INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

X.215

(11/1988)

SERIES X: DATA COMMUNICATION NETWORKS:
OPEN SYSTEMS INTERCONNECTION (OSI) – MODEL
AND NOTATION, SERVICE DEFINITION

SESSION SERVICE DEFINITION FOR OPEN SYSTEMS INTERCONNECTION FOR CCITT APPLICATIONS

Reedition of CCITT Recommendation X.215 published in the Blue Book, Fascicle VIII.4 (1988)

NOTES

1 CCITT Recommendation X.215 was published in Fascicle VIII.4 of the *Blue Book*. This file is an extract from the *Blue Book*. While the presentation and layout of the text might be slightly different from the *Blue Book* version, the contents of the file are identical to the *Blue Book* version and copyright conditions remain unchanged (see below).

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

Recommendation X.215

SESSION SERVICE DEFINITION FOR OPEN SYSTEMS INTERCONNECTION FOR CCITT APPLICATIONS[1]

(Malaga–Torremolinos, 1984, amended at Melbourne, 1988)

The CCITT,

considering

(a) that Recommendation X.200 defines the Reference Model of Open Systems Interconnection for CCITT applications;

(b) that Recommendation X.225 specifies the Session Protocol Specification for Open Systems Interconnection for CCITT applications;

(c) that Recommendation T.62 defines the Control Procedures for Teletex and Group 4 Facsimile Services;

unanimously declares

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

CONTENTS

[1] Recommendation X.215 is technically aligned with ISO 8326 [Information Processing Systems – Open Systems Interconnection – Basic Connection oriented session service definition] which includes corrections resulting from ISO defect reports numbered 8326/4, 8326/6, 8326/9, 8326/11 through 8326/17, 8326/20, 8327/5 and 8327/35 and the addendum 2 to incorporate unlimited user data.

0 Introduction

This Recommendation is one of a set of Recommendations produced to facilitate the interconnection of computer systems. It is related to other Recommendations 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 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 also defines the session service characteristics which the presentation protocol may exploit. The relationship between the Recommendations for the session service, session protocol, transport service, and the presentation protocol is illustrated in Figure 1/X.215.

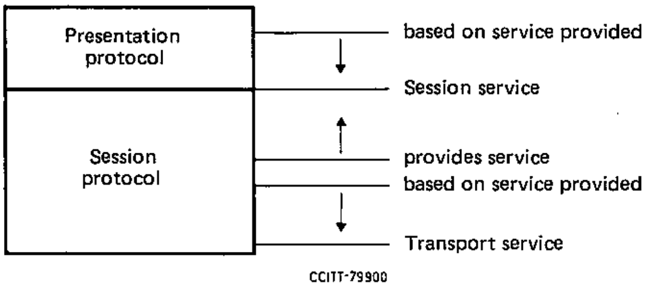


FIGURE 1/X.215

Relationship of this Recommendation to other OSI Recommendations

It is recognized that, with respect to session Quality of Service, (described in § 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 at a later time which reflects further QOS developments and integration.

1 Scope and field of application

This Recommendation 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 is that which is provided by the OSI session protocol (in conjunction with the transport service) and which may be used by any OSI presentation protocol.

This Recommendation 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.

2 References

Recommendation X.200	Reference Model of Open Systems Interconnection for CCITT Applications. (See also ISO 1498–1).
Recommendation X.210	OSI Layer Service Definition Conventions. (See also ISO TR8509).
Recommendation X.214	Transport Service Definition for Open Systems Interconnection for CCITT Applications. (See also ISO 8072).
Recommendation X.225	Basic Connection Oriented Session Protocol Specification for Open Systems Interconnection for CCITT Applications. (See also ISO 8327 and ISO 8327 Addendum 2).
ISO 7498–3	Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 3: Naming and Addressing[2]

3 Definitions

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

3.1 *Reference Model definitions*

This Recommendation is based on the concepts developed in the Basic Reference Model for Open Systems Interconnection (Recommendation X.200), and makes use of the following terms defined in that Recommendation:

a) expedited session service data unit;

b) session connection;

c) Session Layer;

d) session 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 also makes use of the following terms defined in the OSI Service Conventions (Recommendation X.210), as they apply to the Session Layer:

a) service-user;

b) service-provider;

c) primitive;

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

d) request;

e) indication;

f) response;

g) confirm.

3.3 *Session service definitions*

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

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 a source of data during the data transfer phase of a session connection.

3.3.4 **receiving SS-user**

An SS-user that acts as a sink of data during the data phase of a session connection.

Note – An SS-user can be both a 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; 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.

4 Symbols and abbreviations

4.1. *Abbreviations*

SS:	session service
SSAP:	session service access point
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

QOS:	quality of service

4.2 *Service variables*

V(A):	See § 11.4.1.1.
V(M):	See § 11.4.1.2.
V(R):	See § 11.4.1.3.
Vsc:	See § 11.4.1.4.

5 Conventions

This Recommendation uses the descriptive conventions contained in the OSI Service Conventions (Recommendation X.210) except that, where indicated in this Recommendation, 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

This Recommendation uses the abstract model for a layer service defined in the OSI Service Conventions (Recommendation X.210). The model defines the interactions between the SS-user and the SS-provider which take place at the two SSAPs. Information is passed between an SS-user and the SS-provider by service primitives, which may convey parameters.

7 Overview of the session service

7.1 *General overview*

The session service provides the means for organized and synchronized exchange of data between cooperating 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 prearranged point.

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. Each synchronization point is identified by a serial number 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 2/X.215 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.

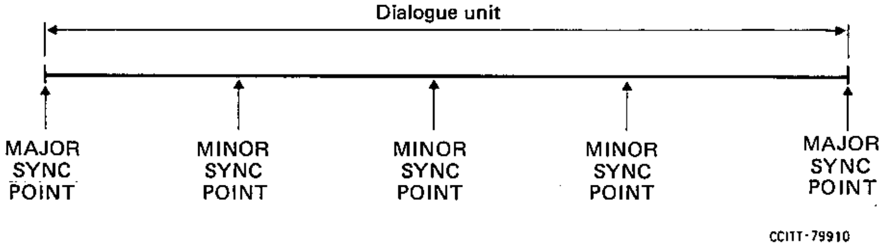


FIGURE 2/X.215

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 3/X.215 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.

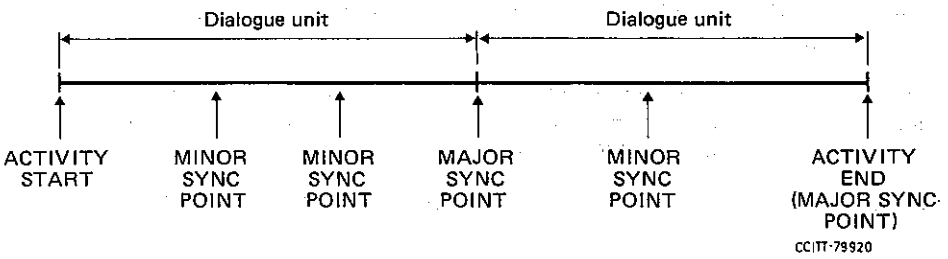


FIGURE 3/X.215

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 to a new value. Resynchronization purges all undelivered data.

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* which is used to set the synchronization point serial number to any value chosen by SS-user.

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

In all other combinations of functional units, no initial synchronization point serial number is proposed.

8 Phases and services of the session service

The session service comprises three phases. The purpose of each phase, and a short description of the associated services is given in this section. The services and the primitives by which they are invoked are defined in §§ 12, 13 and 14.

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 three 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 *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;

j) 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:

k) 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;

l) 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:

m) 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;

n) 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;

o) 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;

p) 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;

q) 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 for the purpose of:

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

b) reference by other Recommendations.

Table 1/X.215 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/X.215.

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 *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.

9.1.10 *Resynchronize functional unit*

The resynchronize functional unit supports the resynchronize service.

9.1.11 *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.

9.1.12 *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.

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

TABLE 1/X.215

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/X.215

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 exchange	Capability data exchange	13.4
Minor synchronize	Minor synchronization point Give tokens Please tokens	13.8 13.5 13.6
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

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:

   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 renegotiated 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 § 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/X.215;

b) parameters which express other session service characteristics, as shown in Table 4/X.215.

These session service QOS parameters are defined in this subsection.

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.

TABLE 3/X.215

Classification of Performance QOS Parameters

Phase	Performance criterion	
	Speed	Accuracy7Reliability
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, duplication/loss) Session connection resilience Transfer falure probability
Session connection release	Session connection release delay	Session connection release failure probability

TABLE 4/X.215

Parameters Specifying Other Session Service Features

Extended control
Session connection protection
Session connection priority
Optimized dialogue transfer

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* is greater than or equal to two, 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.

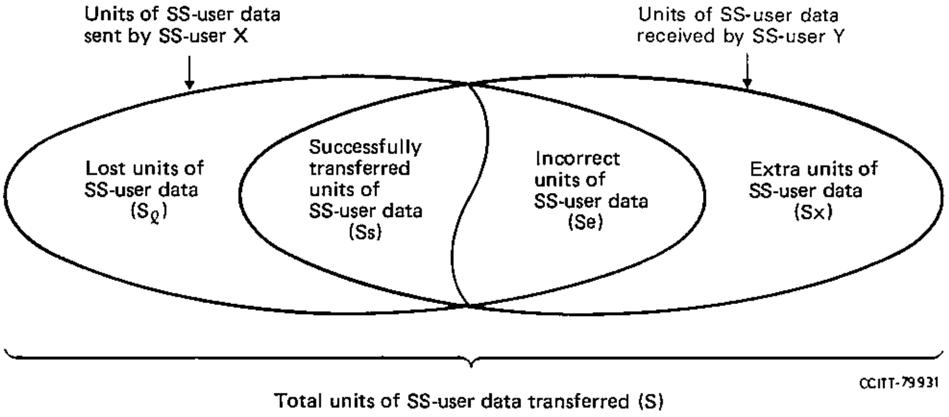
10.3.5 *Residual error rate*

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 4/X.215.

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.


Units of SS-user data sent by SS-user X
Units of SS-user data received by SS-user Y
Lost units of SS-user data (Sℓ)
Successfully transferred units of SS-user data (Ss)
Incorrect units of SS-user data (Se)
Extra units of SS-user data (Sx)
Total units of SS-user data transferred (S)
CCITT-79931

$$\mathrm{RER} = \frac{Se + S\ell + Sx}{S}$$

FIGURE 4/X.215

Components of residual error rate

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.

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/X.215, 6/X.215 and 7/X.215 summarize the primitives and their parameters occurring in each phase of the session service. The parameters are defined in §§ 12, 13 and 14.

TABLE 5/X.215

Session Connection Establishment Phase Primitives

Service	Primitives	Parameters
Session connection	S-CONNECT request	Session connection identifier
	S-CONNECT indication	Calling/Called/Responding session addresses
	S-CONNECT response	Result, QOS, Session requirements
	S-CONNECT confirm	Synchronization point serial number, Initial assignment of tokens, SS-user data.

TABLE 6/X.215

Data Transfer Phase Primitives

Service	Primitives	Parameters
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
Given 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, 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

TABLE 6/X.215 *(cont.)*

Service	Primitives	Parameters
Resynchronize	S-RESYNCHRONIZE request S-RESYNCHRONIZE indication S-RESYNCHRONIZE response S-RESYNCHRONIZE confirm	Resynchronize type, Synchronization point serial number Assignment of tokens, SS-user dat
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/X.215

Session Connection Release Phase Primitives

Service	Primitives	Parameters
Orderly release	S-RELEASE request S-RELEASE indication S-RELEASEresponse S-RELEASEconfirm	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/X.215 defines the conditions under which those service primitives requiring tokens may be issued.

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 a synchronization point serial number, which is used to identify a synchronization point. Synchronization points are assigned valid synchronization point serial numbers in the range 0 to 999998 by the SS-provider. It is the responsibility of the SS-user to ensure that the number assigned by the SS-provider in a synchronization point request does not exceed 999998.

The synchronization point serial number 999999 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 in terms of:

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

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

These operations are summarized in Table A-4/X.215 in Annex A.

TABLE 8/X.215

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 (synchronize-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 (synchronize-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 S-SYNC-MAJOR request	2 2	1 2	nr 1	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.

11.4.1 *Variables*

11.4.1.1 *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).

11.4.1.2 *V(M)*

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

11.4.1.3 *V(R)*

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

11.4.1.4 *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.

11.4.2 *Session connection establishment*

When a session connection is established in which 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 to false.

11.4.3 *Minor synchronization point*

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 is 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.

11.4.4 *Major synchronization point*

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 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.

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.

11.4.5 *Resynchronization*

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 § 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.

11.4.6 *Activity management*

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.

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/X.215 specifies the types of session service primitives and parameters needed for session connection establishment.

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 ISO 7498-3).

12.1.2.3 *Calling Session Address* is the session address of the called entity (see ISO 7498-3).

12.1.2.4 *Responding Session Address* is the session address of the responding entity (see ISO 7498-3).

TABLE 9/X.215

Session Connection Establishment Primitives and Parameters

Primitive	S-CONNECT			
Parameter	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(=)
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.

12.1.2.5 *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) rejected 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;

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.

5) Implementation restriction state in the PICS.

12.1.2.6 *Quality of service* is a list of parameters which are defined and negotiated as described in § 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) major synchronize functional unit;

h) resynchronize functional unit;

i) expedited data functional unit;

j) activity management functional unit;

k) 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 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 *Initial Synchronization Point Serial Number* identifies the initial synchronization point. The conditions for its presence and rules for its negotiation are defined in § 7.6.3. Its value is in the range 0 to 999999.

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 shown in Figure 5/X.215.

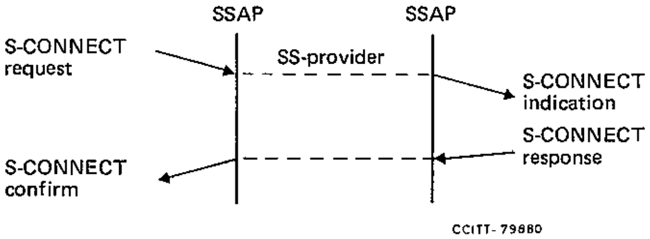


FIGURE 5/X.215

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/X.215.

13.1.2 *Types of primitives and their parameters*

Table 10/X.215 specifies the types of session service primitives and parameters needed for normal data transfer.

TABLE 10/X.215

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.

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.

13.1.3 *Sequence of primitives*

The sequence of primitives in a successful normal data transfer is defined by the time sequence diagram shown in Figure 6/X.215.

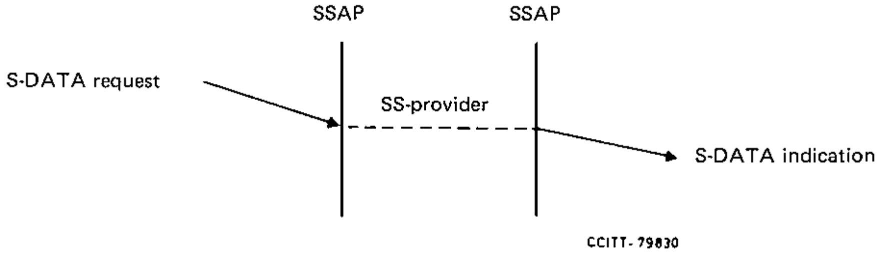


FIGURE 6/X.215

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/X.215 specifies the types of session service primitives and parameters needed for expedited data transfer.

TABLE 11/X.215

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.

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

13.2.3 *Sequence of primitives*

The sequence of primitives in a successful expedited data transfer is defined by the time sequence diagram shown in Figure 7/X.215.

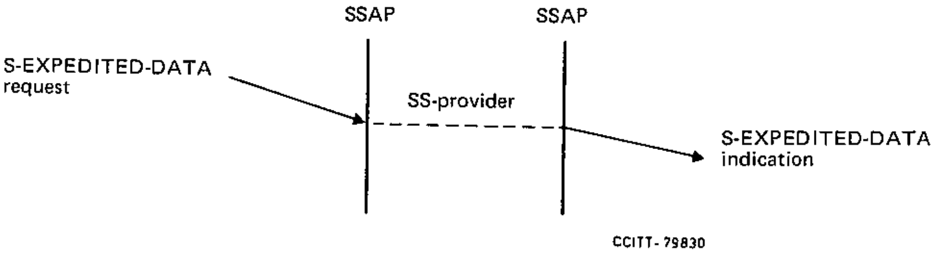


FIGURE 7/X.215

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/X.215 specifies the types of session service primitives and parameters needed for the typed data transfer service.

TABLE 12/X.215

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.

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.

13.3.3 *Sequence of primitives*

The sequence of primitives in a successful typed data transfer is defined by the time sequence diagram shown in Figure 8/X.215.

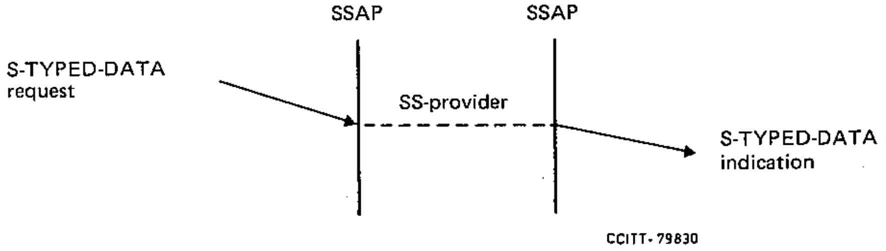


FIGURE 8/X.215

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/X.215.

13.4.2 *Types of primitives and their parameters*

Table 13/X.215 specifies the types of session service primitives and parameters needed for the capability data exchange service.

TABLE 13/X.215

Capability data exchange primitives and parameters

Primitive / Parameter	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.

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

13.4.3 *Sequence of primitives*

The sequence of primitives in a successful capability data exchange is defined by the time sequence diagram shown in Figure 9/X.215.

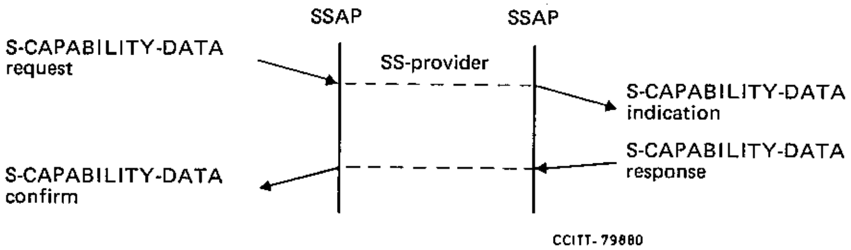

SSAP
SSAP
S-CAPABILITY-DATA request
SS-provider
S-CAPABILITY-DATA indication
S-CAPABILITY-DATA response
S-CAPABILITY-DATA confirm
CCITT-79880

FIGURE 9/X.215

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/X.215.

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/X.215 specifies the types of session service primitives and parameters needed for the give tokens service.

TABLE 14/X.215

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.
U: presence of the parameter is conditional.
C: 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 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 shown in Figure 10/X.215.

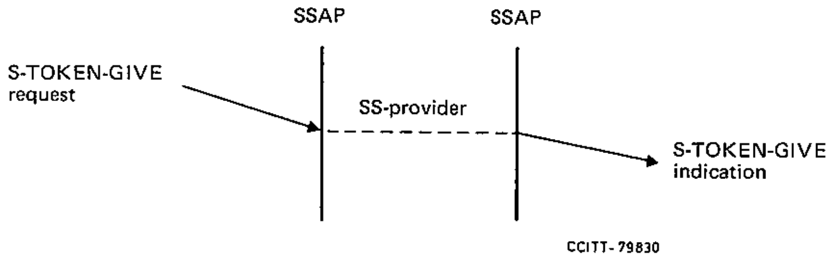


FIGURE 10/X.215

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/X.215.

13.6.2 *Types of primitives and their parameters*

Table 15/X.215 specifies the types of session service primitives and parameters needed for the please tokens service.

TABLE 15/X.215

Please tokens primitives and parameters

Primitive / Parameter	S-Token-Please	
	Request	Indication
Tokens	M	M(=)
SS-user data	U	C(=)

M: presence of the parameter is mandatory.
U: presence of the parameter is conditional.
C: 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 not assigned to but 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 shown in Figure 11/X.215.

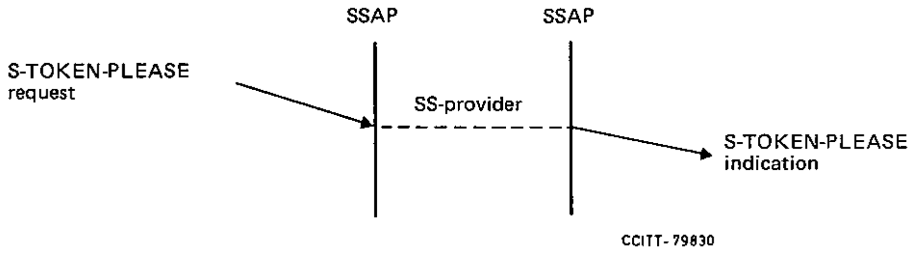


FIGURE 11/X.215

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/X.215 specifies the types of session service primitives and parameter needed for the give control service.

TABLE 16/X.215

Give Control Primitives and Parameters

Parameter \ Primitive	S-Control-Give	
	Request	Indication
SS-user data	U	C(=)

C: presence of the parameter is a user option.
U: presence of the parameter is conditional.
(=): the value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive.

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

13.7.3 *Sequence of primitives*

The sequence of primitives in a successful transfer of tokens is defined by the time sequence diagram shown in Figure 12/X.215.

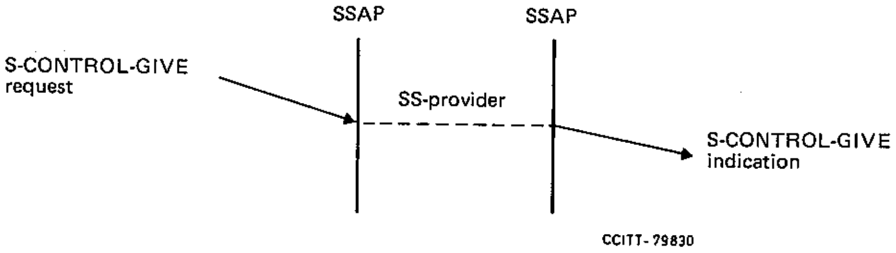


FIGURE 12/X.215

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/X.215.

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.

Note – When the duplex functional unit is selected, additional arrangements between SS-users may be required to correlate minor synchronization point requests and confirms with the flow of data from the SS-user without the synchronize-minor token.

13.8.2 *Types of primitives and their parameters*

Table 17/X.215 specifies the types of session service primitives and parameters needed for the minor synchronization point service.

TABLE 17/X.215

Minor synchronization point primitives and parameters

Primitive / Parameter	S-Sync-Minor			
	Request	Indication	Response	Confirm
Type	M	M(=)		
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;

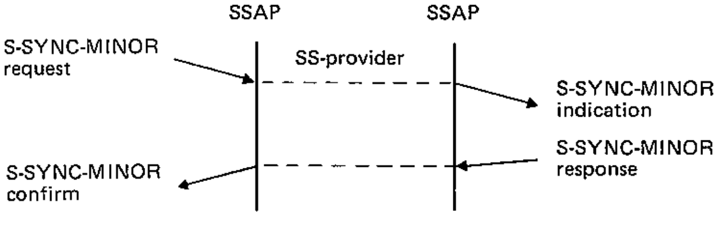
b) optional.

13.8.2.2 *Synchronization Point Serial Number* is defined in § 11.4.3. It is in the range 0 to 999998.

13.8.2.3 *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 shown in Figure 13/X.215.



CCITT-79880

FIGURE 13/X.215

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 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 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/X.215.

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-SYNC-MINOR 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.

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

13.9.2 *Types of primitives and their parameters*

Table 18/X.215 specifies the types of session service primitives and parameters needed for the major synchronization point service.

TABLE 18/X.215

Major synchronization point primitives and parameters

Primitive / Parameter	S-Sync-Major			
	Request	Indication	Response	Confirm
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.

13.9.2.1 *Synchronization Point Serial Number* is defined in § 11.4.4. It is in the range 0 to 999998.

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 shown in Figure 14/X.215.

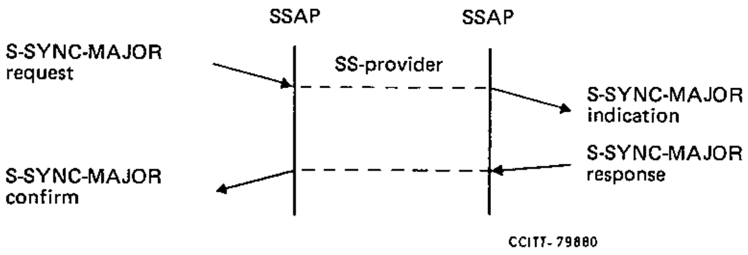


FIGURE 14/X.215

13.10 *Resynchronize service*

13.10.1 *Function*

The resynchronize service is provided to assist orderly reestablishment of communication within the current session connection, typically following an error or lack of response by either of the SS-user 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 of the synchronization point serial number, which will be the next synchronization point serial number to be used.

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 note); or

   3) S-ACTIVITY-DISCARD request (see note); or

   4) S-ACTIVITY-INTERRUPT request (see 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 § 16).

c) all undelivered data are purged;

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 a new value for the synchronization point serial number;

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 § 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 new 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. Where there are unacknowledged minor synchronization points at the time of the S-RESYNCHRONIZE request/indication, they remain unacknowledged;

i) *restart* is used to return to an agreed point which is identified by a past acknowledged 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 responsability of the SS-users;

j) *set* is used to synchronize to any valid synchronization point serial number specified by the SS-users. When there are unacknowledged minor synchronization points at the time of the S-RESYNCHRONIZE request/indication, they remain unacknowledged.

13.10.2 *Types of primitives and their parameters*

Table 19/X.215 specifies the types of session service primitives and parameters needed for the resynchronize service.

TABLE 19/X.215

Resynchronize primitives and parameters

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

M: presence of the parameter is mandatory
C: presencne 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 *Resynchronize Type* is a parameter which specifies one of the resynchronize options. Its value is one of:

a) abandon;

b) restart;

c) set.

13.10.2.2 *Synchronization Point Serial Number* depends on the resynchronize option and is defined in §§ 11.4 and 11.4.5.

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 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 shown in Figure 15/X.215. Collision cases are defined in § 16.

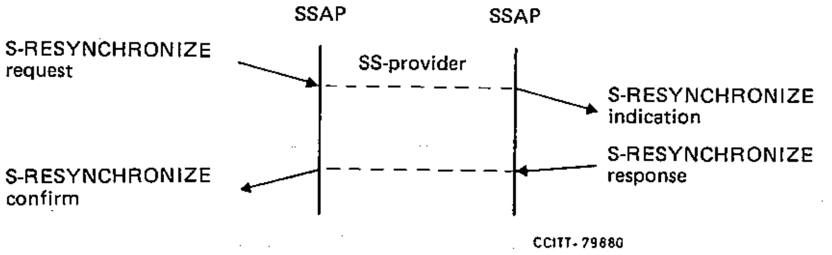


FIGURE 15/X.215

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.

Note 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.

Note 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.

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

Note 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/X.215 specifies the types of session service primitives and parameters needed for the P-exception reporting service.

TABLE 20/X.215

P-exception reporting primitives and parameters

Primitive	S-P-Exception-Report
Parameter	Indication
Reason	M

M: presence of the parameter is mandatory.

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

a) protocol error;

b) non-specific error.

13.11.3 *Sequence of primitives*

The sequence of primitives in a successful P-exception report is defined by the time sequence diagram shown in Figure 16/X.215.

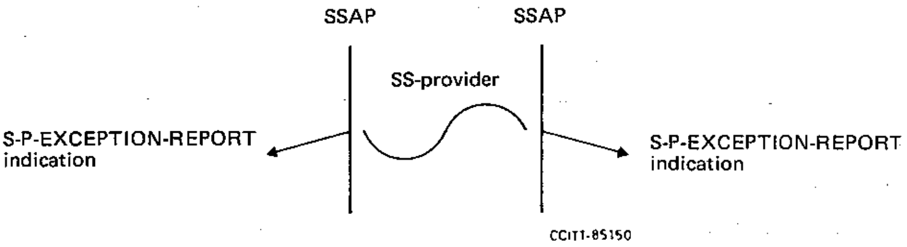

SSAP
SSAP
SS-provider
S-P-EXCEPTION-REPORT indication
S-P-EXCEPTION-REPORT indication
CCITT-85150

FIGURE 16/X.215

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/X.215.

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.

Note 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.

Note 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.

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

Note 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/X.215 specifies the types of session service primitives and parameters needed for the U-exception reporting service.

TABLE 21/X.215

U-exception reporting primitives and parameters

Primitive / Parameter	S-U-Exception-Report	
	Request	Indication
Reason	M	M(=)
SS-user data	U	C(=)

M: presence of the parameter is mandatory.
U: presence of the parameter is conditional.
C: 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 shown in Figure 17/X.215.

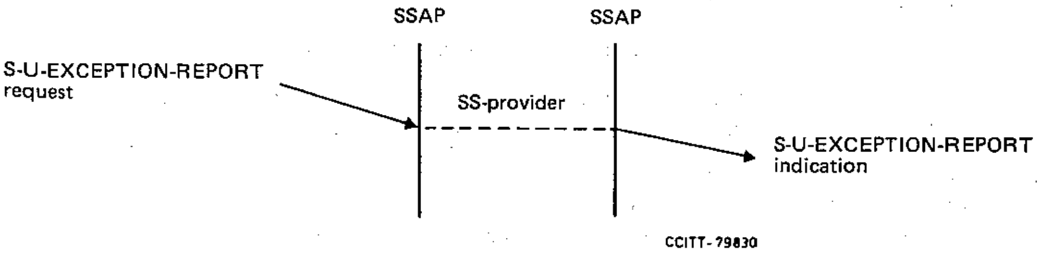


FIGURE 17/X.215

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. The value of the next synchronization point serial number to be used is set to one (see § 11.4.6). The service can only be initiated if no activity is in progress and subject to the token restrictions specified in Table 8/X.215.

13.13.2 *Types of primitives and their parameters*

Table 22/X.215 specifies the types of session service primitives and parameters needed for the activity start service.

TABLE 22/X.215

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.
U: presence of the parameter is conditional.
C: 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 shown in Figure 18/X.215.

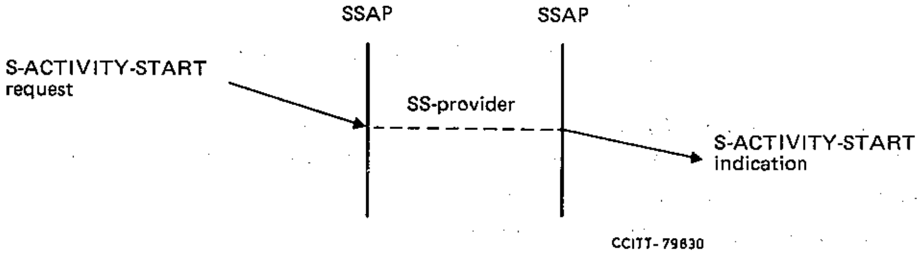


FIGURE 18/X.215

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/X.215.

13.14.2 *Types of primitives and their parameters*

Table 23/X.215 specifies the types of session service primitives and parameters needed for the activity resume.

TABLE 23/X.215

Activity resume primitives and parameters

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

M: presence of the parameter is mandatory.
U: presence of the parameter is conditional.
C: 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 *Synchronization Point Serial Number* is provided by the SS-user and is defined in § 11.4.6.

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 shown in Figure 19/X.215.

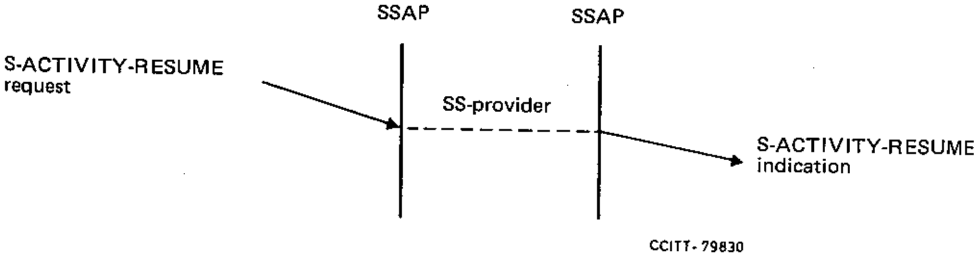


FIGURE 19/X.215

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 if an activity is in progress and subject to the token restrictions specified in Table 8/X.215. 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/X.215 specifies the types of session service primitives and parameters needed for the activity interrupt service.

TABLE 24/X.215

Activity interrupt primitives and parameters

Parameter \ Primitive	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 shown in Figure 20/X.215.

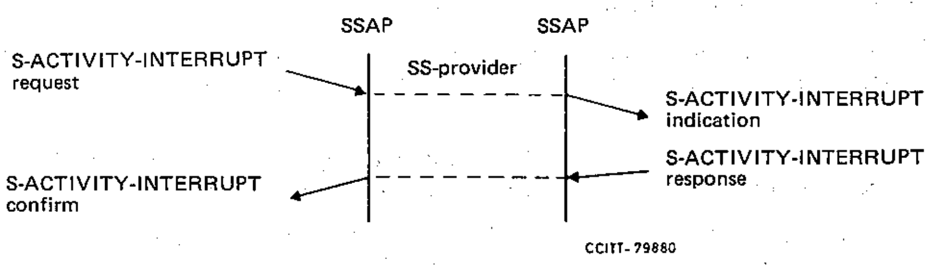

SSAP
SSAP
S-ACTIVITY-INTERRUPT request
SS-provider
S-ACTIVITY-INTERRUPT indication
S-ACTIVITY-INTERRUPT response
S-ACTIVITY-INTERRUPT confirm
CCITT- 79880

FIGURE 20/X.215

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/X.215. 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/X.215 specifies the types of session service primitives and parameters needed for the activity discard service.

TABLE 25/X.215

Activity discard primitives and parameters

Primitive / Parameter	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 shown in Figure 21/X.215.

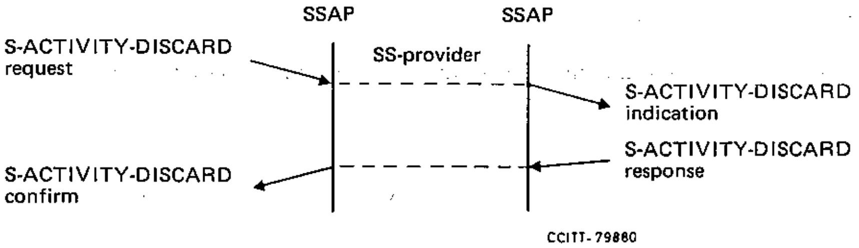

SSAP
SSAP
S-ACTIVITY-DISCARD request
SS-provider
S-ACTIVITY-DISCARD indication
S-ACTIVITY-DISCARD response
S-ACTIVITY-DISCARD confirm
CCITT-79880

FIGURE 21/X.215

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/X.215.

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/X.215 specifies the types of session service primitives and parameters needed for the activity end service.

TABLE 26/X.215

Activity end primitives and parameters

Primitive / Parameter	S-Activity-End			
	Request	Indication	Response	Confirm
Synchronization Point Serial Number	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.

13.17.2.1 *Synchronization Point Serial Number* is defined in § 11.4.6.

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 shown in Figure 22/X.215.

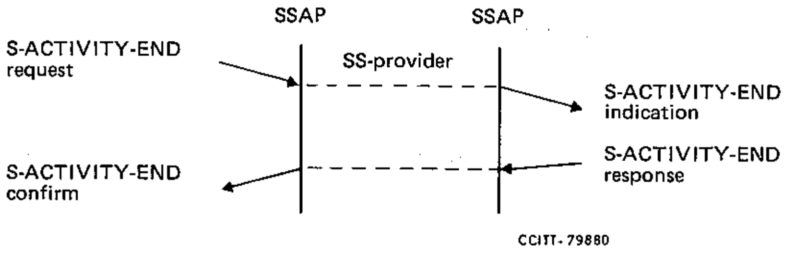


FIGURE 22/X.215

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/X.215. 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/X.215 specifies the types of session service primitives and parameters needed for the orderly release service.

TABLE 27/X.215

Orderly release primitives and parameters

Primitive / Parameter	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 shown in Figure 23/X.215.

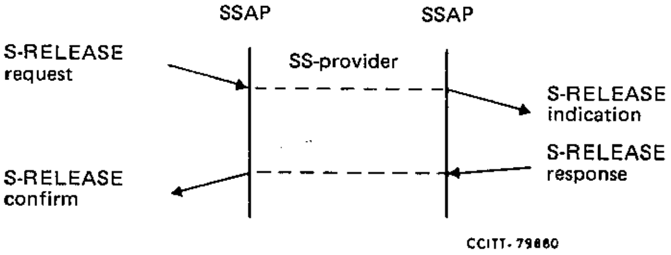


FIGURE 23/X.215

A collision of S-RELEASE requests may occur when no tokens are available. This results in S-RELEASE indications to both SS-user. 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.

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/X.215 specifies the types of session service primitives and parameters needed for the U-abort service.

TABLE 28/X.215

U-abort primitives and parameters

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

C: presence of the parameter is a user option.
U: presence of the parameter is conditional.
(=): the value of the parameter is identical to the value of the corresponding parameter of the preceding SS primitive.

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

14.2.3 *Sequence of primitives*

The sequence of primitives in a successful U-abort is defined by the time sequence diagram shown in Figure 24/X.215.

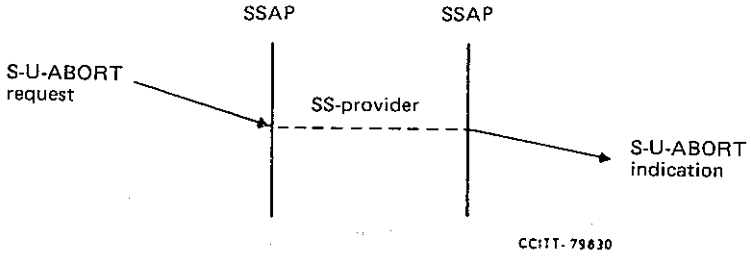


FIGURE 24/X.215

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/X.215 specifies the types of session service primitives and parameters needed for the P-abort service.

TABLE 29/X.215

P-abort primitives and parameters

Primitive / Parameter	S-P-Abort
	Indication
Reason	M

M: presence of the parameter is mandatory.

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.

14.3.3 *Sequence of primitives*

The sequence of primitives in a successful P-abort is defined by the time sequence diagram shown in Figure 25/X.215.

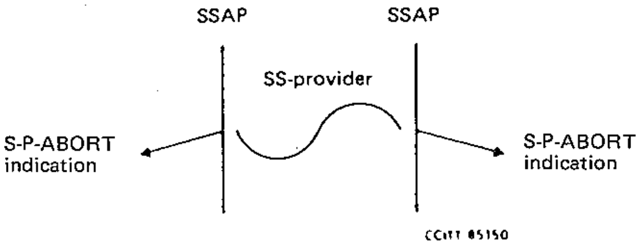


FIGURE 25/X.215

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-EXCEPTION-REPORT request.

Table 30/X.215 defines the indications that may be received which indicate that the SS-user has lost a collision resolved by the SS-provider.

TABLE 30/X.215

Indications resulting from collision resolution

SS-user receives	ER	RR	RS	RA	AI	AD	AB
SS-user is waiting for							
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.

16.2 *Collision resolution by the SS-provider*

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.

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.

ANNEX A

(to Recommendation X.215)

State tables

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 §§ 12, 13 and 14.

Table A-1/X.215 specifies the abbreviated name and name of each incoming event generated by the SS-provider.

Table A-2/X.215 specifies the abbreviated name and name of each state.

Table A-3/X.215 specifies the abbreviated name and name of each outgoing event generated by the SS-user.

Table A-4/X.215 summarizes the operations on the variables V(A), V(M), V(R) and Vsc.

Table A-5/X.215 specifies the specific actions.

Table A-6/X.215 specifies the predicates.

Tables A-7/X.215 to 14/X.215 specify the state tables.

A.2 *Notation for state tables*

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

A.2.2 Specific actions are represented by the notation (n), where n is the number of the specific action in Table A-5/X.215.

A.2.3 Predicates are represented by the notation pn, where n is the number of the predicate in Table A-6/X.215.

A.2.4 Boolean operators are represented by the following notation:

&	AND
^	NOT
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.

A.5.1 *Functional units*

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

fu-dom = {FD, HD, EXCEP, TD, NR, SY, MA, RESYN, EX, ACT, CD}

where:

FD	=	Duplex functional unit
HD	=	Half-duplex functional unit
EXCEP	=	Exceptions functional unit
TD	=	Typed data functional unit
NR	=	Negotiated release functional unit
SY	=	Minor synchronize functional unit
MA	=	Major synchronize functional unit
RESYN	=	Resynchronize functional unit
EX	=	Expedited data functional unit
ACT	=	Activity management functional unit
CD	=	Capability data 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 is defined as:

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

where:

mi = synchronize-minor token

ma = major/activity token

tr = release token

dk = data token

The following boolean functions are defined over tk-dom:

a) AV(t), for t in tk-dom, is a function which defines the availability of the corresponding token and has the following values:

AV(mi) = FU(SY)

AV(dk) = FU(HD)

AV(tr) = FU(NR)

AV(ma) = FU(MA) or FU(ACT)

b) OWNED(t), for t in tk-dom, is a function which defines the assignment of the corresponding token and is defined as:

OWNED(t) = true: if token assigned to the 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 *I*nitiating rights for the behaviour controlled by the token. This applies even if the corresponding token is not available:

I(t) = ^AV(t) OR OWNED(t)

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

A(t) = ^AV(t) OR ^OWNED(t)

e) II(t), for t in tk-dom, is a function which, when true, indicates that the SS-user has *I*nitiating rights as I(t), but this applies to the case when the behaviour may only be initiated if the corresponding token is available and owned:

II(t) = AV(t) AND OWNED(t)

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

AA(t) = AV(t) AND ^OWNED(t)

A.5.3 *SET of tokens*

The following subsets of tk-dom are defined:

RT = {tokens requested in the input event}

GT = {tokens given in the input event}

For the purpose of the following function definitions, two further 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*

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 STA713.

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)*

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)*

V(M) is used by the SS-user and is the next serial number to be used.

A.5.4.6 *V(R)*

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.

Note – Table A-4/X.215 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.

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

TABLE A-1/X.215

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
SCONcnf̣	S-CONNECT (accept) confirm primitive
SCONcnf̃	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
SPTindc	S-TOKEN-PLEASE indication primitive
SRELind	S-RELEASE indication primitive
SRELcnf̣	S-RELEASE (accept) confirm primitive
SRELcnf̃	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
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/X.215

States

Abbreviated name	Name and description
STA 01	Idle, no connection
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/X.215

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
SCONrsp̃	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
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/X.215

Operations on variables

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 <= 999999	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.

> =: 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/X.215

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]	Update Vrsp and, if necessary, Vrspnb
[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, set V(A) = V(M). Set Vsc = true Set V(M) = V(M)̣ 1
[24]	If Vsc = true, set V(A) = V(M). Set Vsc = false Set V(M) = V(M)̣ 1
[25]	Set V(A) = serial numbeṛ 1
[26]	Set V(A) = V(M) = V(R) = 1
[27]	Set V(A) = V(M) = serial numbeṛ 1. Set V(R) = 1
[28]	Set V(A) = V(M) = serial number If Vrsp = a then set V(R) = 0 If Vrsp = s then set V(R) = 0 Set Vrsp = no
[29]	Set the position of the tokens such that all available tokens are owned. Set Vact = false
[30]	Set the position of the tokens such that all available tokens are not owned. Set Vact = false
[31]	If Vsc false, set V(A) = V(M) Set V(M) = V(M)̣ 1
[32]	Set Vdnr = true

TABLE A-6/X.215

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(ACT) OR Vact) & I(dk) & I(mi) & II(ma)
p15	(^FU(ACT) OR Vact) & I(dk) & II (mi)
p18	(^FU(ACT) OR Vact) & FU(SY) & Vsc
p20	serial number = V(M)˜ 1
p21	V(M) > serial number >= V(A)
p25	(FU(SY) OR FU(MA)) & FU(RESYN)
p26	(^FU(ACT) OR Vact)
p28	FU(RESYN)
p29	(^FU(ACT) OR Vact) & FU(RESYN)
p32	serial number >= V(R)
p33	V(M) >= serial number >= V(R)
p34	FU(ACT)
p39	Vact & II(ma)
p43	((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) & ANY(II, 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) & (îFU(ACT) OR ^Vact)
p67	FU(NR)
p69	Vcoll
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

TABLE A-7/X.215

Connection establishment state table

STATE / EVENT	STA01 idle disconnected	STA02A await SCONcnf	STA08 await SCONrsp
SCONcṇf		[5] [11] STA713	
SCONcñf		STA01	
SCONind	STA08		
SCONreq	STA02A		
SCONrsp̣			[5] [11] STA713
SCONrsp̃			STA01

TABLE A-8/X.215

Data transfer state table

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

TABLE A-9/X.215

Synchronization state table

STATE EVENT	STA03 await SRELcnf	STA04A await SSYNMcnf	STA04B await SACTEcnf	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] STA10A
SSYNMreq							p13 [24] STA04A
SSYNMrsp					[22] STA713		
SSYNmcnf	[25] STA03	[25] STA04A	[25] STA04B				[25] STA713
SSYNmind							[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-10/X.215

Resynchronization state table

STATE EVENT	STA03 await SRELcnf	STA04A await SSYNMcnf	STA04B await SACTEcnf	STA05A await SRSYNcnf	STA09 await SRELrsp	STA10A await SSYNMrsp
SRSYNcnf				[28] STA713		
SRSYNind(a)	[16] [19] STA11A	[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		[16] STA11A
SRSYNreq(a)		p28 [16] STA05A			p10& p28&^p34 [16] STA05A	p28 [16] STA05A
SRSYNreq(r)					p10& p25&^p34&p33 [16] STA05A	p25&p33 [16] STA05A
SRSYNreq(s)		p28 [16] STA05A			p10& p25&^p34 [16] STA05A	p25 [16] STA05A
SRSYNrsp						

TABLE A-10/X.215 (*concluded*)

Resynchronization state table

STATE / EVENT	STA10B await SACTErsp	STA11A await SRSYNrsp	STA19 await recovery indication	STA20 await recovery request	STA713 data transfer
SRSYNcnf					
SRSYNind(a)			[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
SRSYNreq(a)	p28 [16] STA05A	p83 [16] STA05A		p28 [16] STA05A	p29 [16] STA05A
SRSYNreq(r)	p25&p33 [16] STA05A	p81&p33 [16] STA05A		p25&p33 [16] STA05A	p25&p26&p33 [16] STA05A
SRSYNreq(s)	p25 [16] STA05A	p82 [16] STA05A		p25 [16] STA05A	p25&p26 [16] STA05A
SRSYNrsp		p43 [28] STA713			

TABLE A-11/X.215

Activity interrupt and discard state table

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-11/X.215 (*concluded*)

Activity interrupt and discard state table

STATE / EVENT	STA11A await SRSYNrsp	STA11B await SACTIrsp	STA11C await SACTDrsp	STA19 await recovery indication	STA20 await recovery request	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
SACTIreq	p34&p39 STA05B				p34&p11 STA05B	p34&p39 STA05B
SACTIrsp		[30] STA713				

TABLE A-12/X.215

Activity start, resume and capability data state table

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-13/X.215

Token management and exceptions state table

STATE EVENT	STA03 await SRELcnf	STA04A await SSYNMcnf	STA04B await SACTEcnf	STA09 await SRELrsp	STA10A await SSYNMrsp	STA10B await SACTErsp
SCGind						
SCGreq						
SGTind		[11] STA04A	[11] STA04B		[11] STA10A	[11] STA10B
SGTreq		p54 [11] STA04A	p54 [11] STA04B		p54 [11] STA10A	p54 [11] STA10B
SPERind	STA20	p03 STA20 ^p03 STA713	p03 STA20 ^p03 STA713			
SPTind	STA03	STA04A	STA04B			
SPTreq				p53 STA09	p53 STA10A	p53 STA10B
SUERind	STA20	p03 STA20 ^p03 STA713	p03 STA20 ^p03 STA713			
SUERreq				p50 STA19	p50 STA19	p50 STA19

TABLE A-13/X.215 (*concluded*)

Token management and exceptions state table

STATE EVENT	STA19 await recovery indication	STA20 await recovery request	STA21 await SCDcnf	STA22 await SCDrsp	STA713 data transfer
SCGind					[11] STA713
SCGreq					p55 [11] STA713
SGTind	p60 [11] STA19 p61 [11] STA713	p60 [11] STA20 p61 [11] STA713	[11] STA21		[11] STA713
SGTreq		p57 [11] STA20 p58 [11] STA713			p54 [11] STA713
SPERind	STA19		STA20		p50 STA713 p51 STA20
SPTind			STA21		STA713
SPTreq				p53 STA22	p53 STA713
SUERind	STA19				p50 STA713 p51 STA20
SUERreq					p50 STA19

TABLE A-14/X.215

Connection release state table

STATE / EVENT	STA03 await SRELcnf	STA09 await SRELrsp	STA713 data transfer	Any other state
SPABind	STA01	STA01	STA01	STA01
SRELcṇf	STA01	[32] STA09		
SRELcñf	STA713			
SRELind	[18] STA09		STA09	
SRELreq			p63 STA03	
SRELrsp		p75 STA01 p69&p95 STA03		
SRELrsp̃		p67 STA713		
SUABind	STA01	STA01	STA01	STA01
SUABreq	STA01	STA01	STA01	STA01

ANNEX B

(to Recommendation X.215)

Usage of the generalized OSI session service to achieve compatibility with Basic T.62

B.1 *Compatibility requirements*

Recommendation T.62 for the Basic Teletex Service was adopted by the CCITT in 1980 and has been used in a number of products available or under development.

It is of essential importance that the existing and future Telematic terminals and systems must be able to interact with OSI-Systems.

This was one of the main guidelines for the development of the generalized OSI session protocol which has been conducted in very close cooperation between CCITT and ISO during the last two years of the 1981-84 study period.

This Annex shows how compatibility between OSI generalized session protocol and the basic T.62 can be achieved.

B.2 *Principles for ensuring compatibility*

The compatibility requirements outlined in the previous section impose that systems using OSI protocols can interact with terminals using Telematic protocols.

The layered structure of both protocols which conforms with the OSI Reference Model and the full compatibility of the transport-oriented layer protocols limits the problem of compatibility to the protocols above the transport-oriented layers.

As far as the higher layer protocols are concerned, the principles used to ensure the required compatibility are the following:

a) The functions of T.62 usable directly in a generalized session protocol are identified.

b) These functions and the corresponding protocol elements are integrated in the generalized OSI session protocol which must remain consistent and to continue to satisfy the various requirements of a generalized OSI session protocol.

c) The additional matters in T.62 related to the particular services and to T.61 are placed in the presentation and application layer on the top of the OSI session protocol. With appropriate application rules using the session services, the T.62 protocol elements can be generated.

The usage rules of the generalized session service are described in the next section followed by the explanation of how these translate precisely in basic T.62, thus ensuring the required compatibility.

B.3 *Usage session service to ensure compatibility with basic T.62*

The following rules specify how an OSI session user entity has to use the generalized session service to give full T.62 basic Teletex/Facsimile service (see also Figures B-1/X.215 and B-2/X.215).

The BAS subset of the generalized session service must be used.

Only the following service primitives must be used:

S-CONNECT

S-RELEASE

S-U-ABORT

S-P-ABORT

S-ACTIVITY-START

S-ACTIVITY-RESUME

S-ACTIVITY-END

S-ACTIVITY-INTERRUPT

S-ACTIVITY-DISCARD

S-SYNC-MINOR

S-U-EXCEPTION-REPORT

S-P-EXCEPTION-REPORT

S-CONTROL-GIVE

S-TOKEN-PLEASE

S-CAPABILITY DATA

S-DATA

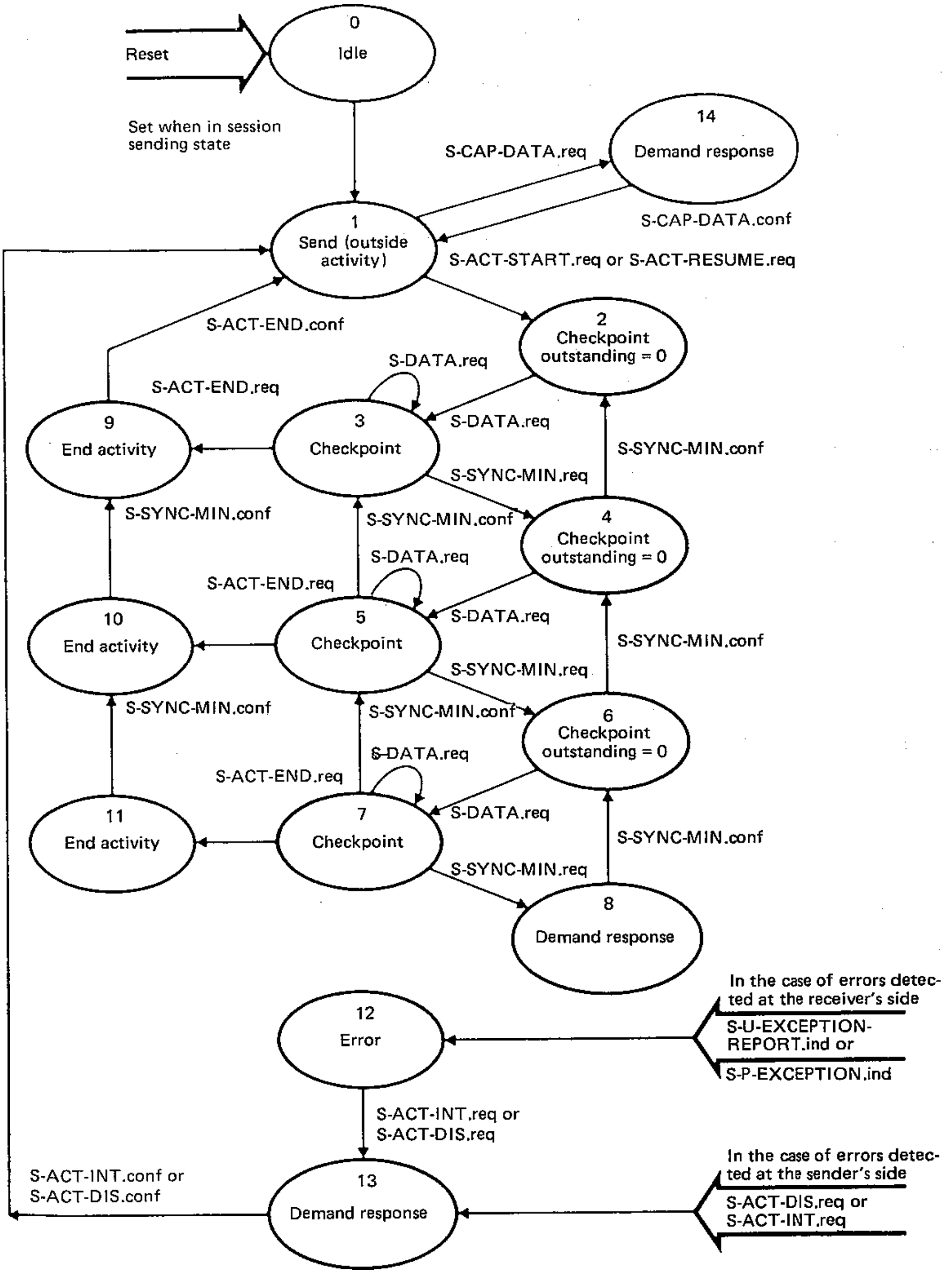
The data taken must be available.

Note – Minor sync and major/activity tokens are always available in BAS. The release token is not available.

B.3.1 *Session connection establishment phase*

The session service user initiates a session connection by using the S-CONNECT request primitive.

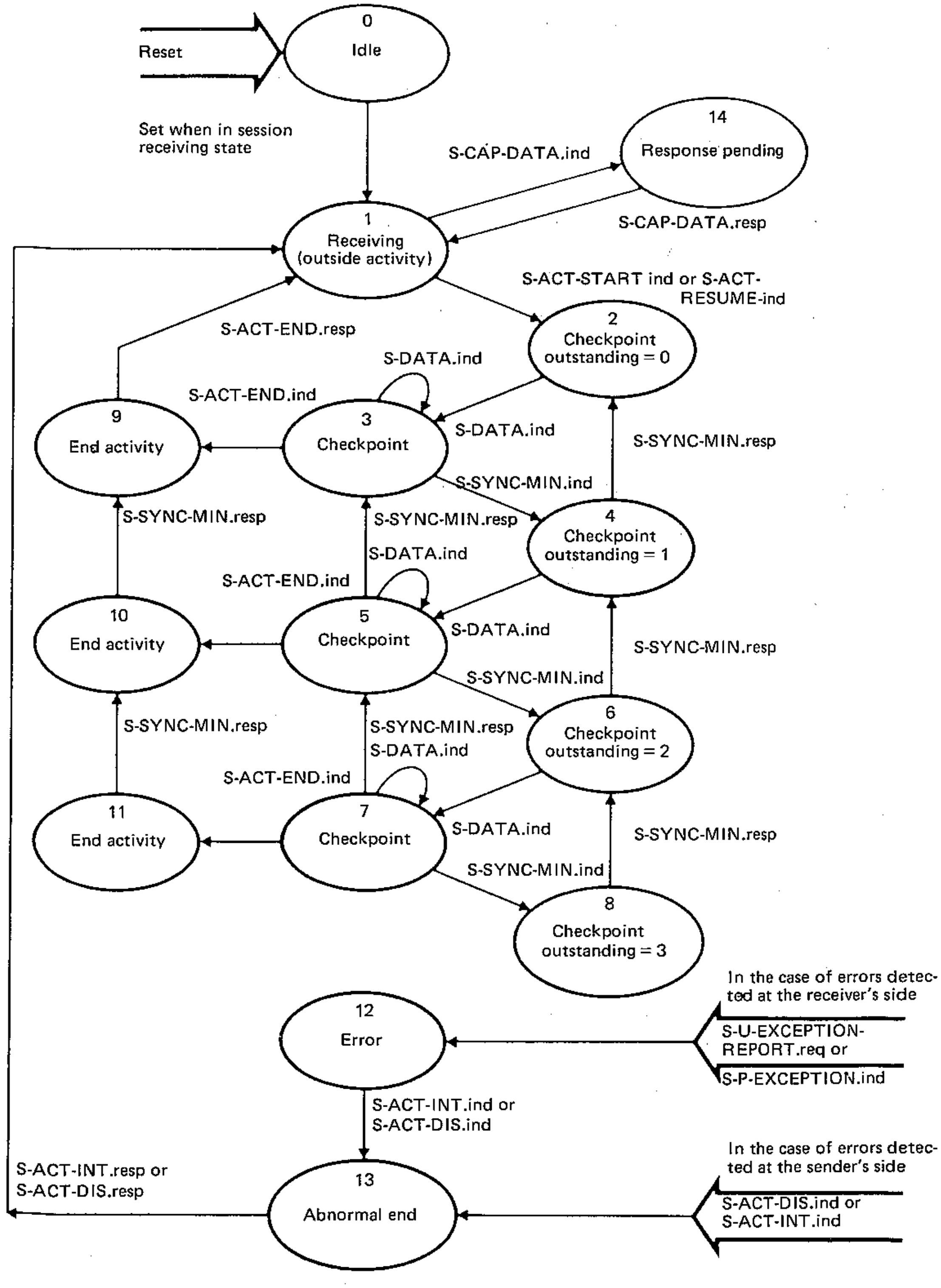
The acceptor SS-user may accept or refuse the connection with the S-CONNECT response primitive. When accepting the connection the SS-USER may request the control by issuing a S-TOKEN-PLEASE request primitive (see § B.3.5).



Note – This diagram assumes a window size of 3.

FIGURE B-1/X.215

Diagram showing Session Service Interactions for the sending Session Service User



Note – This diagram assumes a window size of 3.

CCITT-38390

FIGURE B-2/X.215

Diagram showing Session Service Interactions for the receiving Session Service User

Parameters of the S-CONNECT primitives are used in the following way:

- Connection identifier: supplied by the user in the request and response, its format is the same as defined in T.62 (calling or called terminal identifier, date and time, additional session reference number).

  Note – The four fields of the connection identifier have lengths as identified in Recommendation F.200.
- Calling and called SSAP addresses: the session layer addressing is not used by T.62 equipments.
- Quality of service:
  - no extended control (i.e. no transport expedited)
  - no optimized dialogue transfer (i.e. basic concatenation)
  - other parameters, if available, must be set in order to select class 0 of the transport protocol.
- Session requirements: the following functional units have to be selected:
  - Synchronization minor
  - Activity management
  - Capability data
  - Half duplex
  - Exceptions.
- Initial assignment of tokens: all the available tokens are assigned to the initiator.
- User data: this parameter contains the following information (see also § B.5):
  - miscellaneous session capabilities
  - window size (to be negotiated according to the rules defined in T.62)
  - service identifier
  - non basic terminal capabilities.
- Result: (in response/confirmation) this parameter is used to accept or refuse the session connection.

  In case of refusing the connection, a Reason (session) information may be present in the user data.

B.3.2 *Session connection termination phase*

The session connection can be terminated by means of the S-RELEASE primitives (orderly release) or S-U-ABORT/S-P-ABORT primitives.

Only the initiator of the session connection is allowed to release it by using S-RELEASE request.

No user data have to be supplied and the result parameter in response/confirmation indicates acceptation of the release (since the release token is not available, there is no way to refuse the release).

When requesting S-U-ABORT, the user data parameter must be absent.

Re-use of the transport connection is a local implementation choice, and does not appear at the session service level.

B.3.3 *Document management*

The T.62 document concept is equivalent to the BAS activity concept where the activity identifier is the document number.

The SS-user will use the S-ACTIVITY-START and S-ACTIVITY-RESUME primitives to start or resume documents; when resuming a previously interrupted document it is up to the user to supply and control the linking informations.

The SS-user terminates a document by using the S-ACTIVITY-END confirmed service.

Documents may be interrupted or discarded by using the S-ACTIVITY-INTERRUPT or S-ACTIVITY-DISCARD services.

B.3.3.1 *S-ACTIVITY-START*

- Activity identifier: this parameter contains the document reference number.
- User data: this parameter contains the non basic terminal capabilities, the document type identifier, the service interworking identifier encoded as defined in T.62.

B.3.3.2 *S-ACTIVITY-RESUME*

- Activity identifier: this parameter contains the document reference number.
- Old activity identifier is the same activity identifier that had been supplied when the activity had been started.
- The old session connection identifier identifies the session in which the activity had been started, it must contain the calling terminal identifier, the called terminal identifier, date and time and additional session reference number.
- The user data parameter must contain the non basic terminal capabilities, the document type identifier, the service interworking identifier encoded as in T.62.

Note – It is the responsibility of the receiving terminal to discard any user information that has been duplicated in the process of continuation of an interrupted activity.

B.3.3.3 *S-ACTIVITY-END*

- Synchronization point serial number (request/indication): the last checkpoint reference number of the document.
- No user data are supplied.

The SS-user must not use S-ACTIVITY-END request immediately after having requested a minor synchronization point (in T.62 data must be sent between the last page boundary and the end of the document).

To refuse the checkpoint indicated in S-ACTIVITY-END indication, the SS-user shall use the U-USER-EXCEPTION-REPORT service.

When activating the S-ACTIVITY-END response primitive, the receiving SS-user indicates that:

- it has not detected any error
- it accepts responsibility for the received document
- it is ready to receive a new S-ACTIVITY-START or S-ACTIVITY-RESUME indication.

B.3.3.4 *S-ACTIVITY-DISCARD*

The S-ACTIVITY-DISCARD request primitive shall be used to indicate to the remote entity, the abnormal ending of a document and that the receiver of S-ACTIVITY-DISCARD indication is not held responsible for the part of the document received so far.

Note – S-ACTIVITY-DISCARD indication is an invitation to discard the whole of the document.

The S-ACTIVITY-DISCARD response primitive shall be used to acknowledge the S-ACTIVITY-DISCARD indication and to indicate that the SS-user is ready to receive a new S-ACTIVITY-BEGIN indication.

The SS-user may use the reason parameter in the S-ACTIVITY-DISCARD primitives, only the following reasons shall be indicated:

a) local terminal error

b) unrecoverable procedural error

c) no specific reason stated.

B.3.3.5 *S-ACTIVITY-INTERRUPT*

The S-ACTIVITY-INTERRUPT request shall be used to indicate to the remote entity the point of resynchronization and to abnormally end the document transfer in progress.

The S-ACTIVITY-INTERRUPT response primitive shall be used to acknowledge the S-ACTIVITY-INTERRUPT indication. It confirms to the initiator of S-ACTIVITY-INTERRUPT that the receiver has already accepted responsibility for the received document (up to the last synchronization point for which a positive acknowledgement has been sent).

Linking of parts of an interrupted document is a local operation at the receiver and is therefore not within the responsibility of the session service provider. The SS-user may use the reason parameter in the S-ACTIVITY-INTERRUPT primitive, only one of the following reasons shall be indicated:

a) local terminal error

b) unrecoverable procedural error

c) no specific reason stated.

B.3.4 *Synchronization*

The SS-user will not request major synchronization (since the S-SYNC-MAJOR primitive has not been selected during session connection establishment phase).

In the basic Teletex service, a minor synchronization point must be inserted at each page boundary using the S-SYNC-MINOR request primitive.

The user will use only the minor synchronization service with explicit confirmation requested.

The SS-user must not request an end of activity or a minor synchronization point immediately after having requested a minor synchronization point.

The maximum window size may be negotiated during session connection establishment by using the user data parameter of the S-CONNECT primitive. (The negotiation rules are defined in T.62).

The sender (i.e. the owner of all the tokens) is permitted to recover from an interrupted transmission at only one of two points:

a) A cancellation is achieved by the subsequent use of S-ACTIVITY-RESUME request and S-ACT-DISCARD request and the transmission will be resumed by the S-ACTIVITY-START request.

b) The sender may resume by use of S-ACTIVITY-RESUME starting at the last minor sync point for which an S-SYNC-MINOR confirmation was received.

On this basis, the receiver must be able to resume reception at a minor sync point ranging from the last acknowledged sync point, to the last acknowledged sync point plus one minus the window size.

The S-SYNC-MINOR request primitive is used to indicate the boundary between pages and it also indicates a checkpoint for error recovery purposes. The S-SYNC-MINOR indication invites the SS-user to accept responsibility for the previously received page.

The S-SYNC-MINOR response shall be used to indicate that the user accepts responsibility for the page and acknowledges the minor sync point. Each minor sync point must be explicitly acknowledged.

Note – The rules (i.e. the state machine) to use the session service for synchronization are unaffected by the inclusion or exclusion of the synchronization window mechanism within/from the session layer.

B.3.4.1 *S-SYNC-MINOR*

– Type (request/indication): this parameter must indicate that explicit confirmation is requested.

– Synchronization point serial number: (indication/response/confirm) check point number.

– No user data must be supplied in the request.

– The user data must be supplied in the response with the first octet encoded as follows:

0 further traffic can be accepted

1 ability to receive further traffic is jeopardized.

B.3.4.2 *S-U-EXCEPTION-REPORT*

– Reason:

a) SS-user receiving ability jeoparidized

b) local SS-user error

c) sequence error

d) unrecoverable procedural error

e) non-specific error

– User data parameter must not be supplied.

The receiver of a document may issue an S-U-EXCEPTION-REPORT request at any time after having received an S-SYNC-MINOR indication, or S-ACTIVITY-END indication instead of giving the confirmation.

When receiving an S-U-EXCEPTION-REPORT indication or S-P-EXCEPTION-REPORT indication, the user must request the S-ACTIVITY-INTERRUPT or S-ACTIVITY-DISCARD services.

B.3.5 *Control exchange*

The S-CONTROL-GIVE service is used to give all the available tokens. This can only be used when no activity is in progress. The control give service is unconfirmed (although it is confirmed at the protocol level).

B.3.5.1 *S-TOKEN-PLEASE*

Must only be used to request all the available tokens by requesting only the data token.

When receiving an S-TOKEN-PLEASE indication with data token parameter, the SS-USER may give the control by requesting the give control service.

B.3.5.2 *S-CONTROL-GIVE*

There is no parameter associated with these service primitives.

B.3.5.3 *S-TOKEN-PLEASE*

Tokens: data token only to demand control.

No user data.

B.3.6 *Data transfer phase*

The S-DATA service must be used to send user information only within an activity.

B.3.7 *Capability exchange*

The document capability list is exchanged by using the S-CAPABILITY-DATA primitives.

The list of the capability parameters is described in T.62, § 3.4.5, and these parameters are supplied by means of the user data parameter of the S-CAPABILITY-DATA primitives.

B.4 *Usage of the session service to ensure compatibility with non basic T.62*

The following rules specify how an OSI session service user has to use the generalized session service to ensure compatibility with non basic T.62 (i.e. extension for interactive mode, typed data facility and duplex exchanges).

The rules for ensuring compatibility with basic T.62 remain unchanged except for the services described in this section.

The following additional service primitives may be used:

S-TOKEN-GIVE

S-TYPED-DATA

B.4.1 *Connection establishment phase*

For pure interactive mode, the BCS subset must be selected.

For interactive mode with document transfer, the BAS subset must be selected.

The SS-user indicates in the session requirements parameter which of the duplex and half duplex functional units is selected.

He may optionally propose the use of the typed data functional unit.

B.4.2 *Tokens exchange*

Tokens are never exchanged when an activity is in progress.

The S-CONTROL-GIVE service can be used to give all the available tokens outside activities. This service may not be used if pure interactive mode has been selected at the session establishment phase.

The S-GIVE-TOKEN service must be used to provide interactive token exchange (i.e. unconfirmed at the protocol level). All the available tokens must be given when invoking this service. The S-PLEASE-TOKEN service is used to request all the available tokens (by requesting either data, sync minor and major tokens or only the data token).

B.4.3 *Data transfer*

When interactive mode with document transfer is selected, data may be sent inside and outside activities.

When the duplex functional unit is selected, data may be sent by both users outside activities. Only the SS-user who owns the minor sync token and major/activity token is allowed to send data when an activity is in progress.

S-TYPED-DATA service may be used if the corresponding functional unit has been selected at connection establishment.

B.4.4 *Exception reporting*

In the case of an error occurring during an interactive phase (i.e. outside an Activity in BAS, or in BCS), only the S-U-ABORT service can be used to resolve it.

B.5 *Handling of T.62 PIs and PGIs not defined within the OSI session layer*

Recommendation T.62 defines a number of PIs and PGIs which are not part of the OSI session layer. Some of them are considered as components of valid session SPDUs. For example, Calling Terminal Identifier, Date and Time and Additional Session Reference Number are components of the Session Connection Identifier Parameter of the CONNECT SPDU.

Although the others are recognized within the session layer specification, they are not defined in the OSI session protocol. Therefore, local conventions are needed in order to ensure that the corresponding protocol elements are generated and received in accordance with Recommendation T.62.

The SPDUs and parameters which are subject to such conventions are listed in Table B-1/X.215.

TABLE B-1/X.215

S.62 Parameters subject to local conventions

	CONNECT	ACCEPT	REFUSE	ACTIVITY-START	ACTIVITY-RESUME	CAPABILITY DATA	CAPABILITY DATA ACK
Non-basic session capabilities	X	X	X				
Service identifier	X	X	X				
Inactivity timer	X	X				X	X
Service interworking identifier				X	X		
Acceptance of CDCL parameters							X
Storage capability negotiation						X	X
Document type identifier				X	X		
Non-basic teletex terminal capabilities	X	X	X	X	X	X	X

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks and open system communications**
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems