parameters
PGI		
Reserved for extension	0	See Table C-1/X.225
Session reference	1	Connection Identifier
Non-basic session capabilities	2	See Table C-1/X.225
	3	
	4	
	5	Connect/Accept Item
	6	
	7	
PI		
Service identifier	8	See Table C-1/X.225
Terminal identifier (called terminal)	9	Called SS-user Ref.
Terminal identifier (calling terminal)	10	Calling SS-user Ref.
Date and time	11	Common Reference
Additional session reference number	12	Additional Ref. Info.
Miscellaneous session capabilities	13	See Table C-1/X.225
Window size	14	See Table C-1/X.225
	15	Sync Type Item
Session control functions	16	Token Item
Session termination parameter	17	Transport Disconnect
Inactivity timer	18	See Table C-1/X.225
	19	Protocol options
Session service functions	20	Session Requirements
	21	TSDU Maximum Size
	22	Version number
	23	Initial Serial Number
	24	Prepare Type
	25	Enclosure Item
	26	Token Setting Item
	27	Resync Type
Initiator's reference number	28	See Table C-1/X.225
Acceptor's reference number	29	See Table C-1/X.225
Reactivation/transaction indication	30	See Table C-1/X.225
Suspend reject reason	31	See Table C-1/X.225
PGI		
Reserved for extension	32	See Table C-1/X.225
Document linking	33	Linking Information
	34	
	35	
	36	
	37	
	38	
	39	

TABLE B-2/X.225 *(cont.)*

T.62 parameters	code	X.225 parameters
PI		
Service interworking identifier	40	See Table C-1/X.225
Document reference number	41	Activity identifier
Checkpoint reference number	42	Serial number
Reserved	43	
Acceptance of CDCL parameters	44	See Table C-1/X.225
Storage capacity negotiation	45	See Table C-1/X.225
Receiving ability jeopardized	46	User data (in MIA SPDU)
Reserved	47	
Document type identifier	48	See Table C-1/X.225
Reflect parameter values	49	Reflect parameter values
Reason (session and document)	50	Reason code
	51	Calling session selector
	52	Called/Responding session selector
	53	
	54	
	55	
	56	
	57	
	58	
	59	
	60	Data overflow
	61	
	62	
	63	
PGI		
Reserved for extension	64	See Table C-1/X.225
Nonbasic teletex terminal capabilities	65	See Table C-1/X.225
	66	
	67	
	68	
	69	
	70	
	71	
PI		
Graphic character set	72	See Table C-1/X.225
Control character set	73	See Table C-1/X.225
Teletex page format	74	See Table C-1/X.225
Miscellaneous teletex terminal capabilities	75	See Table C-1/X.225
	76	
Number of dots for character box height	77	See Table C-1/X.225
Number of dots for character box width	78	See Table C-1/X.225
	79	
PGI		
	192	
Session user data	193	User data
	194	Extended User Data

ANNEX C

(to Recommendation X.225)

PGIs and PIs reserved for use by Recommendation T.62

Table C-1/X.225 lists the PGIs and PIs which are not defined in this Recommendation, but which are reserved as they are used in Recommendation T.62 for parameters that are relevant to higher layers than the session layer.

TABLE C-1/X.225

PGIs and PIs reserved for use by Recommendation T.62

PGI	
0	Reserved for extension
1	Non-basic session capabilities
PI	
8	Service identifier
13	Miscellaneous session capabilities
14	Window size
18	Inactivity timer
28	Initiator's reference number
29	Acceptor's reference number
30	Reactivation/transaction indication
31	Suspend reject reason
PGI	
32	Reserved for extension
PI	
40	Service interworking identifier
44	Acceptance of CDCL parameters
45	Storage capacity negotiation
48	Document type identifier
PGI	
64	Reserved for extension
65	Non-basic teletex terminal capabilities
PI	
72	Graphic character set
73	Control character set
74	Teletex page format
75	Miscellaneous teletex terminal capabilities
77	Number of dots for character box height
78	Number of dots for character box width

ANNEX D

(to Recommendation X.225)

Compatibility between protocol version 1 and protocol version 2

Protocol version 2 of the Session Protocol is a superset of protocol version 1 (both of which are specified in this Recommendation). Protocol version 1 of the Session Protocol imposes restrictions on the length of user data fields. Protocol version 2 removes those length restrictions.

An implementation of the session protocol may limit the length of user data supported, based on the requirements of its SS-user or protocol version supported. Any such limitation is stated in the Protocol Implementation Conformance Statement. If no user of a session implementation requires more than 10k of user data during connection establishment, the implementation need not be able to send the CDO SPDU or receive the OA SPDU.

Implementations of protocol version 2 can interwork with implementations of protocol version 1 only by imposing a number of restrictions (all of which are valid, in terms of the conformance statement). These restrictions are:

a) User Data parameter value in the ABORT SPDU shall not exceed 9 octets.

b) Reason Code parameter value in the REFUSE SPDU shall not exceed 513 octets.

c) The User Data PGI unit shall not be present in GIVE TOKENS, GIVE TOKENS CONFIRM, ACTIVITY INTERRUPT, ACTIVITY INTERRUPT ACK, ACTIVITY DISCARD and ACTIVITY DISCARD ACK SPDUs. The User Data PGI unit in all other SPDUs shall not exceed 512 octets.

d) Protocol version 1 shall be proposed in the CONNECT SPDU. In this case the Extended User Data parameter and Data Overflow parameter in the CONNECT SPDU shall not be present.

Note - Protocol version 2 may also be proposed, but to operate validly with a protocol version 1 only implementation, protocol version 1 shall be selected.

As a consequence of protocol version 1 being selected:

e) segmenting, as specified in ¤ 6.3.5 b) does not apply. Only data SSDUs and typed data SSDUs may be segmented;

f) the OVERFLOW ACCEPT SPDU and CONNECT DATA OVERFLOW SPDU are not used.

Note - Implementations of the earlier edition of this Recommendation which specified protocol version 1 can upgrade to and claim conformance with this edition of this Recommendation by stating, in their Protocol Implementation Conformance Statement, the restrictions specified in a) to c) above; and by conforming to the specified procedure for rejecting SPDUs with Òto muchÓ user data (see ¤ A.4.3.1.2) (note that this requires that the implemantation recognises the Extended User Data parameter in the CONNECT SPDU and the Enclosure Item in the ABORT SPDU). This is a minimal implementation of protocol version 2 and will not satisfy the requirements of some ASEs.

APPENDIX I

(to Recommendation X.225)

Differences between Recommendation X.225

and ISO International Standard 8327

I.1 In ISO 8327 the last sentence in the second paragraph of A.1 states ÒIn case of arbitration or dispute this annex takes precedence over clause 7Ó. This statement is not included in this Recommendation.