INTERNATIONAL STANDARD

ISO 8327

First edition
1987-08-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ

Information processing systems — Open Systems Interconnection — Basic connection oriented session protocol specification

Systèmes de traitement de l'information — Interconnexion de systèmes ouverts — Protocole de session en mode connexion

Reference number
ISO 8327 : 1987 (E)

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work.

Draft International Standards adopted by the technical committees are circulated to the member bodies for approval before their acceptance as International Standards by the ISO Council. They are approved in accordance with ISO procedures requiring at least 75 % approval by the member bodies voting.

International Standard ISO 8327 was prepared by Technical Committee ISO/TC 97, *Information processing systems*.

Users should note that all International Standards undergo revision from time to time and that any reference made herein to any other International Standard implies its latest edition, unless otherwise stated.

Printed in Switzerland

Contents

Information processing systems — Open Systems Interconnection — Basic connection oriented session protocol specification

0 Introduction

This International Standard is one of a set of International Standards produced to facilitate the interconnection of computer systems. The set of International Standards covers the services and protocols required to achieve such interconnection.

This International Standard is related to other International Standards in the set as defined by the Reference Model for Open Systems Interconnection (ISO 7498). The Reference Model subdivides the area of standardization for interconnection into a series of layers of specification, each of manageable size. It is most closely related to and lies within the field of application of the Session Service Definition (ISO 8326). It also uses and references the Transport Service Definition (ISO 8072), whose provisions it assumes in order to accomplish the aims of the session protocol. The interrelationship of these International Standards is illustrated in figure 1.

This International Standard 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 International Standard 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 part of an agreement for the admittance of systems into the open systems environment;

d) as a refinement to the understanding of OSI.

As it is expected that the initial users of this International Standard will be designers and implementors of equipment this International Standard contains, in notes or in annexes, guidance on the implementation of the procedures defined herein.

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 International Standard 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 International Standard 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 International Standard are essential as they 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 International Standard with statements of the options provided or required or with statements of the intended purpose of provision or use.

This International Standard contains the following annexes :

a) annex A — State tables;

b) annex B — Relationship to CCITT Recommendation T.62 encoding;

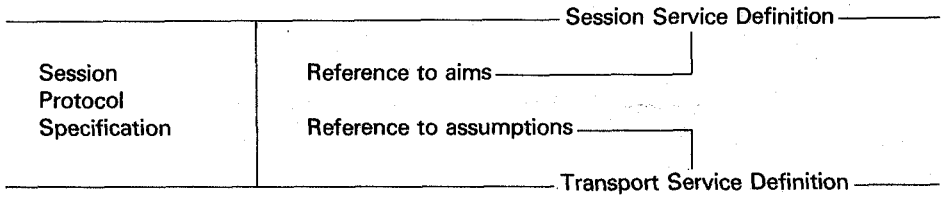


Figure 1 — Relationship between the session protocol and adjacent services

c) annex C — PGIs and PIs reserved for use by Recommendation T.62.

1 Scope and field of application

This International Standard 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.

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.

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.

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

2 References

ISO 7498, *Information processing systems — Open Systems Interconnection — Basic Reference Model.*

ISO 7498/Add. 3, *Information processing systems — Open Systems Interconnection — Basic Reference Model — Addendum 3: Naming including addressing.*[1)]

ISO 8072, *Information processing systems — Open Systems Interconnection — Transport Service Definition.*

ISO 8326, *Information processing systems — Open Systems Interconnection — Basic connection oriented session service definition.*

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

NOTE — CCITT Recommendation T.62 is not essential for the application of this International Standard, 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 annex B and C).

1) At present at the stage of draft; publication anticipated in due course.

Section one : General

3 Definitions

NOTE — The definitions contained in this clause make use of abbreviations defined in clause 4.

3.1 This International Standard is based on the concepts developed in ISO 7498, and makes use of the following terms defined in it :

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 selector (defined in ISO 7498/Add. 3).

3.2 This International Standard is also based on concepts developed in ISO 8326 and makes use of the following terms defined in it :

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 International Standard are used in relation to tokens and are explained in ISO 8326 :

a) assigned;
b) not assigned;
c) available;
d) not available.

3.3 For the purposes of this International Standard, the following definitions also apply.

3.3.1 Session Protocol Machine (SPM) : An abstract machine that carries out the procedures specified in this protocol.

NOTE — A session entity comprises 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 SPU.

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 International Standard with respect to structure and encoding.

3.3.14 invalid SPDU : An SPDU which does not comply with the requirements of this International Standard 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 International Standard, 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 International Standard, 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 the 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 (ISO 8326). 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 (ISO 8072) 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.

NOTE — A session entity comprises one or more SPMs.

5.2 Services provided by the Session Layer

The protocol specified in this International Standard supports the session service defined in ISO 8326. Information is transferred to and from the SS-user using the session service primitives listed in table 1. Table 1 also defines the SPDUs associated with each of the service primitives.

5.3 Services assumed from the Transport Layer

The protocol specified in this International Standard assumes the use of the connection-oriented transport service defined in ISO 8072.

Information is transferred to and from the TS-provider in the transport service primitives listed in table 2.

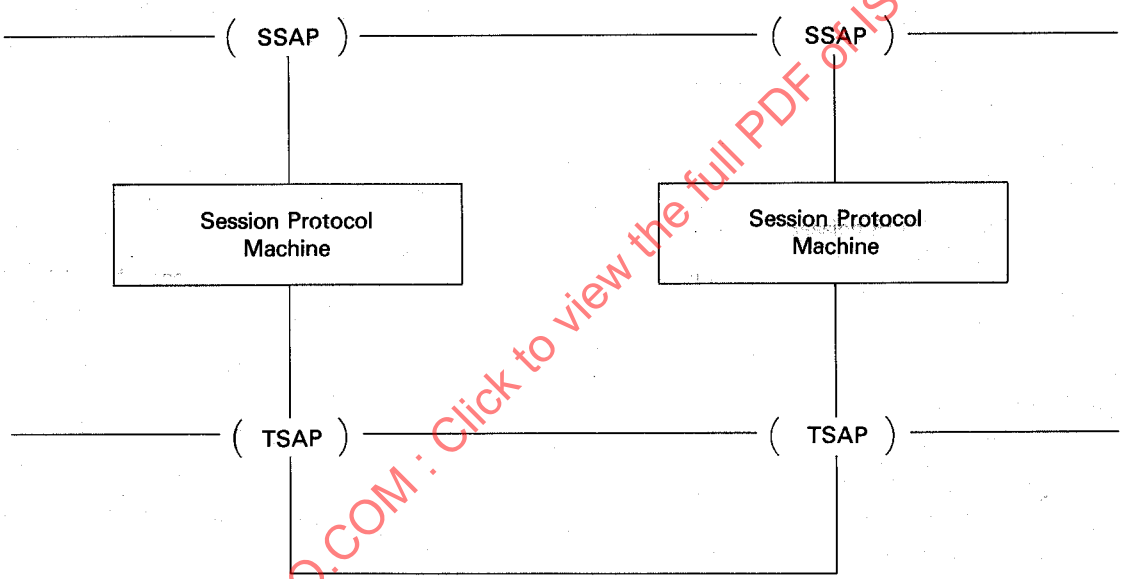


Figure 2 — Model of the Session Layer

Table 1 — 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
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 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
Resynchronization	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

Table 1 (*concluded*)

Service	Primitives	Associated SPDUs
Activity resume	S-ACTIVITY-RESUME request	ACTIVITY RESUME SPDU
	S-ACTIVITY-RESUME indication	ACTIVITY RESUME SPDU
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

Table 2 — Transport service primitives

Primitives	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

Key :

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.

These functions 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 and 7.2);

d) to transfer session selectors (see 7.1 and 7.2) if required;

e) to distinguish between session connections (see 7.1 and 7.2);

f) to transfer a limited amount of transparent user data (see 7.1 and 7.2).

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 ISO 8326 :

a) normal data transfer (see 7.9), 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 7.14 to 7.17), to enable the SS-users to request and transfer tokens which control the exclusive right to exercise certain functions (see table 5);

c) exception reporting (see 7.25 and 7.26), to enable the SS-provider or the SS-user to report exception conditions that are less than those requiring abort;

d) typed data transfer (see 7.11), to enable transfer of information which is not subject to assignment of the data token;

e) minor synchronization point (see 7.18 and 7.19), 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 7.20 and 7.21 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 7.22 and 7.23), 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 7.10), 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 7.27 to 7.34) provides a means explicitly to start, end, assume, 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 7.12 and 7.13), 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 a limited amount of transparent user data.

5.5 Functional units

Functional units are logical groupings of related elements of procedure defined by this International Standard 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.

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.

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 a limited amount 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 points are identified by serial numbers, and clearly separate the data flow before and 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 allows the SS-users to manage synchronized logical pieces of work.

5.6 Tokens

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

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

Table 3 — Functional units

Functional unit	SPDU code	SPDU name	Reference
Kernel	CN	CONNECT (see note 1)	7.1
	AC	ACCEPT (see note 1)	7.2
	RF	REFUSE (see note 1)	7.3
	FN	FINISH	7.4
	DN	DISCONNECT	7.5
	AB	ABORT	7.7
	AA	ABORT ACCEPT (see note 2)	7.8
	DT	DATA TRANSFER	7.9
Negotiated release	NF	NOT FINISHED	7.6
	GT	GIVE TOKENS (see note 4)	7.14
	PT	PLEASE TOKENS (see note 4)	7.15
Half-duplex	GT	GIVE TOKENS (see note 3)	7.14
	PT	PLEASE TOKENS (see note 3)	7.15
Duplex		No additional associated SPDUs	
Expedited data	EX	EXPEDITED DATA	7.10
Typed data	TD	TYPED DATA	7.11
Capability data exchange	CD	CAPABILITY DATA	7.12
	CDA	CAPABILITY DATA ACK	7.13
Minor synchronize	MIP	MINOR SYNC POINT	7.18
	MIA	MINOR SYNC ACK	7.19
	GT	GIVE TOKENS (see note 5)	7.14
	PT	PLEASE TOKENS (see note 5)	7.15
Major synchronize	MAP	MAJOR SYNC POINT	7.20
	MAA	MAJOR SYNC ACK	7.21
	PR	PREPARE (see note 6)	7.24
	GT	GIVE TOKENS (see note 7)	7.14
	PT	PLEASE TOKENS (see note 7)	7.15
Resynchronize	RS	RESYNCHRONIZE	7.22
	RA	RESYNCHRONIZE ACK	7.23
	PR	PREPARE (see note 6)	7.24
Exceptions	ER	EXCEPTION REPORT	7.25
	ED	EXCEPTION DATA	7.26
Activity management	AS	ACTIVITY START	7.27
	AR	ACTIVITY RESUME	7.28
	AI	ACTIVITY INTERRUPT	7.29
	AIA	ACTIVITY INTERRUPT ACK	7.30
	AD	ACTIVITY DISCARD	7.31
	ADA	ACTIVITY DISCARD ACK	7.32
	AE	ACTIVITY END	7.33
	AEA	ACTIVITY END ACK	7.34
	PR	PREPARE (see note 6)	7.24
	GT	GIVE TOKENS (see note 7)	7.14
	PT	PLEASE TOKENS (see note 7)	7.15
	GTC	GIVE TOKENS CONFIRM (see note 8)	7.16
	GTA	GIVE TOKENS ACK (see note 8)	7.17

NOTES

1 An implementation (see clause 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.

2 Reception and correct action is mandatory; transmission is optional if the transport connection is not to be reused (see 7.8.2).

3 Used to manage the data token.

4 Used to manage the release token.

5 Used to manage the synchronize-minor token.

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

7 Used to manage the major/activity token.

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

Table 4 — 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 — 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 END SPDU	2	2	1	nr

Key :

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 "called SS-user choice", 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 versions of the protocol that it is capable of supporting. The highest common version number is used.

5.7.5 Negotiation of maximum TSDU size

Each SPM proposes a maximum TSDU size that the initiator is permitted to send in the data transfer and connection release phases. 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 SSDUs on the session connection.

Each SPM also proposes a maximum TSDU size that the responder is permitted to send in the data transfer and connection release phases. 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 SSDUs on the session connection.

5.8 Local variables

This International Standard 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 clause 7 and are summarized in table 42 in annex A.

Section two : Session protocol specification

6 Use of the transport service

This clause 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 ISO 8326) 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 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 a 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 7.3)
FINISH SPDU (see 7.4)
DISCONNECT SPDU (see 7.5)
ABORT SPDU (see 7.7)
ABORT ACCEPT SPDU (see 7.8)

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)
ACCEPT SPDU (see 7.2)
REFUSE SPDU (see 7.3)
FINISH SPDU (see 7.4)
DISCONNECT SPDU (see 7.5)
NOT FINISHED SPDU (see 7.6)
DATA TRANSFER SPDU (see 7.9)
TYPED DATA SPDU (see 7.11)
CAPABILITY DATA SPDU (see 7.12)
CAPABILITY DATA ACK SPDU (see 7.13)
GIVE TOKENS SPDU (see 7.14)
PLEASE TOKENS SPDU (see 7.15)
GIVE TOKENS CONFIRM SPDU (see 7.16)
GIVE TOKENS ACK SPDU (see 7.17)
MINOR SYNC POINT SPDU (see 7.18)
MINOR SYNC ACK SPDU (see 7.19)
MAJOR SYNC POINT SPDU (see 7.20)
MAJOR SYNC ACK SPDU (see 7.21)
RESYNCHRONIZE SPDU (see 7.22)
RESYNCHRONIZE ACK SPDU (see 7.23)
EXCEPTION REPORT SPDU (see 7.25)
EXCEPTION DATA SPDU (see 7.26)
ACTIVITY START SPDU (see 7.27)
ACTIVITY RESUME SPDU (see 7.28)
ACTIVITY INTERRUPT SPDU (see 7.29)
ACTIVITY INTERRUPT ACK SPDU (see 7.30)
ACTIVITY DISCARD SPDU (see 7.31)
ACTIVITY DISCARD ACK SPDU (see 7.32)
ACTIVITY END SPDU (see 7.33)
ACTIVITY END ACK SPDU (see 7.34)

If the transport expedited flow is not available, the following additional SPDUs are sent on the transport normal flow :

ABORT SPDU (see 7.7)
ABORT ACCEPT SPDU (see 7.8)

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

Each SSDU is mapped one-to-one onto an SPDU, unless segmenting has been selected for that direction of transfer, in which case a data SSDU or a typed SSDU may be mapped onto more than one 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 TDSU 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 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) category 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, 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) onto a single TSDU.

The valid mappings of SPDUs onto TSDUs are illustrated in figure 3.

Any other concatenation of SPDUs is defined as invalid.

6.3.7.1 Processing order of concantenated 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.

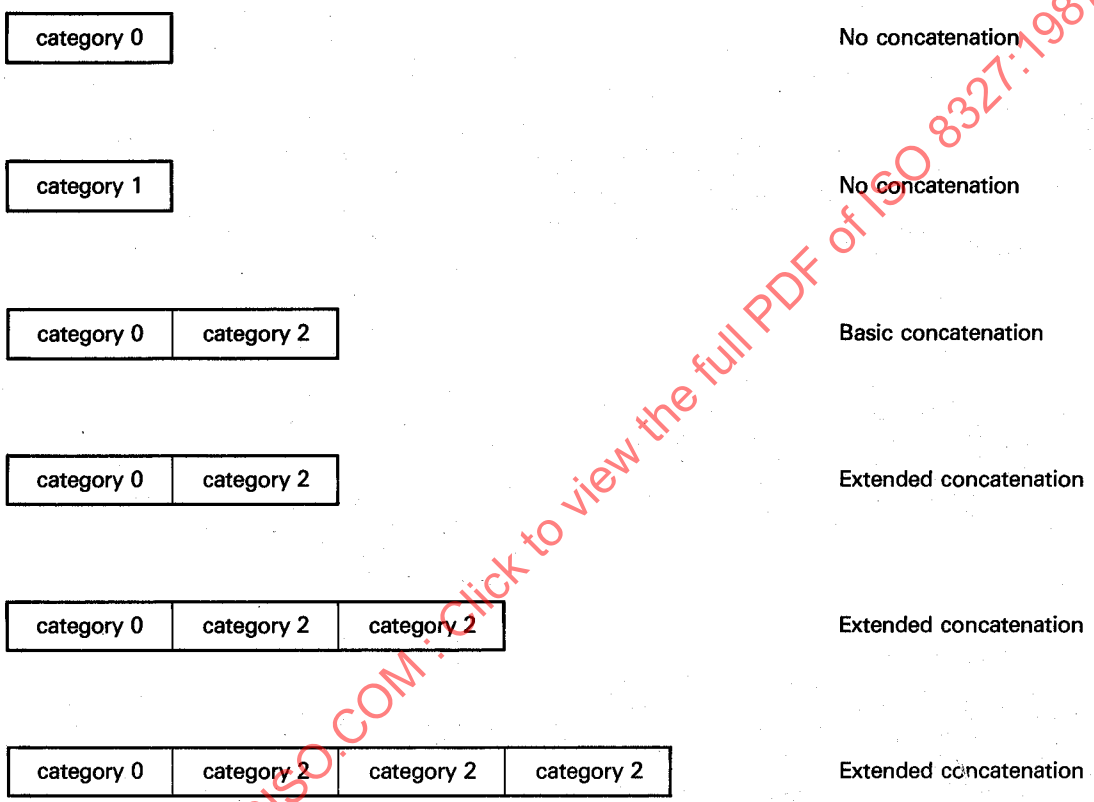


Figure 3 — Illustration of TSDU structures

Table 6 — Category 0, 1 and 2 SPDUs

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

Table 7 — 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* PLEASE TOKENS SPDU	RESYNCHRONIZE SPDU RESYNCHRONIZE ACK SPDU
GIVE TOKENS SPDU GIVE TOKENS SPDU	ACTIVITY START SPDU ACTIVITY RESUME SPDU
GIVE TOKENS SPDU* PLEASE TOKENS SPDU	ACTIVITY DISCARD SPDU ACTIVITY DISCARD ACK SPDU
GIVE TOKENS SPDU* 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* PLEASE TOKENS SPDU	CAPABILITY DATA SPDU CAPABILITY DATA ACK SPDU
PLEASE TOKENS SPDU PLEASE TOKENS SPDU	EXCEPTION REPORT SPDU EXCEPTION DATA SPDU

Key :

* 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.

** Indicates that the Token Item parameter is only present in the GIVE TOKENS SPDU if this DATA TRANSFER SPDU contains a complete SSDU, or the last segment of a segmented SSDU.

Table 8 — Valid extended concatenation of SPDUs

First SPDU	Second SPDU	Third SPDU	Fourth SPDU	Status
GIVE TOKENS SPDU GIVE TOKENS SPDU	MINOR SYNC ACK SPDU MAJOR SYNC ACK SPDU			
GIVE TOKENS SPDU	ACTIVITY END ACK SPDU			
GIVE TOKENS SPDU GIVE TOKENS SPDU GIVE TOKENS SPDU* GIVE TOKENS SPDU* GIVE TOKENS SPDU* GIVE TOKENS SPDU*	ACTIVITY START SPDU ACTIVITY RESUME SPDU ACTIVITY START SPDU ACTIVITY RESUME SPDU ACTIVITY START SPDU ACTIVITY RESUME SPDU	MINOR SYNC POINT SPDU MINOR SYNC POINT SPDU ACTIVITY END SPDU ACTVITY END SPDU MAJOR SYNC POINT SPDU MAJOR SYNC POINT SPDU		
GIVE TOKENS SPDU GIVE TOKENS SPDU GIVE TOKENS SPDU GIVE TOKENS SPDU	MINOR SYNC POINT SPDU MINOR SYNC ACK SPDU MAJOR SYNC POINT SPDU MAJOR SYNC ACK SPDU	DATA TRANSFER SPDU DATA TRANSFER SPDU DATA TRANSFER SPDU DATA TRANSFER SPDU		CL CL CL CL
GIVE TOKENS SPDU GIVE TOKENS SPDU	ACTIVITY START SPDU ACTIVITY RESUME SPDU	DATA TRANSFER SPDU DATA TRANSFER SPDU		CF CF
GIVE TOKENS SPDU GIVE TOKENS SPDU	ACTIVITY END SPDU ACTIVITY END ACK SPDU	DATA TRANSFER SPDU DATA TRANSFER SPDU		CL CL
GIVE TOKENS SPDU GIVE TOKENS SPDU GIVE TOKENS SPDU* GIVE TOKENS SPDU* GIVE TOKENS SPDU* GIVE TOKENS SPDU*	ACTIVITY START SPDU ACTIVTIY RESUME SPDU ACTIVITY START SPDU ACTIVITY RESUME SPDU ACTIVITY START SPDU ACTIVITY RESUME SPDU	MINOR SYNC POINT SPDU MINOR SYNC POINT SPDU ACTIVITY END SPDU ACTIVITY END SPDU MAJOR SYNC POINT SPDU MAJOR SYNC POINT SPDU	DATA TRANSFER SPDU DATA TRANSFER SPDU DATA TRANSFER SPDU DATA TRANSFER SPDU DATA TRANSFER SPDU DATA TRANSFER SPDU	C C C C C C

Key :

* indicates that the Token Item parameter is not present in the GIVE TOKENS SPDU.

Status :

CL : The DATA TRANSFER SPDU contains a complete SPDU 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.

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 7.7)
ABORT ACCEPT SPDU (see 7.8)
EXPEDITED DATA SPDU (see 7.10)
PREPARE SPDU (see 7.24)

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.

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

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.

7 Elements of procedure related to SPDUs

This clause defines valid sequences of operation of the protocol.

A more precise definition of procedures is contained in annex A which incorporates 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 clause.

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) 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) a Calling SS-user Reference parameter;

2) a Common Reference parameter;

3) an Additional Reference Information parameter.

b) a Connect/Accept Item parameter group containing

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

2) a TSDU maximum size parameter which, if present and not zero, indicates that segmenting of SSDUs is requested by the initiator. The initiator proposes the maximum TSDU sizes for each direction of transfer (see 5.7.5). If segmenting is not to be used, this parameter is not present or is zero;

3) a Version Number parameter to identify which versions of this protocol have been implemented;

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

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. The SPM waits until it receives an ACCEPT SPDU or a REFUSE SPDU.

7.1.3 Receiving the CONNECT SPDU

A valid incoming CONNECT SPDU 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.

7.2 ACCEPT SPDU

An SPM receiving a CONNECT SPDU may accept a proposal to establish a session connection by transferring an ACCEPT SPDU to the initiator, on the same transport connection.

7.2.1 Content of ACCEPT SPDU

The ACCEPT SPDU contains

a) 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) a Called SS-user Reference parameter;

2) a Common Reference parameter;

3) an Additional Reference Information parameter.

b) a Connect/Accept Item parameter group containing

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

2) a TSDU Maximum Size parameter which, if present and not zero, indicates that segmenting of SSDUs has been proposed by the responder. The responder proposes alternative values for the maximum TSDU sizes for each direction of transfer (see 5.7.5). These values may be larger or smaller than 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;

3) a Version Number parameter to identify which versions of this protocol have been implemented. The highest version number indicated by both initiator and responder is used;

4) an 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) a 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), regardless of the settings of the token setting item parameter in the CONNECT SPDU [see 7.1.1b)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 shall be returned.

c) a 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) a 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 shall propose which one is to be available. If only one of these functional units was indicated in the CONNECT SPDU, then the ACCEPT SPDU shall indicate that the same functional unit is to be used (or the connection attempt shall 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 SS-user may be present and is derived from the Responding Session Address provided by the responding SS-user;

f) a 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.

7.2.2 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.

7.2.3 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 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 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 7.2.1c)], an S-PLEASE-TOKEN indication is also generated.

7.3 REFUSE SPDU

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

7.3.1 Content of REFUSE SPDU

The REFUSE SPDU contains

a) 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) a Called SS-user Reference parameter;

2) a Common Reference parameter;

3) an Additional Reference Information parameter.

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

c) a 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) a Version Number parameter to identify which versions of this protocol have been implemented by the sending SPM;

e) a 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.

7.3.2 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.

7.3.3 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.

7.4 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.

7.4.1 Content of FINISH SPDU

The FINISH SPDU contains

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

7.4.2 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.

7.4.3 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.

7.5 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.

7.5.1 Content of DISCONNECT SPDU

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

7.5.2 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.

7.5.3 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.

NOTES

1 In the case of collision of a FINISH SPDU and an ABORT SPDU (see 7.7), the ABORT SPDU takes preference and thus the indication in the FINISH SPDU to keep or release the transport connection is ignored.

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.

7.6 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. No confirmation is sought.

7.6.1 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.

7.6.2 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.

7.6.3 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.

7.7 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.

7.7.1 Content of ABORT SPDU

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.

7.7.2 Sending the ABORT SPDU

An S-U-ABORT request, or the detection of a protocol error in any state of the SPM, results in an ABORT SPDU. 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 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.

7.7.3 Receiving the ABORT SPDU

A valid incoming ABORT SPDU 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 7.8).

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 7.8), and

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

7.8 ABORT ACCEPT SPDU

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

7.8.1 Content of ABORT ACCEPT SPDU

The ABORT ACCEPT SPDU contains no parameters.

7.8.2 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.

7.8.3 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.

7.9 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 are allowed (see 6.3.7).

The right to issue a DATA TRANSFER SPDU is subject to the token restrictions specified in table 5.

7.9.1 Content of DATA TRANSFER SPDU

The DATA TRANSFER SPDU contains

a) an Enclosure Item parameter to indicate the beginning and end of the 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) a 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.

7.9.2 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.

7.9.3 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).

It is also valid to receive EXPEDITED SPDUs and PREPARE (MAJOR SYNC ACK) SPDUs and this has no effect on the segmented SSDU being received.

The receipt of any other SPDUs is a protocol error.

7.10 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.

7.10.1 Content of EXPEDITED SPDU

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

7.10.2 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.

7.10.3 Receiving the EXPEDITED SPDU

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

7.11 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 7.9). The same rules for segmenting also apply.

7.11.1 Content of TYPED DATA SPDU

The TYPED DATA SPDU contains

a) an 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) a 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 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. The current state of the SPM is not changed.

7.11.3 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 the 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 receipt of SPDUs other than TYPED DATA SPDUs during the transfer of a segmented SSDU are the same as for the DATA TRANSFER SPDU (see 7.9.3).

7.12 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).

7.12.1 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.

7.12.2 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.

7.12.3 Receiving the CAPABILITY DATA SPDU

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

7.13 CAPABILITY DATA ACK SPDU

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

7.13.1 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.

7.13.2 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.

7.13.3 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.

7.14 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 a parameter field, it is used to indicate concatenation without assignment of tokens and, in this case, the sending and receiving procedures do not apply.

7.14.1 Content of GIVE TOKENS SPDU

The GIVE TOKENS SPDU contains a Token Item parameter which indicates which tokens are being transferred from the sending SS-user to the receiving SS-user.

7.14.2 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.

7.14.3 Receiving the GIVE TOKENS SPDU

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

7.15 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.

7.15.1 Content of PLEASE TOKENS SPDU

The PLEASE TOKENS SPDU contains

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

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

7.15.2 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.

7.15.3 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.

7.16 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.

7.16.1 Content of GIVE TOKENS CONFIRM SPDU

The GIVE TOKENS CONFIRM SPDU contains no parameters.

7.16.2 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 (for example TYPED DATA SPDU) may be sent or received as normal. This SPDU is sent on the transport normal flow.

7.16.3 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.

7.17 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.

7.17.1 Content of GIVE TOKENS ACK SPDU

The GIVE TOKENS ACK SPDU contains no parameters.

7.17.2 Sending the GIVE TOKENS ACK SPDU

A valid incoming GIVE TOKENS CONFIRM SPDU results in a GIVE TOKENS ACK SPDU (see also 7.16.3). The SPM may now transmit SPDUs associated with the token controlled functional units. This SPDU is sent on the transport normal flow.

7.17.3 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.

7.18 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 7.19). 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.

7.18.1 Content of MINOR SYNC POINT SPDU

The MINOR SYNC POINT SPDU contains

a) a Sync Type Item parameter which is used to indicate if an explicit confirmation is required (see 7.19);

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

7.18.2 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.

7.18.3 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.

7.19 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.

7.19.1 Content of MINOR SYNC ACK SPDU

The MINOR SYNC ACK SPDU contains

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

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

7.19.2 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.

7.19.3 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.

7.20 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.

7.20.1 Content of MAJOR SYNC POINT SPDU

The MAJOR SYNC POINT SPDU contains

a) 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) 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);

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

7.20.2 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.

7.20.3 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.

7.21 MAJOR SYNC ACK SPDU

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

7.21.1 Content of MAJOR SYNC ACK SPDU

The MAJOR SYNC ACK SPDU contains

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

7.21.2 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.

7.21.3 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.

7.22 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 "purge" 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.

7.22.1 Content of RESYNCHRONIZE SPDU

The RESYNCHRONIZE SPDU contains

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

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

c) a Serial Number parameter which indicates the serial number to which resynchronization 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) a User Data parameter which allows a limited amount of transparent user data to be transferred.

7.22.2 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 7.22.4.

7.22.3 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 7.22.4) passes an S-RESYNCHRONIZE indication to the SS-user.

7.22.4 Resynchronization contention

The contention between two RESYNCHRONIZE, ACTIVITY INTERRUPT, or ACTIVITY DISCARD SPDUs is resolved according to table 9. 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.

7.23 RESYNCHRONIZE ACK SPDU

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

7.23.1 Content of RESYNCHRONIZE ACK SPDU

The RESYNCHRONIZE ACK SPDU contains

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

b) a 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);

Table 9 — Contention winner

Outgoing SPDU from SPM A \ Incoming SPDU from SPM B	RESYNCHRONIZE (abandon)	RESYNCHRONIZE (set)	RESYNCHRONIZE (restart)	ACTIVITY INTERRUPT	ACTIVITY DISCARD
RESYNCHRONIZE (abandon)	initiator	SPM A	SPM A	SPM B	SPM B
RESYNCHRONIZE (set)	SPM B	initiator	SPM A	SPM B	SPM B
RESYNCHRONIZE (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 the note	see the note
ACTIVITY DISCARD	SPM A	SPM A	SPM A	see the note	see the 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.

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

7.23.2 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 "accepting SS-user's choice" 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.

7.23.3 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.

7.24 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.

7.24.1 Content of PREPARE SPDU

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

7.24.2 Sending the PREPARE SPDU

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

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.

Table 10 — 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 SPDU ACK	RESYNCHRONIZE ACK
ACTIVITY END ACK SPDU	MAJOR SYNC ACK

7.24.3 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. If an EXPEDITED DATA SPDU is validly 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.

7.25 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.

7.25.1 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.

7.25.2 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.

7.25.3 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 8.3.25.3.

7.26 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 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.

7.26.1 Content of EXCEPTION DATA SPDU

The EXCEPTION DATA SPDU contains

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

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

7.26.2 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.

7.26.3 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)

7.27 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.

7.27.1 Content of ACTIVITY START SPDU

The ACTIVITY START SPDU contains

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

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

7.27.2 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.

7.27.3 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.

7.28 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.

7.28.1 Content of ACTIVITY RESUME SPDU

The ACTIVITY RESUME SPDU contains

a) a Linking Information parameter group which contains

1) a Called SS-user Reference parameter;

2) a Calling SS-user Reference parameter;

3) a Common Reference parameter;

4) an Additional Reference Information parameter;

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

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

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

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

7.28.2 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.

7.28.3 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.

7.29 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. Use of this procedure may result in loss of data.

7.29.1 Content of ACTIVITY INTERRUPT SPDU

The ACTIVITY INTERRUPT SPDU may contain a Reason Code parameter which indicates the reason for sending the ACTIVITY INTERRUPT SPDU.

7.29.2 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 INTERRUPT ACK and ABORT SPDUs.

7.29.3 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 7.22) 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.

7.30 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.

7.30.1 Content of ACTIVITY INTERRUPT ACK SPDU

The ACTIVITY INTERRUPT ACK SPDU contains no parameters.

7.30.2 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.

7.30.3 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 7.23) 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.

7.31 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. Use of this procedure may result in the loss of data.

7.31.1 Content of ACTIVITY DISCARD SPDU

The ACTIVITY DISCARD SPDU may contain a Reason Code parameter which indicates the reason for sending the ACTIVITY DISCARD SPDU.

7.31.2 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.

7.31.3 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 7.22) 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.

7.32 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.

7.32.1 Content of ACTIVITY DISCARD ACK SPDU

The ACTIVITY DISCARD ACK SPDU contains no parameters.

7.32.2 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.

7.32.3 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 7.23) 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.

7.33 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.

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

7.33.1 Content of ACTIVITY END SPDU

The ACTIVITY END SPDU contains

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

7.33.2 Sending the ACTIVITY END SPDU

An S-ACTIVITY-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.

7.33.3 Receiving the ACTIVITY END SPDU

A valid incoming ACTIVITY END SPDU [when Vact is true, and 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.

7.34 ACTIVITY END ACK SPDU

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

7.34.1 Content of ACTIVITY END ACK SPDU

The ACTIVITY END ACK SPDU contains

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

7.34.2 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.

7.34.3 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.

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).

NOTES

1 The TSDU structure is illustrated in figure 4. The integrity of this structure is maintained over a transport connection. This International Standard does not define the way in which the TSDU is transmitted.

2 When the structure of an SPDU is illustrated in this International Standard, 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.

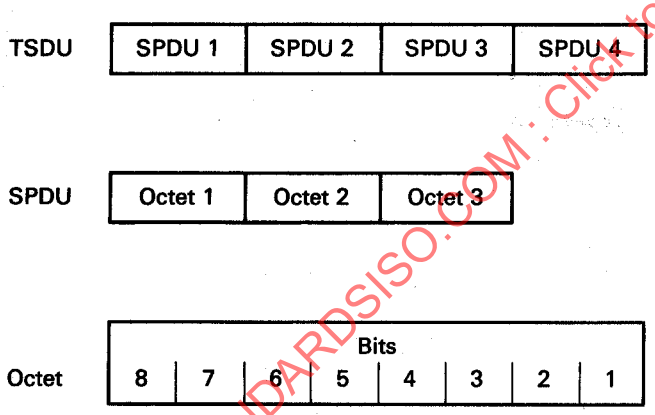


Figure 4 — Illustration of definition of TSDU structure

8.2 SPDU structure

This sub-clause specifies the general structure of SPDUs in terms of their constituent fields. This structure is illustrated in figure 5.

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

Examples of valid SPDU structure are illustrated in figure 6.

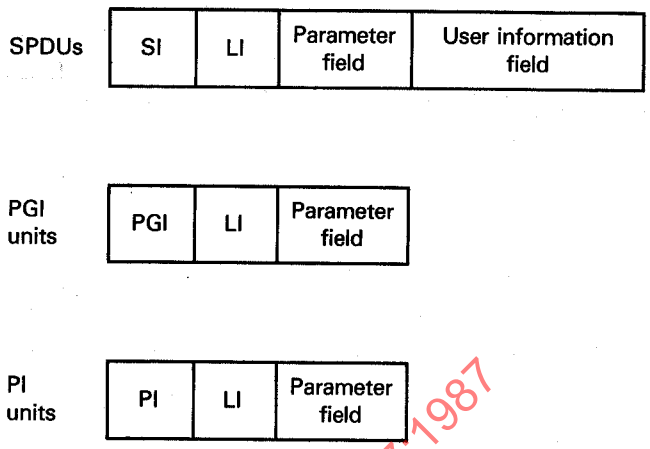


Figure 5 — Illustration of structure of SPDUs, PGI and PI units

8.2.1 SPDUs

SPDUs shall contain, in the following order :

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

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

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 CCITT Recommendation T.62. The protocol specified in this International Standard 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;

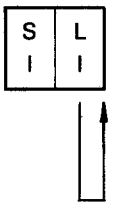
b) the LI field that indicates the length of the associated parameter field defined in 8.2.2c);

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

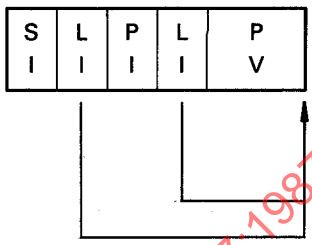
1) a single parameter value (see the note); or

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

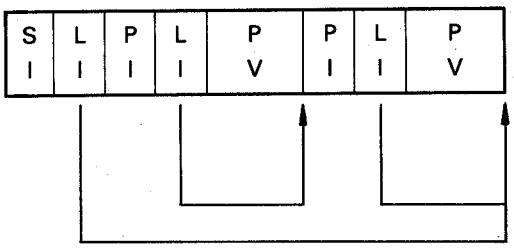
NOTE — A PGI unit with one parameter is structurally equivalent to a PI unit, but the distinction has been retained in order to maintain compatibility with CCITT Recommendation T.62.



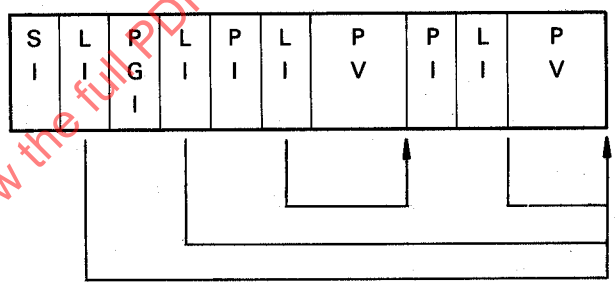
a) LI = 0 (i.e. no parameter field).



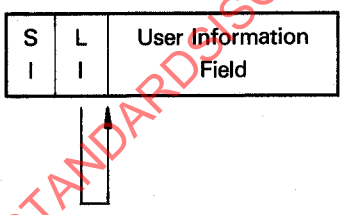
b) LI = 3 and 1 respectively means one parameter of one octet value.



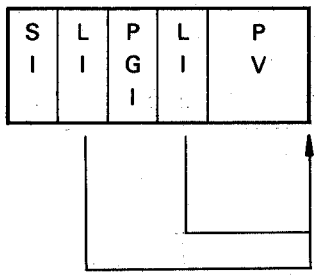
c) Two parameters of 1 and 3 octets respectively. LIs are 8, 1 and 3 respectively.



d) One PGI with two enclosed PIs.



e) The most simple carrier of user information



f) PGI without PI.

Figure 6 — Examples of SPDU structure

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 8.2.3c);

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 the note). A value of zero indicates that the associated parameter field is absent.

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

LI fields indicating lengths within the range 255 to 65535 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 the 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 SSDUs 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

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.

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 to 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 the use of segmenting is proposed. If the TSDU Maximum Size PV is present

a) the first two octets of the PV field shall contain the proposed maximum TSDU size, expressed in octets, in the

Table 11 — 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	7.1.1a)1) 8.3.1.3
			Common Reference	nm	11	64 octets maximum	7.1.1a)2) 8.3.1.4
			Additional Reference Information	nm	12	4 octets maximum	7.1.1a)3) 8.3.1.5
Connect/Accept Item (see 8.3.1.6)	nm	5	Protocol Options	m	19	1 octet	7.1.1b)1) 8.3.1.7
			TSDU Maximum Size	nm	21	4 octets	7.1.1b)2) 8.3.1.8
			Version Number	m	22	1 octet	7.1.1b)3) 8.3.1.9
			Initial Serial Number	nm	23	6 octets maximum	7.1.1b)4) 8.3.1.10
			Token Setting Item	nm	26	1 octet	7.1.1b)5) 8.3.1.11
			Session User Requirements	nm	20	2 octets	7.1.1c) 8.3.1.12
			Calling Session Selector	nm	51	16 octets maximum	7.1.1d) 8.3.1.13
			Called Session Selector	nm	52	16 octets maximum	7.1.1d) 8.3.1.14
User Data	nm	193				512 octets maximum	7.1.1e) 8.3.1.15

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

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 the TSDU maximum size parameter is absent, there shall be no segmenting of SSDUs over the session connection. If either pair of octets has the value zero, there shall be no segmenting in the direction associated with that pair of octets.

8.3.1.9 In the Version Number PV field bit 1 shall have the value 1, indicating that this version of the International Standard is implemented. All other bits are reserved. If this PI unit or PV field is absent, the default shall be this version of this International Standard.

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.

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 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 to 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.

8.3.2 ACCEPT (AC) SPDU

8.3.2.1 The SI field shall contain the value 14.

8.3.2.2 The parameter fields shall be as specified in table 12.

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

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

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

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

Table 12 — 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	7.2.1a)1) 8.3.2.3
			Common Reference	nm	11	64 octets maximum	7.2.1a)2) 8.3.2.4
			Additional Reference Information	nm	12	4 octets maximum	7.2.1a)3) 8.3.2.5
Connect/Accept Item (see 8.3.2.6)	nm	5	Protocol Options	m	19	1 octet	7.2.1b)1) 8.3.2.7
			TSDU Maximum Size	nm	21	4 octets	7.2.1b)2) 8.3.2.8
			Version Number	m	22	1 octet	7.2.1b)3) 8.3.2.9
			Initial Serial Number	nm	23	6 octets maximum	7.2.1b)4) 8.3.2.10
			Token Setting Item	nm	26	1 octet	7.2.1b)5) 8.3.2.11
			Token Item	nm	16	1 octet	7.2.1c) 8.3.2.12
			Session User Requirements	nm	20	2 octets	7.2.1d) 8.3.2.13
			Calling Session Selector	nm	51	16 octets maximum	7.2.1e) 8.3.2.14
			Responding Session Selector	nm	52	16 octets maximum	7.2.1e) 8.3.2.15
User Data	nm	193				512 octets maximum	7.2.1f) 8.3.2.16

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

8.3.2.7 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.

8.3.2.8 The TSDU Maximum Size parameter shall be present if the use of segmenting is proposed by the receiver. The encoding and default for this field is defined in 8.3.1.8.

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

8.3.2.10 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.

8.3.2.11 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 shall be returned. The value "called SS-user's choice" 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.

8.3.2.12 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.

8.3.2.13 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 shall have been set in the CONNECT SPDU. The encoding and default value is defined in 8.3.1.12.

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

8.3.2.15 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 was set to a null value.

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

8.3.3 REFUSE (RF) SPDU

8.3.3.1 The SI field shall contain the value 12.

8.3.3.2 The parameter fields shall be as specified in table 13.

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

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

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

8.3.3.6 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 to 8 are reserved.

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

8.3.3.7 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).

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

8.3.3.9 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 called SS; 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 up to 512 octets of user data;

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;

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).

8.3.4 FINISH (FN) SPDU

8.3.4.1 The SI field shall contain the value 9.

8.3.4.2 The parameter fields shall be as specified in table 14.

Table 13 — 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	7.3.1a)1) 8.3.3.3
			Common Reference	nm	11	64 octets maximum	7.3.1a)2) 8.3.3.4
			Additional Reference Information	nm	12	4 octets maximum	7.3.1a)3) 8.3.3.5
			Transport Disconnect	nm	17	1 octet	7.3.1b) 8.3.3.6
			Session User Requirements	nm	20	2 octets	7.3.1c) 8.3.3.7
			Version Number	nm	22	1 octet	7.3.1d) 8.3.3.8
			Reason Code	nm	50	513 octets maximum	7.3.1.e) 8.3.3.9

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 14 — Parameters of the FINISH SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Transport Disconnect	nm	17	1 octet	7.4.1a) 8.3.4.3
User Data	nm	193				512 octets maximum	7.4.1b) 8.3.4.4

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

8.3.4.3 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 to 8 are reserved.

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

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

8.3.5 DISCONNECT (DN) SPDU

8.3.5.1 The SI field shall contain the value 10.

8.3.5.2 The parameter field shall be as specified in table 15.

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

8.3.6 NOT FINISHED (NF) SPDU

8.3.6.1 The SI field shall contain the value 8.

8.3.6.2 The parameter field shall be as specified in table 16.

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

8.3.7 ABORT (AB) SPDU

8.3.7.1 The SI field shall contain the value 25.

8.3.7.2 The parameter fields shall be as specified in table 17.

Table 15 — Parameters of the DISCONNECT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
User Data	nm	193				512 octets maximum	7.5.1 8.3.5.3

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 16 — Parameters of the NOT FINISHED SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
User Data	nm	193				512 octets maximum	7.6.1 8.3.6.3

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 17 — Parameters of the ABORT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Transport Disconnect	m	17	1 octet	7.7.1a) 8.3.7.3
			Reflect Parameter Values	nm	49	9 octets maximum	7.7.1b) 8.3.7.4
User Data	nm	193				9 octets maximum	7.7.1c) 8.3.7.5

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

8.3.7.3 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 8.3.7.5);

d) bit 3 = 1 : protocol error (see 8.3.7.4);

e) bit 4 = 1 : *no reason*.

Bits 5 to 8 are reserved.

8.3.7.4 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.

8.3.7.5 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.

8.3.8 ABORT ACCEPT (AA) SPDU

8.3.8.1 The SI field shall contain the value 26.

8.3.8.2 There is no parameter field associated with this SPDU.

8.3.9 DATA TRANSFER (DT) SPDU

8.3.9.1 The SI field shall contain the value 1.

8.3.9.2 The parameter field shall be as specified in table 18.

8.3.9.3 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 to 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).

8.3.9.4 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.

8.3.10 EXPEDITED (EX) SPDU

8.3.10.1 The SI field shall contain the value 5.

8.3.10.2 This SPDU contains only a User Information Field as specified in table 19.

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

Table 18 — Parameters of the DATA TRANSFER SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.9.1a) 8.3.9.3
User Information Field						Unlimited	7.9.1b) 8.3.9.4

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 19 — Parameters of the EXPEDITED SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
User Information Field						14 octets maximum	7.10.1 8.3.10.3

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

8.3.11 TYPED DATA (TD) SPDU

8.3.11.1 The SI field shall contain the value 33.

8.3.11.2 The parameter field shall be as specified in table 20.

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 to 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 = 2.

8.3.12 CAPABILITY DATA (CD) SPDU

8.3.12.1 The SI field shall contain the value 61.

8.3.12.2 The parameter field shall be as specified in table 21.

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

8.3.13 CAPABILITY DATA ACK (CDA) SPDU

8.3.13.1 The SI field shall contain the value 62.

8.3.13.2 The parameter field shall be as specified in table 22.

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

Table 20 — Parameters of the TYPED DATA SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Enclosure Item	nm	25	1 octet	7.11.1a) 8.3.11.3
User Information Field						Unlimited	7.11.1b) 8.3.11.4

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 21 — Parameters of the CAPABILITY DATA SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
User Data	nm	193				512 octets maximum	7.12.1 8.3.12.3

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 22 — Parameters of the CAPABILITY DATA ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
User Data	nm	193				512 octets maximum	7.13.1 8.3.13.3

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

8.3.14 GIVE TOKENS (GT) SPDU

8.3.14 The SI field shall contain the value 1.

8.3.14.2 The parameter field shall be as specified in table 23.

8.3.14.3 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.

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

8.3.15 PLEASE TOKENS (PT) SPDU

8.3.15.1 The SI field shall contain the value 2.

8.3.15.2 The parameter fields shall be as specified in table 24.

8.3.15.3 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.

8.3.15.4 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.

8.3.15.5 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 and 8. In this case, the SPDU does not achieve any Please Token function.

Table 23 — Parameters of the GIVE TOKENS SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Token Item	nm	16	1 octet	7.14.1 8.3.14.3

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 24 — Parameters of the PLEASE TOKENS SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Token Item	nm	16	1 octet	7.15.1a) 8.3.15.3
User Data	nm	193				512 octets maximum	7.15.1b) 8.3.15.4

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

8.3.16 GIVE TOKENS CONFIRM (GTC) SPDU

8.3.16.1 The SI field shall contain the value 21.

8.3.16.2 There is no parameter field associated with this SPDU.

8.3.17 GIVE TOKENS ACK (GTA) SPDU

8.3.17.1 The SI field shall contain the value 22.

8.3.17.2 There is no parameter field associated with this SPDU.

8.3.18 MINOR SYNC POINT (MIP) SPDU

8.3.18.1 The SI field shall contain the value 49.

8.3.18.2 The parameter fields shall be as specified in table 25.

8.3.18.3 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 to 8 are reserved.

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

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

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

8.3.19 MINOR SYNC ACK (MIA) SPDU

8.3.19.1 The SI field shall contain the value 50.

8.3.19.2 The parameter fields shall be as specified in table 26.

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

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

Table 25 — Parameters of the MINOR SYNC POINT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Sync Type Item	nm	15	1 octet	7.18.1a) 8.3.18.3
			Serial Number	m	42	6 octets maximum	7.18.1b) 8.3.18.4
User Data	nm	193				512 octets maximum	7.18.1c) 8.3.18.5

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 26 — Parameters of the MINOR SYNC ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Serial Number	m	42	6 octets maximum	7.19.1a) 8.3.19.3
			User Data	nm	46	512 octets maximum	7.19.1b) 8.3.19.4

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

8.3.20 MAJOR SYNC POINT (MAP) SPDU

8.3.20.1 The SI field shall contain the value 41. This is the same value as the SI field for the ACTIVITY END SPDU (see 8.3.33).

8.3.20.2 The parameter fields shall be as specified in table 27.

8.3.20.3 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 to 8 are reserved.

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.

8.3.21 MAJOR SYNC ACK (MAA) SPDU

8.3.21.1 The SI field shall contain the value 42.

8.3.21.2 The parameter fields shall be as specified in table 28.

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.

NOTE — This SPDU is identical to the ACTIVITY END ACK SPDU (see 8.3.34).

Table 27 — Parameters of the MAJOR SYNC POINT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Sync Type Item	m	15	1 octet	7.20.1a) 8.3.20.3
			Serial Number	m	42	6 octets maximum	7.20.1b) 8.3.20.4
User Data	nm	193				512 octets maximum	7.20.1c) 8.3.20.5

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 28 — Parameters of the MAJOR SYNC ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Serial Number	m	42	6 octets maximum	7.21.1a) 8.3.21.3
User Data	nm	193				512 octets maximum	7.21.1b) 8.3.21.4

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

8.3.22 RESYNCHRONIZE (RS) SPDU

8.3.22.1 The SI field shall contain the value 53.

8.3.22.2 The parameter fields shall be as specified in table 29.

8.3.22.3 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.

8.3.22.4 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.

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

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

8.3.23 RESYNCHRONIZE ACK (RA) SPDU

8.3.23.1 The SI field shall contain the value 34.

8.3.23.2 The parameter fields shall be as specified in table 30.

Table 29 — Parameters of the RESYNCHRONIZE SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Token Setting Item	nm	26	1 octet	7.22.1a) 8.3.22.3
			Resync Type	m	27	1 octet	7.22.1b) 8.3.22.4
			Serial Number	m	42	6 octets maximum	7.22.1c) 8.3.22.5
User Data	nm	193				512 octets maximum	7.22.1d) 8.3.22.6

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 30 — Parameters of the RESYNCHRONIZE ACK SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Token Setting Item	nm	26	1 octet	7.23.1a) 8.3.23.3
			Serial Number	m	42	6 octets maximum	7.23.1b) 8.3.23.4
User Data	nm	193				512 octets maximum	7.23.1c) 8.3.23.5

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

8.3.23.3 The Token Setting Item PV field indicates token settings for each token available on the session connection. The bits and encoding are defined in 8.3.22.3. In 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 shall be returned.

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

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

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

8.3.24 PREPARE (PR) SPDU

8.3.24.1 The SI field shall contain the value 7.

8.3.24.2 The parameter field shall be as specified in table 31.

8.3.24.3 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.

All other values are reserved and shall not be used.

8.3.25 EXCEPTION REPORT (ER) SPDU

8.3.25.1 The SI field shall contain the value 0.

8.3.25.2 The parameter field shall be as specified in table 32.

8.3.25.3 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 $1\,024 \leqslant n \leqslant 65\,531$.

NOTE — Not all implementations may be able to deal with field lengths greater than 1 024. 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.

Table 31 — Parameters of the PREPARE SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Prepare Type	m	24	1 octet	7.24.1 8.3.24.3

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 32 — 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	7.25.1 8.3.25.3

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

8.3.26 EXCEPTION DATA (ED) SPDU

8.3.26.1 The SI field shall contain the value 48.

8.3.26.2 The parameter fields shall be as specified in table 33.

8.3.26.3 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.

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

8.3.27 ACTIVITY START (AS) SPDU

8.3.27.1 The SI field shall contain the value 45.

8.3.27.2 The parameter fields shall be as specified in table 34.

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

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

8.3.28 ACTIVITY RESUME (AR) SPDU

8.3.28.1 The SI field shall contain the value 29.

8.3.28.2 The parameter fields shall be as specified in table 35.

8.3.28.3 The Called SS-user Reference PV field shall be as defined by the sending SS-user.

Table 33 — Parameters of the EXCEPTION DATA SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Reason Code	m	50	1 octet	7.26.1a) 8.3.26.3
User data	nm	193				512 octets maximum	7.26.1b) 8.3.26.4

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 34 — Parameters of the ACTIVITY START SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Activity Identifier	m	41	6 octets maximum	7.27.1a) 8.3.27.3
User Data	nm	193				512 octets maximum	7.27.1b) 8.3.27.4

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

8.3.28.4 The Calling SS-user Reference PV field shall be as defined by the sending SS-user.

8.3.28.5 The Common Reference PV field shall be as defined by the sending SS-user.

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

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

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

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

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

8.3.29 ACTIVITY INTERRUPT (AI) SPDU

8.3.29.1 The SI field shall contain the value 25.

8.3.29.2 The parameter fields shall be as specified in table 36.

8.3.29.3 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 35 — 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	7.28.1a)1) 8.3.28.3
			Calling SS-user Reference	nm	10	64 octets maximum	7.28.1a)2) 8.3.28.4
			Common Reference	nm	11	64 octets maximum	7.28.1a)3) 8.3.28.5
			Additional Reference Information	nm	12	4 octets maximum	7.28.1a)4) 8.3.28.6
			Old Activity Identifier	m	41	6 octets maximum	7.28.1a)5) 8.3.28.7
			Serial Number	m	42	6 octets maximum	7.28.1a)6) 8.3.28.8
			New Activity Identifier	m	41	6 octets maximum	7.28.1b) 8.3.28.9
User Data	nm	193				512 octets maximum	7.28.1c) 8.3.28.10

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 36 — Parameters of the ACTIVITY INTERRUPT SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Reason Code	nm	50	1 octet	7.29.1 8.3.29.3

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

8.3.30 ACTIVITY INTERRUPT ACK (AIA) SPDU

8.3.30.1 The SI field shall contain the value 26.

8.3.30.2 There is no parameter field associated with this SPDU.

8.3.31 ACTIVITY DISCARD (AD) SPDU

8.3.31.1 The SI field shall contain the value 57.

8.3.31.2 The parameter field shall be as specified in table 37.

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.

8.3.32 ACTIVITY DISCARD ACK (ADA) SPDU

8.3.32.1 The SI field shall contain the value 58.

8.3.32.2 There is no parameter field associated with this SPDU.

8.3.33 ACTIVITY END (AE) SPDU

8.3.33.1 The SI field shall contain the value 41. This is the same value as the SI field for the MAJOR SYNC POINT SPDU (see 8.3.20).

8.3.33.2 The parameter fields shall be as specified in table 38.

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

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

8.3.34 ACTIVITY END ACK (AEA) SPDU

The ACTIVITY END ACK SPDU is identical to the MAJOR SYNC ACK SPDU (see 8.3.21).

Table 37 — Parameters of the ACTIVITY DISCARD SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Reason Code	nm	50	1 octet	7.31.1 8.3.31.3

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Table 38 — Parameters of the ACTIVITY END SPDU

PGI	m/nm	Code	PI	m/nm	Code	Length	Reference
			Serial Number	m	42	6 octets maximum	7.34.1a) 8.3.33.3
User Data	nm	193				512 octets maximum	7.34.1b) 8.3.33.4

Key :

m : mandatory

nm : not mandatory (see 8.2.6)

Section three : Conformance

9 Conformance to this International Standard

9.1 A system claiming conformance to this International Standard 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 The system shall be capable of

a) initiating a session connection (by sending a CONNECT SPDU) or responding to a CONNECT SPDU (by sending an ACCEPT SPDU or a REFUSE SPDU), 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.

Annex A

State tables

(This annex forms part of the standard.)

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 resultant 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 clause 7. In case of arbitration or dispute, this annex takes precedence over clause 7.

Table 39 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 40 specifies the abbreviated name and name of each state.

Table 41 specifies the abbreviated name, category and name of each outgoing event. The categories are SS-provider event, TS-user event and SPDU event.

Table 42 summarizes the operations on the variables V(A), V(M), V(R) and Vsc.

Table 43 specifies the specific actions.

Table 44 specifies the predicates.

Tables 45 to 53 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 43.

A.2.3 Notes are represented by the notation (n), where n is the number of the note at the foot of table 44.

A.2.4 Predicates are represented by the notation pn, where n is the number of the predicate in table 44.

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.3.2a)].

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 left-hand 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 states 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.4.3 Receipt of SPDUs

A.4.3.1 Valid SPDUs

The SPM shall process valid SPDUs as specified in tables 45 to 53.

This International Standard 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 the notes).

An SPM receiving 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 the notes).

The SPM shall ignore any bits within a parameter field which are specified as reserved in 8.3.

NOTES

1 The received SPDU is processed as though the unknown PGI and/or PI units were not present in the SPDU.

2 These provisions permit communication with systems operating other versions of this protocol.

A.4.3.2 Invalid SPDUs

If an invalid SPDU is received, the SPM shall

a) take the actions defined in A.4.1.2a); or

b) take the actions defined in A.4.1.2b); or

c) take any other action that does not violate the procedures specified in this International Standard; or

d) take no action.

A.5 Definitions of sets and variables

The following sets and variables are specified in this International Standared.

A.5.1 Functional units

The set of all functional units specified in this International Standard 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 exchange 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 International Standard 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; or

5) the ACTIVITY INTERRUPT ACK SPDU is sent or received; or

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 Initiating 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 Accepting 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 Initiating 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 Accepting 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 sets 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 (for example 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 no 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 STA713.

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 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. Vsc is set true when a MINOR SYNC POINT SPDU is received.

NOTE — Table 42 summarizes the operations on V(A), V(M), V(R) and Vsc.

Table 39 — 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
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

Table 39 (*concluded*)

Abbreviated name	Category	Name and description
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
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
MIP	SPDU	MINOR SYNC POINT SPDU
NF	SPDU	NOT FINISHED 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

NOTES

1 If the Token Item parameter is present in the ACCEPT SPDU, both the AC event and the PT event occur.

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 40 — States

Abbreviated name	Name and description
STA01	Idle, no transport connection
STA01A	Wait for the ABORT ACCEPT SPDU
STA01B	Wait for T-CONNECT confirm
STA01C	Idle, transport connected
STA02A	Wait for the ACCEPT SPDU
STA03	Wait for the DISCONNECT SPDU
STA04A	Wait for the MAJOR SYNC ACK SPDU or PREPARE (MAJOR SYNC ACK) SPDU
STA04B	Wait for the ACTIVITY END ACK SPDU or PREPARE (MAJOR SYNC ACK) SPDU
STA05A	Wait for the RESYNCHRONIZE ACK SPDU or PREPARE (RESYNCHRONIZE ACK) SPDU
STA05B	Wait for the ACTIVITY INTERRUPT ACK SPDU or PREPARE (RESYNCHRONIZE ACK) SPDU
STA05C	Wait for the ACTIVITY DISCARD ACK SPDU or PREPARE (RESYNCHRONIZE ACK) SPDU
STA06	Wait for the RESYNCHRONIZE SPDU (resynchronization collision after receiving PREPARE (RESYNCHRONIZE) SPDU)
STA08	Wait for S-CONNECT response
STA09	Wait for S-RELEASE response
STA10A	Wait for S-SYNC-MAJOR response
STA10B	Wait for S-ACTIVITY-END response
STA11A	Wait for S-RESYNCHRONIZE response
STA11B	Wait for S-ACTIVITY-INTERRUPT response
STA11C	Wait for S-ACTIVITY-DISCARD response
STA15A	After PREPARE, wait for the MAJOR SYNC ACK SPDU or the ACTIVITY END ACK SPDU
STA15B	After PREPARE, wait for the RESYNCHRONIZE SPDU or the ACTIVITY INTERRUPT SPDU or the ACTIVITY DISCARD SPDU
STA15C	After PREPARE, wait for the RESYNCHRONIZE ACK SPDU or the ACTIVITY INTERRUPT ACK SPDU or the ACTIVITY DISCARD SPDU
STA16	Wait for T-DISCONNECT indication
STA18	Wait for the GIVE TOKENS ACK SPDU
STA19	Wait for a recovery request or SPDU (initiator of EXCEPTION DATA SPDU)
STA20	Wait for a recovery SPDU or request
STA21	Wait for the CAPABILITY DATA ACK SPDU
STA22	Wait for S-CAPABILITY-DATA response
STA713	Data transfer state

Table 41 — 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
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

Table 41 (*concluded*)

Abbreviated name	Category	Name and description
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
CN	SPDU	CONNECT SPDU
DN	SPDU	DISCONNECT SPDU
DT	SPDU	DATA TRANSFER 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
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
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 42 — Operations on variables

Events	Condition for valid SPDU or primitive	Condition for update of variables	Operations on variables			
			V(A)	V(M)	V(R)	Vsc
SSYNMreq SSYNmreq SACTEreq		if Vsc true	set to V(M)	V(M) + 1	unchanged	false
		if Vsc false	unchanged	V(M) + 1	unchanged	false
MAP SPDU AE SPDU	sn = V(M)	if Vsc true	unchanged	V(M) + 1	unchanged	unchanged
		if Vsc false	set to V(M)	V(M) + 1	unchanged	unchanged
MIP SPDU	sn = V(M)	if Vsc true	unchanged	V(M) + 1	unchanged	true
		if Vsc false	set to V(M)	V(M) + 1	unchanged	true
SSYNMrsp SACTErsp MAA SPDU AEA SPDU	sn = V(M) − 1		set to V(M)	unchanged	set to V(M)	unchanged
SSYNmrsp	Vsc = true and V(M) > sn ⩾ V(A)*		set to sn + 1	unchanged	unchanged	unchanged
MIA SPDU	Vsc = false and V(M) > sn ⩾ V(A)*		set to sn + 1	unchanged	unchanged	unchanged
SRSYNreq	a : not applicable r : V(M) ⩾ sn ⩾ V(R) s : sn ⩽ 999 999	abandon restart set	unchanged unchanged unchanged	unchanged unchanged unchanged	unchanged unchanged unchanged	unchanged unchanged unchanged
RS SPDU	a : sn ⩽ 999 999 r : sn ⩾ V(R) s : sn ⩽ 999 999	abandon restart set	unchanged unchanged unchanged	max (sn, V(M)) unchanged unchanged	unchanged unchanged unchanged	unchanged unchanged unchanged
SRSYNrsp	a : sn = V(M) r : sn as in RS SPDU s : sn ⩽ 999 999	abandon restart set	set to sn set to sn set to sn	set to sn set to sn set to sn	set to 0 unchanged set to 0	unchanged unchanged unchanged
RA SPDU	a : sn ⩾ V(M) r : sn as in RS SPDU s : sn ⩽ 999 999	abandon restart set	set to sn set to sn set to sn	set to sn set to sn set to sn	set to 0 unchanged set to 0	unchanged unchanged unchanged
SACTRreq AR SPDU			set to sn + 1	set to sn + 1	set to 1	unchanged
SACTSreq AS SPDU			set to 1	set to 1	set to 1	unchanged
SCONrsp+ AC SPDU		sn present	set to sn	set to sn	set to 0	false

Key :

sn : synchronization point serial number quoted in SS-user request or SPDU

⩾ : greater than or equal to

⩽ : less than or equal to

* : synchronization point serial number not equal to V(M) − 1 if major synchronization or activity end outstanding

Table 43 — 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

Table 44 — 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(ma)
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)
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)

Table 44 (*concluded*)

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

NOTES TO TABLES 45 TO 53

1 PR is not sent if TEXP is false.

2 The serial number given in the indication is V(M).

3 SxABind means generate event SUABind if bit 2 of the Transport Disconnect PV field in the ABORT SPDU has the value "user abort". Otherwise, SxABind means generate the event SPABind.

Table 45 — Connection establishment state table

State / Event	STA01 idle no TC	STA01A await AA	STA01B await TCONcnf	STA01C idle TC con	STA02A await AC	STA08 await SCONrsp	STA16 await TDISind
AC	//	STA01A	//	TDISreq STA01	SCONcnf+ [5] [11] STA713 [6]		STA16
CN	//	TDISreq [3] STA01	//	^p01 SCONind STA08 p01 TDISreq STA01			TDISreq [3] STA01
RF-nr	//	STA01A	//	TDISreq STA01	SCONcnf– TDISreq STA01		STA16
RF-r	//	STA01A	//	TDISreq STA01	^p02 SCONcnf– TDISreq STA01 p02 SCONcnf– STA01C		STA16
SCONreq	TCONreq [2] STA01B			p01 CN STA02A			
SCONrsp+						AC [5] [11] STA713	
SCONrsp–						^p02 RF-nr [4] STA16 p02 RF-r STA01C	
TCONcnf	//	//	CN STA02A	//	//	//	//
TCONind	TCONrsp [1] STA01C	//	//	//	//	//	//

Table 46 — Data transfer state table

State / Event	STA01A await AA	STA01C idle TC con	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		p05 & p10 SDTind STA03	p05 SDTind STA04	p05 SDTind STA04B	p05 STA05A	p05 STA05B
EX	STA01A	TDISreq STA01	[10] STA02A	p09 SEXind STA03	p08 SEXind STA04A	p08 SEXind STA04B	p08 STA05A	p08 STA05B
TD	STA01A	TDISreq STA01		p06 & p10 STDind STA03	p06 STDind STA04A	p06 STDind STA04B	p06 STA05A	p06 STA05B
SDTreq								
SEXreq								
STDreq								

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

State / Event	STA15C wait after PR-RA	STA16 await TDISind	STA18 await GTA	STA19 await recovery (init)	STA20 await recovery	STA21 await CDA	STA713 data transfer
DT	p05 STA15C	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	STA16	p06 STDind STA18	p06 STA19	p06 STA20	p06 STDind STA21	p06 STDind STA713
SDTreq			p70 DT STA18				p03 DT STA713
SEXreq			p08 EX STA18				p08 EX STA713
STDreq			p06 TD STA18				p06 TD STA713

Table 47 — Synchronization state table

State / Event	STA01A await AA	STA01C idle TC con	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	p16 & p20 SSYNMcnf [14] [22] STA713	p16 & p20 SACTEcnf [14] [22] STA713	STA05A	STA05B	STA05C
MAP	STA01A	TDISreq STA01			p12 STA05A		
PR-MAA	STA01A	TDISreq STA01	STA15A	STA15A	STA05A	STA05B	STA05C
SSYNMreq							
SSYNMrsp							

State / Event	STA06 await after coll	STA10A await SSYNMrsp	STA15A wait after PR-MAA	STA15B wait after PR-RS	STA15C wait after PR-RA
MAA or AEA	STA06		p20 & ^p23 SSYNMcnf [14] [22] STA713 [6] p20 & p23 SACTEcnf [14] [22] STA713 [6]	STA15B	STA15C
MAP	p12 STA06			p12 STA15B	p12 STA15C
PR-MAA					
SSYNMreq				p13 STA15B	
SSYNMrsp		PR-MAA (1) MAA [14] [22] STA713		STA15B	

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 47 (*continued*)

State / Event	STA01A await AA	STA01C idle TC con	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA	STA05A await PR or RA
AE	STA01A	TDISreq STA01				p72 STA05A
MIA	STA01A	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				p14 STA05A
SACTEreq						
SACTErsp						
SSYNmreq						
SSYNmrsp						

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

State / Event	STA10B await SACTErsp	STA15A wait after PR-MAA	STA15B wait after PR-RS	STA15C wait after PR-RA	STA16 await TDISind
AE			p72 STA15B	p72 STA15C	STA16
MIA		p17 & ^p20 & p21 SSYNmcnf [25] STA15A	p17 STA15B	p17 STA15C	STA16
MIP			p14 STA15B	p14 STA15C	STA16
SACTEreq			p71 STA15B		
SACTErsp	PR-MAA (1) AEA [14] [22] STA713				
SSYNmreq			p15 STA15B		
SSYNmrsp	p18 & ^p20 & p21 MIA [25] STA10B		p18 & p21 STA15B		

Table 47 (*concluded*)

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 48 — Resynchronization state table

State / Event	STA01A await AA	STA01C idle TC con	STA02A await AC	STA03 await DN	STA04A await PR or MAA	STA04B await PR or AEA
PR-RA	STA01A	TDISreq STA01				
PR-RS	STA01A	TDISreq STA01	[10] STA02A	p10 STA15B	STA15B	STA15B
RA	STA01A	TDISreq STA01				
RS-a	STA01A	TDISreq STA01		p10 & ^p34 & p35 [19] SRSYNind (2) [16] STA11A	p35 [19] SRSYNind (2) [16] STA11A	p35 [19] SRSYNind (2) [16] STA11A
RS-r	STA01A	TDISreq STA01		p10 & ^p34 & p35 & p32 SRSYNind [16] STA11A	p32 & p35 SRSYNind [16] STA11A	p32 & p35 SRSYNind [16] STA11A
RS-s	STA01A	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 48 (*continued*)

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 48 (*continued*)

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
PR-RA						
PR-RS	STA15B	STA15B		[10] STA15A		[10] STA15C
RA						p36 & p49 SRSYNcnf [28] [11] STA713 [6]
RS-a	p35 [19] SRSYNind (2) [16] STA11A				p29 [19] SRSYNind (2) [16] STA11A	
RS-r					p32 & p29 SRSYNind [16] STA11A	
RS-s	p35 SRSYNind [16] STA11A				p29 SRSYNind [16] STA11A	
SRSYNreq(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	
SRSYNreq(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	
SRSYNreq(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	
SRSYNrsp			p43 PR-RA (1) RA [28] [11] STA713			