INTERNATIONAL TELECOMMUNICATION UNION

ITU-T X.215

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(11/95)

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

OPEN SYSTEMS INTERCONNECTION – SERVICE DEFINITIONS

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – SESSION SERVICE DEFINITION

ITU-T Recommendation X.215

(Previously "CCITT Recommendation")

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. Some 179 member countries, 84 telecom operating entities, 145 scientific and industrial organizations and 38 international organizations participate in ITU-T which is the body which sets world telecommunications standards (Recommendations).

The approval of Recommendations by the Members of ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, 1993). In addition, the World Telecommunication Standardization Conference (WTSC), which meets every four years, approves Recommendations submitted to it and establishes the study programme for the following period.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC. The text of ITU-T Recommendation X.215 was approved on the 21st of November 1995. The identical text is also published as ISO/IEC International Standard 8326.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

(February 1994)

ORGANIZATION OF X-SERIES RECOMMENDATIONS

Subject area	Recommendation Series
PUBLIC DATA NETWORKS	
Services and Facilities	X.1-X.19
Interfaces	X.20-X.49
Transmission, Signalling and Switching	X.50-X.89
Network Aspects	X.90-X.149
Maintenance	X.150-X.179
Administrative Arrangements	X.180-X.199
OPEN SYSTEMS INTERCONNECTION	
Model and Notation	X.200-X.209
Service Definitions	X.210-X.219
Connection-mode Protocol Specifications	X.220-X.229
Connectionless-mode Protocol Specifications	X.230-X.239
PICS Proformas	X.240-X.259
Protocol Identification	X.260-X.269
Security Protocols	X.270-X.279
Layer Managed Objects	X.280-X.289
Conformance Testing	X.290-X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300-X.349
Mobile Data Transmission Systems	X.350-X.369
Management	X.370-X.399
MESSAGE HANDLING SYSTEMS	X.400-X.499
DIRECTORY	X.500-X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600-X.649
Naming, Addressing and Registration	X.650-X.679
Abstract Syntax Notation One (ASN.1)	X.680-X.699
OSI MANAGEMENT	X.700-X.799
SECURITY	X.800-X.849
OSI APPLICATIONS	
Commitment, Concurrency and Recovery	X.850-X.859
Transaction Processing	X.860-X.879
Remote Operations	X.880-X.899
OPEN DISTRIBUTED PROCESSING	X.900-X.999

CONTENTS

Summary

This Recommendation | International Standard specifies the services which are provided by the session layer to the presentation layer at the boundary between the session layer and the presentation layer of the OSI reference model. This includes:

a) the primitive actions and events of the service;

b) the parameters associated with each primitive action and event; and

c) the relationship between the valid sequence of these actions and events.

Introduction

This Recommendation | International Standard is one of a set of Recommendations | International Standards produced to facilitate the interconnection of computer systems.

This Recommendation | International Standard is related to other Recommendations | International Standards in the set as defined by the Reference Model for Open Systems Interconnection. The Reference Model subdivides the area of standardization for interconnection into a series of layers of specification, each of manageable size.

The purpose of this Recommendation | International Standard is to define the service provided to the Presentation Layer at the boundary between the Session and Presentation Layers of the Reference Model. The session service is provided by the session protocol making use of the services available from the Transport Layer. This Recommendation | International Standard also defines the session service characteristics which the presentation protocol may exploit. The relationship between the Recommendations | International Standards for the session service, session protocol, transport service, and the presentation protocol is illustrated in Figure Intro.1 below.

It is recognized that, with respect to session Quality of Service (described in clause 10), work is still in progress to provide an integrated treatment of QOS across all of the layers of the OSI Reference Model and to ensure that the individual treatments in each layer service satisfy overall QOS objectives in a consistent manner. As a consequence, an addendum may be added to this Recommendation | International Standard at a later time which reflects further QOS developments and integration.

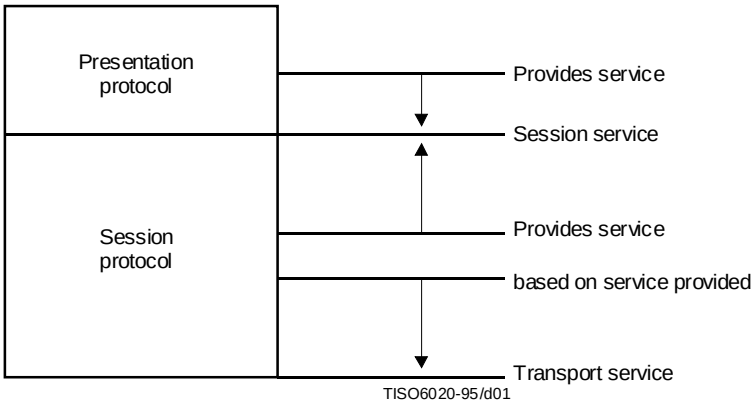


Figure Intro. 1 – Relationship of this Recommendation | International Standard to other OSI Standards

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – OPEN SYSTEMS INTERCONNECTION – SESSION SERVICE DEFINITION

SECTION 1 – GENERAL

1 Scope

This Recommendation | International Standard defines in an abstract way the externally visible service provided by the OSI Session Layer in terms of:

a) the primitive actions and events of the service;

b) the parameter data associated with each primitive action and event;

c) the relationship between, and the valid sequence of these actions and events.

The service defined in this Recommendation | International Standard is that which is provided by the OSI session protocol (in conjunction with the transport service) and which may be used by the OSI presentation protocol.

This Recommendation | International Standard does not specify individual implementations or products, nor does it constrain the implementation of entities and interfaces within a computer system. There is, therefore, no conformance to this Recommendation | International Standard.

The text pertaining to the symmetric synchronization functional unit is not applicable to the support of ITU-T applications.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and International Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and International Standards listed below. Members of ISO and IEC maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

– ITU-T Recommendation X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.*

– ITU-T Recommendation X.210 (1993) | ISO/IEC 10731:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: Conventions for the definition of OSI services.*

– ITU-T Recommendation X.214 (1993) | ISO/IEC 8072:1994, *Information technology – Open Systems Interconnection – Transport service definition.*

– ITU-T Recommendation X.225 (1995) | ISO/IEC 8327-1:1996, *Information technology – Open Systems Interconnection – Connection-oriented Session protocol: Protocol specification.*

– ITU-T Recommendation X.226 (1994) | ISO/IEC 8823-1:1994, *Information technology – Open Systems Interconnection – Connection-oriented presentation protocol: Protocol specification.*

– ITU-T Recommendation X.235 (1995) | ISO/IEC 9548-1:1995, *Information technology – Open Systems Interconnection – Connectionless Session protocol: Protocol specification.*

2.2 Paired Recommendations | International Standards equivalent in technical content

– CCITT Recommendation X.650 (1992), *Open Systems Interconnection (OSI) – Reference Model for naming and addressing.*

ISO 7498-3:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 3: Naming and addressing.*

3 Definitions

For the purposes of this Recommendation | International Standard, the following definitions apply.

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

3.1 Reference Model definitions

This Recommendation | International Standard is based on the concepts developed in ITU-T Rec. X.200 | ISO/IEC 7498-1, and makes use of the following terms defined in it:

a) expedited-session-service-data-unit;

b) session connection;

c) Session Layer;

d) session service;

e) session-service-access-point;

f) session-service-data-unit;

g) Transport Layer;

h) duplex;

i) half-duplex.

3.2 Service convention definitions

This Recommendation | International Standard also makes use of the following terms defined in ITU-T Rec. X.210 | ISO/IEC 10731, as they apply to the Session Layer:

a) service-user;

b) service-provider;

c) primitive;

d) request;

e) indication;

f) response;

g) confirm.

3.3 Session service definitions

3.3.1 **calling SS-user**: An SS-user that initiates a session connection establishment request.

3.3.2 **called SS-user**: An SS-user with whom a calling SS-user wishes to establish a session connection.

NOTE – Calling SS-users and called SS-users are defined with respect to a single connection. An SS-user can be both a calling and called SS-user simultaneously.

3.3.3 sending SS-user: An SS-user that acts as either a source of data during the data transfer phase of a session connection or a source of data during a particular instance of session-connectionless-mode transmission.

3.3.4 receiving SS-user: An SS-user that acts as either a sink of data during the data transfer phase of a session connection or a sink of data during a particular instance of session-connectionless-mode transmission.

NOTE – An SS-user can be both sending and a receiving SS-user simultaneously.

3.3.5 requestor; requesting SS-user: An SS-user that initiates a particular action.

3.3.6 acceptor; accepting SS-user: An SS-user that accepts a particular action.

3.3.7 token: An attribute of a session connection which is dynamically assigned to one SS-user at a time to permit certain services to be invoked.

3.3.8 conditional (parameter): A parameter whose presence in a request or response depends on conditions defined in the text of this Recommendation | International Standard; and whose presence in an indication or confirm is mandatory if that parameter was present in the preceding session service primitive, or absent if that parameter was absent in the preceding session service primitive.

3.3.9 proposed parameter: The value for a parameter proposed by an SS-user in an S-CONNECT request or an S CONNECT response that it wishes to use on the session connection.

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

3.3.11 session-connectionless-mode transmission; session-connectionless transmission: The transmission of a unit of data in a single self-contained session service without establishing a session connection (as defined in ITU-T Rec. X.200 | ISO/IEC 7498-1).

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply.

4.1 Data units

SSDU	Session service data unit
NSSDU	Normal data session service data unit
TSSDU	Typed data session service data unit
XSSDU	Expedited session service data unit

4.2 Miscellaneous

SS	Session service
SSAP	Session service access point
QOS	Quality of Service

4.3 Service variables

V(A)	See 11.4.1.1.1
V(M)	See 11.4.1.1.2
V(R)	See 11.4.1.1.3
Vsc	See 11.4.1.1.4

5 Conventions

This Recommendation | International Standard uses the descriptive conventions contained in the OSI Service Conventions (ITU-T Rec. X.210 | ISO/IEC 10731) except that, where indicated in this Recommendation | International Standard, parameter values associated with a service primitive may be passed in a direction opposite to the direction of the service primitive.

6 Model of the session service

6.1 Model of the layer service

This Recommendation | International Standard uses the abstract model for a layer service defined in ITU-T Rec. X.210 | ISO/IEC 10731. The model defines the interactions between the SS-users and the SS-providers which take place at the two SSAPs. Information to passed between an SS-user and the SS-provider by service primitives, which may convey parameters.

There are two types of session service:

a) A connection-mode service, which is defined in clause 2.

b) A connectionless-mode service, which is defined in clause 3. The connectionless-mode service defines the feature given in 7.1 e). This service is invoked by the connectionless presentation P-UNIT-DATA service.

6.2 Model of session connectionless-mode transmission

This Recommendation | International Standard uses the abstract model of connectionless-mode transmission defined in ITU-T Rec. X.200 | ISO/IEC 7498-1. A defining characteristic of session connectionless-mode transmission is the independent nature of each invocation of the connectionless-mode session service.

7 Overview of the session service

7.1 General overview

The session service provides the means for organized and synchronized exchange of data between co-operating SS-users. It provides its users with means to:

a) establish a connection with another SS-user, exchange data with that user in a synchronized manner, and release the connection in an orderly manner;

b) negotiate for the use of tokens to exchange data, synchronize and release the connection, and to arrange for data exchange to be half-duplex or duplex;

c) establish synchronization points within the dialogue and, in the event of errors, resume the dialogue from an agreed synchronization point;

d) interrupt a dialogue and resume it later at a pre-arranged point.

In addition, the following service is described:

e) a means by which a single unit of data is transmitted from one source SSAP to another SSAP in a single session service access, without first establishing or later releasing a session connection.

7.2 Token concept

A token is an attribute of a session connection which is dynamically assigned to one SS-user at a time to permit certain services to be invoked. It is the right to exclusive use of the service.

Four tokens are defined:

a) the data token;

b) the release token;

c) the synchronize-minor token;

d) the major/activity token.

A token is always in one of the following states:

e) available, in which case it is always

   1) assigned to one SS-user, who then has the exclusive right to use the associated service (provided that no other restrictions apply); and

   2) not assigned to the other SS-user, who does not have the right to use the service but may acquire it later; or

f) not available to either SS-user, in which case neither SS-user has the exclusive use of the associated service. The service then becomes inherently available to both SS-users (data transfer and release), or otherwise unavailable to both SS-users (synchronization and activities).

Restrictions related to the availability and assignment of tokens are defined in 11.2.

7.3 Synchronization and dialogue unit concepts

SS-users may insert synchronization points into the data they are transmitting. There are two methods to identify a synchronization point. A synchronization point can be identified by a single serial number if the right to send all synchronization points is token controlled. Alternatively, synchronization points can be identified by two serial numbers, one for each direction of flow. This dual numbering scheme, called symmetric synchronization, allows the SS-users to independently place synchronization points in their sending flows. Serial numbers are maintained by the SS-provider (see 11.4).

Any semantics which SS-users may give to their synchronization points are transparent to the SS-provider.

There are two types of synchronization points:

a) minor synchronization points;

b) major synchronization points.

Major synchronization points are used to structure the exchange of data into a series of dialogue units. The characteristic of a dialogue unit is that all communication within it is completely separated from all communication before and after it. A major synchronization point indicates the end of one dialogue unit and the beginning of the next. Each major synchronization point is confirmed explicitly.

Minor synchronization points are used to structure the exchange of data within a dialogue unit. Figure 1 illustrates how a dialogue unit is structured through the use of minor synchronization points. Each minor synchronization point may or may not be confirmed explicitly. A minor synchronization point inserted in association with the data separation service protects all data sent before the minor synchronization point from being discarded by a subsequent resynchronize request.

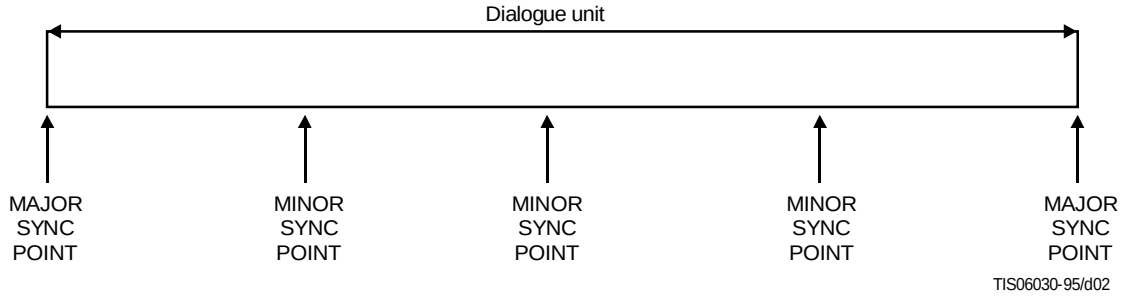


Figure 1 – Example of a structured dialogue unit

7.4 Activity concept

The activity concept allows SS-users to distinguish between different logical pieces of work called activities. Each activity consists of one or more dialogue units. Only one activity is allowed on a session connection at a time, but there may be several consecutive activities during a session connection. An activity may also span more than one session connection. An activity can be interrupted and then resumed on the same or on a subsequent session connection. This can be considered as a form of resynchronization.

Figure 2 shows how an activity may be structured into dialogue units through the use of major synchronization points. In addition, the SS-users may transfer data outside an activity.

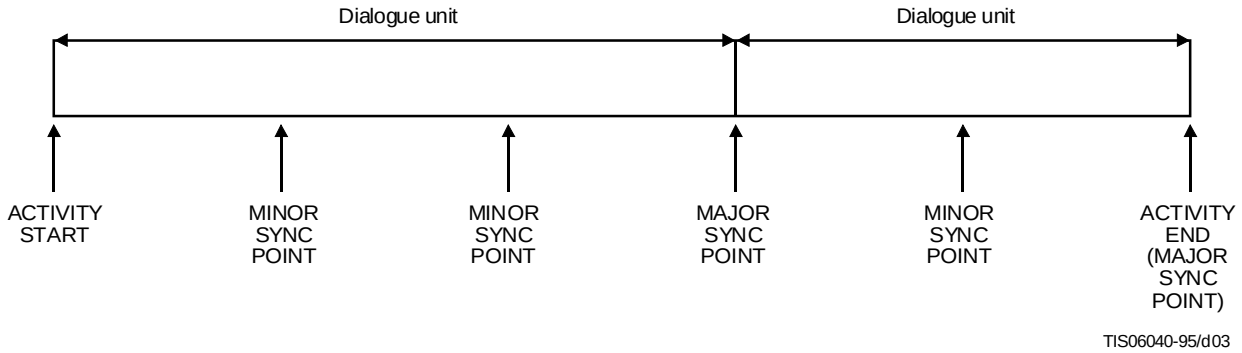


Figure 2 – Example of a structured activity

7.5 Resynchronization

Resynchronization may be initiated by either SS-user. It sets the session connection to a defined state, and therefore includes reassignment of tokens and setting the synchronization point serial number(s) to new value(s).

When symmetric synchronization is used, the SS-user can request to resynchronize one direction of data flow, or both directions. If both directions are requested, then both serial numbers are assigned new values; otherwise only the serial number associated with the requested direction of flow is assigned a new value. Resynchronization purges all undelivered data for the requested direction(s) of flow.

When symmetric synchronization is not used, both directions of flow are always resynchronized. The single synchronization point serial number is assigned a new value. All undelivered data is purged.

Three options are provided:

a) abandon option which is used to set the synchronization point serial number to an unused value;

b) restart option which is used to set the synchronization point serial number to any used value which is greater than the synchronization point serial number which identifies the last acknowledged major synchronization point;

c) set option is used to set the synchronization point serial number to any value chosen by SS-user.

When symmetric synchronization is used, a resynchronize option (restart, set or abandon) is provided by the SS-user for each direction of data flow on which resynchronization is requested.

7.6 Negotiation

Negotiation takes place between both SS-users during the session connection establishment phase according to the following rules.

7.6.1 Negotiation of functional units

The kernel functional unit (see clause 9) is always used. Each SS-user proposes the use or non-use of each of the other functional units. A functional unit is selected only if both SS-users propose use of the functional unit and it is supported by the SS-provider. Specific negotiation rules are given in 12.1.2.

7.6.2 Negotiation of initial token settings

When the calling SS-user proposes use of a functional unit that requires a token, it also proposes the initial token settings:

a) calling SS-user side;

b) called SS-user side;

c) called SS-user choice.

If the use of the functional unit is selected, the token is set to:

d) the side proposed by the called SS-user, if “called SS-user choice” is proposed by the calling SS-user; or

e) in all other cases, the side proposed by the calling SS-user.

7.6.3 Negotiation of initial synchronization point serial number

When a calling SS-user proposes any of the minor synchronize, symmetric synchronize, major synchronize or resynchronize functional units, but does not propose the activity functional unit, it also proposes values for the initial synchronization point serial number(s). Two serial numbers are proposed if the symmetric synchronize functional unit is proposed; otherwise one serial number is proposed. If two serial numbers are proposed, the First Initial Synchronization Point Serial Number is associated with the calling SS-user’s sending flow and the Second Initial Synchronization Point Serial Number is associated with the calling SS-user’s receiving flow.

The calling SS-user may also propose value(s) for the initial synchronization point serial number(s) even if the activity management functional unit is proposed provided that any of the minor synchronize, symmetric synchronize, major synchronize or resynchronize functional units are also proposed. If the called SS-user selects use of any of the minor synchronize, symmetric synchronize, major synchronize or resynchronize functional units, but does not select use of the activity management functional unit, it returns value(s) for the initial synchronization point serial number(s) which may or may not be the same as the values proposed by the calling SS-user. If the symmetric synchronize functional unit is selected, the called SS-user returns values for two serial numbers; otherwise a value for one serial number is returned. The values returned by the called SS-user are used as the initial synchronization point serial number(s) for the new session connection.

In all other combinations of functional units, no initial synchronization point serial number(s) are proposed.

8 Phases and services of the session service

The connection-mode session service comprises three phases. The purpose of each phase, and a short description of the associated services are given in 8.1 to 8.3. The services and the primitives by which they are invoked are defined in clauses 12, 13 and 14.

The connectionless-mode session service is used to transfer a single SSDU from the sending SS-user to the receiving SS-user. The service primitives associated with it are defined in clause 17.

NOTE – The amount of SS-user data which can be transferred in certain primitives may be limited due to protocol restrictions imposed by the SS-provider.

8.1 Session connection establishment phase

The session connection establishment phase is concerned with establishing a connection between two SS-users. It has one service associated with it:

The session connection service (see 12.1) is used to set up a session connection and to negotiate tokens and parameters to be used for the connection.

8.2 Data transfer phase

The data transfer phase is concerned with the exchange of data between the two SS-users connected in the session connection establishment phase.

There are four services associated with data transfer:

a) The normal data transfer service (see 13.1) allows the transfer of normal data SSDUs (NSSDUs) over a session connection. Its use is controlled by the data token if the half-duplex functional unit has been selected.

b) The expedited data transfer service (see 13.2) allows the transfer of expedited SSDUs (XSSDUs) over a session connection free from the token and flow control constraints of the normal data transfer service, typed data transfer service and capability data exchange service.

c) The typed data transfer service (see 13.3) is used to transfer typed data SSDUs (TSSDUs) independent of the availability and assignment of the data token.

d) The capability data exchange service (see 13.4) is used to exchange confirmed SS-user data while not within an activity.

There are three services concerned with token management:

e) The give tokens service (see 13.5) allows an SS-user to surrender one or more specific tokens to the other SS-user.

f) The please tokens service (see 13.6) allows an SS-user to request the other SS-user to transfer one or more specific tokens to it.

g) The give control service (see 13.7) allows an SS-user to surrender all available tokens to the other SS-user.

There are five services associated with synchronization and resynchronization:

h) The minor synchronization point service (see 13.8) allows the SS-user to separate the flow of NSSDUs and TSSDUs transmitted before the service was invoked from the subsequent flow of NSSDUs and TSSDUs. Its use is controlled by the synchronize-minor token.

i) The symmetric synchronization point service (see 13.8) allows the SS-user to independently define minor synchronization points on each SS-user's sending flow. The synchronize-minor token is not available.

j) The data separation service (see 13.8) adds functionality to the minor synchronization point service and the symmetric synchronize service to protect all data sent before the minor synchronization point from being discarded by a subsequent resynchronize request. There are no additional services associated with the data separation functional unit.

k) The major synchronization point service (see 13.9) allows the SS-user to confine the flow of sequentially transmitted NSSDUs, TSSDUs and XSSDUs in each direction within a dialogue unit. Its use is controlled by the major/activity token.

l) The resynchronize service (see 13.10) is used to set the session connection to a previous or to a new synchronization point and to reassign the available tokens. This service may cause loss of NSSDUs, TSSDUs and XSSDUs.

There are two services for reporting errors or unanticipated situations:

m) The provider-initiated exception reporting service (see 13.11) (P-exception reporting service) permits SS-users to be notified of exception conditions or SS-provider protocol errors. This service may cause loss of NSSDUs, TSSDUs and XSSDUs.

n) The user-initiated exception reporting service (see 13.12) (U-exception reporting service) is used by the SS-user to report an exception condition when the data token is available but not assigned to the SS-user. This service may cause loss of NSSDUs, TSSDUs and XSSDUs.

There are five services associated with activities:

o) The activity start service (see 13.13) is used to indicate that a new activity is entered. Its use is controlled by the major/activity token.

p) The activity resume service (see 13.14) is used to indicate that a previously interrupted activity is re-entered. Its use is controlled by the major/activity token.

q) The activity interrupt service (see 13.15) allows an activity to be abnormally terminated with the implication that the work so far achieved is not to be discarded and may be resumed later. Its use is controlled by the major/activity token. This service may cause loss of NSSDUs, TSSDUs and XSSDUs.

r) The activity discard service (see 13.16) allows an activity to be abnormally terminated with the implication that the work so far achieved is to be discarded, and not resumed. Its use is controlled by the major/activity token. This service may cause loss of NSSDUs, TSSDUs and XSSDUs.

s) The activity end service (see 13.17) is used to end an activity (and set a major synchronization point). Its use is controlled by the major/activity token.

Using the activity services may lead to a state where no activity is in progress on the session connection. When activity services are employed, but no activity is in progress, only the activity start, activity resume, token management, capability data, typed data, normal data, expedited data, abort and release services may be invoked by the SS-users.

8.3 Session connection release phase

The session connection release phase is concerned with releasing a previously established session connection. It has three services associated with it:

a) The orderly release service (see 14.1) provides a means of achieving the orderly release of a session connection.

b) The user-initiated abort service (see 14.2) (U-Abort service) is used to initiate the release of a session connection in a way that will terminate any outstanding service request. This service may cause loss of NSSDUs, TSSDUs and XSSDUs.

c) The provider-initiated abort service (see 14.3) (P-Abort service) is used by the SS-provider to indicate the release of the session connection for internal reasons. This service may cause loss of NSSDUs, TSSDUs and XSSDUs. Any outstanding service request is terminated.

9 Functional units and subsets

9.1 Functional units

Functional units are logical groupings of related services defined by this Recommendation | International Standard for the purpose of:

a) negotiation of SS-user requirements during the session connection establishment phase;

b) reference by other Recommendation | International Standards.

Table 1 specifies the association of tokens and functional units. When a functional unit implies the availability of a token, services concerned with the management of that token are provided in order to be able to request and transfer the available tokens.

The services associated with each functional unit are specified in Table 2.

Table 1 – Functional units using tokens

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 2 – Services associated with each functional unit

Functional unit	Service(s)	Reference
Kernel (non-negotiable)	Session connection Normal data transfer Orderly release U-Abort P-Abort	12.1 13.1 14.1 14.2 14.3
Negotiated release	Orderly release Give tokens Please tokens	14.1 13.5 13.6
Half-duplex	Give tokens Please tokens	13.5 13.6
Duplex	No additional service	
Expedited data	Expedited data transfer	13.2
Typed data	Typed data transfer	13.3
Capability data	Capability data exchange	13.4
Minor synchronize	Minor synchronization point Give tokens Please tokens	13.8 13.5 13.6
Symmetric synchronize	Symmetric synchronize	13.8
Data separation	Data separation (associated with the minor synchronization point service or the symmetric synchronize service)	13.8
Major synchronize	Major synchronization point Give tokens Please tokens	13.9 13.5 13.6
Resynchronize	Resynchronize	13.10
Exceptions	Provider exception reporting User exception reporting	13.11 13.12
Activity management	Activity start Activity resume Activity interrupt Activity discard Activity end Give tokens Please tokens Give control	13.13 13.14 13.15 13.16 13.17 13.5 13.6 13.7

9.1.1 Kernel functional unit

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

9.1.2 Negotiated release functional unit

The negotiated release functional unit supports the negotiated orderly release service. The release token is available when this functional unit has been selected.

9.1.3 Half-duplex functional unit

The half-duplex functional unit supports the half-duplex service. The data token is available when this functional unit has been selected. It is not possible to select both this functional unit and the duplex functional unit for use on the same session connection.

9.1.4 Duplex functional unit

The duplex functional unit supports the duplex service. It is not possible to select both this functional unit and the half-duplex functional unit for use on the same session connection.

9.1.5 Expedited data functional unit

The expedited data functional unit supports the session expedited data transfer service.

9.1.6 Typed data functional unit

The typed data functional unit supports the typed data transfer service.

9.1.7 Capability data exchange functional unit

The capability data exchange functional unit supports the capability data exchange service. This functional unit can only be selected when the activity management functional unit has been selected.

9.1.8 Minor synchronize functional unit

The minor synchronize functional unit supports the minor synchronization point service. The synchronize-minor token is available when this functional unit has been selected.

9.1.9 Symmetric synchronize functional unit

The symmetric synchronize functional unit supports the symmetric synchronization point service which allows the SS-users to independently place minor synchronization points in their sending flows. Correlation of these synchronization points is an SS-user responsibility. The synchronize-minor token is not available. It is not possible to select this functional unit and the minor synchronize functional unit for use on the same session connection.

It is possible to select the symmetric synchronize functional unit with either the duplex functional unit or the half-duplex functional unit. If the half-duplex functional unit is selected, the data token is available; if the duplex functional unit is selected, the data token is not available. The synchronize-minor token is not available when the symmetric synchronize functional unit is selected.

9.1.10 Data separation functional unit

The data separation functional unit supports the data separation service and adds functionality to the minor synchronization point service and the symmetric synchronize service. The data separation functional unit can only be selected when either the minor synchronize functional unit has been selected or the symmetric synchronize functional unit has been selected. It is not valid to select both this functional unit and the activity management functional unit for use on the same session connection.

9.1.11 Major synchronize functional unit

The major synchronize functional unit supports the major synchronization point service. The major/activity token is available when this functional unit has been selected.

9.1.12 Resynchronize functional unit

The resynchronize functional unit supports the resynchronize service.

9.1.13 Exceptions functional unit

The exceptions functional unit supports the user exception and provider exception reporting services.

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

9.1.14 Activity management functional unit

The activity management functional unit supports the activity management services and the give control service. The major/activity token is available when this functional unit has been selected.

9.2 Subsets

A subset is defined as a combination of the kernel functional unit together with any other set of functional units provided that:

a) if the capability data functional unit is included in the subset, then the activity management functional unit is also included in the subset; and

b) if the exceptions functional unit is included in the subset, then the half-duplex functional unit is also included in the subset;

c) if the data separation functional unit is included in a subset, one of the minor synchronize functional unit and the symmetric synchronize functional unit is also included in the subset.

NOTE – This Recommendation | International Standard contains no requirements for the registration of subsets. Users of this Recommendation | International Standard may define subsets to meet their session service needs. Other Recommendations | International Standards may identify subsets that comply with the above definition.

10 Quality of session service

The term “Quality of Service” (QOS) refers to certain characteristics of a session connection as observed between the session connection endpoints. QOS describes aspects of a session connection which are attributable solely to the SS-provider; such aspects are independent of SS-user behaviour (which is beyond the control of the SS-provider). SS-user behaviour does not impact the QOS provided.

Once a session connection is established, the SS-users at the two ends have the same knowledge and understanding of what the QOS over the session connection is.

10.1 Determination of QOS

QOS is described in terms of QOS parameters.

The definition of the QOS parameters associated with the session service is given in 10.3. These definitions provide both SS-users and the SS-provider with a common understanding of the QOS characteristics.

Two types of session service QOS parameters are identified:

a) Those which are negotiated during the session connection establishment phase:

1) session connection protection (see 10.3.9);

2) session connection priority (see 10.3.10);

3) residual error rate (see 10.3.5);

4) throughput, for each direction of transfer (see 10.3.3);

5) transit delay, for each direction of transfer (see 10.3.4);

6) optimized dialogue transfer (see 10.3.13); and

7) extended control (see 10.3.12).

b) Those which are not negotiated during the session connection establishment phase but whose values are selected and/or known by other methods (for example, *a priori* knowledge and agreement, or by means of management functions) not defined in this Recommendation | International Standard:

1) session connection establishment delay (see 10.3.1);

2) session connection establishment failure probability (see 10.3.2);

3) transfer failure probability (see 10.3.6);

4) session connection release delay (see 10.3.7);

5) session connection release failure probability (see 10.3.8);

6) session connection resilience (see 10.3.11).

The negotiation procedures for parameters listed in 10.1 a) are defined in 10.2. Once the session connection is established, the selected QOS parameters are not re-negotiated during the lifetime of the session connection. The SS-user should be aware that changes in QOS during a session connection are not signalled in the session service.

10.2 Session connection QOS negotiation procedures

QOS negotiation is described in terms of parameters which can be conveyed by the S-CONNECT primitives during the session connection establishment phase (see clause 12). For the parameters which are negotiated during the session connection establishment phase [see 10.1 a)], the parameter values and negotiation rules are defined as follows:

a) In the S-CONNECT request primitive, the calling SS-user can specify:

   1) For session connection protection, session connection priority, extended control, and optimized dialogue transfer, a single parameter value which is the "desired" QOS; for extended control and optimized dialogue transfer, one of the two values "feature desired" or "feature not desired" is conveyed.

      NOTE – If the calling SS-user proposes use of the expedited data functional unit, the extended control parameter has the value "feature desired".

   2) For residual error rate, and for each direction of throughput and transit delay, two parameter values which are the "desired" QOS and the "lowest acceptable" QOS to which the calling SS-user will agree.

b) In the S-CONNECT indication primitive, for each of the negotiated parameters, an "available" value is conveyed which is specified as follows:

   1) For session connection protection, if the SS-provider agrees to provide a QOS value equivalent to the "desired" value specified in the S-CONNECT request, then the SS-provider specifies that value as "available"; if the SS-provider does not agree to provide the "desired" QOS requested, the SS-provider refuses to establish the session connection by issuing the S-CONNECT (reject) confirm primitive to the calling SS-user.

   2) For session connection priority, the SS-provider specifies the QOS value it is willing to provide (a value which is equal to or better than the "desired" value specified in the S-CONNECT request) as "available".

   3) For the residual error rate and each direction of throughput and transit delay, if the SS-provider agrees to provide a value of QOS which is equal to or better than the "lowest acceptable" QOS value specified in the S-CONNECT request, then the SS-provider specifies the value as "available"; if the SS-provider does not agree to provide this QOS, then the SS-provider refuses to establish the session connection by issuing the S-CONNECT (reject) confirm primitive to the calling SS-user.

   4) For extended control and optimized dialogue transfer, if the "desired" value in the S-CONNECT request primitive is "feature not desired" then "feature not desired" is specified as "available"; if the "desired" value is "feature desired" and the SS-provider agrees to provide the feature on the session connection, then "feature desired" is specified as "available"; otherwise if the SS-provider does not agree to provide the feature, "feature not desired" is specified as "available".

c) In the S-CONNECT response primitive, for each of the negotiated parameters, an "agreed" value is conveyed which is specified as follows:

   1) For optimized dialogue transfer, if the "available" value in the S-CONNECT indication primitive is "feature not desired" and the called SS-user agrees not to have the feature provided on the session connection, then "feature not desired" is specified as "agreed"; otherwise the SS-user may reject establishment of the session connection; if the "available" value in the indication primitive is "feature desired" and the SS-user agrees to have the feature provided, then "feature desired" is specified as "agreed"; otherwise, if the SS-user does not agree to provision of the feature, the value "feature not desired" is specified as "agreed".

   2) For each of the other parameters, if the called SS-user agrees to the QOS value specified as "available" in the S-CONNECT indication primitive, then the identical value is specified as "agreed"; if the SS-user does not agree to the "available" value, the SS-user may reject establishment of the session connection.

d) In the S-CONNECT confirm primitive, for each of the negotiated parameters, an "agreed" value is conveyed which is identical to the "agreed" value conveyed in the S-CONNECT response.

10.3 Definition of QOS parameters

QOS parameters can be classified as:

a) parameters which express session service performance parameters, as shown in Table 3;

b) parameters which express other session service characteristics, as shown in Table 4.

These session service QOS parameters are defined in this subclause.

Table 3 – Classification of Performance of QOS Parameters

Phase	Performance criterion	
	Speed	Accuracy/reliability
Session connection establishment	Session connection establishment delay	Session connection establishment failure probability (misconnection/session connection refusal)
Data transfer	Throughput Transit delay	Residual error rate (corruption) Session connection resilience Transfer failure probability
Session connection release	Session connection release delay	Session connection release failure probability

Table 4 – Parameters specifying other session service features

Extended Control
Session Connection Protection
Session Connection Priority
Optimized Dialogue Transfer

10.3.1 Session connection establishment delay

Session connection establishment delay is the maximum acceptable delay between an S-CONNECT request and the corresponding S-CONNECT confirm primitive.

NOTE – This delay includes SS-user dependent components.

10.3.2 Session connection establishment failure probability

Session connection establishment failure probability is the ratio of total session connection establishment failures to total session connection establishment attempts in a measurement sample.

Session connection establishment failure is defined to occur when a requested session connection is not established within the specified maximum acceptable session connection establishment delay as a result of misconnection, session connection refusal, or excessive delay on the part of the SS-provider. Session connection establishment attempts which fail as a result of error, session connection refusal, or excessive delay on the part of an SS-user are excluded in calculating session connection establishment failure probability.

10.3.3 Throughput

Throughput is defined for each direction of transfer, in terms of a sequence of at least two SSDUs successfully transferred by an S-DATA request/S-DATA indication or S-TYPED-DATA request/S-TYPED-DATA indication sequence of primitives. Given such a sequence of n SSDUs, where $n \geq 2$, the throughput is defined to be the smaller of:

a) the number of SS-user data octets contained in the last $n - 1$ SSDUs divided by the time between the first and last S-DATA or S-TYPED-DATA request in the sequence; and

b) the number of SS-user data octets contained in the last $n - 1$ SSDUs divided by the time between the first and the last S-DATA or S-TYPED-DATA indications in the sequence.

Successful transfer of the octets in a transmitted SSDU is defined to occur when the bits are delivered to the intended receiving SS-user without error, in the proper sequence, prior to release of the session connection by the receiving SS-user.

Throughput is only meaningful for a sequence of complete SSDUs and each specification is based on a previously stated average SSDU size.

Throughput is specified separately for each direction of transfer on a session connection. In each direction, a specification of throughput will consist of a "maximum throughput" value and an "average throughput" value. The "maximum throughput" value represents the maximum rate at which the SS-provider can continuously accept and deliver SSDUs, in the absence of sending SS-user input delays or flow control applied by the receiving SS-user. Thus, the sequence of SSDUs in the calculation above are defined to be presented continuously at the maximum rate. The "average throughput" value represents the expected transfer rate on a session connection including the effects of expected user-attributable delays (e.g. non-continuous SSDU input, receiving SS-user flow control). Thus, the sequence of SSDUs in the calculation above are defined to be presented at a rate which includes components representing "average" user delays.

It is possible for either the input or the output of a sequence of SSDUs to be excessively delayed by the SS-users. Such occurrences are excluded in calculating "average throughput" values.

For each direction of transfer, and for each of the "maximum throughput" and "average throughput" specifications, the throughput QOS for a particular session connection is negotiated between the SS-users and the SS-provider (see 10.2).

Throughput on a session connection relates only to the transfer of normal data and typed data over the session connection. There is no specification of the throughput for data which is transferred in association with the issue of any other session service primitives (e.g. S-CONNECT, S-CAPABILITY-DATA, etc.).

10.3.4 Transit delay

Transit delay is the elapsed time between the completion of any session service request primitive and the corresponding session service indication primitive occurring during the data transfer phase of a session connection. Elapsed time values are calculated only on service primitive pairs which are successfully completed.

Successful completion of a service primitive pair is defined to occur when the issue of the request primitive by one SS-user results in the issue of the corresponding indication primitive to the peer user (including any SS-user data associated with the primitive) which is without error, and in a proper sequence with respect to other primitives, prior to release of the session connection by the receiving SS-user.

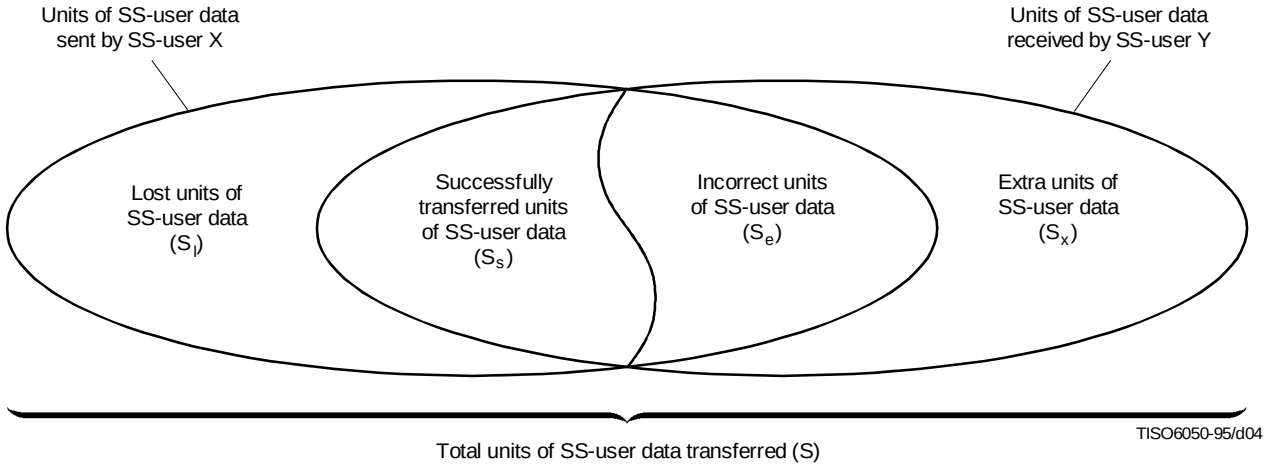
In duplex and half-duplex session connections, transit delay is specified independently for each direction of transfer. In general, each transit delay specification defines both the average value and the maximum value expected for a session connection. Each specification of transit delay assumes a previously stated average size for SS-user data included in the service primitive pair.

An attempt to measure the transit delay for an individual service primitive pair may be greatly influenced if the receiving SS-user exercises flow control. Such occurrences are excluded in calculating both average and maximum transit delay values.

10.3.5 Residual error rate

Residual error rate is the ratio of total incorrect, lost, and duplicate units of SS-user data to the total units of SS-user data transferred across the session service boundary in association with any SS-primitive issued in the data transfer phase of a

session connection during a measurement period. The relationship between these quantities for a particular SS-user pair is defined in Figure 3.



$$RER = \frac{S_e + S_l + S_x}{S}$$

Figure 3 – Components of residual error rate

10.3.6 Transfer failure probability

Transfer failure probability is the ratio of total transfer failures to total transfer samples observed during a performance measurement.

A transfer sample is a discrete observation of SS-provider performance in handling service requests made by the SS-user. A transfer sample begins with the initiation of session service requests during the data transfer phase and continues until the outcome of a given number of service requests have been determined. These service requests may include the transfer of SS-user data or other service requests (such as S-ACTIVITY-START request, S-TOKEN-PLEASE request, etc.) made by the SS-user. A transfer sample will normally correspond to the duration of an individual session connection.

A transfer failure is a transfer sample in which the observed performance is worse than a specified minimum acceptable level. Transfer failures are identified by comparing the measured values for applicable supported performance parameters with specified transfer failure thresholds. The three supported performance parameters which may apply are throughput, transit delay, and residual error rate.

In systems where session service QOS is reliably monitored by the SS-provider, transfer failure probability can be estimated by the probability of an S-P-ABORT or an S-P-EXCEPTION-REPORT during a transfer sample.

10.3.7 Session connection release delay

Session connection release delay is the maximum acceptable delay between an SS-user initiated S-U-ABORT request and the successful release of a particular session connection. Session connection release delay is normally specified independently for each SS-user.

Issue of an S-U-ABORT request by either SS-user marks the beginning of the session connection release delay for both users. Successful release for one SS-user is defined to occur when that SS-user is first able to initiate a new session connection. Successful release is signalled to the SS-user not initiating the S-U-ABORT request by an S-U-ABORT indication. The SS-user initiating the S-U-ABORT request will normally receive a similar signal of local significance.

10.3.8 Session connection release failure probability

Session connection release failure probability is the ratio of total SS-user initiated abort requests resulting in session connection release failure to total SS-user initiated abort requests included in a measurement sample. Session connection release failure probability is normally specified independently for each SS-user.

Session connection release failure is defined to occur, for a particular SS-user, if that SS-user is not successfully released (as defined in 10.3.7) within the specified maximum session connection release delay as a result of error or excessive delay on the part of the SS-provider. Session connection release attempts which fail as a result of error, release refusal, or excessive delay on the part of an SS-user are excluded in calculating session connection release failure probability.

10.3.9 Session connection protection

Session connection protection is the extent to which an SS-provider attempts to prevent unauthorized monitoring or manipulation of SS-user originated information. Session connection protection is specified qualitatively by selecting one of the following session connection protection options:

a) no protection features;

b) protection against passive monitoring;

c) protection against modification, replay, addition or deletion;

d) both b) and c).

10.3.10 Priority

The specification of priority is concerned with the relationship between session connections. This parameter specifies the relative importance of a session connection with respect to:

a) the order in which session connections are to have their QOS degraded, if necessary; and

b) the order in which session connections are to be broken to recover resources, if necessary.

This parameter only has meaning in the context of some management entity or structure able to judge relative importance. The number of priority levels is limited.

10.3.11 Session connection resilience

Session connection resilience parameters specify the probability of:

a) an SS-provider initiated non-orderly release of a session connection (i.e. issue of an S-P-ABORT indication); and

b) an SS-provider exception report (i.e. issue of an S-P-EXCEPTION-REPORT indication) during a specified time interval on an established session connection.

10.3.12 Extended control parameter

The extended control parameter allows the SS-users to make use of the resynchronize, abort, activity interrupt and activity discard services when normal flow is congested.

NOTE – When the expedited data functional unit has been selected, the extended control QOS is always provided to the SS-users.

10.3.13 Optimized dialogue transfer

The optimized dialogue transfer QOS parameter permits the concatenated transfer of certain session service requests. How this concatenation of service requests is achieved is a local implementation matter.

NOTE – This QOS parameter invokes the SS-provider extended concatenation protocol option.

SECTION 2 – DEFINITION OF CONNECTION ORIENTED SESSION SERVICE PRIMITIVES

11 Introduction to session service primitives

11.1 Summary of primitives

Each of the services constituting the session service is achieved by invoking a sequence of session service primitives. Tables 5, 6 and 7 summarize the primitives and their parameters occurring in each phase of the session service. The parameters are defined in clauses 12, 13 and 14.

Table 5 – Session connection establishment phase primitives

Service	Primitives	Parameter
Session connection	S-CONNECT request S-CONNECT indication S-CONNECT response S-CONNECT confirm	Session Connection Identifier, Calling Session Address, Called Session Address, Responding Session Address, Result, QOS, Session Requirements, Synchronization Point Serial Number, Initial Assignment of Tokens, SS-user data

Table 6 – Data transfer phase primitives

Service	Primitives	Parameter
Normal data transfer	S-DATA request S-DATA indication	SS-user data
Expedited data transfer	S-EXPEDITED-DATA request S-EXPEDITED-DATA indication	SS-user data
Typed data transfer	S-TYPED-DATA request S-TYPED-DATA indication	SS-user data
Capability data exchange	S-CAPABILITY-DATA request S-CAPABILITY-DATA indication S-CAPABILITY-DATA response S-CAPABILITY-DATA confirm	SS-user data
Give tokens	S-TOKEN-GIVE request S-TOKEN-GIVE indication	Tokens, SS-user data
Please tokens	S-TOKEN-PLEASE request S-TOKEN-PLEASE indication	Tokens, SS-user data
Give control	S-CONTROL-GIVE request S-CONTROL-GIVE indication	SS-user data
Minor synchronization point	S-SYNC-MINOR request S-SYNC-MINOR indication S-SYNC-MINOR response S-SYNC-MINOR confirm	Type, Data separation, Synchronization point serial number, SS-user data
Major synchronization point	S-SYNC-MAJOR request S-SYNC-MAJOR indication S-SYNC-MAJOR response S-SYNC-MAJOR confirm	Synchronization point serial number, SS-user data
Resynchronize	S-RESYNCHRONIZE request S-RESYNCHRONIZE indication S-RESYNCHRONIZE response S-RESYNCHRONIZE confirm	Resynchronize type, Synchronization point serial number, Assignment of Tokens, SS-user data
P-exception report	S-P-EXCEPTION-REPORT indication	Reason
U-exception reporting	S-U-EXCEPTION-REPORT request S-U-EXCEPTION-REPORT indication	Reason, SS-user data
Activity start	S-ACTIVITY-START request S-ACTIVITY-START indication	Activity identifier, SS-user data
Activity resume	S-ACTIVITY-RESUME request S-ACTIVITY-RESUME indication	Activity identifier, Old activity identifier, Synchronization point serial number, Old session connection identifier, SS-user data
Activity interrupt	S-ACTIVITY-INTERRUPT request S-ACTIVITY-INTERRUPT indication S-ACTIVITY-INTERRUPT response S-ACTIVITY-INTERRUPT confirm	Reason, SS-user data
Activity discard	S-ACTIVITY-DISCARD request S-ACTIVITY-DISCARD indication S-ACTIVITY-DISCARD response S-ACTIVITY-DISCARD confirm	Reason, SS-user data
Activity end	S-ACTIVITY-END request S-ACTIVITY-END indication S-ACTIVITY-END response S-ACTIVITY-END confirm	Synchronization point serial number, SS-user data

Table 7 – Session connection release phase primitives

Service	Primitives	Parameter
Orderly release	S-RELEASE request S-RELEASE indication S-RELEASE response S-RELEASE confirm	Result, SS-user data
U-abort	S-U-ABORT request S-U-ABORT indication	SS-user data
P-abort	S-P-ABORT indication	Reason

11.2 Token restrictions on sending primitives

Table 8 defines the conditions under which those service primitives requiring tokens may be issued.

Table 8 – Token restrictions on service primitives

Service primitives	Data token	Synchronize minor token	Major/activity token	Release token
S-RELEASE request S-RELEASE response (negative)	2 nr	2 nr	2 nr	2 0
S-DATA request (half-duplex) S-DATA request (duplex)	1 3	nr nr	nr nr	nr nr
S-CAPABILITY-DATA request	2	2	1	nr
S-TOKEN-GIVE request (data token) S-TOKEN-GIVE request (sync-minor token) S-TOKEN-GIVE request (major/activity token) S-TOKEN-GIVE request (release token)	1 nr nr nr	nr 1 nr nr	nr nr 1 nr	nr nr nr 1
S-TOKEN-PLEASE request (data token) S-TOKEN-PLEASE request (sync-minor token) S-TOKEN-PLEASE request (major/activity token) S-TOKEN-PLEASE request (release token)	0 nr nr nr	nr 0 nr nr	nr nr 0 nr	nr nr nr 0
S-CONTROL-GIVE request	2	2	1	2
S-SYNC-MINOR request, without symmetric synchronization S-SYNC-MINOR request, with symmetric synchronization S-SYNC-MAJOR request	2 nr 2	1 3 2	nr nr 1	nr nr nr
S-U-EXCEPTION-REPORT request	0	nr	nr	nr
S-ACTIVITY-START request S-ACTIVITY-RESUME request S-ACTIVITY-INTERRUPT request S-ACTIVITY-DISCARD request S-ACTIVITY-END request	2 2 nr nr 2	2 2 nr nr 2	1 1 1 1 1	nr nr nr nr nr

0	Token available and not assigned to the SS-user who initiated the service primitive
1	Token available and assigned to the SS-user who initiated the service primitive
2	Token not available or token assigned to the SS-user who initiated the service primitive
3	Token not available
nr	No restriction

11.3 Sequencing of primitives

All SS-user requests and responses are delivered by the SS-provider in the order in which they are submitted by the SS-user, except for the following:

a) S-EXPEDITED-DATA;

b) S-RESYNCHRONIZE;

c) S-ACTIVITY-INTERRUPT;

d) S-ACTIVITY-DISCARD;

e) S-U-ABORT,

which may be delivered earlier than previously submitted primitives, but not later than subsequently submitted primitives.

11.4 Synchronization point serial number management

Certain primitives carry one or two synchronization point serial numbers to identify a synchronization point. The upper limit of a serial number is defined by the SS-users. It is referred to as the Upper Limit Serial Number and is in the range 1 000 000 to infinity. The Upper Limit Serial Number is negotiated during the session connection establishment phase. The means by which the Upper Limit Serial Number is passed between the SS-users and the SPM is a local matter.

Synchronization points are assigned valid synchronization point serial numbers in the range 0 to (10**Upper Limit Serial Number) − 2 by the SS-provider. It is the responsibility of the SS-user to ensure that the number(s) assigned by the SS-provider in a synchronization point request do not exceed (10**Upper Limit Serial Number) − 2.

The synchronization point serial number (10**Upper Limit Serial Number) − 1 is also a valid synchronization point serial number for use by the SS-user, but only in the following services, which require the synchronization point serial number of the next synchronization point:

a) Session Connection Service;

b) Resynchronization Service.

The management of synchronization point serial numbers is defined in this Recommendation | International Standard in terms of:

c) operations on abstract local variables V(M), V(A), V(R) and Vsc, managed by the SS-provider; and

d) primitives issued by the SS-user in order to invoke these operations.

These operations are summarized in Table A.4.

A synchronization point may be identified in either of the following two ways, depending on whether the symmetric synchronize functional unit has been selected:

e) *Single serial number synchronization points:* Only one SS-user is allowed to define synchronization points at any given time, as controlled by the synchronize-minor and major/activity tokens. A single serial number is used to identify all synchronization points; it is incremented each time either SS-user issues a synchronization request. Single serial number synchronization points are used when the symmetric synchronize functional unit has not been selected. With this option, only the First Synchronization Point Serial Number parameter is used with the service primitives.

f) *Symmetric serial number synchronization points:* Both SS-users are allowed to define minor synchronization points on their sending directions of data flow. Correlation of two minor synchronization points (one for each direction of flow) is an SS-user responsibility. The synchronize-minor token is not available. Only the serial number associated with the direction of flow on which a minor synchronization point travels is incremented. Symmetric serial number synchronization points are used when the symmetric synchronize functional unit has been selected. With this option, both the First and Second Synchronization Point Serial Number parameters are used with the service primitives. The First Synchronization Point Serial Number is used to indicate points on the requesting SS-user's sending flow; the Second Synchronization Point Serial Number is used to indicate points on the opposite direction of flow.

With symmetric serial number synchronization points, a minor synchronization point is associated with a single direction of data flow and can be represented by one synchronization point serial number. Major synchronization points and the ends of activities are always defined by two correlated synchronization point serial numbers, one for each direction of data flow.

When the data token is available and the symmetric synchronize functional unit has been selected, ownership of the data token is not required to issue a minor synchronization point. This permits a half-duplex user to indicate minor synchronization points in its typed-data flow against the turn, when the typed data functional unit is selected. If an SS-user using half-duplex mode does not wish to synchronize the typed data flow, the SS-user should restrict the right to issue minor synchronization points to the owner of the data token.

When both the major synchronize and symmetric synchronize functional units are selected, the major/activity token is available. Acknowledgement of a major synchronization defines a serial number for the direction of flow on which the confirmation travels. Only the owner of the major/activity token (and the data token, if available) is allowed to make an S-SYNC-MAJOR request.

For the case when the symmetric synchronize functional unit has not been selected, 11.4.1 defines management of single serial synchronization point numbers. For the case when the symmetric synchronize functional unit has been selected, 11.4.2 defines management of symmetric serial number synchronization points.

11.4.1 Management of single serial synchronization point numbers

This subclause defines the synchronization variables and how they are managed when the symmetric synchronize functional unit has not been selected for use on the session connection.

11.4.1.1 Variables

The synchronization variables V(A), V(M), and V(R) are each single number variables when the symmetric synchronize functional unit has not been selected.

11.4.1.1.1 V(A)

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

11.4.1.1.2 V(M)

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

11.4.1.1.3 V(R)

V(R) is the lowest serial number to which resynchronization restart is permitted.

11.4.1.1.4 Vsc

Vsc is used 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.

11.4.1.2 Session connection establishment

When a session connection is established in which the symmetric synchronize functional unit has not been selected and at least one of the following functional units has been selected:

a) minor synchronize functional unit; or

b) major synchronize functional unit; or

c) resynchronize functional unit

and the activity management functional unit has not been selected, V(M) and V(A) are set to the initial synchronization point serial number of the response/confirm primitives. V(R) is set to zero. Vsc is set false.

11.4.1.3 Minor synchronization point

When the minor synchronize functional unit has been selected without the symmetric synchronize functional unit, the minor synchronize service has the following effect on synchronization variables.

When an S-SYNC-MINOR request is issued, the associated synchronization point serial number, which is indicated to the SS-user, is equal to V(M). V(R) remains unchanged. V(A) is set to V(M) if Vsc is true, otherwise V(A) remains unchanged. V(M) is then incremented by one and Vsc is set to false.

When an S-SYNC-MINOR indication is received, the associated synchronization point serial number, which is indicated to the SS-user, is equal to V(M). V(R) remains unchanged. V(A) is set to V(M) if Vsc is false, otherwise it remains unchanged. V(M) is then incremented by one and Vsc is set to true.

When an S-SYNC-MINOR response is issued, Vsc must be true and the associated synchronization point serial number, which is supplied by the SS-user, must be less than V(M) and equal to or greater than V(A). V(A) is set to the serial number plus one. V(M), V(R) and Vsc remain unchanged.

When an S-SYNC-MINOR confirm is received, Vsc is false and the associated synchronization point serial number, which is indicated to the SS-user, is less than V(M) and equal to or greater than V(A). V(A) is set to the serial number plus one. V(M), V(R) and Vsc remain unchanged.

11.4.1.4 Major synchronization point

When the major synchronize functional unit has been selected without the symmetric synchronize functional unit, the major synchronize service has the following effect on synchronization variables.

When an S-SYNC-MAJOR request is issued, the associated synchronization point serial number, which is indicated to the SS-user, is equal to V(M). V(R) remains unchanged. V(A) is set to V(M) if Vsc is true, otherwise it remains unchanged. V(M) is then incremented by one and Vsc is set to false.

When an S-SYNC-MAJOR indication is received, the associated synchronization point serial number, which is indicated to the SS-user, is equal to V(M). V(R) and Vsc remain unchanged. V(A) is set to V(M) if Vsc is false, otherwise it remains unchanged. V(M) is then incremented by one.

When an S-SYNC-MAJOR response is issued, the associated synchronisation point serial number is equal to V(M) minus one. No synchronization point serial number is passed with this primitive. V(A) and V(R) are set to V(M). V(M) and Vsc remain unchanged.

When an S-SYNC-MAJOR confirm is received, the associated synchronization point serial number is equal to V(M) minus one. No synchronization point serial number is passed with this primitive. V(A) and V(R) are set to V(M). V(M) and Vsc remain unchanged.

11.4.1.5 Resynchronization

When the resynchronize functional unit has been selected without the symmetric synchronize functional unit, the resynchronize service has the following effect on synchronization variables.

When an S-RESYNCHRONIZE request is issued:

a) if the option is “abandon”, there is no associated synchronization point serial number;

b) if the option is “restart”, the associated synchronization point serial number, which is supplied by the SS-user, must be greater than or equal to V(R) and less than or equal to V(M);

c) if the option is “set”, the associated synchronization point serial number, which is supplied by the SS-user, may have any valid value.

For all options, V(A), V(M), V(R) and Vsc remain unchanged.

When an S-RESYNCHRONIZE indication is received:

d) If the option is “abandon”, the associated synchronization point serial number, which is indicated to the SS-user, is greater than or equal to V(M). V(M) is set to the serial number contained in the indication.

e) If the option is “restart”, the associated synchronization point serial number, which is indicated to the SS-user, is greater than or equal to V(R). If the synchronization point serial number is greater than V(M) (see Note), the SS-user either responds to the S-RESYNCHRONIZE indication [see g)] or generates a collision (see clause 16).

NOTE – This situation can arise if the extended control QOS is provided and the S-RESYNCHRONIZE request caused an earlier S-SYNC-MINOR request to be discarded by the SS-provider.

f) If the option is “set”, the associated synchronization point serial number, which is indicated to the SS-user, may have any valid value.

For all options, V(A), V(R) and Vsc remain unchanged. For the “restart” and “set” options, V(M) remains unchanged.

When an S-RESYNCHRONIZE response is issued:

g) if the option is “abandon” or “restart”, the associated synchronization point serial number, which is supplied by the SS-user, must be equal to the value received in the S-RESYNCHRONIZE indication;

h) if the option is “set”, the associated synchronization point serial number, which is supplied by the SS-user, may have any valid value.

V(A) and V(M) are set to the synchronization point serial number and Vsc remains unchanged. V(R) is set to zero for the options “abandon” and “set”; it remains unchanged for the “restart” option.

When an S-RESYNCHRONIZE confirm is received:

i) if the option is “abandon”, the associated synchronization point serial number, which is indicated to the SS-user, is greater than or equal to V(M);

j) if the option is “restart”, the associated synchronization point serial number, which is indicated to the SS-user, is equal to the synchronization point serial number in the corresponding request;

k) if the option is “set”, the associated synchronization point serial number, which is indicated to the SS-user, may have any valid value.

V(A) and V(M) are set to the synchronization point serial number and Vsc remains unchanged. V(R) is set to zero for the options “abandon” and “set”; it remains unchanged for the “restart” option.

11.4.1.6 Activity management

When the activity management functional unit has been selected without the symmetric synchronize functional unit, the activity management service has the following effect on synchronization variables.

When an S-ACTIVITY-START request is issued, or when an S-ACTIVITY-START indication is received, V(A), V(M) and V(R) are set to one and Vsc remains unchanged.

When an S-ACTIVITY-RESUME request is issued, or when an S-ACTIVITY-RESUME indication is received, V(A) and V(M) are set to the synchronization point serial number supplied by the SS-user plus one; V(R) is set to one and Vsc remains unchanged.

The management of V(A), V(M), V(R) and Vsc for S-ACTIVITY-END request, indication, response and confirm is identical to that for S-SYNC-MAJOR request, indication, response and confirm respectively.

The use of S-ACTIVITY-DISCARD and S-ACTIVITY-INTERRUPT primitives has no implication on V(A), V(M), V(R) and Vsc.

11.4.2 Management of symmetric serial number synchronization point numbers

This subclause defines the synchronization variables and how they are managed when the symmetric synchronize functional unit has been selected for use on the session connection. For both minor and major synchronization services, the serial numbers are managed by the SS-provider and given to the SS-user.

11.4.2.1 Variables

The synchronization variables V(A), V(M), and V(R) are each dual number variables (an ordered pair) when the symmetric synchronize functional unit has been selected.

11.4.2.1.1 V(A) = V(As), V(Ar)

V(As) is the lowest serial number on the SS-user’s sending flow to which a synchronization point confirmation is expected. No confirmation is expected to be received when V(As) = V(Ms).

V(Ar) is the lowest serial number on the SS-user's receiving flow for which a confirmation has not yet been sent. No confirmation can be sent when V(Ar) = V(Mr).

11.4.2.1.2 V(M) = V(Ms), V(Mr)

V(Ms) is the serial number of the next synchronization point to be sent.

V(Mr) is the serial number of the next synchronization point to be received.

11.4.2.1.3 V(R) = V(Rs), V(Rr)

V(Rs) is the lowest serial number on the SS-user's sending flow to which resynchronization restart is permitted.

V(Rr) is the lowest serial number on the SS-user's receiving flow to which resynchronization restart is permitted.

11.4.2.1.4 Vsc

Vsc is not used if the symmetric synchronize functional unit has been selected.

11.4.2.2 Session connection establishment

When any session connection is established in which the symmetric synchronize functional unit has been selected and the activity management functional unit has not been selected, V(Ms) and V(Mr) are set to the initial synchronization point serial numbers of the response/confirm primitives. V(As) and V(Ar) are also set to these numbers. V(Rs) and V(Rr) are set to zero. The dual number definitions of these numbers apply if the symmetric synchronize functional unit has been selected.

11.4.2.3 Minor synchronization point

When the symmetric synchronize functional unit has been selected, the minor synchronize service has the following effect on synchronization variables.

When an S-SYNC-MINOR request is issued, the associated synchronization point serial number, which is indicated to the SS-user, is equal to V(Ms). V(Rs) and V(As) remain unchanged. V(Ms) is then incremented by one.

When an S-SYNC-MINOR indication is received, the associated synchronization point serial number, which is indicated to the SS-user, is equal to V(Mr). V(Rr) and V(Ar) remain unchanged. V(Mr) is then incremented by one.

When an S-SYNC-MINOR response is issued, the associated synchronization point serial number, which is supplied by the SS-user, must be less than V(Mr) and equal to or greater than V(Ar). V(Ar) is set to the serial number plus one. V(Mr) and V(Rr) remain unchanged.

When an S-SYNC-MINOR confirm is received, the associated synchronization point serial number, which is indicated to the SS-user, is less than V(Ms) and equal to or greater than V(As). V(As) is set to the serial number plus one. V(Ms) and V(Rs) remain unchanged.

11.4.2.4 Major synchronization point

When the major synchronize functional unit and the symmetric synchronize functional unit have been selected, the major synchronize service has the following effect on synchronization variables.

Two serial numbers are associated with a major synchronization point. The First Synchronization Point Serial Number is associated with the sending flow of the requestor, and the Second Synchronization Point is associated with the receiving flow of the requestor. The first serial number is defined by the SS-provider at the time the S-SYNC-MAJOR request is issued. The second serial number is defined by the SS-provider at the time the S-SYNC-MAJOR response is issued.

When an S-SYNC-MAJOR request is issued, the associated sending flow synchronization point serial number, which is indicated to the SS-user, is equal to V(Ms). V(Rs) remains unchanged. V(Ms) is then incremented by one.

When an S-SYNC-MAJOR indication is received, the associated receiving flow synchronization point serial number, which is indicated to the SS-user, is equal to V(Mr). V(Rr) remains unchanged. V(Mr) is then incremented by one.

When an S-SYNC-MAJOR response is issued, the associated sending flow synchronization point serial number, which is indicated to the SS-user, is equal to V(Ms). V(Ms) is then incremented by one. V(As) and V(Rs) are set to V(Ms).

V(Ar) and V(Rr) are set to V(Mr).

When an S-SYNC-MAJOR confirm is received, the associated receiving flow synchronization point serial number, which is indicated to the SS-user, is equal to V(Mr). V(Mr) is then incremented by one. V(As) and V(Rs) are set to V(Ms). V(Ar) and V(Rr) are set to V(Mr). V(Ms) remains unchanged.

After an SS-user has made an S-SYNC-MAJOR request, it may receive minor synchronization points and user data (if the full duplex functional unit is selected). The receiving SS-user is not permitted to respond to these synchronization points or data while waiting on the S-SYNC-MAJOR confirm.

11.4.2.5 Resynchronization

When the resynchronize functional unit and the symmetric synchronize functional unit have been selected, the resynchronize service has the following effect on synchronization variables.

The First Synchronization Point Serial Number is associated with the sending flow of the requestor, and the Second Synchronization Point is associated with the receiving flow of the requestor.

When the symmetric synchronize functional unit has been selected, the SS-user may request to resynchronize one or both directions of flow. The SS-user supplies, with the S-RESYNCHRONIZE request, a resynchronize option (abandon, restart or set) for each requested direction of flow. The option for a direction of flow determines how the synchronization point serial number and variables associated with that direction of flow are handled.

When an S-RESYNCHRONIZE request is issued:

a) if the option for a direction of flow is “abandon”, there are no associated serial numbers;

b) if the option for a direction of flow is “restart”, the associated sending (or receiving) flow synchronization point serial number must be greater than or equal to V(Rs) [or V(Rr)], and less than or equal to V(Ms) [or V(Mr)];

c) if the option for a direction of flow is “set”, the associated synchronization point serial number may have any valid value.

For all options, V(As), V(Ar), V(Ms), V(Mr), V(Rs), and V(Rr) remain unchanged.

When an S-RESYNCHRONIZE indication is received:

d) If the option for a direction of flow is “abandon”, the associated sending (or receiving) synchronization point serial number, which is indicated to the SS-user, is greater than or equal to V(Ms) [or V(Mr)]. V(Ms) [or V(Mr)] is set to the serial number contained in the indication.

e) If the option for a direction of flow is “restart”, the associated sending (or receiving) synchronization point serial number, which is indicated to the SS-user, is greater than or equal to V(Rs) [or V(Rr)]. If the synchronization point is greater than V(Ms) [or V(Mr)], the SS-user either responds to the S-RESYNCHRONIZE indication [see g)] or generates a collision (see clause 16).

f) If the option for a direction of flow is “set”, the associated sending (or receiving) synchronization point serial number, which is indicated to the SS-user, may have any valid value.

For all options, V(As), V(Ar), V(Rs) and V(Rr) remain unchanged. For the “set” option, V(Ms) and V(Mr) remain unchanged.

When an S-RESYNCHRONIZE response is issued:

g) if the option for a direction of flow is “abandon” or “restart”, the associated sending (or receiving) synchronization point serial number, which is supplied by the SS-user, must be equal to that received in the S-RESYNCHRONIZE indication;

h) if the option for a direction of flow is “set”, the associated sending (or receiving) synchronization point serial number, which is supplied by the SS-user, may have any valid value.

If the sending flow is being resynchronized, V(As) and V(Ms) are set to the synchronization point serial number for the SS-user sending flow; if the receiving flow is being resynchronized, V(Ar) and V(Mr) are set to the synchronization point serial number for the SS-user receiving flow. V(Rs) and/or V(Rr) are set to zero if the resynchronize option for the associated flow is “abandon” or “set”; they remain unchanged for the “restart” option.

When an S-RESYNCHRONIZE confirm is received:

j) if the option for a direction of flow is “abandon”, the associated sending (or receiving) synchronization point serial number, which is indicated to the SS-user, is greater than or equal to V(Ms) [or V(Mr)];

k) if the option for a direction of flow is “restart”, the associated sending (or receiving) synchronization point serial number, which is indicated to the SS-user, is equal to or less than the synchronization point serial number in the corresponding request;

l) if the option for a direction of flow is “set”, the associated sending (or receiving) synchronization point serial number, which is indicated to the SS-user, may have any valid value.

V(As) and V(Ms) are set to the synchronization point serial number for the SS-user sending flow if the sending flow is being resynchronized; V(Ar) and V(Mr) are set to the synchronization point serial number for the SS-user receiving flow if the receiving flow is being resynchronized. V(Rs) and/or V(Rr) are set to zero if the resynchronize option for the associated direction of flow is “abandon” or “set”; they remain unchanged for the “restart” option.

11.4.2.6 Activity management

When the activity management functional unit and the symmetric synchronize functional unit have been selected, the activity management service has the following effect on synchronization variables.

The First Synchronization Point Serial Number is associated with the sending flow of the requestor, and the Second Synchronization Point is associated with the receiving flow of the requestor.

When an S-ACTIVITY-START request is issued, or when an S-ACTIVITY-START indication is received, V(As), V(Ar), V(Ms), V(Mr), V(Rs) and V(Rr) are set to one.

When an S-ACTIVITY-RESUME request is issued, or when an S-ACTIVITY-RESUME indication is received, V(As) and V(Ms) are set to the synchronization point serial number associated with the SS-user’s sending flow, and V(Ar) and V(Mr) are set to the synchronization point serial number associated with the SS-user’s receiving flow. V(Rs) and V(Rr) are set to one.

The management of V(As), V(Ar), V(Ms), V(Mr), V(Rs), and V(Rr) for S-ACTIVITY-END request, indication, response, and confirm is identical to that for S-SYNC-MAJOR request, indication, response, and confirm respectively.

The use of S-ACTIVITY-DISCARD and S-ACTIVITY-INTERRUPT primitives has no implication on V(As), V(Ar), V(Ms), V(Mr), V(Rs), and V(Rr).

11.5 Data separation

The data separation service adds functionality to the minor synchronization point service and the symmetric synchronize service to protect data sent before the minor synchronization points from being discarded in the event of a resynchronization. When the data separation functional unit has been selected, the associated service is invoked by the SS-user by setting the data separation parameter to true in the S-SYNC-MINOR request.

This service has no specific effect on the variables described in 11.4.2.1.

12 Session connection establishment phase

12.1 Session connection service

12.1.1 Function

The session connection service enables two SS-users to establish a session connection between themselves.

Simultaneous attempts by both SS-users to establish a session connection between themselves may result in two session connections. An SS-user may always reject an unwanted connection. No architectural restrictions are placed on the number of concurrent session connections between two SS-users.

This service allows the SS-users to exchange the values of session connection parameters. By the end of the session connection establishment phase the SS-users have agreed on a set of parameter values concerning the session connection.

12.1.2 Types of primitives and their parameters

Table 9 specifies the types of session service primitives and parameters needed for session connection establishment.

Table 9 – Session connection establishment primitives and parameters

Parameter \ Primitive	S-CONNECT			
	Request	Indication	Response	Confirm
Session Connection Identifier	U	C(=)	U	C(=)
Calling Session Address	M	M		
Called Session Address	M	M		
Responding Session Address			M	M
Result			M	M(=)
Quality of Service	M	M	M	M
Session Requirements	M	M(=)	M	M(=)
First Initial Synchronization Point Serial Number	C	C(=)	C	C(=)
Second Initial Synchronization Point Serial Number	C	C(=)	C	C(=)
Initial assignment of tokens	C	C(=)	C	C(=)
SS-user data	U	C(=)	U	C(=)

M Presence of the parameter is mandatory
C Presence of the parameter is conditional
U Presence of the parameter is a user option
blank The parameter is absent
(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

12.1.2.1 Session Connection Identifier is a parameter which is provided by the SS-users to enable them to identify the session connection. The session connection identifier is transparent to the SS-provider. This parameter consists of:

a) Calling SS-user Reference (request and indication only) with a maximum of 64 octets;

b) Called SS-user Reference (response and confirm only) with a maximum of 64 octets;

c) Common Reference with a maximum of 64 octets;

d) Additional Reference Information with a maximum of 4 octets.

12.1.2.2 Calling Session Address is the session address of the calling entity (see CCITT Rec. X.650 | ISO/IEC 7498-3).

12.1.2.3 Called Session Address is the session address of the called entity (see CCITT Rec. X.650 | ISO/IEC 7498-3).

12.1.2.4 Responding Session Address is the session address of the responding entity (see CCITT Rec. X.650 | ISO/IEC 7498-3).

12.1.2.5 Result is a parameter indicating the success or failure of the connection establishment request. Its value can be one of:

a) accept;

b) reject by called SS-user, where the reason for failure in the result parameter is one of:

1) reason not specified;

2) rejection by called SS-user due to temporary congestion;

3) rejection by called SS-user; the user data field may be used to provide further information;

c) reject by SS-provider where the reason of failure in the result parameter is one of:

1) reason not specified;

2) SS-provider congestion;

3) Called Session Address unknown;

4) called SS-user not attached to SSAP;

5) implementation restriction stated in the PICS.

Reasons 3) and 4) may be regarded as persistent.

Only value a) or b) can be present in a response. Any of the values may be present in a confirm.

12.1.2.6 Quality of Service is a list of parameters which are defined and negotiated as described in clause 10.

12.1.2.7 Session Requirements is a list of functional units subject to the restrictions defined in 9.2 and are chosen from:

a) half-duplex functional unit;

b) duplex functional unit;

c) exceptions functional unit;

d) typed data functional unit;

e) negotiated release functional unit;

f) minor synchronize functional unit;

g) symmetric synchronize functional unit;

h) data separation functional unit;

i) major synchronize functional unit;

j) resynchronize functional unit;

k) expedited data functional unit;

l) activity management functional unit;

m) capability data exchange functional unit.

The session requirements specified in the response indicate the called SS-user session requirements to the requestor. The acceptor may not propose both the half-duplex and the duplex functional units in the response. If only one of the half-duplex or duplex functional units was proposed in the indication, then the acceptor proposes the same functional unit in the response or refuses the connection. If the capability data exchange functional unit is proposed, the activity management functional unit is also proposed. If the data separation functional unit is proposed by either the requestor or by the acceptor, at least one of the minor synchronize functional unit and the symmetric synchronize functional unit is also proposed. If the data separation functional unit is proposed by the acceptor, the activity management functional unit is not proposed. If the exceptions functional unit is proposed, the half-duplex functional unit is also proposed. With these exceptions, additional SS-user session requirements which were not included in the indication, may be included in the response. SS-user session requirements that are proposed in both the indication and the response are the ones selected for use on the session connection.

12.1.2.8 First and Second Initial Synchronization Point Serial Numbers identify the initial synchronization point. The conditions for the presence of one or both initial synchronization point serial numbers and rules for negotiation are defined in 7.6.3. A serial number is in the range 0 to (10**Upper Limit Serial Number) – 1 (see 11.4).

12.1.2.9 Initial Assignment of Tokens is a list of the initial sides to which the available tokens are assigned. The parameter is only required if the corresponding tokens are available. For each available token, the value in a request/indication may be one of:

a) requestor side;

b) acceptor side;

c) acceptor chooses.

The parameter in a response/confirm is absent, unless the value in the request/indication is c), in which case the acceptor replies with a) or b).

12.1.2.10 SS-user data is a parameter containing an unlimited number of octets of user information.

12.1.3 Sequence of primitives

The sequence of primitives for session connection establishment, whether accepted or rejected, is defined by the time sequence diagram in Figure 4.

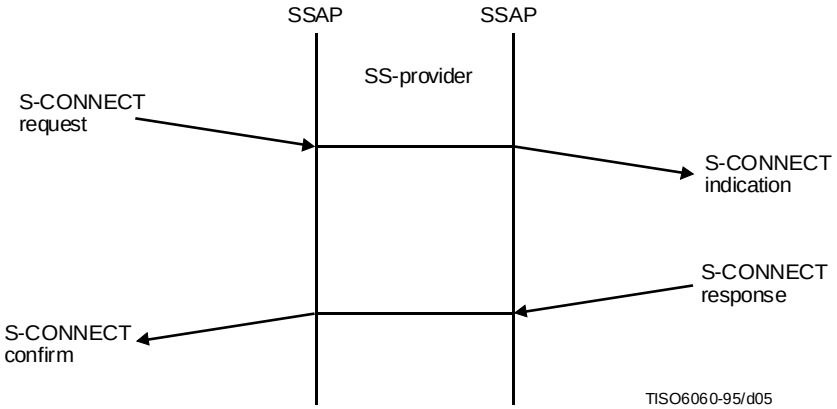


Figure 4 – Session connection establishment time sequence diagram

13 Data transfer phase

13.1 Normal data transfer service

13.1.1 Function

The normal data transfer service allows both SS-users to transfer NSSDUs over the session connection. The SS-provider should deliver each NSSDU to the SS-user as soon as possible. This service is always available on every session connection.

Use of this service is subject to the token restrictions specified in Table 8.

13.1.2 Types of primitives and their parameters

Table 10 specifies the types of session service primitives and parameters needed for normal data transfer.

SS-user data parameter is an NSSDU. The size of an NSSDU is an integral number of octets greater than zero and unlimited in length.

Table 10 – Normal data transfer primitives and parameters

Primitive / Parameter	S-DATA	
	Request	Indication
SS-user data	M	M(=)
M Presence of the parameter is mandatory		
(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive		

13.1.3 Sequence of primitives

The sequence of primitives in a successful normal data transfer is defined by the time sequence diagram in Figure 5.

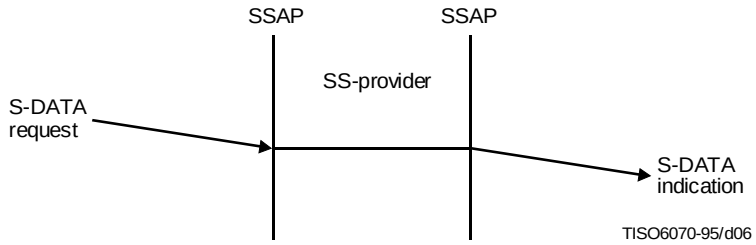


Figure 5 – Normal data transfer time sequence diagram

13.2 Expedited data transfer service

13.2.1 Function

The expedited data transfer service allows SS-users to transfer XSSDUs over the session connection. The transfer of an XSSDU is free from the token and flow control constraints of the normal data transfer service, typed data transfer service and the capability data exchange service.

The SS-provider guarantees that an XSSDU will not be delivered after any subsequently submitted NSSDU or TSSDU on that session connection. The size of an XSSDU is limited.

13.2.2 Types of primitives and their parameters

Table 11 specifies the types of session service primitives and parameters needed for expedited data transfer.

SS-user data parameter is an XSSDU. The size of an XSSDU is 1 to 14 octets.

Table 11 – Expedited data transfer primitives and parameters

Primitive / Parameter	S-EXPEDITED-DATA	
	Request	Indication
SS-user data	M	M(=)
M Presence of the parameter is mandatory		
(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive		

13.2.3 Sequence of primitives

The sequence of primitives in a successful expedited data transfer is defined by the time sequence diagram in Figure 6.

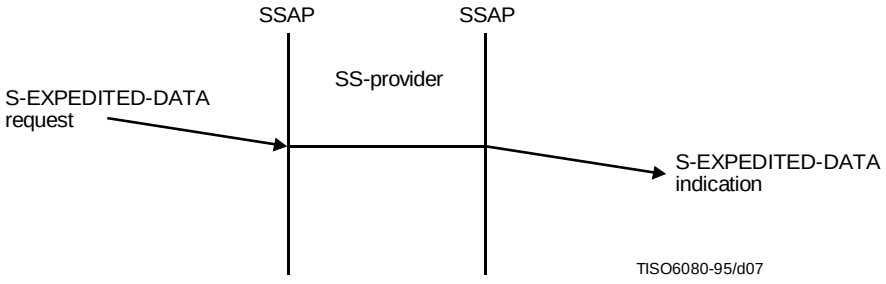


Figure 6 – Expedited data transfer time sequence diagram

13.3 Typed data transfer service

13.3.1 Function

The typed data transfer service permits the SS-users to transfer TSSDUs over the session connection. Typed data transfers are subject to the same service restrictions as normal data transfers, except that typed data transfers are not subject to token restrictions.

13.3.2 Types of primitives and their parameters

Table 12 specifies the types of session service primitives and parameters needed for the typed data transfer service.

SS-user data parameter is a TSSDU. The size of a TSSDU is an integral number of octets greater than zero and unlimited in length.

Table 12 – Typed data primitives and parameters

Primitive / Parameter	S-TYPED-DATA	
	Request	Indication
SS-user data	M	M(=)

M Presence of the parameter is mandatory

(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.3.3 Sequence of primitives

The sequence of primitives in a successful typed data transfer is defined by the time sequence diagram in Figure 7.

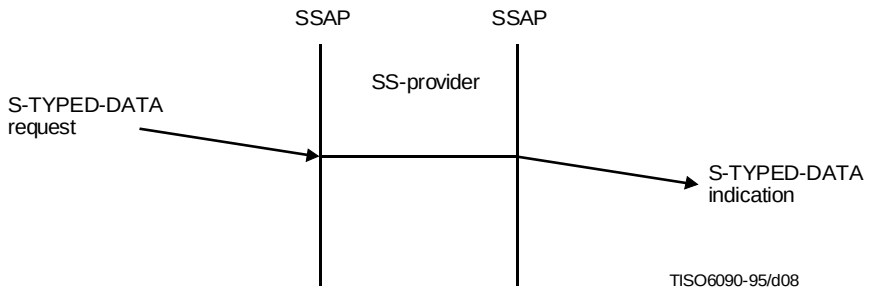


Figure 7 – Typed data transfer time sequence diagram

13.4 Capability data exchange service

13.4.1 Function

The capability data exchange service allows SS-users to exchange user data while not within an activity. The service can only be initiated if activity services are available but no activity is in progress. Use of this service is subject to the token restrictions specified in Table 8.

13.4.2 Types of primitives and their parameters

Table 13 specifies the types of session service primitives and parameters needed for the capability data exchange service.

SS-user data is a parameter containing an unlimited number of octets of user information.

Table 13 – Capability data exchange primitives and parameters

Parameter \ Primitive	S-CAPABILITY-DATA			
	Request	Indication	Response	Confirm
SS-user data	U	C(=)	U	C(=)

C Presence of the parameter is conditional

U Presence of the parameter is a user option

(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.4.3 Sequence of primitives

The sequence of primitives in a successful capability data exchange is defined by the time sequence diagram in Figure 8.

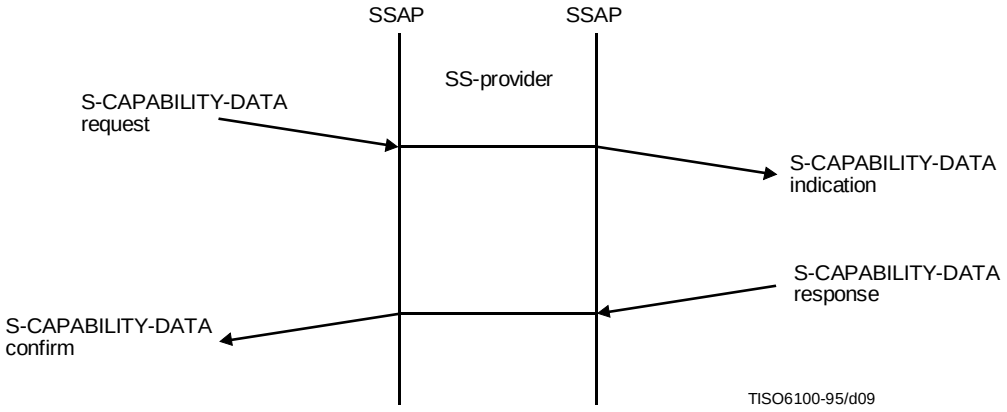


Figure 8 – Capability data exchange time sequence diagram

13.5 Give tokens service

13.5.1 Function

The give tokens service allows an SS-user to surrender one or more tokens to the other SS-user, subject to the token restrictions specified in Table 8.

The initial assignment of the tokens is established when the session connection is established (see 7.6.2).

13.5.2 Types of primitives and their parameters

Table 14 specifies the types of session service primitives and parameters needed for the give tokens service.

Table 14 – Give tokens primitives and parameters

Parameter \ Primitive	S-TOKEN-GIVE	
	Request	Indication
Tokens	M	M(=)
SS-user data	U	C(=)

M Presence of the parameter is mandatory
C Presence of the parameter is conditional
U Presence of the parameter is a user option
(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.5.2.1 Tokens is a list of tokens assigned to this SS-user to be transferred to the other user. The value is any combination of:

a) data token;

b) synchronize-minor token;

c) major/activity token;

d) release token.

13.5.2.2 SS-user data is a parameter containing an unlimited number of octets of user information.

13.5.3 Sequence of primitives

The sequence of primitives in a successful transfer of tokens is defined by the time sequence diagram in Figure 9.

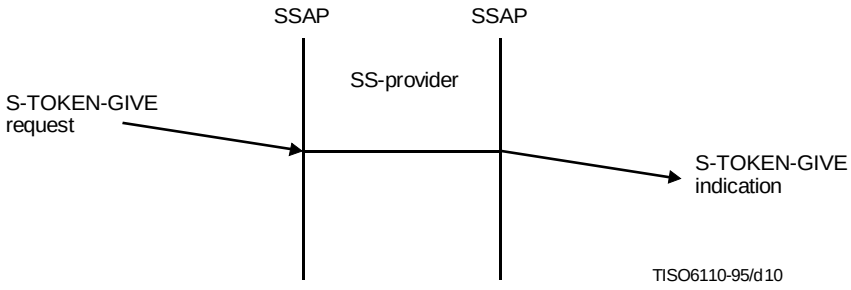


Figure 9 – Give tokens time sequence diagram

13.6 Please tokens service

13.6.1 Function

The please tokens service allows an SS-user to request specific tokens, subject to the token restrictions specified in Table 8.

13.6.2 Types of primitives and their parameters

Table 15 specifies the types of session service primitives and parameters needed for the please tokens service.

Table 15 – Please tokens primitives and parameters

Parameter \ Primitive	S-TOKEN-PLEASE	
	Request	Indication
Tokens	M	M(=)
SS-user data	U	C(=)

M Presence of the parameter is mandatory
C Presence of the parameter is conditional
U Presence of the parameter is a user option
(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.6.2.1 Tokens is a list of available tokens requested by the SS-user. The value is any combination of:

a) data token;
b) synchronize-minor token;
c) major/activity token;
d) release token.

13.6.2.2 SS-user data is a parameter containing an unlimited number of octets of user information.

13.6.3 Sequence of primitives

The sequence of primitives in a successful request for tokens is defined by the time sequence diagram in Figure 10.

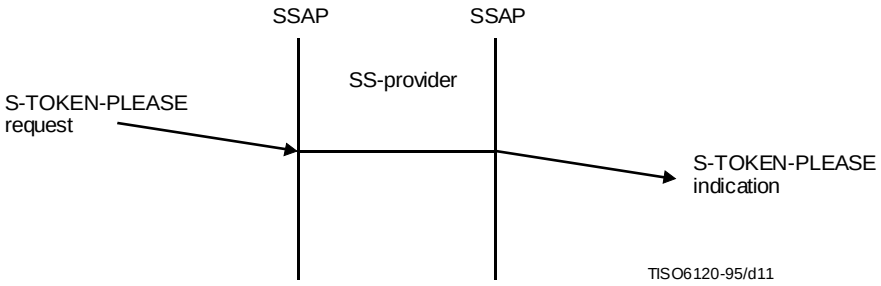


Figure 10 – Please tokens time sequence diagram

13.7 Give control service

13.7.1 Function

The give control service allows an SS-user to surrender the entire set of available tokens. This service is an integral part of the activity management concept. This service can only be requested when activity management functional unit has been selected, but no activity is in progress.

13.7.2 Types of primitives and their parameters

Table 16 specifies the types of session service primitives and parameters needed for the give control service.

SS-user data is a parameter containing an unlimited number of octets of user information.

Table 16 – Give control primitives and parameters

Parameter \ Primitive	S-CONTROL-GIVE	
	Request	Indication
SS-user data	U	C(=)

C Presence of the parameter is conditional

U Presence of the parameter is a user option

(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.7.3 Sequence of primitives

The sequence of primitives in a successful transfer of tokens is defined by the time sequence diagram in Figure 11.

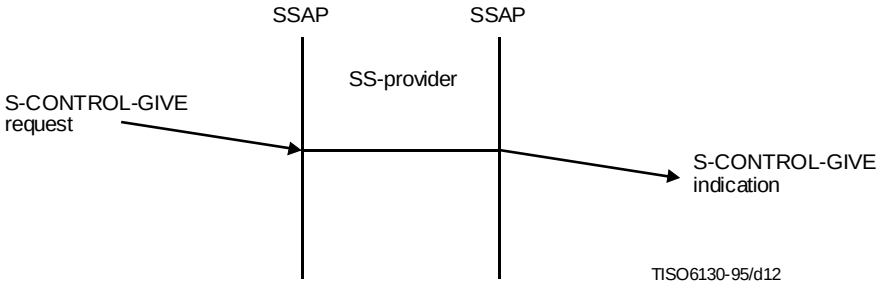


Figure 11 – Give control time sequence diagram

13.8 Minor synchronization point service

13.8.1 Function

The minor synchronization point service allows SS-users to define minor synchronization points in the flow of NSSDUs and TSSDUs. If the activity management functional unit has been selected, this service can only be initiated within an activity. Use of this service is subject to the token restrictions specified in Table 8.

The requestor may request explicit confirmation of a minor synchronization point request through the use of the Type parameter. However, the SS-provider does not require that an explicit confirmation be issued. The acceptor may issue a confirmation even if explicit confirmation is not requested.

Responses are issued in the order in which the corresponding indications were received. A further minor synchronization point request may be made while previous minor synchronization points are unconfirmed.

The confirmation of a minor or major synchronization point confirms all previously unconfirmed minor synchronization points. The number of unconfirmed minor synchronization points is not limited by the SS-provider.

Any semantics associated with request and confirmation of a minor synchronization point have no connotations to the SS-provider.

Minor synchronization points may be issued if the minor synchronize or symmetric synchronize functional units have been selected. When the minor synchronize functional unit is selected, only the owner of the synchronize-minor token (and the data token, if available) may issue a minor synchronize point. When the symmetric synchronize functional unit is selected, the synchronize-minor token is not available, and ownership of the data token (when available) is not required to issue a minor synchronize point.

The data separation service adds functionality to the minor synchronization point service and the symmetric synchronize service to protect data sent before the minor synchronization points from being discarded in the event of a resynchronization. When the data separation functional unit has been selected, the associated service is invoked by the SS-user by setting the data separation parameter to true in the S-SYNC-MINOR request.

13.8.2 Types of primitives and their parameters

Table 17 specifies the types of session service primitives and parameters needed for the minor synchronization point service.

Table 17 – Minor synchronization point primitives and parameters

Primitive / Parameter	S-SYNC-MINOR			
	Request	Indication	Response	Confirm
Type	M	M(=)		
Data separation	C	C(=)		
First Synchronization Point Serial Number	M	M(=)	M	M(=)
SS-user data	U	C(=)	U	C(=)

M	Presence of the parameter is mandatory
C	Presence of the parameter is conditional
U	Presence of the parameter is a user option
blank	The parameter is absent
(=)	The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.8.2.1 Type is a parameter which indicates whether or not explicit confirmation is requested by the SS-user and is transparent to the SS-provider. Its value is one of:

a) explicit;

b) optional.

13.8.2.2 Data separation is a parameter which indicates whether or not data separation is requested by the SS-user. Its value is one of:

a) true;

b) false.

The data separation parameter is set to true to indicate that data separation is requested. The data separation parameter is only present when the data separation functional unit has been selected. The data separation parameter value, if present, is indicated in the indication for information.

13.8.2.3 First Synchronization Point Serial Number is defined in 11.4, 11.4.1.3 and 11.4.2.3. It is in the range 0 to (10**Upper Limit Serial Number) – 2 (see 11.4).

13.8.2.4 SS-user data is a parameter containing an unlimited number of octets of user information.

13.8.3 Sequence of primitives

The sequence of primitives for confirmation of a minor synchronization point is defined by the time sequence diagram in Figure 12.

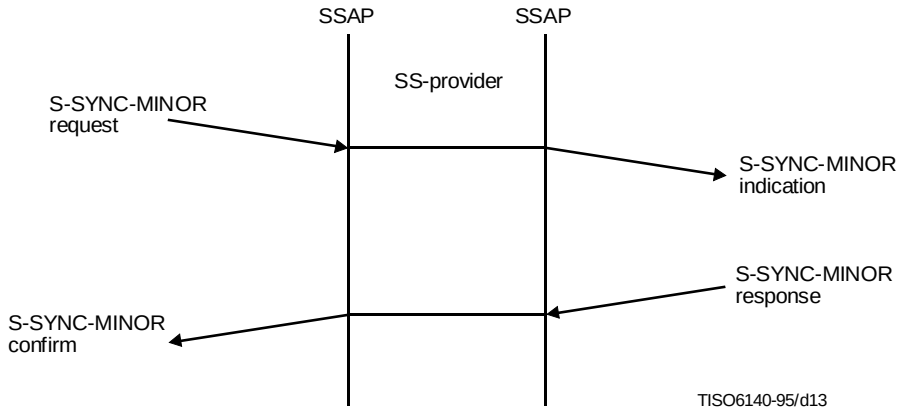


Figure 12 – Minor synchronization point time sequence diagram

The response and confirm may be absent even if the Type parameter is set to explicit in the indication.

The successful confirmation of the minor synchronization point may also be achieved by issuing (instead of the S-SYNC-MINOR response to the synchronization point specified in the S-SYNC-MINOR indication):

a) an S-SYNC-MINOR response to a subsequent S-SYNC-MINOR indication;

b) an S-SYNC-MAJOR response to a subsequent S-SYNC-MAJOR indication;

c) an S-SYNC-MINOR request for a subsequent minor synchronization point (provided that the synchronize-minor token, if available, has been passed from the other SS-user);

d) an S-SYNC-MAJOR request for a subsequent major synchronization point (provided that the synchronize-minor token, if available, and, if necessary, the major/activity token have been passed from the other SS-user).

13.9 Major synchronization point service

13.9.1 Function

The major synchronization point service allows the requestor to define major synchronization points in the flow of NSSDUs, TSSDUs and XSSDUs, to completely separate the flow before and after the major synchronization point. If the activity management functional unit has been selected, this service may only be initiated within an activity. Use of this service is subject to the token restrictions specified in Table 8.

After making the S-SYNC-MAJOR request, the requestor is not able to initiate any services, except for S-TOKEN-GIVE request, S-ACTIVITY-INTERRUPT request, S-ACTIVITY-DISCARD request, S-U-ABORT request or S-RESYNCHRONIZE request until the S-SYNC-MAJOR confirm is received.

After receiving the S-SYNC-MAJOR indication, in addition to any existing restrictions, the acceptor is not able to initiate S-SYNC-MAJOR request, S-ACTIVITY-INTERRUPT request, S-ACTIVITY-DISCARD request, S-ACTIVITY-END request or S-RELEASE request until an S-SYNC-MAJOR response is issued. The acceptor is only able to initiate a S-SYNC-MINOR request in this case if the symmetric synchronize functional unit has been selected.

Expedited data transfer services initiated by the acceptor after issuing a S-SYNC-MAJOR response are not indicated before the S-SYNC-MAJOR confirm.

When the symmetric synchronize functional unit has not been selected, a major synchronization point is defined by a single serial number.

If the symmetric synchronize functional unit has been selected, a major synchronization point is defined by two correlated serial numbers, one for each direction of data flow. The First Synchronization Point Serial Number is defined by the S-SYNC-MAJOR request and indication; this number is associated with the requestor’s sending data flow. The Second Synchronization Point Serial Number is defined by the S-SYNC-MAJOR response and confirm. This number is associated with the requestor's receiving flow.

13.9.2 Types of primitives and their parameters

Table 18 specifies the types of session service primitives and parameters needed for the major synchronization point service.

Table 18 – Major synchronization point primitives and parameters

Primitive / Parameter	S-SYNC-MAJOR			
	Request	Indication	Response	Confirm
First Synchronization Point Serial Number	M	M(=)		
Second Synchronization Point Serial Number			C	C(=)
SS-user data	U	C(=)	U	C(=)

M	Presence of the parameter is mandatory
C	Presence of the parameter is conditional
U	Presence of the parameter is a user option
blank	The parameter is absent
(=)	The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.9.2.1 First Synchronization Point Serial Number is defined in 11.4.1.4 (when the symmetric synchronize functional unit is not selected) and in 11.4.2.4 (when the symmetric synchronize functional unit is selected). It is in the range 0 to (10**Upper Limit Serial Number) – 2 (see 11.4).

Second Synchronization Point Serial Number is defined in 11.4.2.4. It is only used when the symmetric synchronize functional unit is selected. It is in the range 0 to (10**Upper Limit Serial Number) – 2 (see 11.4).

13.9.2.2 SS-user data is a parameter containing an unlimited number of octets of user information.

13.9.3 Sequence of primitives

The sequence of primitives in the successful definition of a major synchronization point is defined by the time sequence diagram in Figure 13.

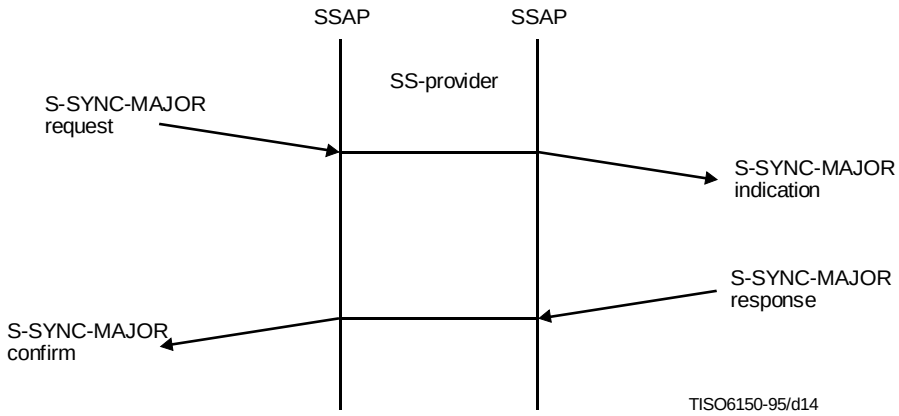


Figure 13 – Major synchronization point time sequence diagram

13.10 Resynchronize service

13.10.1 Function

The resynchronize service is provided to assist orderly re-establishment of communication within the current session connection, typically following an error or lack of response by either of the SS-users or the SS-provider, or disagreements between SS-users. Requesting the service sets the session connection to an agreed defined state, including the positions of the available tokens and the value(s) of the synchronization point serial number(s), which will be the next synchronization point serial number(s) to be used.

If the symmetric synchronize functional unit is selected, then resynchronization can be requested for one or both directions of data flow. If both directions of data flow are requested by the SS-user in this case, then new values will be assigned to both synchronization point serial numbers.

The service may be initiated by either SS-user and has the following characteristics:

a) After issuing the S-RESYNCHRONIZE request, the requestor is not able to initiate any services except S-U-ABORT request, until the S-RESYNCHRONIZE confirm is received.

b) After having received an S-RESYNCHRONIZE indication, the acceptor may only issue:

   1) S-RESYNCHRONIZE response; or

   2) S-RESYNCHRONIZE request (see the Note); or

   3) S-ACTIVITY-DISCARD request (see the Note); or

   4) ACTIVITY-INTERRUPT request (see the Note); or

   5) S-U-ABORT request.

   NOTE – These requests cause a collision of resynchronize requests and therefore the SS-user can only issue the request if he is going to be the collision winner (see clause 16).

c) Service primitives not yet delivered to the SS-user are treated as follows:

   1) If the data separation functional unit has been selected and an S-RESYNCHRONIZE request has been issued:

      i) If all S-SYNC-MINOR requests with the data separation parameter set to true are acknowledged, or no S-SYNC-MINOR requests with the data separation parameter set to true have been issued, all undelivered service primitives are discarded.

      ii) If an S-SYNC-MINOR request with the data separation parameter set to true was issued previously and is unacknowledged, all undelivered service primitives are delivered. After the point where the most recently issued S-SYNC-MINOR request with the data separation parameter set to true is acknowledged, all undelivered service primitives are discarded.

      iii) If an S-SYNC-MINOR request with the data separation parameter set to true was issued previously and is unacknowledged, the confirm corresponding to an S-SYNC-MAJOR response and the indications corresponding to subsequent requests issued by the peer SS-user are discarded. The exception to this is that the indication corresponding to a subsequent S-EXPEDITED-DATA request may be delivered.

   2) If neither the data separation functional unit nor the symmetric synchronize functional unit has been selected, after issuing an S-RESYNCHRONIZE request, all undelivered service primitives are discarded.

   3) If the symmetric synchronize functional unit has been selected and an S-RESYNCHRONIZE request has been issued or an S-RESYNCHRONIZE indication has been received, undelivered service primitives are discarded for the requested direction(s) of flow and confirmations of synchronization points for the requested direction(s) of flow are also discarded.

   Discarding of undelivered service primitives ends with the receipt of an S-RESYNCHRONIZE confirm, S-U-ABORT indication or S-P-ABORT indication.

d) Means are provided for the requesting SS-user either to set or to let the acceptor set a new assignment of each available token.

e) Means are provided to assign new value(s) for the synchronization point serial number(s).

f) When there is an unacknowledged major synchronization point at the time of the S-RESYNCHRONIZE indication, this point remains unacknowledged. In any case, no confirmations should be issued until the

resynchronization is complete and until new indications for synchronization points have been received.

g) Collision of resynchronize requests is resolved, so that only one of the colliding requests is confirmed (see clause 16).

The Resynchronize Type parameter is used to indicate the resynchronize option:

h) "Abandon" is used to request the SS-provider to resynchronize the session connection to a synchronization point which is greater than or equal to V(M). The new synchronization point serial number will be greater than any previous value used on this session connection.

i) "Restart" is used to return to an agreed point which is identified by a past acknowledged or unacknowledged synchronization point serial number. This point cannot be earlier than the last confirmed major synchronization point. The necessary securing of state information associated with the point is the responsibility of the SS-users.

j) "Set" is used to synchronize to any valid synchronization point serial number specified by the SS-users.

If the symmetric synchronize functional unit has been selected and the requesting SS-user has requested both directions of flow, then two Resynchronize Type parameters are specified by the requesting SS-user (one for each direction of flow). The First Resynchronize Type parameter corresponds to the requesting SS-user's sending flow; the Second Resynchronize Type parameter corresponds to the opposite direction of flow. If only one direction of flow is requested, then only the Resynchronize Type parameter associated with that direction of flow is provided by the requesting SS-user.

Only the First Resynchronize Type parameter is used when the symmetric synchronize functional unit has not been selected. In this case, it is a mandatory parameter.

13.10.2 Types of primitives and their parameters

Table 19 specifies the types of session service primitives and parameters needed for the resynchronize service.

Table 19 – Resynchronize primitives and parameters

Primitive / Parameter	S-RESYNCHRONIZE			
	Request	Indication	Response	Confirm
First Resynchronize Type	C	C	C	C(=)
First Synchronization Point Serial Number	C	C	C	C(=)
Second Resynchronize Type	C	C	C	C(=)
Second Synchronization Point Serial Number	C	C	C	C(=)
Assignment of Tokens	C	C(=)	C	C(=)
SS-user data	U	C(=)	U	C(=)

M Presence of the parameter is mandatory
C Presence of the parameter is conditional
U Presence of the parameter is a user option
blank The parameter is absent
(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.10.2.1 First Resynchronize Type is a parameter which specifies one of the resynchronize options. The conditions for its presence are defined in 13.10.1.

Second Resynchronize Type also specifies one of the resynchronize options. The conditions for its presence are defined in 13.10.1.

The value for each of these parameters is one of:

a) abandon;

b) restart;

c) set.

13.10.2.2 First Synchronization Point Serial Number depends on the resynchronize option and is defined in 11.4 and 11.4.1.5 (when the symmetric synchronize functional unit is not selected) and 11.4.2.5 (when the symmetric synchronize functional unit is selected).

Second Synchronization Point Serial Number depends on the resynchronize option and is defined in 11.4.2.5. It is only used when the symmetric synchronize functional unit is selected.

13.10.2.3 Assignment of Tokens is a list of the available tokens for the session connection with values for their assignment following the resynchronization. For each available token, the value in a request/indication is one of:

a) requestor side;

b) acceptor side;

c) acceptor chooses.

The value for a response/confirm is the same as in the request/indication unless that value is c), in which case the acceptor chooses a) or b).

13.10.2.4 SS-user data is a parameter containing an unlimited number of octets of user information.

13.10.3 Sequence of primitives

The sequence of primitives in a successful resynchronization without collision is defined by the time sequence diagram in Figure 14. Collision cases are defined in clause 16.

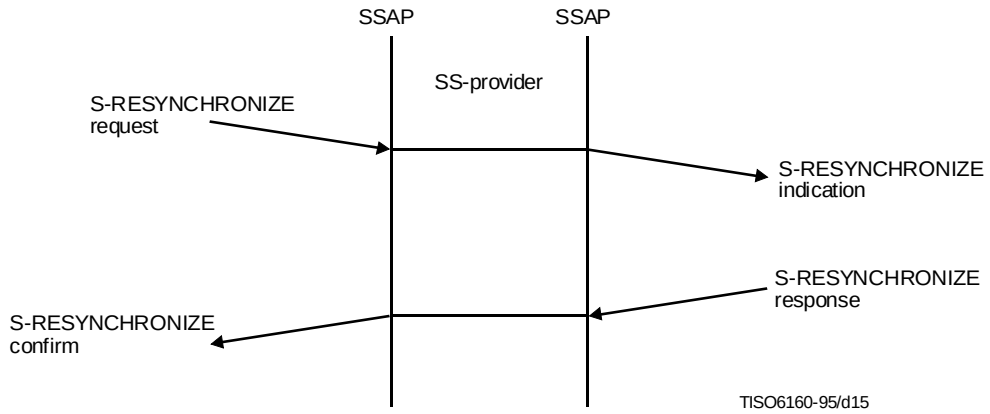


Figure 14 – Resynchronization time sequence diagram

13.11 P-exception reporting service

13.11.1 Function

The P-exception reporting service permits SS-users to be notified of unanticipated situations not covered by other services. If a service cannot be completed due to SS-provider protocol errors or malfunctions, the P-exception reporting service is used to indicate this to both SS-users.

If used with the activity management service, the P-exception reporting service is only permitted while an activity is in progress or waiting for S-CAPABILITY-DATA confirm.

Following an S-P-EXCEPTION-REPORT indication, and until the error condition is cleared:

a) NSSDUs, TSSDUs and XSSDUs will be discarded by the SS-provider;

b) synchronization point indications will not be given to the SS-users.

On receipt of an S-P-EXCEPTION-REPORT indication, either SS-user initiates one of the following services to clear the error:

c) resynchronize;

d) abort;

e) activity interrupt or activity discard;

f) give the data token (see Notes).

The SS-users are not permitted to initiate any other services until the error is cleared.

NOTES

1 It is not recommended that the error condition be cleared by passing the data token when the resynchronize and/or activity management functional units have been selected.

2 If the error condition is cleared by passing the data token, data and synchronization point serial numbers may be lost. However, the SS-provider will keep track of the serial numbers of the synchronization points which have been discarded. Therefore, the synchronization point serial number indicated to the SS-user in a synchronization point request/indication made after the error condition has been cleared will reflect the fact that synchronization points have been discarded during the error condition.

3 XSSDUs sent after the S-TOKEN-GIVE request will be discarded if they overtake the request.

4 Tokens other than the data token may be transferred at the same time.

13.11.2 Types of primitives and their parameters

Table 20 specifies the types of session service primitives and parameters needed for the P-exception reporting service.

Reason is a parameter specifying the reason for the exception report. Its value is one of:

a) protocol error;

b) non-specific error.

Table 20 – P-exception reporting primitives and parameters

Primitive / Parameter	S-P-EXCEPTION-REPORT
	Indication
Reason	M

M Presence of the parameter is mandatory

13.11.3 Sequence of primitives

The sequence of primitives in a successful P-exception report is defined by the time sequence diagram in Figure 15.

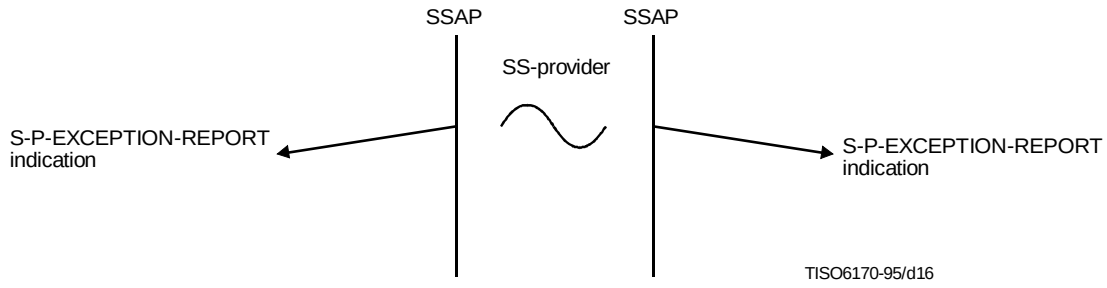


Figure 15 – P-exception report time sequence diagram

13.12 U-exception-reporting service

13.12.1 Function

The U-exception reporting service permits an SS-user to report an exception condition subject to the token restrictions specified in Table 8.

If used with the activity management service, the U-exception reporting service is only permitted while an activity is in progress.

Following an S-U-EXCEPTION-REPORT request, and until the error condition is cleared:

a) NSSDUs, TSSDUs and XSSDUs will be discarded by the SS-provider;

b) synchronization point indications will not be given to the requestor of the S-U-EXCEPTION-REPORT;

c) the requestor is only permitted to issue S-U-ABORT request.

On receipt of an S-U-EXCEPTION-REPORT indication, the acceptor initiates one of the following services to clear the error:

d) resynchronize;

e) abort;

f) activity interrupt or activity discard;

g) give the data token (see Notes).

The acceptor is not permitted to initiate any other services until the error is cleared.

NOTES

1 It is not recommended that the error condition be cleared by passing the data token when the resynchronize and/or activity management functional units have been selected.

2 If the error condition is cleared by passing the data token, data and synchronization point serial numbers may be lost. However, the SS-provider will keep track of the serial numbers of the synchronization points which have been discarded. Therefore, the synchronization point serial number indicated to the SS-user in a synchronization point request/indication made after the error condition has been cleared will reflect the fact that synchronization points have been discarded during the error condition.

3 XSSDUs sent after the S-TOKEN-GIVE request will be discarded if they overtake the request.

4 Tokens other than the data token may be transferred at the same time.

13.12.2 Types of primitives and their parameters

Table 21 specifies the types of session service primitives and parameters needed for the U-exception reporting service.

Table 21 – U-exception reporting primitives and parameters

Parameter \ Primitive	S-U-EXCEPTION-REPORT	
	Request	Indication
Reason	M	M(=)
SS-user data	U	C(=)

M	Presence of the parameter is mandatory
C	Presence of the parameter is conditional
U	Presence of the parameter is a user option
(=)	The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.12.2.1 Reason is a parameter specifying the reason for the exception report and is transparent to the SS-provider. Its value is one of:

a) SS-user receiving ability jeopardized (i.e. data received may not be handled correctly);

b) local SS-user error;

c) sequence error;

d) demand data token;

e) unrecoverable procedural error;

f) non-specific error.

13.12.2.2 SS-user data is a parameter containing an unlimited number of octets of user information.

13.12.3 Sequence of primitives

The sequence of primitives in a successful U-exception report is defined by the time sequence diagram in Figure 16.

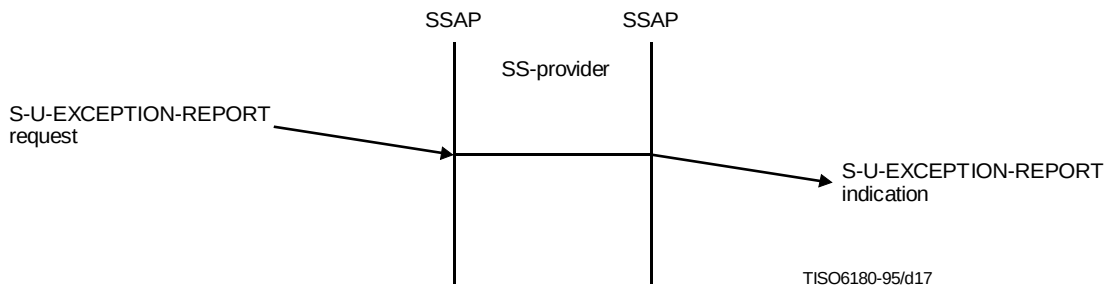


Figure 16 – U-exception report time sequence diagram

13.13 Activity start service

13.13.1 Function

The activity start service allows an SS-user to indicate that a new activity is entered. Values for the next synchronization point serial number(s) are set to one (see 11.4.1.6 and 11.4.2.6). The service can only be initiated if no activity is in progress and subject to the token restrictions specified in Table 8.

13.13.2 Types of primitives and their parameters

Table 22 specifies the types of session service primitives and parameters needed for the activity start service.

Table 22 – Activity start primitives and parameters

Primitive / Parameter	S-ACTIVITY-START	
	Request	Indication
Activity Identifier	M	M(=)
SS-user data	U	C(=)

M Presence of the parameter is mandatory
C Presence of the parameter is conditional
U Presence of the parameter is a user option
(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.13.2.1 Activity Identifier is a parameter which is provided by the SS-users to enable them to identify the new activity and is transparent to the SS-provider. This parameter has a maximum of 6 octets.

13.13.2.2 SS-user data is a parameter containing an unlimited number of octets of user information.

13.13.3 Sequence of primitives

The sequence of primitives in a successful activity start is defined by the time sequence diagram in Figure 17.

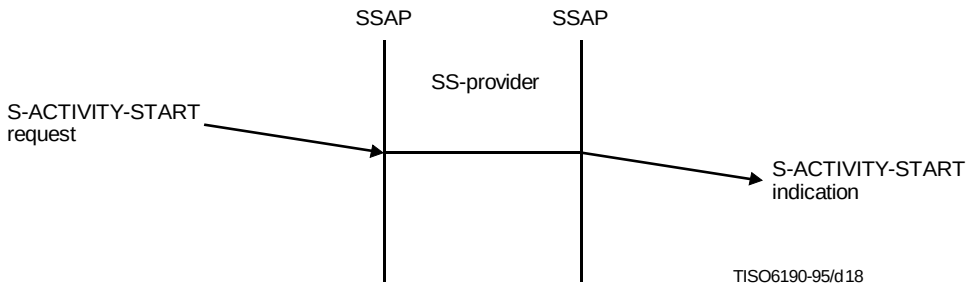


Figure 17 – Activity start time sequence diagram

13.14 Activity resume service

13.14.1 Function

The activity resume service allows an SS-user to indicate that a previously interrupted activity is resumed. A new activity identifier is provided by the SS-user together with the identifier of the activity being resumed and the next synchronization point serial number to be used minus one. In the case when the resumed activity was originally started on another session connection, the session connection identifier of that session connection is also provided by the SS-user.

The service can only be initiated if no activity is in progress and subject to the token restrictions specified in Table 8.

If the symmetric synchronize functional unit has been selected, the SS-user provides two synchronization point serial numbers, one for each direction of flow in the activity being resumed. Otherwise, only one synchronization point serial number (First Synchronization Point Serial Number) is provided.

13.14.2 Types of primitives and their parameters

Table 23 specifies the types of session service primitives and parameters needed for the activity resume service.

Table 23 – Activity resume primitives and parameters

Primitive / Parameter	S-ACTIVITY-RESUME	
	Request	Indication
Activity Identifier	M	M(=)
Old Activity Identifier	M	M(=)
First Synchronization Point Serial Number	M	M(=)
Second Synchronization Point Serial Number	C	C(=)
Old Session Connection Identifier	U	C(=)
SS-user data	U	C(=)

M Presence of the parameter is mandatory
C Presence of the parameter is conditional
U Presence of the parameter is a user option
(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.14.2.1 Activity Identifier is a parameter which is provided by the SS-users to enable them to give a new identifier to the activity being resumed and is transparent to the SS-provider. This parameter has a maximum of 6 octets.

13.14.2.2 Old Activity Identifier is the original identifier of the activity being resumed and is transparent to the SS-provider.

13.14.2.3 First Synchronization Point Serial Number is provided by the SS-user and is defined in 11.4.1.6 and 11.4.2.6.

Second Synchronization Point Serial Number is provided by the SS-user if the symmetric synchronize functional unit has been selected and is defined in 11.4.2.6.

13.14.2.4 Old Session Connection Identifier is the session connection identifier of the session connection in which the activity being resumed was originally started and is transparent to the SS-provider. It consists of:

a) Calling SS-user Reference with a maximum of 64 octets;

b) Called SS-user Reference with a maximum of 64 octets;

c) Common Reference with a maximum of 64 octets;

d) Additional Reference Information with a maximum of 4 octets.

13.14.2.5 SS-user data is a parameter containing an unlimited number of octets of user information.

13.14.3 Sequence of primitives

The sequence of primitives in a successful activity resume is defined by the time sequence diagram in Figure 18.

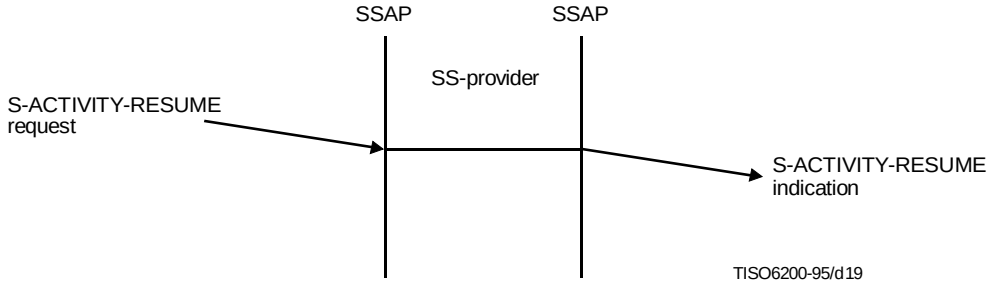


Figure 18 – Activity resume time sequence diagram

13.15 Activity interrupt service

13.15.1 Function

The activity interrupt service allows an SS-user to abnormally terminate the current activity so that work achieved before the interruption is not cancelled, and may be resumed later.

The service can only be initiated subject to the token restrictions specified in Table 8 and:

a) an activity is in progress; or

b) when waiting for an S-CAPABILITY-DATA confirm when the extended control QOS is not provided to the SS-user.

After receipt of the confirm, all available tokens are assigned to the SS-user which issued the request.

After issuing an S-ACTIVITY-INTERRUPT request, the requestor is not able to initiate any services, except S-U-ABORT request, until the S-ACTIVITY-INTERRUPT confirm is received.

After receiving an S-ACTIVITY-INTERRUPT indication, the acceptor is not able to initiate any services, except S-U-ABORT request, until the S-ACTIVITY-INTERRUPT response is issued.

Use of this service may cause loss of data which has not yet been delivered to the SS-user.

13.15.2 Types of primitives and their parameters

Table 24 specifies the types of session service primitives and parameters needed for the activity interrupt service.

Table 24 – Activity interrupt primitives and parameters

Primitive / Parameter	S-ACTIVITY-INTERRUPT			
	Request	Indication	Response	Confirm
Reason			U	C(=)
SS-user data	U	C(=)	U	C(=)

C	Presence of the parameter is conditional
U	Presence of the parameter is a user option
blank	The parameter is absent
(=)	The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.15.2.1 Reason is a parameter specifying the reason for the activity interrupt and is transparent to the SS-provider. Its value is one of:

a) SS-user receiving ability jeopardized (i.e. data received may not be handled correctly);

b) local SS-user error;

c) sequence error;

d) demand data token;

e) unrecoverable procedural error;

f) non-specific error.

13.15.2.2 SS-user data is a parameter containing an unlimited number of octets of user information.

13.15.3 Sequence of primitives

The sequence of primitives in a successful activity interrupt is defined by the time sequence diagram in Figure 19.

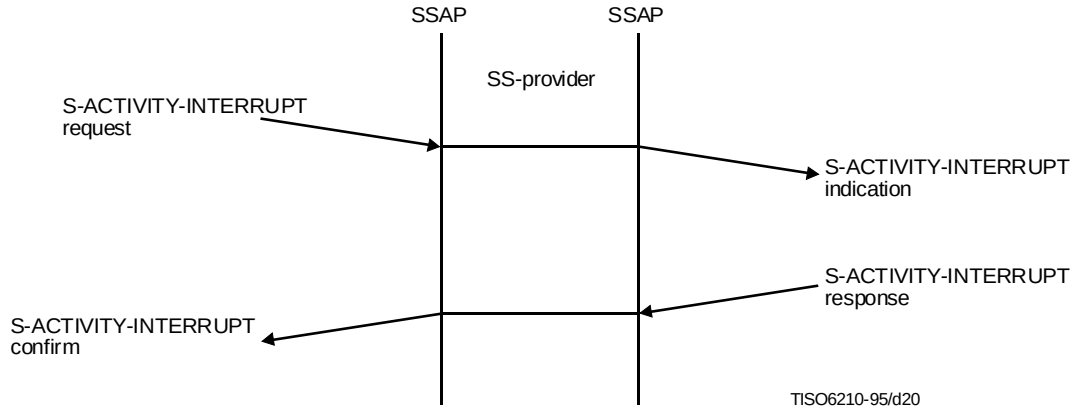


Figure 19 – Activity interrupt time sequence diagram

13.16 Activity discard service

13.16.1 Function

The activity discard service allows an SS-user to abnormally terminate the current activity. There is an implied meaning to the SS-user that the previous content of this activity is cancelled, but this is not controlled by the SS-provider.

The service can only be initiated if an activity is in progress and subject to the token restrictions specified in Table 8. After receipt of the confirm, all available tokens are assigned to the SS-user which issued the request.

After issuing an S-ACTIVITY-DISCARD request, the requestor is not able to initiate any services, except S-U-ABORT request, until the S-ACTIVITY-DISCARD confirm is received.

After receiving an S-ACTIVITY-DISCARD indication, the acceptor is not able to initiate any services, except S-U-ABORT request, until the S-ACTIVITY-DISCARD response is issued.

Use of this service may cause loss of data which has not yet been delivered to the SS-user.

13.16.2 Types of primitives and their parameters

Table 25 specifies the types of session service primitives and parameters needed for the activity discard service.

Table 25 – Activity discard primitives and parameters

Parameter \ Primitive	S-ACTIVITY-DISCARD			
	Request	Indication	Response	Confirm
Reason	U	C(=)		
SS-user data	U	C(=)	U	C(=)

C Presence of the parameter is conditional
U Presence of the parameter is a user option
blank The parameter is absent
(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.16.2.1 Reason is a parameter specifying the reason for the activity discard and is transparent to the SS-provider. Its value is one of:

a) SS-user receiving ability jeopardized (i.e. data received may not be handled correctly);

b) local SS-user error;

c) sequence error;

d) demand data token;

e) unrecoverable procedural error;

f) non-specific error.

13.16.2.2 SS-user data is a parameter containing an unlimited number of octets of user information.

13.16.3 Sequence of primitives

The sequence of primitives in a successful activity discard is defined by the time sequence diagram in Figure 20.

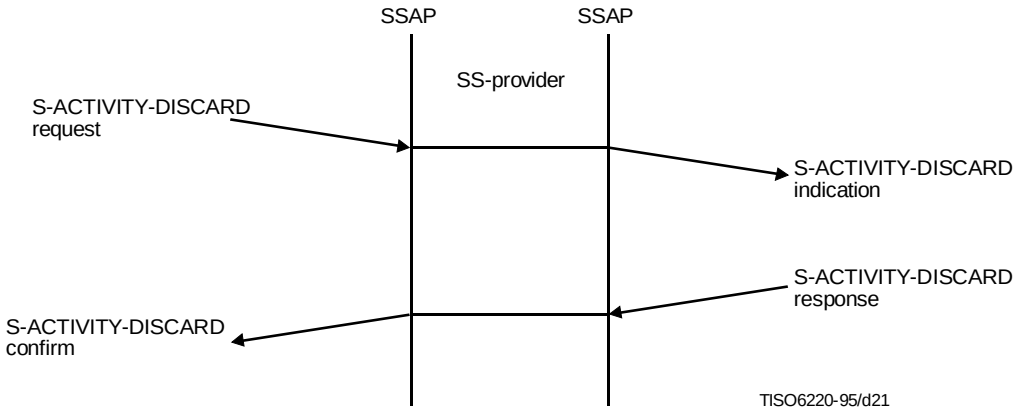


Figure 20 – Activity discard time sequence diagram

13.17 Activity end service

13.17.1 Function

The activity end service allows an SS-user to indicate the end of an activity, and has the effect of setting a major synchronization point. This service can only be invoked if an activity is in progress and subject to the token restrictions specified in Table 8.

After issuing the S-ACTIVITY-END request, in addition to any existing restrictions, the requestor is not able to initiate any services, except for S-U-ABORT request, S-ACTIVITY-INTERRUPT request, S-ACTIVITY-DISCARD request or S-TOKEN-GIVE request until the S-ACTIVITY-END confirm is received.

After receiving the S-ACTIVITY-END indication, in addition to any existing restrictions, the acceptor is not able to initiate S-SYNC-MAJOR request, S-SYNC-MINOR request, S-ACTIVITY-INTERRUPT request, S-ACTIVITY-DISCARD request, S-ACTIVITY-END request or S-RELEASE request until the S-ACTIVITY-END response is issued.

If the activity management functional unit has been selected, the SS-user is not allowed to initiate any services, except activity start, activity resume, token management, capability data, expedited data, typed data, normal data, release or abort, until an activity is started or resumed.

13.17.2 Types of primitives and their parameters

Table 26 specifies the types of session service primitives and parameters needed for the activity end service.

Table 26 – Activity end primitives and parameters

Parameter \ Primitive	S-ACTIVITY-END			
	Request	Indication	Response	Confirm
First Synchronization Point Serial Number	M	M(=)		
Second Synchronization Point Serial Number			C	C(=)
SS-user data	U	C(=)	U	C(=)

M	Presence of the parameter is mandatory
C	Presence of the parameter is conditional
U	Presence of the parameter is a user option
blank	The parameter is absent
(=)	The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

13.17.2.1 First Synchronization Point Serial Number is defined in 11.4.1.6 and 11.4.2.6.

Second Synchronization Point Serial Number is present when the symmetric synchronize functional unit has been selected, in order to identify the serial number in the direction of flow on which the confirmation travels. It is defined in 11.4.2.6.

13.17.2.2 SS-user data is a parameter containing an unlimited number of octets of user information.

13.17.3 Sequence of primitives

The sequence of primitives in a successful normal termination of an activity is defined by the time sequence diagram in Figure 21.

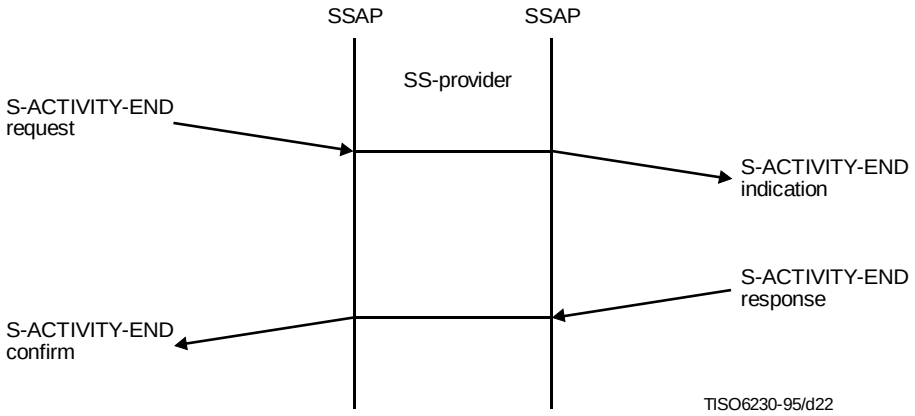


Figure 21 – Activity end time sequence diagram

14 Session connection release phase

14.1 Orderly release service

14.1.1 Function

The orderly release service is always provided and allows either SS-user to release the session connection in an orderly manner. This is done cooperatively between the two SS-users without the loss of data after all in-transit data have been delivered and accepted by both SS-users.

Use of this service is subject to the token restrictions specified in Table 8. If the release token is available the acceptor may refuse the release and continue the session connection without loss of data. If the release token is not available, the acceptor cannot refuse the release.

14.1.2 Types of primitives and their parameters

Table 27 specifies the types of session service primitives and parameters needed for the orderly release service.

Table 27 – Orderly release primitives and parameters

Parameter \ Primitive	S-RELEASE			
	Request	Indication	Response	Confirm
Result			M	M(=)
SS-user data	U	C(=)	U	C(=)

M	Presence of the parameter is mandatory
C	Presence of the parameter is conditional
U	Presence of the parameter is a user option
blank	The parameter is absent
(=)	The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

14.1.2.1 Result is a parameter indicating whether or not the session release is granted. Its value may be one of:

a) affirmative;

b) negative.

The latter value may be given only if the release token is available.

14.1.2.2 SS-user data is a parameter containing an unlimited number of octets of user information.

14.1.3 Sequence of primitives

The sequence of primitives in a successful orderly session release is defined by the time sequence diagram in Figure 22.

A collision of S-RELEASE requests may occur when no tokens are available. This results in S-RELEASE indications to both SS-users. In this case, the calling SS-user should send the S-RELEASE response after receiving the S-RELEASE indication from the called SS-user. The called SS-user should not send his S-RELEASE response before receiving the S-RELEASE confirm from the calling SS-user.

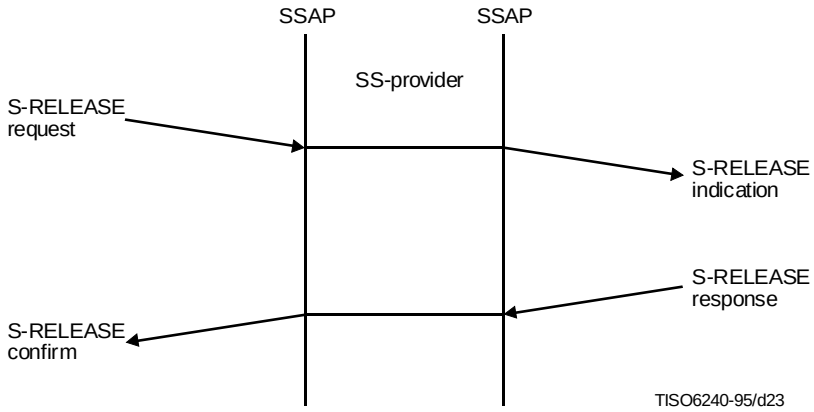


Figure 22 – Orderly session release time sequence diagram

14.2 U-abort service

14.2.1 Function

The U-abort service provides the means by which either SS-user can instantaneously release the session connection and have the other SS-user informed of this release. Use of this service will cause loss of undelivered data.

14.2.2 Types of primitives and their parameters

Table 28 specifies the types of session service primitives and parameters needed for the U-abort service.

SS-user data is a parameter containing an unlimited number of octets of user information.

Table 28 – U-abort primitives and parameters

Primitive / Parameter	S-U-ABORT	
	Request	Indication
SS-user data	U	C(=)

C Presence of the parameter is conditional

U Presence of the parameter is a user option

(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

14.2.3 Sequence of primitives

The sequence of primitives in a successful U-abort is defined by the time sequence diagram in Figure 23.

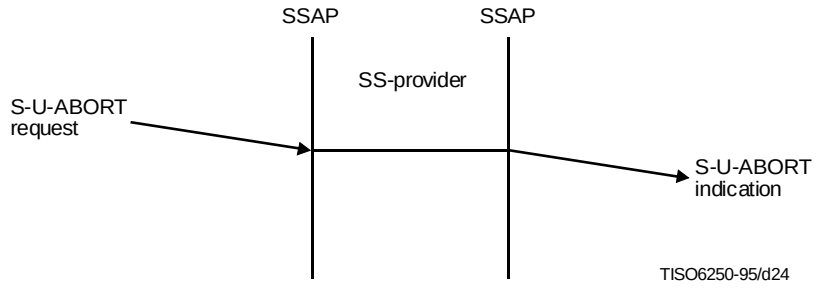


Figure 23 – U-abort time sequence diagram

14.3 P-abort service

14.3.1 Function

The P-abort service provides the means by which the SS-provider may indicate the release of the session connection for reasons internal to the SS-provider. Use of this service will cause loss of undelivered data. A reason code of limited size is passed from the SS-provider to the SS-user.

14.3.2 Types of primitives and their parameters

Table 29 specifies the types of session service primitives and parameters needed for the P-abort service.

Reason is a parameter indicating the reason for the abort. Its value is one of:

a) transport disconnect;

b) protocol error;

c) undefined;

d) implementation restriction stated in the PICS.

Table 29 – P-abort primitives and parameters

Primitive / Parameter	S-P-ABORT
	Indication
Reason	M
M Presence of the parameter is mandatory	

14.3.3 Sequence of primitives

The sequence of primitives in a successful P-abort is defined by the time sequence diagram in Figure 24.

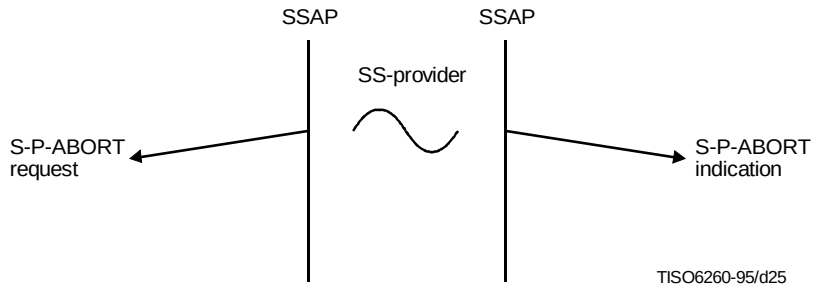


Figure 24 – P-abort time sequence diagram

15 Sequences of primitives

15.1 State tables

Annex A contains state tables which define the constraints on the sequences in which the session service primitives may occur. The constraints determine the order in which the session services occur, but do not fully specify when they may occur. Other constraints will affect the ability of an SS-user or the SS-provider to issue a primitive at any particular time.

15.2 Sequences of primitives at one session connection end-point

The possible sequences of primitives at one session connection end-point may be derived directly from the state tables in Annex A.

16 Collision

16.1 Collision as viewed by the SS-user

The SS-provider resolves collisions between those requests that may destroy SS-user data. If a collision occurs, one of the SS-users will receive an unexpected indication while awaiting one of the following:

a) S-RESYNCHRONIZE confirm;

b) S-ACTIVITY-INTERRUPT confirm;

c) S-ACTIVITY-DISCARD confirm;

d) clearing the error state after issuing an S-U-EXCEPTION REPORT request.

Table 30 defines the indications that may be received which indicate that the SS-user has lost a collision resolved by the SS-provider.

16.2 Collision resolution by the SS-provider

16.2.1 Collision resolution when the symmetric synchronize functional unit is not selected

When the symmetric synchronize functional unit has not been selected, the SS-provider resolves colliding SS-user requests according to the following rules.

In the case of collision between two of the following types of requests, the first in the list takes precedence.

a) S-U-ABORT request;

b) S-ACTIVITY-DISCARD request;

c) S-ACTIVITY-INTERRUPT request;

d) S-RESYNCHRONIZE (abandon) request;

e) S-RESYNCHRONIZE (set) request;

f) S-RESYNCHRONIZE (restart) request;

g) S-U-EXCEPTION-REPORT request.

Table 30 – Indications resulting from collision resolution

SS-user receives / SS-user is waiting for	ER	RR	RS	RA	AI	AD	AB
Clearing error state after S-U-EXCEPTION-REPORT request		X	X	X	X	X	X
S-RESYNCHRONIZE (restart) confirm		X	X	X	X	X	X
S-RESYNCHRONIZE (set) confirm			X	X	X	X	X
S-RESYNCHRONIZE (abandon) confirm				X	X	X	X
S-ACTIVITY-INTERRUPT confirm							X
S-ACTIVITY-DISCARD confirm							X

X indication may be received
Blank indication will not be received
AB S-P-ABORT indication or S-U-ABORT indication
AD S-ACTIVITY-DISCARD indication
AI S-ACTIVITY-INTERRUPT indication
ER S-U-EXCEPTION-REPORT indication or S-P-EXCEPTION-REPORT indication
RA S-RESYNCHRONIZE (abandon) indication
RR S-RESYNCHRONIZE (restart) indication
RS S-RESYNCHRONIZE (set) indication

Possible collisions of the same request are handled as follows:

h) If two S-RESYNCHRONIZE (abandon) requests collide, the calling SS-user request takes precedence;

i) If two S-RESYNCHRONIZE (restart) requests collide, the request with the lowest serial number takes precedence. If the serial numbers are equal, the calling SS-user request takes precedence;

j) If two S-RESYNCHRONIZE (set) requests collide, the calling SS-user request takes precedence.

16.2.2 Collision resolution when the symmetric synchronize functional unit is selected

When symmetric synchronization is selected and two S-RESYNCHRONIZE requests collide, each direction of flow is considered separately, with the result for that direction of flow being the highest priority request. The request options are prioritized below, with the highest priority listed first:

a) S-RESYNCHRONIZE (abandon) request;

b) S-RESYNCHRONIZE (set) request;

c) S-RESYNCHRONIZE (restart) request.

Possible collisions of the same request for a given direction of flow are handled as follows:

d) if two S-RESYNCHRONIZE (abandon) requests collide, the result is that a resynchronize abandon will be performed on that direction of flow; the session provider determines the new serial number;

e) if two S-RESYNCHRONIZE (restart) requests collide, the result is the lower of the numbers offered by the two SS-users for that direction of flow;

f) if two S-RESYNCHRONIZE (set) requests collide, the result is the serial number proposed by the calling SS-user request for that direction of flow.

If neither SS-user requested resynchronization for a particular direction of flow, that flow is not affected.

If only one SS-user requested resynchronization for a particular direction of flow, then the result is the option and serial number provided by that SS-user for that direction of flow.

If all results for the affected direction(s) of flow were provided by one and only one SS-user, then that SS-user's request "wins" the collision and the other request is discarded. Otherwise, the parameters of the calling SS-user's request are set equal to the collision results for each affected direction of flow, independent of which SS-user provided the result. These results are used in the Resynchronize indication, response, and confirmation.

If an S-RESYNCHRONIZE request for the requestor's sending flow only collides with an incoming S-MAJOR-SYNC indication, a session layer collision has occurred. The incoming S-MAJOR-SYNC indication will be issued to the receiving SS-user, but the SS-user is not permitted to issue a S-SYNC-MAJOR response in this case.

SECTION 3 – DEFINITION OF CONNECTIONLESS-MODE SESSION SERVICE PRIMITIVES

17 S-UNIT-DATA

17.1 Function

Session-connectionless-mode transmission service primitives can be used to transmit an independent, self-contained SSDU from one SSAP to another SSAP in a single session service access. The SSDU is independent in the sense that it bears no relationship to any other SSDU transmitted through the invocation of the connectionless-mode service or the connection-mode service. It is self-contained in that all of the information required to deliver the SSDU is presented to the SS-provider, together with the user data to be transmitted, in a single service access; thus no initial establishment or subsequent release of a session connection is required, provided that the SS-users exist and are known to the SS-provider.

An SSDU transmitted using session-connectionless-mode transmission is not considered by the SS-provider to be related in any way to any other SSDU. In particular, although the session service maintains the integrity of individual SSDUs, it does not necessarily guarantee delivery to the receiving SS-user in the order in which they are presented by the sending SS-user, or at all.

No means are provided for detection of transmission errors by the sending or receiving SS-user.

No means are provided by which the receiving SS-user may control the rate at which the sending SS-user may send SSDUs (peer-to-peer flow control). The SS-provider will not maintain any state information relative to any aspect of the flow of information between any specific combination of SSAPs. Flow control exerted by the SS-provider upon the sending SS-user can only be described in terms of a specific interface.

Any negotiation between SS-users is outside the scope of the connectionless-mode session service.

17.2 Types of Primitives and Parameters

Table 31 specifies the types of session service primitives and parameters needed for the session-connectionless-mode transmission service.

Table 31 – S-Unit-Data primitives and parameters

Primitive / Parameter	S-UNIT-DATA	
	Request	Indication
Calling Session Address	M	M
Called Session Address	M	M
Quality of Service	M	
SS-user data	M	M(=)

M Presence of the parameter is mandatory

(=) The value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive

17.2.1 Calling Session Address and Called Session Address are session-service-access-point addresses. The connection-mode and connectionless-mode session services both use the same session-service-access-point addressing scheme, as described in 12.1.2.

17.2.2 Quality of Service is a list of sub-parameters. The definition of the sub-parameters related to the quality of the connectionless-mode session service is found in clause 10.

17.2.3 SS-user data allows the transmission of SS-user data between SS-users, without modification by the SS-provider. The SS-user may transmit any integral number of octets up to a limit due to implementation restriction stated in the PICS.

17.3 Sequence of Primitives

The sequence of primitives in a successful session-connectionless-mode transmission is defined by the time sequence diagram in Figure 25.

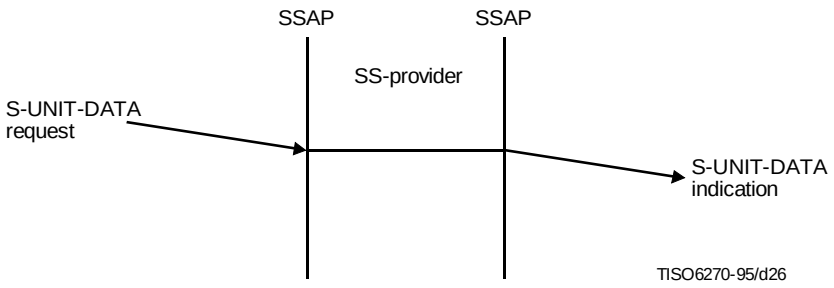


Figure 25 – Session connectionless-mode transmission time sequence diagram

Annex A

State tables

(This annex forms an integral part of this Recommendation | International Standard)

A.1 General

This annex describes the session service in terms of state tables. The state tables show the state of an SS-user, the events that occur at the session service boundary, the actions taken by the SS-user and the resultant state.

These state tables do not constitute a formal definition of the session service; they are included to provide a more precise definition of the relationships between session service primitives defined in clauses 12, 13 and 14.

Table A.1 specifies the abbreviated name and name of each incoming event generated by the SS-provider.

Table A.2 specifies the abbreviated name and name of each state.

Table A.3 specifies the abbreviated name and name of each outgoing event generated by the SS-user.

Table A.4 summarizes the operations on the variables V(A), V(M), V(R) and Vsc when the symmetric synchronize functional unit has not been selected.

Table A.5 summarizes the operations on the variables V(A), V(M), V(R) and Vsc when the symmetric synchronize functional unit has been selected.

Table A.6 specifies the specific actions.

Table A.7 specifies the predicates.

Tables A.8 to A.15 specify the state tables when the symmetric synchronize functional unit has not been selected.

Tables A.16 to A.23 specify the state tables when the symmetric synchronize functional unit has been selected.

A.2 Notation for state tables

A.2.1 Incoming events, states and outgoing events are represented by their abbreviated names.

A.2.2 Specific actions are represented by the notation [n], where n is the number of the specific action in Table A.6.

A.2.3 Predicates are represented by the notation pn, where n is the number of the predicate in Table A.7.

A.2.4 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 or outgoing event which is invalid is left blank.

A.3.2 The intersection of each state and incoming or outgoing event which is valid contains entries which are either:

a) an action list which:

   1) may contain specific actions;

   2) always contains the resultant state; or

b) one or more conditional action lists, each consisting of:

   1) a predicate expression comprising predicates and boolean operators;

   2) an action list [as in A.3.2, a)].

NOTE – The action lists and conditional action lists use the notation in A.2.

A.4 Actions to be taken by the SS-user

The state tables define the action to be taken by the SS-user.

A.4.1 Invalid intersections

If the intersection of the state and an incoming or outgoing event is invalid, any action taken by the SS-user 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 SS-user 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 SS-user 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 SS-user shall take one of the actions defined in A.4.1.

A.5 Definitions of sets and variables

The following sets and variables are specified in this Recommendation | International Standard.

A.5.1 Functional units

The set of all functional units specified in this Recommendation | International Standard is defined as:

fu-dom = {FD, HD, EXCEP, TD, NR, SY, SS, DS, 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
SS	Symmetric synchronize functional unit
DS	Data separation 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 S-CONNECT response is issued or the S-CONNECT confirm is received.

A.5.2 Tokens

The set of all tokens specified in this Recommendation | International Standard is defined as

tk-dom = {mi, ma, tr, dk}

where

mi is the synchronize-minor token

ma is the major/activity token

tr is the release token

dk is the 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 SS-user;
- OWNED(t) = false:if token not assigned to the SS-user.

OWNED(t) is not defined if AV(t) = false. OWNED(t) is set when:

1) the S-CONNECT response is issued or the S-CONNECT confirm is received; or
2) the S-RESYNCHRONIZE response is issued or the S-RESYNCHRONIZE confirm is received; or
3) the S-TOKEN-GIVE request is issued or the S-TOKEN-GIVE indication is received; or
4) the S-CONTROL-GIVE request is issued or the S-CONTROL-GIVE indication is received;
5) the S-ACTIVITY-INTERRUPT response is issued or the S-ACTIVITY-INTERRUPT confirm is received;
6) the S-ACTIVITY-DISCARD response is issued or the S-ACTIVITY-DISCARD confirm is received.

c) I(t), for t in tk-dom, is a function which, when true, indicates that the SS-user has Initiating rights for the behaviour controlled by the token. This applies even if the corresponding token is not available:

$$I(t) = \neg AV(t) \text{ OR } OWNED(t)$$

d) A(t), for t in tk-dom, is a function which, when true, indicates that the SS-user has Accepting rights for the behaviour controlled by the token. This applies even if the corresponding token is not available:

$$A(t) = \neg AV(t) \text{ OR } OWNED(t)$$

e) II(t), for t in tk-dom, is a function which, when true, indicates that the SS-user 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:

$$II(t) = AV(t) \text{ AND } OWNED(t)$$

f) AA(t), for t in tk-dom, is a function which, when true, indicates that the SS-user has Accepting rights as A(t), but only if the corresponding token is available, but not owned:

$$AA(t) = AV(t) \text{ AND } \neg OWNED(t)$$

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 (e.g. on receipt of a S-RELEASE indication)

b) ANY(f, s), for f in F and s in S:

ANY(f, s) = true: any f(t) = true for t in s and 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 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 S-ACTIVITY-START request or S-ACTIVITY-RESUME request is issued or the S-ACTIVITY-START indication or S-ACTIVITY-RESUME indication is received [only possible when FU(ACT) = true];

c) Vact is set false when the S-ACTIVITY-DISCARD response or S-ACTIVITY-INTERRUPT response is issued or the S-ACTIVITY- DISCARD confirm or S-ACTIVITY-INTERRUPT confirm is received;

d) Vact is set false when the S-ACTIVITY-END response is issued or the S-ACTIVITY-END confirm is received.

A.5.4.2 Vrsp and Vrspnb

If the symmetric synchronize functional unit has not been selected, 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.

Vrspnb indicates the serial number in the case of resynchronize restart.

Vrsp and, if necessary Vrspnb, are set when an S-RESYNCHRONIZE request is issued or an S-RESYNCHRONIZE indication is received. Vrsp is set to no when the SS-user goes to STA 713.

A.5.4.3 Vcoll

Vcoll is a boolean variable having the following values:

Vcoll = true: a collision of S-RELEASE requests has been detected;

Vcoll = false: there has not been a collision of S-RELEASE requests.

This variable is set false during the session connection establishment phase.

A.5.4.4 V(A)

If the symmetric synchronize functional unit has not been selected, V(A) is used by the SS-user and is the lowest serial number to which a synchronization point confirmation is expected. No confirmation is expected when V(A) = V(M).

A.5.4.5 V(M)

If the symmetric synchronize functional unit has not been selected, V(M) is used by the SS-user and is the next serial number to be used.

A.5.4.6 V(R)

If the symmetric synchronize functional unit has not been selected, V(R) is used by the SS-user and is the lowest serial number to which resynchronization restart is permitted.

A.5.4.7 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 session connection establishment phase and when an S-SYNC-MINOR request is issued. Vsc is set true when an S-SYNC-MINOR indication is received.

Vsc is not used if the symmetric synchronize functional unit has been selected.

NOTE – Table A.4 summarizes the operations on V(A), V(M), V(R) and Vsc.

A.5.4.8 Vdnr

Vdnr is a boolean variable having the following values:

Vdnr = true: an S-RELEASE confirm has been received in STA09 (following a collision of S-RELEASE requests);

Vdnr = false: no S-RELEASE confirm has been received.

Vdnr is set false during the session connection establishment phase.

A.5.4.9 Vrsps, Vrspr, Vspnbs, and Vrspnbr

When a single serial numbering scheme is used, Vrspnb indicates the serial number in the case of resynchronize restart.

When symmetric synchronization is used, Vrspnbr indicates the serial number for the SS-user's receiving flow and Vrspnbs indicates the serial number for the SS-user's sending flow in the case of resynchronize restart. Vrspr indicates the type of resynchronize (a, s, r, no) for the receiving flow; Vrsps indicates the type of resynchronize for the sending flow.

Vrsps and Vrspr and, if necessary Vrspnbs and Vrspnbr, are set when one of the following events occurs: a Resynchronize request or indication, an Activity Interrupt request or indication, or an Activity Discard request or indication. In the case of a collision, the variables are updated to contain the prevailing values (see A.5.4.14).

A.5.4.10 V(As), V(Ar)

When the symmetric synchronize functional unit is selected, V(As) and V(Ar) are used by the SS-provider to manage symmetric synchronization point confirmations.

V(As) is the lowest serial number on the SS-user's sending data flow to which a synchronization point confirmation is expected to be received. No confirmation is expected to be received when V(As) = V(Ms).

V(Ar) is the lowest serial number on the SS-user's receiving data flow for which a confirmation has not yet been sent. No confirmation will be sent by the SS-user when V(Ar) = V(Mr).

A.5.4.11 V(Ms), V(Mr)

When the symmetric synchronize functional unit is selected, V(Ms) and V(Mr) are used to maintain the next symmetric synchronization serial number to be used on the sending and receiving data flows respectively.

V(Ms) is the serial number of the next synchronization point to be sent. V(Mr) is the serial number of the next synchronization point to be received.

A.5.4.12 V(Rs), V(Rr)

When the symmetric synchronize functional unit is selected, V(Rs) and V(Rr) are used to maintain the lowest serial numbers to which resynchronize restart is permitted.

V(Rs) is the lowest serial number on the SS-user's sending data flow to which resynchronize restart is permitted.

V(Rr) is the lowest serial number on the SS-user's receiving data flow to which resynchronize restart is permitted.

A.5.4.13 Rcv-flow-in-resync – Snd-flow-in-resync

Rcv-flow-in-resync and Snd-flow-in-resync are boolean variables used by the SS-user to maintain whether the receiving and sending flows, respectively, are in the process of being resynchronized. If true, the appropriate flow is in the resynchronize process.

When the resynchronize functional unit has been selected without the symmetric synchronize functional unit, resynchronization on a single direction of flow is not permitted. In this case, both boolean variables are always set when resynchronization is in progress.

A.5.4.14 SS-userwinner

When both SS-users have requested resynchronization, the boolean function SS-userwinner is calculated to determine which SS-user wins against the colliding event.

When the symmetric synchronize functional unit has been selected, the SS-userwinner condition is calculated as follows:

a) The resynchronize type and serial number values are evaluated for each direction of flow according to the parameters of the received event. The local SS-user's value for Vrsps (or Vrspr) is compared to the other SS-user's Vrsps (or Vrspr) with the following ordering rule:

   dsc prevails over int;

   int prevails over a;

   a prevails over s;

   s prevails over r;

   r prevails over no.

   If both are equal to r for a direction of flow, then the local SS-user's value for Vrspnbs (or Vrspnbr) is compared to the other SS-user's value and the lower value prevails.

b) If the local SS-user's values of Vrsps, Vrspr, Vrspnbs and/or Vrspnbr all prevail, then the SS-userwinner condition is true (in this case, the local SS-user wins the collision).

c) If the other SS-user's values of Vrsps, Vrspr, Vrspnbs and/or Vrspnbr all prevail, then the SS-userwinner condition is false (in this case, the local SS-user loses the collision).

d) Otherwise:

   1) if the local SS-user is the initiator of the session connection, then the SS-userwinner condition is true;

   2) if the local SS-user is the acceptor of the session connection, then the SS-userwinner condition is false.

If the SS-user is winner (SS-userwinner condition is true), then the local resynchronization wins against the colliding one.

If the SS-user is not winner (SS-userwinner condition is false), then the local resynchronization loses to the colliding one.

In all cases, Vrsps, Vrspr, Vrspnbs and Vrspnbr are updated to reflect the prevailing values.

NOTE – Table A.4 summarizes the operations on V(A), V(M), V(R) and Vsc when the symmetric synchronize functional unit has not been selected. Table A.5 summarizes the operations on V(As), V(Ar), V(Ms), V(Mr), V(Rs) and V(Rr) when symmetric synchronization is in use.

Table A.1 – Events generated by the SS-provider

Abbreviated name	Name and description
SACTDind	S-ACTIVITY-DISCARD indication primitive
SACTDcnf	S-ACTIVITY-DISCARD confirm primitive
SACTEind	S-ACTIVITY-END indication primitive
SACTEcnf	S-ACTIVITY-END confirm primitive
SACTIind	S-ACTIVITY-INTERRUPT indication primitive
SACTIcnf	S-ACTIVITY-INTERRUPT confirm primitive
SACTRind	S-ACTIVITY-RESUME indication primitive
SACTSind	S-ACTIVITY-START indication primitive
SCDind	S-CAPABILITY-DATA indication primitive
SCDcnf	S-CAPABILITY-DATA confirm primitive
SCGind	S-CONTROL-GIVE indication primitive
SCONind	S-CONNECT indication primitive
SCONcnf+	S-CONNECT (accept) confirm primitive
SCONcnf−	S-CONNECT (reject) confirm primitive
SDTind	S-DATA indication primitive
SEXind	S-EXPEDITED-DATA indication primitive
SGTind	S-TOKEN-GIVE indication primitive
SPABind	S-P-ABORT indication primitive
SPERind	S-P-EXCEPTION-REPORT indication primitive
SPTind	S-TOKEN-PLEASE indication primitive
SRELind	S-RELEASE indication primitive
SRELcnf+	S-RELEASE (accept) confirm primitive
SRELcnf−	S-RELEASE (reject) confirm primitive
SRSYNind	S-RESYNCHRONIZE indication primitive
SRSYNcnf	S-RESYNCHRONIZE confirm primitive
SSYNMind	S-SYNC-MAJOR indication primitive
SSYNMcnf	S-SYNC-MAJOR confirm primitive
SSYNmind	S-SYNC-MINOR indication primitive
SSYNmdind	S-SYNC-MINOR (data separation) indication primitive
SSYNmcnf	S-SYNC-MINOR confirm primitive
STDind	S-TYPED-DATA indication primitive
SUABind	S-U-ABORT indication primitive
SUERind	S-U-EXCEPTION-REPORT indication primitive

Table A.2 – States

Abbreviated name	Name and description
STA 01	Idle
STA 02A	Wait for S-CONNECT confirm
STA 03	Wait for S-RELEASE confirm
STA 04A	Wait for S-SYNC-MAJOR confirm
STA 04B	Wait for S-ACTIVITY-END confirm
STA 05A	Wait for S-RESYNCHRONIZE confirm
STA 05B	Wait for S-ACTIVITY-INTERRUPT confirm
STA 05C	Wait for S-ACTIVITY-DISCARD confirm
STA 08	Wait for S-CONNECT response
STA 09	Wait for S-RELEASE response
STA 10A	Wait for S-SYNC-MAJOR response
STA 10B	Wait for S-ACTIVITY-END response
STA 11A	Wait for S-RESYNCHRONIZE response
STA 11B	Wait for S-ACTIVITY-INTERRUPT response
STA 11C	Wait for S-ACTIVITY-DISCARD response
STA 19	Wait for a recovery indication
STA 20	Wait for a recovery request
STA 21	Wait for S-CAPABILITY-DATA confirm
STA 22	Wait for S-CAPABILITY DATA response
STA 713	Data transfer state

Table A.3 – Events generated by the SS-user

Abbreviated name	Name and description
SACTDreq	S-ACTIVITY-DISCARD request primitive
SACTDrsp	S-ACTIVITY-DISCARD response primitive
SACTEreq	S-ACTIVITY-END request primitive
SACTErsp	S-ACTIVITY-END response primitive
SACTIreq	S-ACTIVITY-INTERRUPT request primitive
SACTIrsp	S-ACTIVITY-INTERRUPT response primitive
SACTRreq	S-ACTIVITY-RESUME request primitive
SACTSreq	S-ACTIVITY-START request primitive
SCDreq	S-CAPABILITY-DATA request primitive
SCDrsp	S-CAPABILITY-DATA response primitive
SCGreq	S-CONTROL-GIVE request primitive
SCONreq	S-CONNECT request primitive
SCONrsp+	S-CONNECT (accept) response primitive
SCONrsp−	S-CONNECT (reject) response primitive
SDTreq	S-DATA request primitive
SEXreq	S-EXPEDITED-DATA request primitive
SGTreq	S-TOKEN-GIVE request primitive
SPTreq	S-TOKEN-PLEASE request primitive
SRELreq	S-RELEASE request primitive
SRELrsp+	S-RELEASE (accept) response primitive
SRELrsp−	S-RELEASE (reject) response primitive
SRSYNreq(a)	S-RESYNCHRONIZE (abandon) request primitive
SRSYNreq(r)	S-RESYNCHRONIZE (restart) request primitive
SRSYNreq(s)	S-RESYNCHRONIZE (set) request primitive
SRSYNrsp	S-RESYNCHRONIZE response primitive
SSYNMreq	S-SYNC-MAJOR request primitive
SSYNMrsp	S-SYNC-MAJOR response primitive
SSYNmreq	S-SYNC-MINOR request primitive
SSYNmdreq	S-SYNC-MINOR (data separation) request primitive
SSYNmrsp	S-SYNC-MINOR response primitive
STDreq	S-TYPED-DATA request primitive
SUABreq	S-U-ABORT request primitive
SUERreq	S-U-EXCEPTION-REPORT request primitive

Table A.4 – Operations on variables when the symmetric synchronize functional unit has not been selected

Events	Condition for valid 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
SSYNMind SACTEind		if Vsc true	unchanged	V(M) + 1	unchanged	unchanged
		if Vsc false	set to V(M)	V(M) + 1	unchanged	unchanged
SSYNmind		if Vsc true	unchanged	V(M) + 1	unchanged	true
		if Vsc false	set to V(M)	V(M) + 1	unchanged	true
SSYNMrsp SACTErsp	sn = V(M) − 1		set to V(M)	unchanged	set to V(M)	unchanged
SSYNMcnf SACTEcnf			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
SSYNmcnf	Vsc = false and V(M) > sn ≥ V(A)*		set to sn + 1	unchanged	unchanged	unchanged
SRSYNreq	r: V(M) ≥ sn ≥ V(R)		unchanged	unchanged	unchanged	unchanged
SRSYNind		abandon restart set	unchanged unchanged unchanged	set to sn unchanged unchanged	unchanged unchanged unchanged	unchanged unchanged unchanged
SRSYNrsp SRSYNcnf	a: sn as in SRSYNind r: sn as in SRSYNind s: sn ≤ (10**ULSN) − 1	abandon restart set	set to sn set to sn set to sn	set to sn set to sn set to sn	0 unchanged 0	unchanged unchanged unchanged
SACTRreq SACTRind			set to sn + 1	set to sn + 1	set to 1	unchanged
SACTSreq SACTSind			set to 1	set to 1	set to 1	unchanged
SCONrsp+ SCONcnf+		sn present	set to sn	set to sn	0	false

sn Synchronization point serial number quoted in session service primitive

ULSN Upper limit serial number negotiated during the session connection establishment phase

≥ 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 A.5 – Operations on variables when the symmetric synchronize functional unit has been selected

Events	Condition for valid primitive	Condition for update of variables	Operations on variables					
			V(Ms + 1)	V(Mr)	V(As)	V(Ar)	V(Rs)	V(Rr)
SSYNMreq SSYNmreq SACTEreq			V(Ms) + 1	unchanged	unchanged	unchanged	unchanged	unchanged
SSYNMind SACTEind	snr = V(Mr)		unchanged	V(Mr) + 1	unchanged	unchanged	unchanged	unchanged
SSYNmind	snr = V(Mr)		unchanged	V(Mr) + 1	unchanged	unchanged	unchanged	unchanged
SSYNMrsp SACTErsp	sns = V(Ms) snr = V(Mr) – 1		V(Ms) + 1	unchanged	set to V(Ms)	set to V(Mr)	set to V(Ms)	set to V(Mr)
SSYNMcnf SACTEcnf	sns = V(Ms) – 1 snr = V(Mr)		unchanged	V(Mr) + 1	set to V(Ms)	set to V(Mr)	set to V(Ms)	set to V(Mr)
SSYNmrsp SSYNmcnf	V(Mr) > snr ≥ V(Ar) V(Ms) > sns ≥ V(As)		unchanged unchanged	unchanged unchanged	unchanged set to sns + 1	set to snr + 1 unchanged	unchanged unchanged	unchanged unchanged
SRSYNreq *	r: V(Mr) ≥ snr ≥ V(Rr) V(Ms) ≥ sns ≥ V(Rs)		unchanged	unchanged	unchanged	unchanged	unchanged	unchanged
SRSYNind *	a: snr, sns r: snr ≥ V(Rr) sns ≥ V(Rs)		unchanged	unchanged	unchanged	unchanged	unchanged	unchanged
SRSYNrsp *	a: not applicable r: sn as in SRSYNind s: sn ≤ (10**ULSN) – 1	abandon restart set	set to sns set to sns set to sns	set to snr set to snr set to snr	set to sns set to sns set to sns	set to snr set to snr set to snr	0 unchanged 0	0 unchanged 0
SRSYNcnf *	a: snr ≥ V(Mr) sns ≥ V(Ms) r: sn as in SRSYNcnf s: sn ≤ (10**ULSN) – 1	abandon restart set	set to sns set to sns set to sns	set to snr set to snr set to snr	set to sns set to sns set to sns	set to snr set to snr set to snr	0 unchanged 0	0 unchanged 0
SACTRreq SACTRind			set sns + 1	set snr + 1	set sns + 1	set snr + 1	set to 1	set to 1
SACTSreq SACTSind			set to 1	set to 1	set to 1	set to 1	set to 1	set to 1
SCONrsp+ SCONcnf+		snr and sns present	set to sns	set to snr	set to sns	set to snr	0	0

sns	Synchronization point serial number quoted in session service primitive for the sending flow
snr	Synchronization point serial number quoted in session service primitive for the receiving flow
ULSN	Upper limit serial number negotiated during the session connection establishment phase
≥	Greater than or equal to
≤	Less than or equal to
*	If the Resync type is only specified for one direction of flow, then only those variables associated with that direction of flow are checked and updated

Table A.6 – Specific actions

[5]	Set V(A) = V(M) = serial number in S-CONNECT response or S-CONNECT confirm Set V(R) = 0 Set Vcoll = false Set Vrsp = no Set Vsc = false Set FU(f) for f in fu-dom according to session user requirements in S-CONNECT response or S-CONNECT confirm If FU(ACT) = true, set Vact = false Set Vdnr = false
[11]	Update the position of the tokens
[12]	Set Vact = true
[14]	Set Vact = false

[16]	If ¬FU(SS), then
	Update Vrsp
	If RS-r, update Vrspnb
	Set Rcv-flow-in-resync = true
	Snd-flow-in-resync = true
	If FU(SS), then
	Update Vrsps
	If Vrsps = r, update Vrspnbs
	If Vrsps ≠ no, set Snd-flow-in-resync = true
	If FU(SS), then
	Update Vrspr
	If Vrspr = r, update Vrspnbr
	If Vrspr ≠ no, set Rcv-flow-in-resync = true

[17]	Set Vrsp = no
[18]	Set Vcoll = true
[19]	Set V(M) = serial number
[22]	Set V(R) = V(A) = V(M)
[23]	If Vsc = false, then set V(A) = V(M) Set Vsc = true Set V(M) = V(M) + 1
[24]	If Vsc = true, then 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 Rcv-flow-in-resync = false Snd-flow-in-resync = false
[29]	Set the position of the tokens such that all available tokens are owned. Set Vact = false
[30]	Set the position of the tokens such that all available tokens are not owned. Set Vact = false
[31]	If Vsc = false, then set V(A) = V(M) Set V(M) = V(M) + 1
[32]	Set Vdnr = true

Table A.6 *(concluded)*

[62]	Set V(Mr) = V(Mr) + 1
[63]	Set V(Ms) = V(Ms) + 1
[64]	Set V(Rs) = V(As) = V(Ms) Set V(Rr) = V(Ar) = V(Mr)
[65]	Set V(As) = serial number + 1
[66]	Set V(Ar) = serial number + 1
[68]	Set V(As) = V(Ms) = sending flow serial number, if present If Vrsps = a, then set V(Rs) = 0 If Vrsps = s, then set V(Rs) = 0 Set Vrsps = no Set Snd-flow-in-resync = false Set V(Ar) = V(Mr) = receiving flow serial number, if present If Vrspr = a, then set V(Rr) = 0 If Vrspr = s, then set V(Rr) = 0 Set Vrspr = no Set Rcv-flow-in-resync = false
[69]	Set V(As) = V(Ms) = sending flow serial number + 1 Set V(Rs) = 1 Set V(Ar) = V(Mr) = receiving flow serial number + 1 Set V(Rr) = 1
[70]	Set V(As) = V(Ms) = V(Rs) = 1 Set V(Ar) = V(Mr) = V(Rr) = 1
[71]	Set V(As) = V(Ms) = sending flow serial number in SCONrsp orSCONcnf Set V(Rs) = 0 Set Vrsps = no Set Snd-flow-in-resync = false Set V(Ar) = V(Mr) = receiving flow serial number in SCONrsp or SCONcnf Set V(Rr) = 0 Set Vrspr = no Set Rcv-flow-in-resync = false Set Vcoll = false

Table A.7 – Predicates

p03	I(dk)
p04	FU(FD) & ¬Vcoll
p06	FU(TD)
p07	FU(TD) & ¬Vcoll
p08	FU(EX)
p09	FU(EX) & ¬Vcoll
p10	¬Vcoll
p11	II(ma)
p13	FU(MA) & [¬FU(ACT) OR Vact] & I(dk) & I(mi) & II(ma)
p15	[¬FU(ACT) OR Vact] & I(dk) & II(mi)
p16	No extended control QOS
p18	[¬FU(ACT) OR Vact] & [FU(SS) OR FU(SY)] & [FU(SS) OR Vsc]
p20	¬FU(SS) & [serial number = V(M) – 1]
p21	¬FU(SS) & [V(M) > serial number ≥ V(A)]
p24	SS-userwinner
p25	[FU(SY) OR FU(MA) OR FU(SS)] & FU(RESYNC)
p26	[¬FU(ACT) OR Vact]
p28	FU(RESYN)
p29	[¬FU(ACT) OR Vact] & FU(RESYN)
p32	[¬FU(SS) & [[type ≠ r] OR [serial number ≥ V(R)]]] OR [FU(SS) & [[rcv flow type ≠ r] OR [rcv flow serial number ≥ V(Rr)]] & [[snd flow type ≠ r] OR [snd flow serial number ≥ V(Rs)]]]
p33	V(M) ≥ serial number ≥ V(R)
p34	FU(ACT)
p39	Vact & II(ma)
p43	¬FU(SS)& [[[Vrsp = r] & [serial number = Vrspnb]] OR [[Vrsp = a] & serial number as in SRSYNind] OR [Vrsp = s]]
p45	[FU(ACT) & ¬Vact] & I(dk) & I(mi) & I(ma)
p47	FU(CD) & [FU(ACT) & ¬Vact] & I(dk) & I(mi) & OWNED(ma)
p50	FU(EXCEP) & [¬FU(ACT) OR Vact] & AA(dk)
p51	FU(EXCEP) & [¬FU(ACT) OR Vact] & II(dk)
p53	ALL(AV, RT) & RT not empty
p54	ALL(II, GT)
p55	[FU(ACT) & ¬Vact] & ALL(I, tk-dom)
p57	ALL(II, GT) & (dk not in GT)
p58	ALL(II, GT) & (dk in GT)
p60	ALL(AA, GT) & (dk not in GT)
p61	ALL(AA, GT) & (dk in GT)
p63	ALL(I, tk-dom) & [¬FU(ACT) OR ¬Vact]
p65	ANY(AV, tk-dom)
p67	FU(NR)
p69	Vcoll

Table A.7 *(concluded)*

p71	FU(ACT) & Vact & I(dk) & I(mi) & II(ma)
p75	[Vcoll & Vdnr] OR ¬Vcoll
p81	[Vrsp = r] & [[Vrspnb > serial number] OR [[Vrspnb = serial number] & p95]]
p82	[Vrsp = r] OR [p95 & p99]
p83	[Vrsp = s] OR p82
p95	SS-user is initiator of the session connection
p99	Resynchronize Type parameter in SRSYNreq is equal to Vrsp
p175	FU(SS) & [serial number = V(Mr) − 1]
p178	FU(SS)
p179	FU(SS) & [V(Mr) > serial number ≥ V(Ar)]
p180	[¬FU(SS) & [type ≠ r]] OR [FU(SS) & [[rcv flow type = a] OR [rcv flow type = s]] & [[snd flow type = a] OR [snd flow type = s]]]
p185	Rcv-flow-in-resync
p186	Snd-flow-in-resync
p187	FU(SS) & [[Vrspr ≠ r] OR [rcv flow serial number = Vrspnbr]] & [[Vrsps ≠ r] OR [snd flow serial number = Vrspnbs]]

Table A.8 – Connection establishment state table without the symmetric synchronize functional unit

State Event	STA01 idle disconn	STA02A await SCONcnf	STA08 await SCONrsp
SCONcnf+		[5][11] STA713	
SCONcnf−		STA01	
SCONind	STA08		
SCONreq	STA02A		
SCONrsp+			[5][11] STA713
SCONrsp−			STA01

Table A.9 – Data transfer state table without the symmetric synchronize functional unit

State / Event	STA03 await SRELcnf	STA04A await SSYNMcnf	STA04B await SACTEcnf	STA05A await SRSYNcnf	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp	STA713 data transfer
SDTind	STA03	STA04A	STA04B	STA05A				STA713
SDTreq					p04 STA09	p03 STA10A	p03 STA10B	p03 STA713
SEXind	STA03	STA04A	STA04B	STA05A				STA713
SEXreq					p09 STA09	p08 STA10A	p08 STA10B	p08 STA713
STDind	STA03	STA04A	STA04B	STA05A				STA713
STDreq					p07 STA09	p06 STA10A	p06 STA10B	p06 STA713

Table A.10 – Synchronization state table without the symmetric synchronize functional unit

State / Event	STA03 await SRELcnf	STA04A await SSYNMcnf	STA04B await SACTEcnf	STA05A await SRSYNcnf	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp	STA713 data transfer
SACTEcnf			[14][22] STA713					
SACTEind								[31] STA10B
SACTEreq								p71 [24] STA04B
SACTErsp							[14][22] STA713	
SSYNMcnf		[22] STA713						
SSYNMind				[31] STA05A				[31] STA10A
SSYNMreq								p13 [24] STA04A
SSYNMrsp						[22] STA713		
SSYNmcnf	[25] STA03	[25] STA04A	[25] STA04B	[25] STA05A				[25] STA713
SSYNmdind				[23] STA05A				[23] STA713
SSYNmdreq								p15 [24] STA713
SSYNmind				[23] STA05A				[23] STA713
SSYNmreq								p15 [24] STA713
SSYNmrsp					p18&p21 [25] STA09	p18&¬p20&p21 [25] STA10A	p18&¬p20&p21 [25] STA10B	p18&p21 [25] STA713

Table A.11 – Resynchronization state table without the symmetric synchronize functional unit

State / Event	STA03 await SRELcnf	STA04A await SSYNMcnf	STA04B await SACTEcnf	STA05A await SRSYNcnf	STA09 await SRELrsp
SRSYNcnf				[28] STA713	
SRSYNind(a)	[16][19] STA11A	[16][19] STA11A	[16][19] STA11A	[16][19] STA11A	
SRSYNind(r)	[16] STA11A	[16] STA11A	[16] STA11A	[16] STA11A	
SRSYNind(s)	[16] STA11A	[16] STA11A	[16] STA11A	[16] STA11A	
SRSYNreq(a)		p28 [16] STA05A			p10&p28&¬p34 [16] STA05A
SRSYNreq(r)					p10&p25&¬p34&p33 [16] STA05A
SRSYNreq(s)		p28 [16] STA05A			p10&p25&¬p34 [16] STA05A
SRSYNrsp					

Table A.11 *(concluded)*

State / Event	STA10A await SSYNMrsp	STA10B await SACTErsp	STA11A await SRSYNrsp	STA19 await rec ind	STA20 await rec req	STA713 data transfer
SRSYNcnf						
SRSYNind(a)	[16][19] STA11A			[16][19] STA11A	[16][19] STA11A	[16][19] STA11A
SRSYNind(r)				[16] STA11A	[16] STA11A	[16] STA11A
SRSYNind(s)	[16] STA11A			[16] STA11A	[16] STA11A	[16] STA11A
SRSYNreq(a)	p28 [16] STA05A	p28 [16] STA05A	p83 [16] STA05A		p28 [16] STA05A	p29 [16] STA05A
SRSYNreq(r)	p25&p33 [16] STA05A	p25&p33 [16] STA05A	p181&p33 [16] STA05A		p25&p33 [16] STA05A	p25&p26&p33 [16] STA05A
SRSYNreq(s)	p25 [16] STA05A	p25 [16] STA05A	p82 [16] STA05A		p25 [16] STA05A	p25&p26 [16] STA05A
SRSYNrsp			p43 [28] STA713			

Table A.12 – Activity interrupt and discard state table without the symmetric synchronize functional unit

State / Event	STA04A await SSYNMcnf	STA04B await SACTEcnf	STA05A await SRSYNcnf	STA05B await SACTIcnf	STA05C await SACTDcnf	STA10A await SSYNMrsp	STA10B await SACTErsp
SACTDcnf					[29] STA713		
SACTDind			STA11C			STA11C	STA11C
SACTDreq	p34&p39 STA05C	p39 STA05C					
SACTDrsp							
SACTIcnf				[29] STA713			
SACTIind			STA11B			STA11B	STA11B
SACTIreq	p34&p39 STA05B	p39 STA05B					
SACTIrsp							

Table A.12 ***(concluded)***

State / Event	STA11A await SRSYNrsp	STA11B await SACTIrsp	STA11C await SACTDrsp	STA19 await rec ind	STA20 await rec req	STA21 await SCDcnf	STA22 await SCDrsp	STA713 data transfer
SACTDcnf								
SACTDind				STA11C	STA11C			STA11C
SACTDreq	p34&p39 STA05C				p34&p11 STA05C			p34&p39 STA05C
SACTDrsp			[30] STA713					
SACTIcnf								
SACTIind				STA11B	STA11B		STA11B	STA11B
SACTIreq	p34&p39 STA05B				p34&p11 STA05B	p16 STA05B		p34&p39 STA05B
SACTIrsp		[30] STA713						

Table A.13 – Activity start, resume and capability data state table without the symmetric synchronize functional unit

State / Event	STA21 await SCDcnf	STA22 await SCDrsp	STA713 data transfer
SACTRind			[12][27] STA713
SACTRreq			p45 [12][27] STA713
SACTSind			[12][26] STA713
SACTSreq			p45 [12][26] STA713
SCDcnf	STA713		
SCDind			STA22
SCDreq			p47 STA21
SCDrsp		STA713	

Table A.14 – Token management and exceptions state table without the symmetric synchronize functional unit

State / Event	STA03 await SRELcnf	STA04A await SSYNMcnf	STA04B await SACTEcnf	STA05A await SRSYNcnf	STA09 await SRELrsp	STA10A await SSYNMrsp
SCGind						
SCGreq						
SGTind		[11] STA04A	[11] STA04B	STA05A		[11] STA10A
SGTreq		p54 [11] STA04A	p54 [11] STA04B			p54 [11] STA10A
SPERind	STA20	p03 STA20 ¬p03 STA713	p03 STA20 ¬p03 STA713			
SPTind	STA03	STA04A	STA04B	STA05A		
SPTreq					p53 STA09	p53 STA10A
SUERind	STA20	p03 STA20 ¬p03 STA713	p03 STA20 ¬p03 STA713			
SUERreq					p50 STA19	p50 STA19

Table A.14 *(concluded)*

State / Event	STA10B await SACTErsp	STA19 await rec ind	STA20 await rec req	STA21 await SCDcnf	STA22 await SCDrsp	STA713 data transfer
SCGind						[11] STA713
SCGreq						p55 [11] STA713
SGTind	[11] STA10B	p60 [11] STA19 p61 [11] STA713	p60 [11] STA20 p61 [11] STA713	[11] STA21		[11] STA713
SGTreq	p54 [11] STA10B		p57 [11] STA20 p58 [11] STA713			p54 [11] STA713
SPERind		STA19		STA20		p50 STA713 p51 STA20
SPTind				STA21		STA713
SPTreq	p53 STA10B				p53 STA22	p53 STA713
SUERind		STA19				p50 STA713 p51 STA20
SUERreq	p50 STA19					p50 STA19

Table A.15 – Connection release state table without the symmetric synchronize functional unit

State / Event	STA03 await SRELcnf	STA09 await SRELrsp	STA713 data transfer	Any other state
SPABind	STA01	STA01	STA01	STA01
SRELcnf+	STA01	[32] STA09		
SRELcnf−	STA713			
SRELind	[18] STA09		STA09	
SRELreq		¬p65 [18] STA09	p63 STA03	
SRELrsp+		p75 STA01 p69&p95 STA03		
SRELrsp−		p67 STA713		
SUABind	STA01	STA01	STA01	STA01
SUABreq	STA01	STA01	STA01	STA01

Table A.16 – Connection establishment state table with the symmetric synchronize functional unit

State / Event	STA01 idle disconn	STA02A await SCONcnf	STA08 await SCONrsp
SCONcnf+		¬p178 [5][11] STA713 p178 [71][11] STA713	
SCONcnf−		STA01	
SCONind	STA08		
SCONreq	STA02A		
SCONrsp+			¬p178 [5][11] STA713 p178 [71][11] STA713
SCONrsp−			STA01

Table A.17 – Data transfer state table with the symmetric synchronize functional unit

Event / State	STA03 await SRELcnf	STA04A await SSYNMcnf	STA04B await SACTEcnf	STA05A await SRSYNcnf	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp	STA11A await SRSYNrsp	STA713 data transfer
SDTind	STA03	STA04A	STA04B	STA05A				STA11A	STA713
SDTreq				p03&¬p186 STA05A	p04 STA09	p03 STA10A	p03 STA10B	p03&¬p186 STA11A	p03 STA713
SEXind	STA03	STA04A	STA04B	STA05A				STA11A	STA713
SEXreq				p08&¬p186 STA05A	p09 STA09	p08 STA10A	p08 STA10B	p08&¬p186 STA11A	p08 STA713
STDind	STA03	STA04A	STA04B	STA05A				STA11A	STA713
STDreq				p06&¬p186 STA05A	p07 STA09	p06 STA10A	p06 STA10B	p06&¬p186 STA11A	p06 STA713

Table A.18 – Synchronization state table with the symmetric synchronize functional unit

State Event	STA03 await SRELcnf	STA04A await SSYNMcnf	STA04B await SACTEcnf	STA05A await SRSYNcnf	STA09 await SRELrsp
SSYNMcnf		¬p178 [22] STA713 p178 [62][64] STA713			
SSYNMind				¬p178 [31] STA05A p178 [62] STA05A	
SSYNMreq					
SSYNMrsp					
SSYNmcnf	¬p178 [25] STA03 p178 [65] STA03	¬p178 [25] STA04A p178 [65] STA04A	¬p178 [25] STA04B p178 [65] STA04B	¬p178 [25] STA05A p178 [65] STA05A	
SSYNmdind				¬p178[23] STA05A p178 [62] STA05A	
SSYNmdreq					
SSYNmind	[62] STA03	[62] STA04A	[62] STA04B	[62] STA05A	
SSYNmreq				¬p186 [63] STA05A	p26&p178 [63] STA09
SSYNmrsp				¬p185&p179 [66] STA05A	p18&p21 [25] STA09 p18&p179 [66] STA09

Table A.18 *(continued)*

State Event	STA10A await SSYNMrsp	STA10B await SACTErsp	STA11A await SRSYNrsp	STA713 data transfer
SSYNMcnf				
SSYNMind				¬p178 [31] STA10A p178 [62] STA10A
SSYNMreq				p13&¬p178 [24] STA04A p13&p178 [63] STA04A
SSYNMrsp	¬p178 [22] STA713 p178 [63][64] STA713			
SSYNmcnf			[65] STA11A	¬p178 [25] STA713 p178 [65] STA713
SSYNmdind				[23] STA713
SSYNmdreq				p15 [24] STA713
SSYNmind			[62] STA11A	¬p178 [23] STA713 p178 [62] STA713
SSYNmreq	p26&p178 [63] STA10A	p26&p178 [63] STA10B	¬p186 [63] STA11A	p15 [24] STA713 p178&p26 [63] STA713
SSYNmrsp	p18&¬p20&p21 [25] STA10A p18&¬p175&p179 [66] STA10A	p18&¬p20&p21 [25] STA10B p18&¬p175&p179 [66] STA10B	¬p186&p179 [66] STA11A	p18&p21 [25] STA713 p26&p179 [66] STA713

Table A.18 *(concluded)*

State Event	STA04B await SACTEcnf	STA10B await SACTErsp	STA713 data transfer
SACTEcnf	¬p178 [14][22] STA713 p178 [14][62][64] STA713		
SACTEind			¬p178 [31] STA10B p178 [62] STA10B
SACTEreq			p71&¬p178 [24] STA04B p71&p178 [63] STA04B
SACTErsp		¬p178 [14][22] STA713 p178 [62][64] STA713	

Table A.19 – Resynchronization state table with the symmetric synchronize functional unit

State Event	STA03 await SRELcnf	STA04A await SSYNMcnf	STA04A await SSYNMcnf	STA05A await SRSYNcnf	STA09 await SRELrsp
SRSYNcnf				¬p178 [28] STA713 p178 [68] STA713	
SRSYNind	[16] STA11A	[16] STA11A	[16] STA11A	[16] STA11A	
SRSYNreq		p28&p180 [16] STA05A			p28&¬p34&p32 [16] STA05A
SRSYNrsp					

Table A.19 *(concluded)*

State / Event	STA10A await SSYNMrsp	STA10B await SACTErsp	STA11A await SRSYNrsp	STA19 await rec ind	STA20 await rec req	STA713 data transfer
SRSYNcnf						
SRSYNind	[16] STA11A			[16] STA11A	[16] STA11A	[16] STA11A
SRSYNreq	p28&p32 [16] STA05A	p28&p32 [16] STA05A	p24&p32 [16] STA05A		p28&p32 [16] STA05A	p29&p32 [16] STA05A
SRSYNrsp			p43 [28] STA713 p187 [68] STA713			

Table A.20 – Activity interrupt and discard state table with the symmetric synchronize functional unit

State / Event	STA04A await SSYNMcnf	STA04B await SACTEcnf	STA05A await SRSYNcnf	STA05B await SACTIcnf	STA05C await SACTDcnf	STA10A await SSYNMrsp	STA10B await SACTErsp
SACTDcnf					[29] STA713		
SACTDind			STA11C			STA11C	STA11C
SACTDreq	p34&p39 STA05C	p39 STA05C					
SACTDrsp							
SACTIcnf				[29] STA713			
SACTIind			STA11B			STA11B	STA11B
SACTIreq	p34&p39 STA05B	p39 STA05B					
SACTIrsp							

Table A.20 ***(concluded)***

State / Event	STA11A await SRSYNrsp	STA11B await SACTIrsp	STA11C await SACTDrsp	STA19 await rec ind	STA20 await rec req	STA21 await SCDcnf	STA22 await SCDrsp	STA713 data transfer
SACTDcnf								
SACTDind				STA11C	STA11C			STA11C
SACTDreq	p34&p39 STA05C				p34&p11 STA05C			p34&p39 STA05C
SACTDrsp			[30] STA713					
SACTIcnf								
SACTIind				STA11B	STA11B		STA11B	STA11B
SACTIreq	p34&p39 STA05B				p34&p11 STA05B	p16 STA05B		p34&p39 STA05B
SACTIrsp		[30] STA713						

Table A.21 – Activity start, resume and capability data state table with the symmetric synchronize functional unit

State Event	STA21 await SCDcnf	STA22 await SCDrsp	STA713 data transfer
SACTRind			¬p178 [12][27] STA713 p178 [12][69] STA713
SACTRreq			p45&¬p178 [12][27] STA713 p45&p178 [12][69] STA713
SACTSind			¬p178 [12][26] STA713 p178 [12][70] STA713
SACTSreq			p45&¬p178 [12][26] STA713 p45&p178 [12][70] STA713
SCDcnf	STA713		
SCDind			STA22
SCDreq			p47 STA21
SCDrsp		STA713	

Table A.22 – Token management and exceptions state table with the symmetric synchronize functional unit

State / Event	STA03 await SRELcnf	STA04A await SSYNMcnf	STA04B await SACTEcnf	STA05A await SRSYNcnf	STA09 await SRELrsp	STA10A await SSYNMrsp
SCGind						
SCGreq						
SGTind		[11] STA04A	[11] STA04B	STA05A		[11] STA10A
SGTreq		p54 [11] STA04A	p54 [11] STA04B			p54 [11] STA10A
SPERind	STA20	p03 STA20 ¬p03 STA713	p03 STA20 ¬p03 STA713			
SPTind	STA03	STA04A	STA04B	STA05A		
SPTreq					p53 STA09	p53 STA10A
SUERind	STA20	p03 STA20 ¬p03 STA713	p03 STA20 ¬p03 STA713			
SUERreq					p50 STA19	p50 STA19

Table A.22 ***(concluded)***

State / Event	STA10B await SACTErsp	STA19 await rec ind	STA20 await rec req	STA21 await SCDcnf	STA22 await SCDrsp	STA713 data transfer
SCGind						[11] STA713
SCGreq						p55 [11] STA713
SGTind	[11] STA10B	p60 [11] STA19 p61 [11] STA713	p60 [11] STA20 p61 [11] STA713	[11] STA21		[11] STA713
SGTreq	p54 [11] STA10B		p57 [11] STA20 p58 [11] STA713			p54 [11] STA713
SPERind		STA19		STA20		p50 STA713 p51 STA20
SPTind				STA21		STA713
SPTreq	p53 STA10B				p53 STA22	p53 STA713
SUERind		STA19				p50 STA713 p51 STA20
SUERreq	p50 STA19					p50 STA19

Table A.23 – Connection release state table with the symmetric synchronize functional unit

State Event	STA03 await SRELcnf	STA09 await SRELrsp	STA713 data transfer	Any other state
SPABind	STA01	STA01	STA01	STA01
SRELcnf+	STA01	[32] STA09		
SRELcnf−	STA713			
SRELind	[18] STA09		STA09	
SRELreq		¬p65 [18] STA09	p63 STA03	
SRELrsp+		p75 STA01 p69&p95 STA03		
SRELrsp−		p67 STA713		
SUABind	STA01	STA01	STA01	STA01
SUABreq	STA01	STA01	STA01	STA01