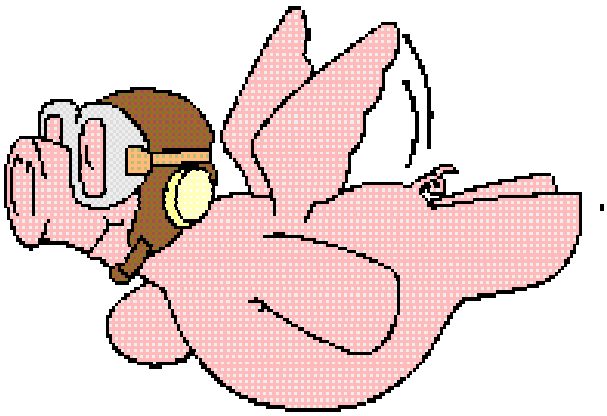
The ATN SARPs



Subvolume Four

Upper Layer Communications Service (ULCS)

Second Edition
(Final Editor's Draft)

Please note that this is the final editor's draft of the "Manual of Technical Provisions for the Aeronautical Telecommunication Network (ATN) – ICAO DOC 9705/AN956 - as circulated within the ATNP. This text will be passed to ICAO for publication. However, it should be noted that this text in no way replaces the ICAO version, nor can it be considered to be of equal status. The official definitive version is that published in hardcopy by ICAO and all claims of compliance must be made against that version.

This PDF version has been prepared for the ATNP Working Groups by FANS Information Services Ltd. – http://www.fans-is.com

Please check our web site regularly for updates to the draft SARPs

Errata and Disclaimer

Please note that this document has been prepared from a number of separate files and no attempt has been made to ensure continuity of page numbers. You may therefore find some overlap between page numbers.

This document has been prepared on a “best efforts” basis and no warrantee is offered as to its correctness.

FOREWORD

The material contained in this document was originally developed as the detailed part of the first set of Standards and Recommended Practices (SARPs) for the aeronautical telecommunication network (ATN) which has commonly been referred to as the CNS/ATM-1 Package. It was intended to make the material an appendix to the new Chapter 3 of Annex 10, Volume III, Part I, containing broad, general, stable and mostly regulatory-type provisions (the core part of new ATN SARPs).

In December 1997, the Air Navigation Commission (ANC), while conducting the final review of draft ATN SARPs, noted that actual implementation and operational experience was yet to be gained by the international civil aviation community. In this regard, the ANC agreed that the detailed part of ATN SARPs should be published as an ICAO manual (to be updated annually, if necessary), while retaining its SARPs-style language. The ANC will review the status of the document, in its entirety or in parts, after sufficient implementation and operational experience has been gained and the requirements for further standardization, in the interests of safety, regularity and efficiency of international civil aviation have been better ascertained.

This document consists of five Sub-Volumes:

Sub-Volume I — Introduction and System Level Requirements
Sub-Volume II — Air-Ground Applications
Sub-Volume III — Ground-Ground Applications
Sub-Volume IV — Upper Layer Communications Service (ULCS)
Sub-Volume V — Internet Communications Service (ICS)

Provisions contained in Sub-Volumes II, III, IV and V have been developed in accordance with system requirements specified in Sub-Volume I.

In line with the agreement by the ANC that the document should be updated on a yearly basis (if deemed necessary), the Second Edition has been published to incorporate changes necessitated by continuing validation and actual implementation activities.

TABLE OF CONTENTS

SUB-VOLUME IV. UPPER LAYER COMMUNICATIONS SERVICE

NOTE ON THE SECOND EDITION

The list below shows the parts of this sub-volume that are different from similar parts of the first edition.

Reference	Nature of change
4.3.2.1	Modification
Table 4.3-2	Modification
Table 4.3-4	Modification
4.3.3.4.5.1.1	Modification
4.3.3.6.3.1.1	Modification
4.3.3.6.5.2.2.2 b)	Modification
Table 4.6-9	Modification

4.1 INTRODUCTION

4.1.1 Scope and Objectives

4.1.1.1 The initial version of the ATN Upper Layer (UL) communications service is specified here.

4.1.1.2 The UL specification supports all current ATN applications except the ATS Message Application. This specification is designed to optimise the use of communications bandwidth, and consequently restricts the functionality available from the OSI Session and Presentation layers.

4.1.1.3 The ATN requirements are addressed for Session Layer (Layer 5), Presentation Layer (Layer 6), and a part of the Application Layer (Layer 7) of the OSI reference model. Figure 4.1-1 shows a conceptual view of the scope of the UL communications service. The remaining part of the Application Layer is the province of the individual ATN applications (i.e. the ADS, CM, CPDLC and FIS (ATIS) specifications for air-ground applications, and the ICC (AIDC) specifications for ground-ground applications).

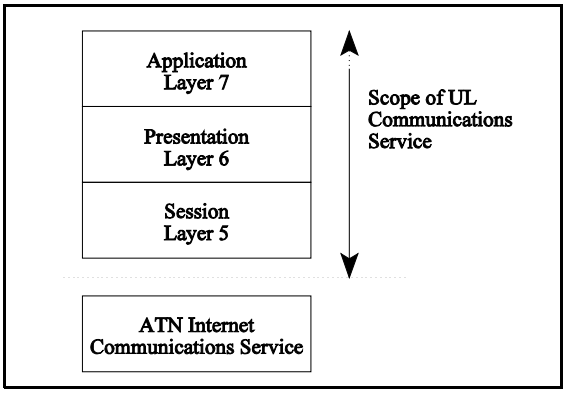


Figure 4.1-1. Conceptual view of the scope of the UL Communications Service

4.1.2 Background

4.1.2.1 The communication aspects of the ATN applications are modelled as Application Entities (AEs) (see 4.1.4.2). Figure 4.1-2 illustrates an example of the application layer structure for the ATN applications.

4.1.2.2 The specification of the UL communications service includes a profile for the protocols in the upper layers, an AE structure and a number of common application services.

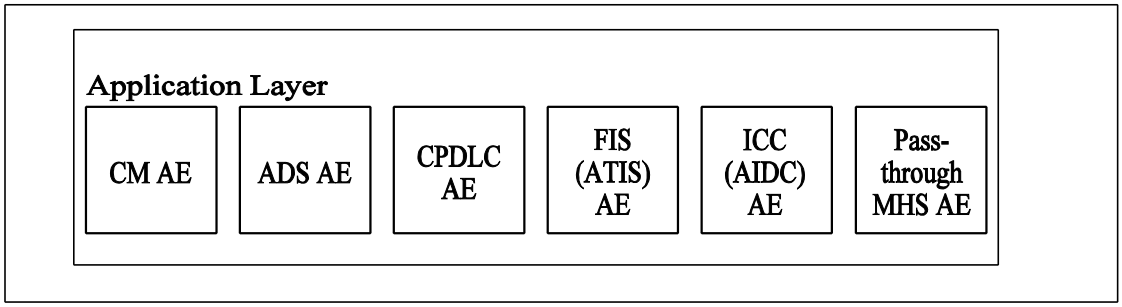


Figure 4.1-2. Conceptual view of Application Layer

4.1.3 Structure of UL Communications Service Specification

4.1.3.1 This specification is structured as follows:

a) Introduction (4.1) contains the purpose and structure of the UL Communications Service Specification, and a background to the functionality defined herein.

b) Dialogue Service Description (4.2) describes the abstract service which is defined for application specifications to refer to in order to provide a common communications service.

c) Application Entity (AE) Description (4.3) describes the Application Entity and specifies the Control Function (CF) which co-ordinates the operation of the various Application Service Elements (ASEs). It also describes the names which are assigned to various upper layer entities.

d) Session Layer Requirements (4.4) describes the requirements for the OSI Session Layer, in the form of a Profile Requirements List (PRL).

e) Presentation Layer Requirements (4.5) describes the requirements for the OSI Presentation Layer, in the form of a PRL.

f) ACSE Specification (4.6) describes the requirements for the Association Control Service Element (ACSE).

4.1.4 Upper Layer Functionality

4.1.4.1 Upper Layer Profile Overview

4.1.4.1.1 A profile is specified for the connection-oriented protocols of Session layer, Presentation layer and the Association Control Service Element (ACSE).

4.1.4.1.2 The Session portion of the specified profile is based on the efficiency enhancements to the Session protocol which are standardised in ISO/IEC 8327-1: 1996 / Amd. 1: 1997

4.1.4.1.3 The Presentation portion of the specified profile is based on the efficiency enhancements to the Presentation protocol which are standardised in ISO/IEC 8823-1: 1994 / Amd. 1: 1997

4.1.4.1.4 As a consequence of using the Session and Presentation protocol efficiency enhancements, the protocol control information transferred by these protocols amounts to two octets in each direction during the connection establishment phase, and zero octets at all other times.

4.1.4.1.5 The ACSE portion of the specified profile is based on ISO/IEC 8650-1, including the extensibility notation as specified as Amendment 1 to that standard.

4.1.4.2 Application Entity (AE) Structure

4.1.4.2.1 The specified AE structure is based on the application layer structure defined in ISO/IEC 9545, where the concepts of Application Service Element (ASE), Application Service Object (ASO) and Control Function (CF) are defined.

4.1.4.2.2 Figure 4.1-3 shows the generic structure of an AE with arrows representing the abstract service boundaries of the various elements. The “upper” service boundary is the abstract service provided by an ASE to its user(s). The “lower” service boundary is the abstract service which is provided to the ASE by the CF.

4.1.4.2.3 The ASE is an element engineered to perform a required task. ISO/IEC 9545 describes how two or more ASEs may be combined, together with a CF to co-ordinate their operation to form an ASO. In turn, an ASO may be combined with other ASOs or ASEs with a CF to form larger ASOs. The AE is the outermost ASO.

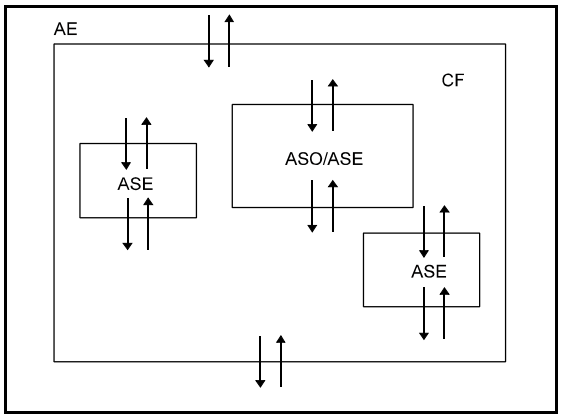


Figure 4.1-3. Generic Application Entity structure

4.1.4.3 Application Services

4.1.4.3.1 For each of the current ATN applications a specific ASE exists, and is defined in the relevant ATN Application specification. The generic name "ATN-App ASE" is used for these specific ASEs.

4.1.4.3.2 Various abstract services are specified. The services are provided at abstract service boundaries. The abstract service provided by the AE to the Application-user (i.e. the service provided at the upper boundary of the AE) is specified in 4.3. In the AE structure specified here, this service is a pass-through to the ATN-App ASE.

4.1.4.3.3 Figure 4.1-4 shows the AE structure which is used to model the ATN applications. This is described in detail in 4.3.

4.1.4.3.4 The Dialogue Service (DS) as defined in 4.2 is the abstract service which the ATN-App ASEs use to interact with the UL communications service. That is, the DS is the combination of specific internal primitives made available by the CF at the lower boundary of the ATN ASE/ASO - it is the application's "world view". In order to provide this service, the CF uses the services of ACSE.

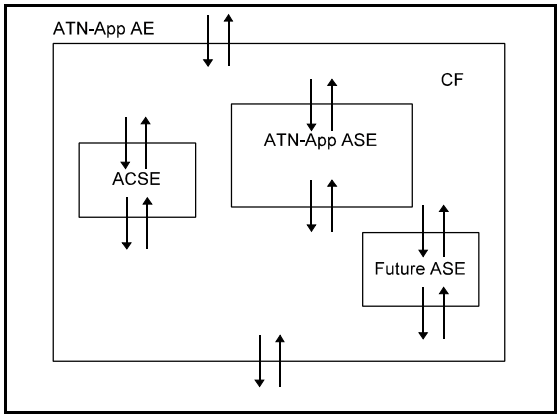


Figure 4.1-4. ATN-specific AE structure

4.2 DIALOGUE SERVICE DESCRIPTION

4.2.1 Scope of Dialogue Service

4.2.1.1 Implementations of the ATN-App ASE, together with the UL elements which provide the Dialogue Service (DS), shall exhibit the behaviour defined in this abstract service definition.

Note 1.— The Dialogue Service is the abstract service which is used by an ATN-App ASE at its lower service boundary. There is no requirement to implement the DS in any product. ATN end systems will in general be designed in such a way that it is impossible to detect (from external access) whether or not an interface corresponding to the DS has been built.

Note 2.— The DS is described from the viewpoint of the ATN-App ASE, using abstract service definition conventions. The abstract service definition is a descriptive technique used to specify the behaviour exhibited by part of the ATN application layer. Specifications of application service elements (ASEs), such as the specifications of ADS, CPDLC, CM and ATIS, may include common functionality by reference to the DS. The DS allows ATN-App ASEs to be specified without the need to consider some of the complexities of some aspects of the underlying communications.

Note 3.— The DS supports a communication relationship between two peers for a duration which exists until the peers agree to terminate the relationship or the relationship is aborted.

*Note 4.— The DS defines a service which may be used to support an ATN-App ASE at its "lower" service boundary. Such an ASE is denoted a **DS-User**. The DS-User can be specified to use the DS in a variety of ways that can be defined in terms of reliability characteristics. A number of user-visible service levels can thus be offered, including for example the following:*

a) An unconfirmed service, which allows individual messages to be transmitted after a dialogue has been set up.

b) A confirmed service, which provides end-to-end confirmation that a message sent by one DS-User was received and acknowledged by the peer DS-User.

Note 5.— An implementation of the DS provider will typically be responsible for detection of errors such as:

a) Invalid primitive (primitive unknown or error in parameter(s))

b) Invalid sequence (primitive issued at inappropriate time)

c) Insufficient resources on submission

d) Invalid or unreachable recipient on submission

e) Data field too large on receive (local implementation constraint exceeded)

f) Invalid or unreachable recipient on receive.

Note 6.— An implementation of an ATN application which makes use of the DS has to be designed with error handling procedures for local error conditions.

4.2.2 Service Primitives

4.2.2.1 Implementations which claim to support the DS functionality shall exhibit the behaviour defined by the service primitives in Table 4.2-1.

Table 4.2-1. Summary of Dialogue Service primitives

Service	Description
D-START	This is a confirmed service used to establish the binding between the communicating DS-Users.
D-DATA	This unconfirmed service is used by a DS-User to send a message from that DS-User to the peer DS-User.
D-END	This is a confirmed service used to provide the orderly unbinding between the communicating DS-Users, such that any data in transit between the partners is delivered before the unbinding takes effect.
D-ABORT	This unconfirmed service can be invoked to abort the relationship between the communicating DS-Users. Any data in transit between them may be lost.
D-P-ABORT	This unconfirmed service is used to indicate to the DS-User that the dialogue service provider has aborted the relationship with the peer DS-User. Any data in transit between the communicating DS-Users may be lost.

Note.— Table 4.2-2 lists the parameters used when invoking the services.

Table 4.2-2. Parameters of the Dialogue Service primitives

Service	Parameters
D-START	Called Peer ID Calling Peer ID DS-User Version Number Security Requirements Quality-of-Service Result Reject Source User Data
D-DATA	User Data
D-END	Result User Data
D-ABORT	Originator User Data
D-P-ABORT	(no parameters)

4.2.3 Service Definition

4.2.3.1 Sequence of Primitives

4.2.3.1.1 Implementations which claim to support the DS functionality shall exhibit behaviour allowing two communicating DS-Users to:

a) establish a dialogue;

b) exchange user data;

c) terminate a dialogue in an orderly or abnormal fashion; and

d) be informed of DS abnormal dialogue termination due to the underlying communications failure;

consistent with the appropriate use of the corresponding service primitives.

Note 1.— Either DS-User may send data at any time after the initial D-START exchange, by using the D-DATA service. Under normal circumstances, a dialogue is released by a DS-User invoking the D-END service. A dialogue is abnormally released with the D-ABORT service. If the underlying service provider abnormally releases the dialogue, the DS-Users which are aware of the dialogue will be notified with the D-P-ABORT service.

Note 2.— For the purposes of this service definition, it is only valid for the DS-User to issue and be prepared to receive primitives for one Dialogue according to the permitted sequences of DS primitives shown in Table 4.2-3, where intersections marked "Y" show possible primitives which may occur after the primitive in the column heading.

Table 4.2-3. Sequence of DS primitives for one Dialogue at one DS-User

The DS primitive X -> may be followed by the DS primitive Y	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***
1 D-START req													
2 D-START cnf (accepted)	*Y*												
3 D-START ind								*Y*		*Y*	*Y*	*Y*	*Y*
4 D-START rsp (accepted)			*Y*										
5 D-DATA req		*Y*		*Y*	*Y*	*Y*			*Y*				
6 D-DATA ind		*Y*		*Y*	*Y*	*Y*	*Y*						
7 D-END req		*Y*		*Y*	*Y*	*Y*							
8 D-END cnf (accepted)							*Y*						
9 D-END ind		*Y*		*Y*	*Y*	*Y*							
10 D-END rsp (accepted)									*Y*				
11 D-ABORT req	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*		*Y*				

The DS primitive X -> may be followed by the DS primitive Y	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***
12 D-ABORT ind	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*		*Y*				
13 D-P-ABORT ind	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*	*Y*		*Y*				

Note 3.— For compactness, each DS primitive is given a number in the column headings in Table 4.2-3; the numbers have the meanings assigned in the row headings. For simplicity, where D-START and D-END response and confirmation primitives are used, Table 4.2-3 only shows the case where the D-START or D-END request is accepted by the peer. If a D-START request is rejected, then a DS-User may not issue or receive any other primitives apart from D-START request or indication. If a D-END request is rejected, then a DS-User may continue to issue and receive primitives as if the dialogue had just been established. A D-START request results in a new instance of communication with the peer DS-User, so could occur at any time. Table 4.2-3 only applies to a single instance of communication.

4.2.3.2 The D-START service

4.2.3.2.1 The behaviour defined by the D-START service primitive shall be provided to enable the setting up of a dialogue between two DS-Users.

Note 1.— D-START is a confirmed service which is invoked by a DS-User (the dialogue-initiator) to start a dialogue with a peer DS-User. D-START request, indication, response and confirmation primitives are defined, as illustrated in Figure 4.2-1.

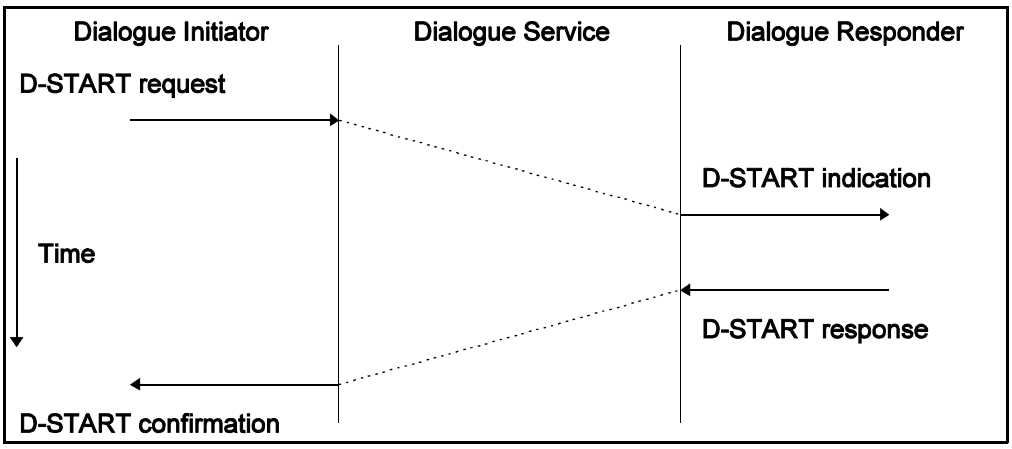


Figure 4.2-1. D-START sequence diagram

Note 2.— The initiating DS-User issues a D-START request primitive. It is not then valid to issue any other primitives (except D-ABORT) until a D-START confirmation is received. When the responding DS-User receives the D-START indication primitive, it must decide whether or not to accept this instantiation of the dialogue service. It may issue only a D-START response or a D-ABORT request primitive. The D-START response and confirmation primitives contain a Result parameter which defines whether the responding DS-User accepts or rejects the request. If the responding DS-User accepts the request, then the dialogue is established. If it rejects the request, then no dialogue exists. The parameters of the D-START primitives are specified in Table 4.2-4.

Table 4.2-4. D-START parameters

Parameter Name	*Req*	*Ind*	*Rsp*	*Cnf*
Called Peer ID	*M*			
Calling Peer ID	*U*	*C(=)*		
DS-User Version Number	*U*	*C(=)*	*U*	*C(=)*
Security Requirements	*U*	*C(=)*	*U*	*C(=)*
Quality Of Service	*M*	M(=)	U	M(=)
Result			*M*	*M*
Reject Source				*C*
User Data	*U*	*C(=)*	*U*	*C(=)*

Note 3.— The Called Peer ID parameter is used in the D-START service to specify the name of the intended peer DS-User, and takes an abstract value corresponding to either a 24-bit ICAO aircraft-id or an ICAO facility designator.

Note 4.— The Calling Peer Id parameter is optionally used in the D-START service to specify the name of the initiating DS-User, and is either absent or takes an abstract value corresponding to either a 24-bit ICAO aircraft-id or an ICAO facility designator. Its presence in the indication primitive is conditional upon it being specified by the DS-User in the request primitive.

Note 5.— The DS-User version number allows peer DS-Users to exchange version information. The parameter is optional in the request and response primitives. Its presence in the indication primitive is conditional upon it being specified by the DS-User in the request primitive, and its presence in the confirmation primitive is conditional upon it being specified by the DS-User in the response primitive. If present, it may take any abstract value in the range 1 to 255.

Note 6.— The Security Requirements parameter allows the DS-Users to exchange requirements for security. The parameter is optional in the request and response primitives. Its presence in the indication primitive is conditional upon it being specified by the DS-User in the request primitive, and its presence in the confirmation primitive is conditional upon it being specified by the DS-User in the response primitive.

Note 7.— The Quality Of Service parameter allows the initiating DS-User to specify in the request primitive its requirements for the quality of service (QOS) to be provided for the dialogue. For ATN, the parameter is not modified by the DS-provider, so the value in the indication primitive is equal to the value in the request. The only QOS component which may be modified by the DS-User in the response primitive is Residual Error Rate (see Note 10). Otherwise, the QOS parameter in the response primitive is assumed by the CF to be equal to the value in the indication primitive. The value of the QOS parameter in the confirmation primitive is equal to that present or assumed in the response primitive. The following QOS parameters may be specified:

a) Routing Class - valid values are defined in Table 5.6-1

b) Priority - valid values are defined in Table 1-2

c) *Residual Error Rate (RER) - valid values are "low" and "high".*

Note 8.— If the Routing Class parameter is not provided by the DS-User in the D-START Request primitive, and the DS-User is an ATS application as specified in 2.1 - 2.4, then the default value "ATSC: No Traffic Type Policy Preference" is assumed. If the DS-User is not an ATS application as specified in 2.1 - 2.4, then the default traffic type "General Communications" is assumed.

Note 9.— If a Priority value is not provided by the DS-User in the D-START Request primitive, then the default value "network/systems administration" is assumed.

Note 10.— For the RER parameter, "low" means a low error rate, i.e. a high quality connection, and "high" means a higher error rate, i.e. a lower quality connection. The high RER allows non-use of the transport checksum in the ATN. A limited negotiation is possible, such that if the RER value received in the indication primitive is "high", the DS-User may set the value in the response primitive to either "low" or "high".

Note 11.— The Result parameter specifies whether the requested dialogue start has been accepted. It can take one of the abstract values:

a) accepted;

b) rejected (transient); or

c) rejected (permanent).

Note 12.— The Reject Source parameter is present if the Result parameter has one of the values "rejected (transient)" or "rejected (permanent)". It specifies who rejected the start of the dialogue, and can have one of the abstract values:

a) DS user; or

b) DS provider.

Note 13.— The User Data parameter allows the peer DS-Users to exchange data during the D-START service invocation. Its presence in the indication primitive is conditional upon it being specified by the DS-User in the request primitive, and its presence in the confirmation primitive is conditional upon it being specified by the DS-User in the response primitive.

4.2.3.3 The D-DATA service

4.2.3.3.1 The behaviour defined by the D-DATA service primitive shall be provided to enable the exchange of information between two DS-Users.

Note 1.— D-DATA is an unconfirmed service which provides data transfer between peer DS-Users. The D-START service must first have been successfully completed to establish the communication relationship between the peers. Request and indication primitives are defined, as illustrated in Figure 4.2-2.

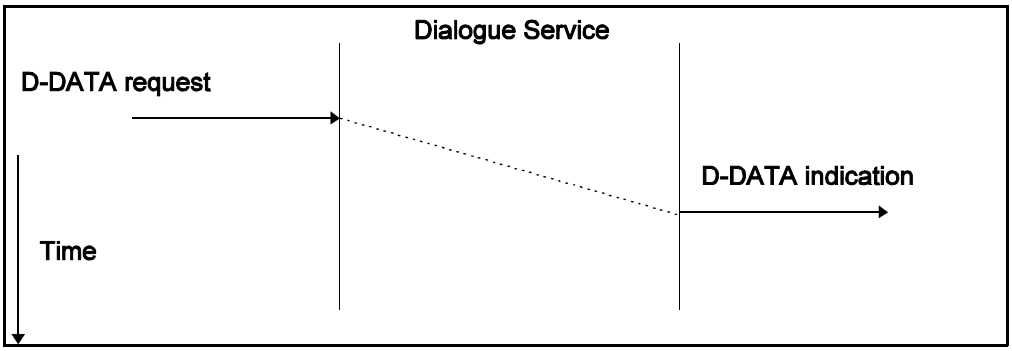


Figure 4.2-2. D-DATA sequence diagram

Note 2.— The parameters of the D-DATA primitives are specified in Table 4.2-5.

Table 4.2-5. D-DATA parameters

Parameter Name	*Req*	*Ind*
User Data	*M*	*M(=)*

Note 3.— The User Data parameter contains the data to be transferred from a DS-User to its peer, using an existing dialogue.

4.2.3.4 **The D-END service**

4.2.3.4.1 The behaviour defined by the D-END service primitive shall be provided to enable the orderly termination of a dialogue between two DS-Users.

Note 1.— D-END is a confirmed service which causes the end of a dialogue. It may be invoked by either of the communicating partners. When the D-END service is invoked, the DS performs an orderly release, whereby any service previously invoked is completed before the dialogue is terminated. The D-END service defines request, indication, response and confirmation primitives, as illustrated in Figure 4.2-3.

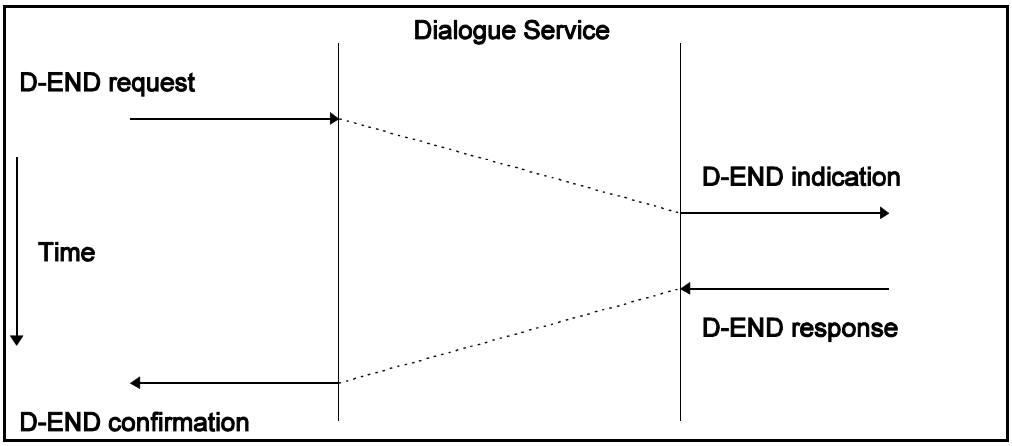


Figure 4.2-3. D-END sequence diagram

Note 2.— The DS-User which wishes to terminate the dialogue issues a D-END request primitive. After issuing a D-END request primitive, the DS-User must not then issue any other service primitive (except D-ABORT if required), until a D-END confirmation is received. After issuing a D-END request primitive, the DS-User must be prepared to continue receiving D-DATA indications from the peer user, until a D-END confirmation primitive is received.

Note 3.— If the D-END confirmation contains a result code of "accepted" then the dialogue no longer exists. If the D-END confirmation contains a result code of "rejected" then the peer DS-User does not wish to terminate the dialogue, and both DS-Users are then free to use the dialogue as if the D-END service had never been invoked.

Note 4.— When a DS-User receives a D-END indication primitive, it may continue to send data using the D-DATA service, but it may at some time issue a D-END response primitive, with a result code of "accepted" or "rejected". After issuing a D-END response primitive with result "accepted", a DS-User must not issue any other service primitive, as the dialogue no longer exists. After issuing a D-END response primitive with result "rejected", a DS-User may issue any other service primitive, as if the D-END service had never been invoked.

Note 5.— The parameters of the D-END primitives are specified in Table 4.2-6.

Table 4.2-6. D-END parameters

Parameter Name	*Req*	*Ind*	*Rsp*	*Cnf*
Result			*M*	*M(=)*
User Data	*U*	*C(=)*	*U*	*C(=)*

*Note 6.— The Result parameter specifies whether the requested dialogue end has been accepted. It can take one of the abstract values: "**accepted**" or "**rejected**".*

Note 7.— The User Data parameter contains the data to be transferred from a DS-User to its peer, using an existing dialogue. Its presence in the confirmation primitive is conditional upon it being specified by the DS-User in the response primitive.

Note 8.— In the event of service disruption (e.g. by D-P-ABORT), the invoker of the D-END response primitive will never know that any associated User Data failed to be delivered, as the service is already terminated.

Note 9.— A D-END collision occurs when both peers issue a D-END request primitive near-simultaneously, such that neither peer has yet received the D-END indication primitive corresponding to the remote peer's D-END request. The collision is handled by the CF on behalf of the DS-User. However, one result of the collision handling is that any User Data present in the D-END request will be delivered to the peer in a D-END confirmation primitive, rather than the usual D-END indication. This means that the peer will be unable to react to the contents of the User Data parameter, as the dialogue will have terminated. When a DS-User application is designed such that either peer may terminate the dialogue, then the application can not require a response to any User Data which is sent on a D-END request primitive. The following sequence diagram illustrates the D-END collision from the viewpoint of the two DS-Users:

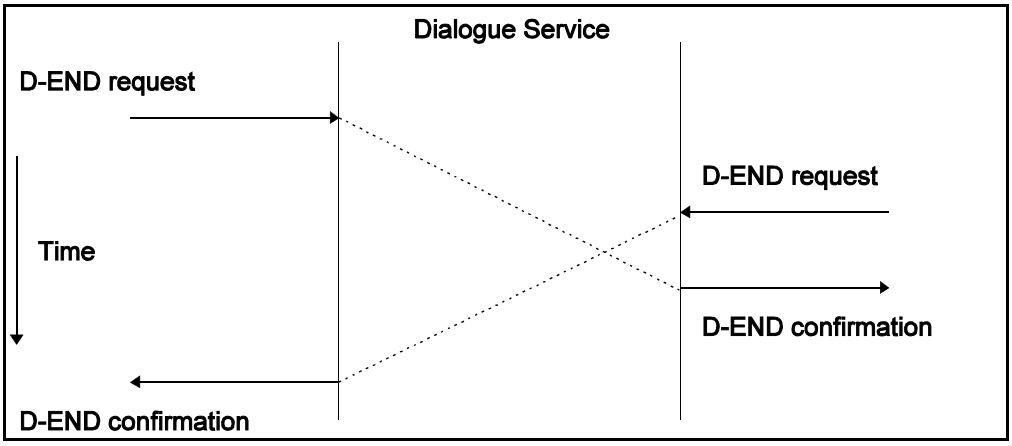


Figure 4.2-4. D-END collision sequence diagram (DS-User view)

4.2.3.5 The D-ABORT service

4.2.3.5.1 The behaviour defined by the D-ABORT service primitive shall be provided to enable the abnormal release of a dialogue between two DS-Users, by either DS-User.

Note 1.— The D-ABORT service request and indication primitives are as illustrated in Figure 4.2-5.

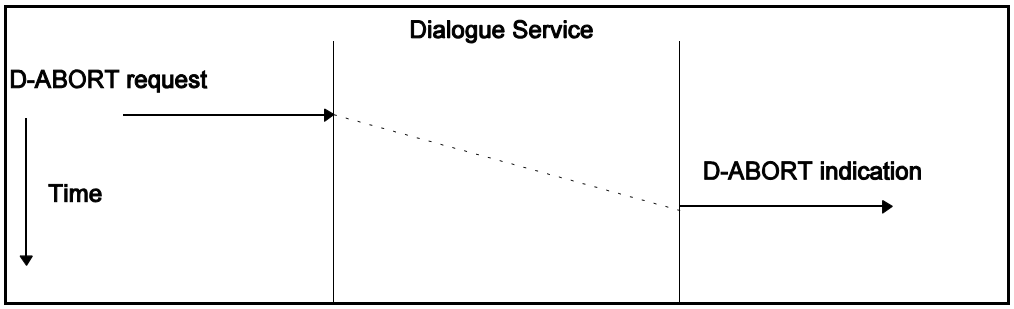


Figure 4.2-5. D-ABORT sequence diagram

Note 2.— When a dialogue is aborted, data in transfer may be lost. The parameters of the D-ABORT primitives are specified in Table 4.2-7.

Table 4.2-7. D-ABORT parameters

Parameter Name	*Req*	*Ind*
Originator	*U*	*C(=)*
User Data	*U*	*C(=)*

Note 3.— The Originator parameter is used to distinguish the source of the abort. Its presence in the indication primitive is conditional upon it being specified by the DS-User in the request primitive. It can take one of the following abstract values:

a) User — the abort originated from the Application-user; or

b) Provider — the abort originated in the ATN-App AE (including the ATN-App ASE).

Note 4.— If the D-ABORT Originator parameter is not specified, the default value "Provider" is assumed.

Note 5.— The User Data parameter contains the data to be transferred from a DS-User to its peer, using an existing dialogue. Its presence in the indication primitive is conditional upon it being specified by the DS-User in the request primitive. There is no guarantee that the peer will receive the User Data; the sender will not be informed if the User Data is not delivered.

4.2.3.6 The D-P-ABORT service

4.2.3.6.1 The behaviour defined by the D-P-ABORT service primitive shall be provided to indicate an abnormal release of a dialogue by the supporting communications service.

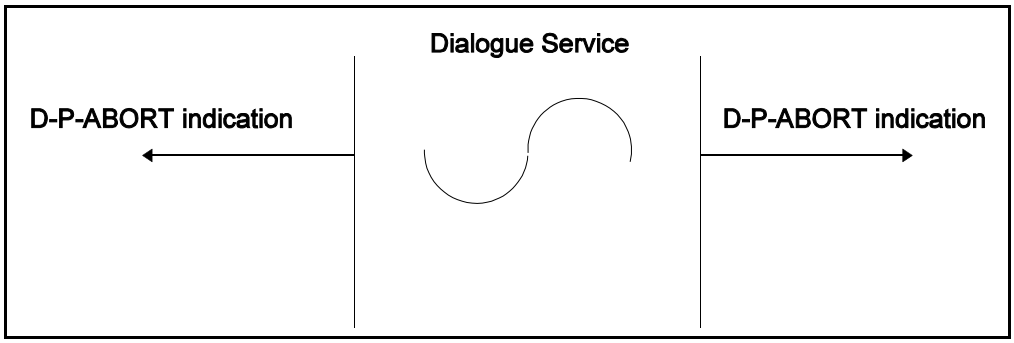


Figure 4.2-6. D-P-ABORT sequence diagram

Note 1.— For the D-P-ABORT service, only an indication primitive is defined, as illustrated in Figure 4.2-6.

Note 2.— The D-P-ABORT service allows the supporting communications service to indicate to the DS-Users that it aborted the dialogue. When the dialogue is aborted, any data in transit may be lost. The D-P-ABORT primitive has no parameters.

4.3 APPLICATION ENTITY (AE) DESCRIPTION

4.3.1 Introduction

4.3.1.1 The ATN-App AE shall exhibit external behaviour as if implemented according to the model shown in Figure 4.3-1, with the protocols defined in ACSE and the ATN-App ASE specifications.

Note 1.— As indicated in 4.1, the AE is described in terms of the Service which it displays to the Application-user, and in terms of the Control Function (CF) which mediates the interactions of the components of the AE.

Note 2.— Figure 4.3-1 also indicates which paragraph describes the behaviour of the CF in response to events at various service boundaries.

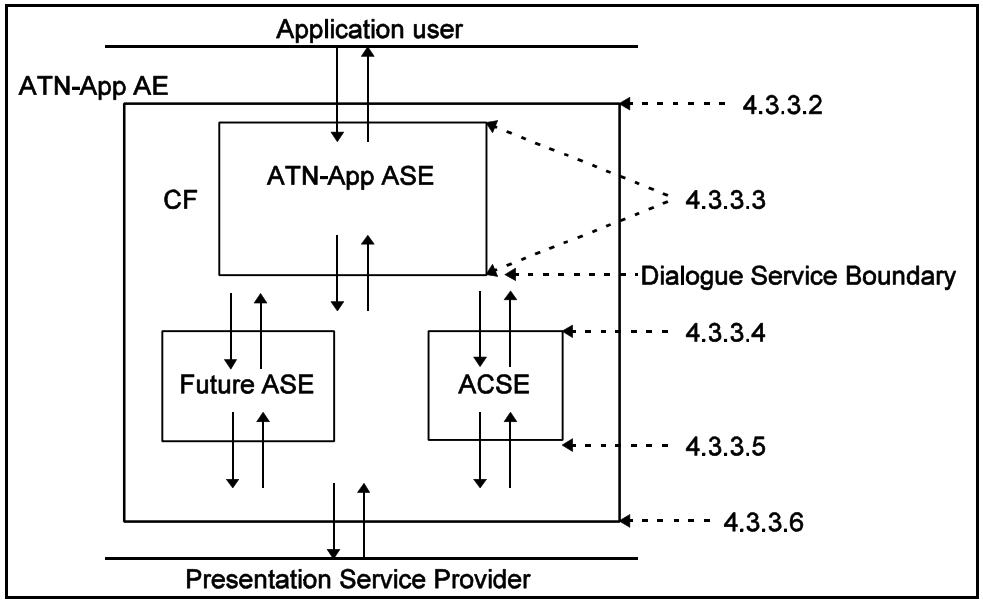


Figure 4.3-1. Components of ATN-App AE

Note 3.— The "Future ASE" component in Figure 4.3-1 indicates how additional functionality can be added in future versions of the ATN Upper Layer Architecture. ACSE provides the basic mechanisms for establishing and releasing an application association. The service provided by the ATN-App AE represents an abstract description of the Application Programming Interface (API) seen by the Application user. The CF of the ATN-App AE specifies how the interactions at the ATN-App AE Service boundary invoke the appropriate service primitives of the constituent ASEs, which in turn generate the actual protocol. The CF also specifies how the constituent ASEs interact with the supporting communications service.

Note 4.— A CF specification is not a service definition of the ATN-App AE or its components. It only defines the actions of the CF as a result of service invocations visible to the CF. Thus, the specification is organised around specifying the response of the CF to these inputs. 4.3.3.2 specifies the actions that result from the inputs of the Application-user. 4.3.3.4 and 4.3.3.5 specify the actions that result from the service invocations of the ACSE component ASE. 4.3.3.3 specifies the actions that result from the service invocations of the ATN-App ASE component ASE. 4.3.3.6 specifies the actions that result from the inputs from the supporting service.

Note 5.— The CF specification describes the overall behaviour of the ATN-App AE. It is not a requirement that an identifiable CF entity be realised in an implementation.

Note 6.— This CF specification assumes that the embedded ASEs (ATN-App ASE and ACSE) are modelled as atomic entities, such that when an input event is invoked by the CF, that event is processed to completion by the ASE and the CF responds to any resulting output events from the ASE, all within the same logical processing thread. This model avoids the need to specify further transient states within the CF. It does not imply any particular implementation architecture.

Note 7.— In the current version of the ATN Upper Layers, the service interface presented to the Application-user is a simple pass-through to the ATN-App ASE. That is, the Application-user passes request and response primitives directly to, and receives indication and confirmation primitives directly from, the ATN-App ASE.

Note 8.— The CF described here supports the four air-ground applications currently defined, and the MHS pass-through application. The specification of the CF for the AIDC application is included in the AIDC specification.

Note 9.— For the purposes of this specification, the ATN-App AE is modelled such that a new instance of communication (effectively a new AE invocation) is implicitly created (a) for each request from the AE-User that will require a new association (i.e., that will result in a D-START request being invoked), and (b) for each indication from the underlying communications service that a new connection is requested. The AE invocation ceases to exist when the underlying communications service connection is disconnected and the CF is idle (i.e., in the NULL state).

4.3.2 Application Level Naming and Context Definition

4.3.2.1 ATN Naming Hierarchy

Note 1.— Names, in the form of object identifiers (OIDs), are assigned here to the defined ATN entities.

Note 2.— ISO/IEC 9834-1 specifies the top of the hierarchical OID name space. At the first level, provision is made for ISO, International Telecommunication Union - Telecommunication Standardisation Sector (ITU-T) and joint ISO/ITU-T sub-name spaces. The ISO name space is further subdivided into:

a) standard (0)

b) registration-authority (1)

c) member-body (2)

d) identified-organisation (3)

Note 3.— ICAO has requested and obtained the allocation of an International Code Designator (ICD), according to ISO 6523. The ICD obtained, name and number "icao (27)", uniquely identifies ICAO and allows ICAO to establish its own object identifier name space within the International Organisation arc using the prefix: { iso (1) identified-organisation (3) icao (27) }. Similarly, IATA has obtained an ICD of "iata (19)"; values assigned under the IATA name space are out of scope.

4.3.2.1.1 Within the ICAO name space, the initial allocation of object identifiers shall follow the structure and values defined here.

Note 1.— In the future, it is likely that the ATN object identifier tree will have further levels of structure, and that fully location-independent values will be assigned.

Note 2.— The ATN naming hierarchy is illustrated in Figure 4.3-2.

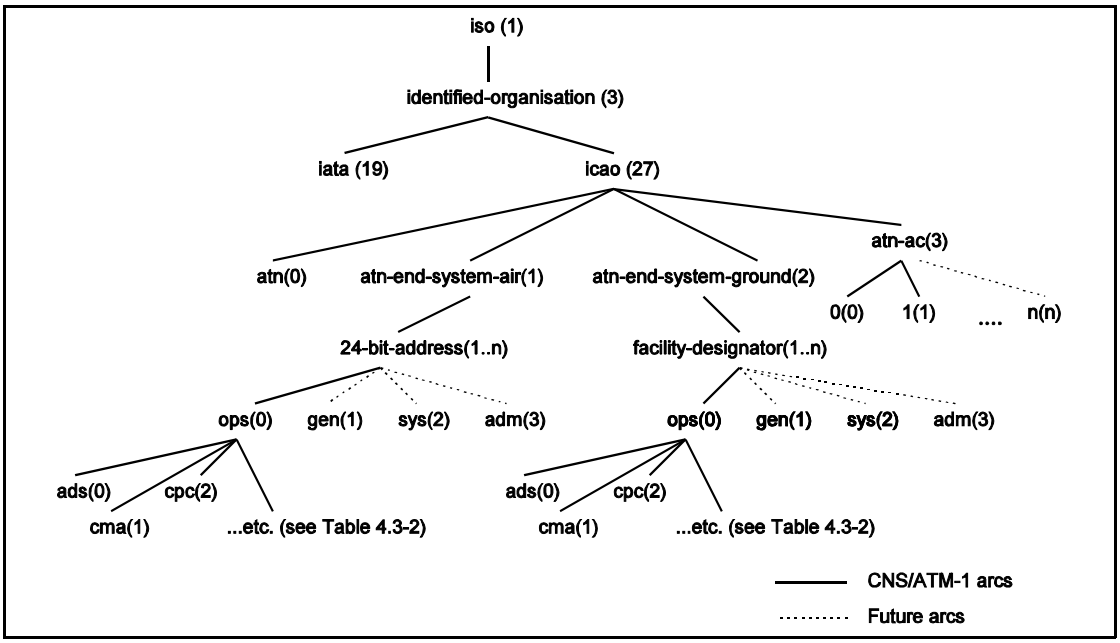


Figure 4.3-2. ATN Naming Hierarchy

4.3.2.1.2 Immediately under the ICAO arc, the values specified in Table 4.3-1 shall be used to specify the next level of the naming hierarchy.

Table 4.3-1. Top-level ICAO Identifiers

Name and numeric value	Description
atn (0)	General ATN identifiers
atn-end-system-air (1)	ATN aircraft end systems. The following OID component beneath this arc is a 24-bit ICAO aircraft identifier
atn-end-system-ground (2)	ATN ground end systems. The following OID component beneath this arc is an ICAO facility designator
atn-ac (3)	ATN application context names

4.3.2.2 Application Process titles

Note.— Application process titles are allocated underneath either of the Object Identifier arcs:

{ atn-end-system-air (1) } or { atn-end-system-ground (2) }.

Immediately subordinate to this arc is an arc whose value is an INTEGER derived from either the 24-bit ICAO aircraft address or the ICAO facility designator, as described in 4.3.2.4. Immediately beneath that arc is an arc whose value is determined by the category of the ATN application. For the present, only the following name and value are defined for the application category:

{ operational (0) }.

4.3.2.2.1 Each application category on each ATN end system shall be assigned an unambiguous application process title (AP-title).

4.3.2.2.2 The AP-title shall be an Object Identifier type (i.e. an AP-title-form2 as defined in ISO/IEC 8650-1).

4.3.2.2.3 Application Process titles shall be of the form:
either:
{iso (1) identified-organisation (3) icao (27) atn-end-system-air (1) <end-system-id> (n) operational (0)}
or:
{iso (1) identified-organisation (3) icao (27) atn-end-system-ground (2) <end-system-id> (n) operational (0)}
where:
<end-system-id> is the ICAO 24-bit address for aircraft end systems, or the ICAO facility designator for ground end systems.
(n) is an INTEGER value derived from the <end-system-id>.

Note.— The algorithm for deriving the INTEGER n from the <end-system-id> is defined in 4.3.2.4.

4.3.2.3 Application Entity Titles

4.3.2.3.1 Each ATN application entity shall be assigned an unambiguous application entity title (AE Title).

4.3.2.3.2 For ATN, an AE Title shall be an Object Identifier type as defined in ISO/IEC 8824-1 (i.e. an AE-title-form2 as defined in ISO/IEC 8650-1).

Note.— The AE Title is composed of an Application Process title (AP-title) and an AE-qualifier.

4.3.2.3.3 The AE-qualifier component of the AE Title shall be an INTEGER type (i.e. an AE-qualifier-form2 as defined in ISO/IEC 8650-1).

4.3.2.3.4 The AE-qualifier value arc of the AE Title object identifier represents the ATN application type (e.g. "ADS" or "CMA"), and shall take one of the values specified in Table 4.3-2.

Table 4.3-2. Assigned AE-Qualifier values

ATN ASE type	**ATN AE-Qualifier name and numeric value**
Automatic Dependent Surveillance	ADS (0)
Context Management Application	CMA (1)
Controller Pilot Data Link Communication	CPC (2)
Automatic Terminal Information Services (ATIS)	ATI (3)
Type A Gateway	GWA (4)
Systems Management Application (SMA)	SMA (5)
ATS Inter-Facility Data Communications (AIDC)	IDC (6)
ATS Message Application	AMS (7)
AFTN-AMHS Gateway	GWB (8)
ATS Message User Agent	AUA (9)
ADS Report Forwarding	ARF (10)
Aviation Routine Weather Report (METAR)	MET (11)
Generic ATN Communication Service AE (GACS)	GAC (12)

4.3.2.3.5 Thus, AE-titles conforming to this definition shall be of the form:

either:

{ iso (1) identified-organisation (3) icao (27) atn-end-system-air (1) <end-system-id> (n) operational (0) <ae-qualifier> (m) }

or

{ iso (1) identified-organisation (3) icao (27) atn-end-system-ground (2) <end-system-id> (n) operational (0) <ae-qualifier> (m) }

where:

<end-system-id> is the ICAO 24-bit address for aircraft end systems, or the ICAO facility designator for ground end systems.

(n) is an INTEGER value derived from the <end-system-id>

<ae-qualifier> is the name form of the AE qualifier from Table 4.3-2

(m) is the number form of the AE qualifier from Table 4.3-2.

Note.— The algorithm for deriving the INTEGER n from the <end-system-id> is defined in 4.3.2.4.

4.3.2.4 Encoding of End System Identifiers

Note 1.— Where <end-system-id> appears as a component of an Object Identifier, the encoding of the OID subidentifier value is obtained as defined in the following text.

Note 2.— For ground stations, the <end-system-id> is derived from an eight-letter facility designator, e.g., "LFPODLHX". The syntax of the first four letters is defined in ICAO Doc 7910 "Location Indicators"; the syntax of the remaining letters is defined in ICAO Doc 8585 "Designators for Aircraft Operating Agencies, Aeronautical Authorities and Services."

4.3.2.4.1 For aircraft, the <end-system-id> naming arc shall be the binary value of the 24-bits comprising the ICAO aircraft identifier, expressed as an INTEGER in the range $0..(2^{24}-1)$ and encoded as an Object Identifier subidentifier as defined in ISO/IEC 8825-1.

4.3.2.4.2 For ground stations, the encoding of the <end-system-id> naming arc shall be derived from the ICAO facility designator, which is a sequence of characters from the restricted character set (A..Z), as follows:

a) Each character is encoded into one octet where:

1) the most significant bit (bit 8) indicates whether the character is the last in the sequence: it is set to zero in the last octet and one in each preceding octet;

2) the next most significant bit (bit 7) is set to zero;

3) the six least significant bits (bits 6 - 1) contain the character encoded as a 6-bit value such that A is encoded as the binary value 000001, B is encoded as 000010, and so on up to Z which is encoded as 011010

Note.— This coding gives compatibility with the Basic Encoding Rules for an Object Identifier subidentifier in ISO/IEC 8825-1. The character coding is equivalent to the "6-bit ASCII" subset of International Alphabet Number 5 (IA5) defined by ITU, which is adopted in SSR Mode S specifications. If required, the encoding can be extended to include numeric characters, with 0 to 9 encoded as binary values 110000 to 111001 respectively, and the space character can be encoded as binary value 100000.

b) The <end-system-id> is the concatenation of these octets.

Note.— Conceptually, bits 7 -1 from each octet are concatenated to form an unsigned binary number whose most significant bit is bit 7 of the first octet and whose least significant bit is bit 1 of the last octet.

4.3.2.5 Application Context Names

Note 1.— The Application Context describes the ASE/ASO types which are present in the AE, including those aspects not distinguished by ASO type (e.g. version and policy aspects). The abstract syntax of the APDUs and the control function are described here. The Application Context name is an identifier which is used to refer to a defined Application Context. The syntax of the Application Context name is defined in ISO/IEC 8650-1 as an Object Identifier.

Note 2.— The application context name is used here only to distinguish between different versions of an application context within the scope of a given AE type, as identified by the AE Title.

4.3.2.5.1 The Application Context name shall be used to indicate the version and policy aspects relative to the AE with which it is associated.

4.3.2.5.2 Each Application Context shall be assigned an Application Context name.

4.3.2.5.3 Application Context names shall have the following structure:

{iso (1) identified-organisation (3) icao (27) atn-ac (3) version-<n> (n)}
where n is an INTEGER in the range 0..255.

Note.— The value n = 0 is reserved for use by the CF.

4.3.2.6 Presentation Context Identification

Note.— The Null Encoding presentation protocol option has been selected for the most efficient encoding of presentation PDUs, as defined in 4.5. As a consequence, the conventional presentation protocol mechanisms which enable users of the presentation service to distinguish the presentation context of received APDUs are not available. Therefore, an alternative, application layer, mechanism is defined here.

4.3.2.6.1 All User Data which is passed across the presentation service boundary shall be encoded using the unaligned variant of the Packed Encoding Rules (PER) for ASN.1 (ISO/IEC 8825-2).

4.3.2.6.2 When in the data transfer phase, in order to be able to distinguish APDUs which are defined in different abstract syntax modules, the presentation User Data encoding shall assume the Full Encoding option of ISO/IEC 8823-1, augmented with the PER-visible constraints defined in ISO/IEC 8823-1: 1994/Amd. 1: 1997 as follows:

Note 1.— ISO/IEC 8823-1 specifies two choices for the encoding of User-data:

User-data ::= CHOICE {
[APPLICATION 0] IMPLICIT Simply-encoded-data,
[APPLICATION 1] IMPLICIT Fully-encoded-data }

Simply-encoded-data ::= OCTET STRING

```
Fully-encoded-data ::= SEQUENCE SIZE (1, ...) OF PDV-list
-- contains one or more presentation-data-value-list (PDV-list) values
PDV-list ::= SEQUENCE
{ transfer-syntax-name          Transfer-syntax-name OPTIONAL,
 presentation-context-identifier  Presentation-context-identifier,
 presentation-data-values  CHOICE
        { single-ASN1-type      [0] ABSTRACT-SYNTAX.&Type
                (CONSTRAINED BY {
                -- Type corresponding to presentation context identifier -- }) ,
        octet-aligned           [1] IMPLICIT OCTET STRING,
        arbitrary               [2] IMPLICIT BIT STRING }
                        -- contains one or more presentation data values from the same
                                        -- presentation context.
        }

Transfer-syntax-name    ::= OBJECT IDENTIFIER -- not used for ATN Upper Layers

Presentation-context-identifier::= INTEGER (1..127, ... )
```

Note 2.— The use of Full Encoding is specified in order to overcome the fact that: (a) the use of presentation protocol efficiency enhancements removes the ability of presentation layer to perform the necessary demarcation, and (b) the use of ASN.1 Packed Encoding Rules means that it would not have been possible to assign unique ASN.1 tag values to individual APDUs to distinguish between them, as PER does not encode tags.

4.3.2.6.3 Only the presentation-context-identifier and presentation-data-values fields shall be present in the encoded presentation User Data.

4.3.2.6.4 Only the “arbitrary” (BIT STRING) choice for presentation-data-values in the PDV-list SEQUENCE shall be used in the encoded presentation User Data.

4.3.2.6.5 The values of Presentation-context-identifier which are pre-defined in Table 4.3-3 shall be used in the encoding of presentation User Data; the presentation-context-identifiers are not dynamically assigned by the presentation service.

Table 4.3-3. Presentation Context identifiers

Presentation-context-identifier value	Short name	Description
1	acse-apdu	ACSE abstract syntax as defined in ISO/IEC 8650-1
2		reserved for future use
3	user-ase-apdu	abstract syntax as defined in individual ATN application specifications
other		reserved for future use

4.3.2.6.6 With the sole exception of the P-CONNECT service (which is used exclusively by ACSE), upon receiving User Data from the presentation service, the CF shall:

a) decode the Fully-encoded-data and use the presentation-context-identifier value to determine the target ASE.

b) if the target ASE is ACSE, decode the header of the embedded presentation-data-value to determine the APDU type, and

c) if the decoding in a) and b) fails for any reason (presentation-context-identifier not recognised, presentation-data-value does not use the “arbitrary” CHOICE value, or unrecognised APDU type) then issue a P-U-ABORT request to the supporting service and behave as if a P-U-ABORT indication with no parameters has been received;

d) otherwise, pass the presentation-data-value (i.e., acse-apdu or user-ase-apdu) to the target ASE by invoking the appropriate indication or confirmation primitive at the lower ASE service boundary, as specified in 4.3.3.

4.3.2.6.7 Except for P-CONNECT primitives issued by ACSE, when an ASE issues a request or response primitive at its lower service boundary which would otherwise map onto a presentation service primitive, the CF shall:

a) embed the User Data into a Fully-encoded-data type, using the presentation-context-identifier value corresponding to the source ASE

b) pass the Fully-encoded User Data to the presentation service by invoking the appropriate primitive, as specified in 4.3.3.

4.3.3 Control Function Specification

4.3.3.1 ATN-App CF State Definitions

4.3.3.1.1 The ATN-App AE shall behave as if it has a Control Function which can exist only in one of the following states:

a) Null (STA0) — This is the state of the CF when there is no association in existence.

b) Association Pending (STA1) — The CF enters this state either when the ATN-App ASE has made a request to establish a dialogue and is waiting for notification from its peer OR an indication has been received that the peer has made a request to establish a dialogue.

c) Data Transfer (STA2) — The CF enters this state once the establishment phase is complete. An association has successfully been established and the communicating partners are free to send and receive data.

d) Release Pending (STA3) — The CF enters this state when the ATN-App ASE has requested the termination of the dialogue OR an indication has been received that the peer has made a request to terminate the dialogue.

e) Release Collision (STA4) — The CF enters this state when both communicating partners have requested the termination of the dialogue near-simultaneously.

4.3.3.1.2 CF State Table

4.3.3.1.2.1 The ATN-App AE CF shall behave as if it has a control function in accordance with the state table specified in Table 4.3-4, which shows diagrammatically the state transitions and actions performed by the CF in response to incoming events.

Note.— The following conventions are used in Table 4.3-4:

a) Incoming events are shown in the first two columns of the state table, and are enumerated in Table 4.3-6.

b) When an input event occurs and the state table indicates an action, the CF performs that action.

c) Each cell in the state table shows:

1) optionally, one or more predicates, denoted "pN", where N is an integer. The state and action which follow the predicate are only valid if the predicate is TRUE. The inverse (logical NOT) of a predicate is indicated by the prefix "~" (tilde character).

2) the new state that the CF enters after the action has been performed

3) the action, if any, which the CF performs. The possible actions are outlined in Table 4.3-7.

d) Blank cells indicate error conditions.

e) When an input event occurs and the state table indicates a state transition, the CF enters the new state after any associated action has been performed.

4.3.3.1.2.2 For the purpose of specifying CF behaviour, embedded ASEs (ATN-App ASE and ACSE) shall be treated as atomic entities, such that when an input event is invoked by the CF, that event is processed to completion by the ASE and the CF responds to any resulting output events from the ASE, all within the same logical processing thread.

Note.— This provision avoids the need to specify further transient states within the CF. It does not imply any particular implementation architecture.

4.3.3.1.2.3 The following combinations of input events and CF states shall be treated as error conditions:

a) The occurrence of an input event other than those listed in Table 4.3-6; or

b) A combination of input event and CF state which corresponds to a blank cell in Table 4.3-4; or

c) A combination of input event and CF state which corresponds to a cell in Table 4.3-4 containing one or more predicates, none of which evaluates to TRUE.

4.3.3.1.2.4 The error handling shall result in the association being aborted, if one exists, and a notification being given to the Application user.

4.3.3.1.2.5 In the event of a conflict between the actions implied by the state table and the text in the following paragraphs, the text shall take precedence.

Table 4.3-4. ATN-App CF State Table

Event Source	**State-->** **Event**	**STA0** **Null**	**STA1** **Assoc. Pending**	**STA2** **Data Transfer**	**STA3** **Release Pending**	**STA4** **Release Collision**
From ATN-App	ATN-APP function req	STA0 ATN-App ASE req	STA1 ATN-App ASE req	STA2 ATN-App ASE req	STA3 ATN-App ASE req	STA4 ATN-App ASE req
	ATN-APP function rsp	STA0 ATN-App ASE rsp	STA1 ATN-App ASE rsp	STA2 ATN-App ASE rsp	STA3 ATN-App ASE rsp	STA4 ATN-App ASE rsp
From ATN-App ASE (upper)	ATN-APP function ind	STA0 ATN-App ind	STA1 ATN-App ind	STA2 ATN-App ind	STA3 ATN-App ind	STA4 ATN-App ind
	ATN-APP function cnf	STA0 ATN-App cnf	STA1 ATN-App cnf	STA2 ATN-App cnf	STA3 ATN-App cnf	STA4 ATN-App cnf
From ATN-App ASE (lower)	D-START req	p0: STA1 A-ASSOC req				
	D-START rsp+		~p1: STA1 A-ASSOC rsp+			
	D-START rsp-		~p1: STA1 A-ASSOC rsp-			
	D-DATA req			STA2 P-DATA req (User)	~p2: STA3 P-DATA req (User)	
	D-END req			STA3 A-RELEASE req		
	D-END rsp+				~p2: STA3 A-RELEASE rsp+	
	D-END rsp-				~p2: STA3 A-RELEASE rsp-	
	D-ABORT req		STA1 A-ABORT req	STA2 A-ABORT req	STA3 A-ABORT req	STA4 A-ABORT req

Event Source	**State--> Event**	**STA0 Null**	**STA1 Assoc. Pending**	**STA2 Data Transfer**	**STA3 Release Pending**	**STA4 Release Collision**
From ACSE (upper)	A-ASSOCIATE ind		STA1 D-START ind			
	A-ASSOCIATE cnf+		STA2 D-START cnf+			
	A-ASSOCIATE cnf-		STA0 D-START cnf-			
	A-RELEASE ind				STA3 D-END ind	p1: STA4 A-RELEASE rsp+ ~p1: STA4
	A-RELEASE cnf+				STA0 D-END cnf+ P-U-ABORT req	p1: STA0 D-END cnf+ P-U-ABORT req ~p1: STA4 D-END cnf+ A-RELEASE rsp+
	A-RELEASE cnf-				STA2 D-END cnf-	
	A-ABORT ind		STA0 D-ABORT ind	STA0 D-ABORT ind	STA0 D-ABORT ind	STA0 D-ABORT ind
	A-P-ABORT ind		STA0 D-P-ABORT ind	STA0 D-P-ABORT ind	STA0 D-P-ABORT ind	STA0 D-P-ABORT ind
From ACSE (lower)	P-CONNECT req		STA1 P-CONN req			
	P-CONNECT rsp+		STA2 P-CONN rsp+			
	P-CONNECT rsp-		STA0 P-CONN rsp-			
	P-RELEASE req				STA3 P-DATA req (RLRQ)	
	P-RELEASE rsp+				STA0 P-DATA req(RLRE+)	p1: STA4 P-DATA req (RLRE+) ~p1: STA0 P-DATA req (RLRE+)
	P-RELEASE rsp-				STA2 P-DATA req (RLRE-)	
	P-U-ABORT req (data)	STA0 P-U-ABORT req	STA0 P-U-ABORT req	STA0 P-DATA req (ABRT)	STA0 P-U-ABORT req	STA0 P-U-ABORT req
	P-U-ABORT req (no data)		STA0 P-U-ABORT req	STA0 P-U-ABORT req	STA0 P-U-ABORT req	STA0 P-U-ABORT req

Event Source	State--> Event	STA0 Null	STA1 Assoc. Pending	STA2 Data Transfer	STA3 Release Pending	STA4 Release Collision
From supporting service	P-CONNECT ind	p0: STA1 P-CONN ind				
	P-CONNECT cnf+		STA1 P-CONN cnf+			
	P-CONNECT cnf-		STA1 P-CONN cnf-			
	P-DATA ind (RLRQ)	p3: STA0		STA3 P-RELEASE ind	p2: STA4 P-RELEASE ind	
	P-DATA ind (RLRE+)	p3: STA0			STA3 P-RELEASE cnf+	STA4 P-RELEASE cnf+
	P-DATA ind (RLRE-)	p3: STA0			STA3 P-RELEASE cnf-	
	P-DATA ind (ABRT)	p3: STA0 P-U-ABORT req ~p3: STA0		STA2 P-U-ABORT ind P-U-ABORT req	STA3 P-U-ABORT ind P-U-ABORT req	STA4 P-U-ABORT ind P-U-ABORT req
	P-DATA ind (User)	p3: STA0		STA2 D-DATA ind	p2: STA3 D-DATA ind	
	P-U-ABORT ind	STA0	STA1 P-U-ABORT ind	STA2 P-U-ABORT ind	STA3 P-U-ABORT ind	STA4 P-U-ABORT ind
	P-P-ABORT ind	STA0	STA1 P-P-ABORT ind	STA2 P-P-ABORT ind	STA3 P-P-ABORT ind	STA4 P-P-ABORT ind

Table 4.3-5. Predicates used in Table 4.3-4

Predicate	Meaning
p0	This is a new instance of communication, i.e., no previous association exists (effectively, a new AE invocation is created).
p1	This CF is the initiator CF, i.e., the CF which issued an A-ASSOCIATE request primitive.
~p1	This CF is the responder CF, i.e., the CF which received an A-ASSOCIATE indication primitive.
p2	This CF is the Release Initiator, i.e., the CF issued an A-RELEASE request primitive.
~p2	This CF is the Release Responder, i.e., the CF received an A-RELEASE indication primitive.
p3	This CF is the “Abort+Data” initiator, i.e., the CF issued a P-DATA request containing an ABRT APDU and is awaiting disconnection by the peer.
~p3	This CF has not initiated an Abort containing user data.

Table 4.3-6. Incoming Event List

Abbreviated name	Source	Description
ATN-APP function req	upper AE service boundary	Application-specific Request primitive issued by the Application User
ATN-APP function rsp		Application-specific Response primitive issued by the Application User
ATN-APP function ind	ATN-App ASE (upper service boundary)	Application-specific Indication primitive issued by the Application ASE
ATN-APP function cnf		Application-specific Confirmation primitive issued by the Application ASE
D-START req	ATN-App ASE (lower service boundary)	D-START Request primitive issued by DS-User
D-START rsp+		D-START Response primitive issued by DS-User, with Result = accepted
D-START rsp-		D-START Response primitive issued by DS-User, with Result = rejected (transient) or rejected (permanent)
D-DATA req		D-DATA Request primitive issued by DS-User
D-END req		D-END Request primitive issued by DS-User
D-END rsp+		D-END Response primitive issued by DS-User, with Result = accepted
D-END rsp-		D-END Response primitive issued by DS-User, with Result = rejected
D-ABORT req		D-ABORT Request primitive issued by DS-User
A-ASSOCIATE ind	ACSE (upper service boundary)	A-ASSOCIATE Indication primitive issued by ACSE service
A-ASSOCIATE cnf+		A-ASSOCIATE Confirmation primitive issued by ACSE service, with Result = accepted
A-ASSOCIATE cnf-		A-ASSOCIATE Confirmation primitive issued by ACSE service, with Result = rejected (transient) or rejected (permanent)
A-RELEASE ind		A-RELEASE Indication primitive issued by ACSE service
A-RELEASE cnf+		A-RELEASE Confirmation primitive issued by ACSE service, with Result = affirmative
A-RELEASE cnf-		A-RELEASE Confirmation primitive issued by ACSE service, with Result = negative
A-ABORT ind		A-ABORT Indication primitive issued by ACSE service
A-P-ABORT ind		A-P-ABORT Indication primitive issued by ACSE service
P-CONNECT req	ACSE (lower service boundary)	P-CONNECT Request primitive issued by ACSE Protocol Machine (ACPM)
P-CONNECT rsp+		P-CONNECT Response primitive issued by ACPM, with Result = acceptance
P-CONNECT rsp-		P-CONNECT Response primitive issued by ACPM, with Result = user-rejection or provider-rejection
P-RELEASE req		P-RELEASE Request primitive issued by ACPM
P-RELEASE rsp+		P-RELEASE Response primitive issued by ACPM, with Result = affirmative
P-RELEASE rsp-		P-RELEASE Response primitive issued by ACPM, with Result = negative

Abbreviated name	Source	Description
P-U-ABORT req (data)		P-U-ABORT Request primitive issued by ACPM, with the User Data parameter present.
P-U-ABORT req (no data)		P-U-ABORT Request primitive issued by ACPM, with the User Data parameter empty or absent.
P-CONNECT ind	supporting service	P-CONNECT Indication primitive issued by presentation service provider
P-CONNECT cnf+		P-CONNECT Confirmation primitive issued by presentation service provider, with Result = acceptance
P-CONNECT cnf-		P-CONNECT Confirmation primitive issued by presentation service provider, with Result = user-rejection or provider-rejection
P-DATA ind (RLRQ)		P-DATA Indication primitive issued by presentation service provider, with a RLRQ APDU as User-Data
P-DATA ind (RLRE+)		P-DATA Indication primitive issued by presentation service provider, with a RLRE APDU as User-Data, with the reason field set to “normal”
P-DATA ind (RLRE-)		P-DATA Indication primitive issued by presentation service provider, with a RLRE APDU as User-Data, with the reason field set to “not-finished”
P-DATA ind (ABRT)		P-DATA Indication primitive issued by presentation service provider, with an ABRT APDU as User-Data
P-DATA ind (User)		P-DATA Indication primitive issued by presentation service provider, with an ATN-APP APDU (e.g. an ADS-ASE protocol data unit) as User-Data
P-U-ABORT ind		P-U-ABORT Indication primitive issued by presentation service provider
P-P-ABORT ind		P-P-ABORT Indication primitive issued by presentation service provider

Table 4.3-7. Outgoing Event List

Abbreviated name	Target	Description
ATN-App ind	upper AE service boundary	Application-specific Indication primitive mapped transparently from the upper service boundary of the ATN-App ASE.
ATN-App cnf		Application-specific Confirmation primitive mapped transparently from the upper service boundary of the ATN-App ASE.
ATN-App ASE req	upper ATN-App ASE service boundary	Application-specific Request primitive mapped transparently from the upper AE service boundary
ATN-App ASE rsp		Application-specific Response primitive mapped transparently from the upper AE service boundary.
D-START ind	DS-User	D-START Indication primitive issued.
D-START cnf+		D-START Confirmation primitive issued, with the Result parameter set to the abstract value “accepted”
D-START cnf-		D-START Confirmation primitive issued, with the Result parameter set to the abstract value “rejected (transient)” or “rejected (permanent)”, according to the A-ASSOCIATE Confirmation primitive which was received.
D-DATA ind		D-DATA Indication primitive issued.
D-END ind		D-END Indication primitive issued.

Abbreviated name	Target	Description
D-END cnf+		D-END Confirmation primitive issued, with the Result parameter set to the abstract value “accepted”.
D-END cnf-		D-END Confirmation primitive issued, with the Result parameter set to the abstract value “rejected”.
D-ABORT ind		D-ABORT Indication primitive issued.
D-P-ABORT ind		D-P-ABORT Indication primitive issued.
A-ASSOC req	ACSE service provider	A-ASSOCIATE Request primitive issued
A-ASSOC rsp+		A-ASSOCIATE Response primitive issued, with Result = “accepted”
A-ASSOC rsp-		A-ASSOCIATE Response primitive issued, with Result = “rejected (transient)” or “rejected (permanent)”, according to the D-START response primitive which was received.
A-RELEASE req		A-RELEASE Request primitive issued.
A-RELEASE rsp+		A-RELEASE Response primitive issued, with Result = “affirmative” and Reason = “normal”
A-RELEASE rsp-		A-RELEASE Response primitive issued, with Result = “negative” and Reason = “not-finished”
A-ABORT req		A-ABORT Request primitive issued.
P-CONN ind	lower ACSE service boundary	P-CONNECT Indication primitive invoked.
P-CONN cnf+		P-CONNECT Confirmation primitive invoked, with the Result parameter set to “acceptance”.
P-CONN cnf-		P-CONNECT Confirmation primitive invoked, with the Result parameter set to “user-rejection”.
P-RELEASE ind		P-RELEASE Indication primitive invoked.
P-RELEASE cnf+		P-RELEASE Confirmation primitive invoked, with the Result parameter set to “affirmative”.
P-RELEASE cnf-		P-RELEASE Confirmation primitive invoked, with the Result parameter set to “negative”.
P-U-ABORT ind		P-U-ABORT Indication primitive invoked.
P-P-ABORT ind		P-P-ABORT Indication primitive invoked.
P-CONN req	supporting service	P-CONNECT Request primitive issued.
P-CONN rsp+		P-CONNECT Response primitive issued, with the Result parameter set to “acceptance”.
P-CONN rsp-		P-CONNECT Response primitive issued, with the Result parameter set to “user-rejection”.
P-DATA req (RLRQ)		P-DATA Request primitive issued. The User Data parameter contains a RLRQ APDU.
P-DATA req (RLRE+)		P-DATA Request primitive issued. The User Data parameter contains a RLRE APDU, with the reason field set to “normal”.
P-DATA req (RLRE-)		P-DATA Request primitive issued. The User Data parameter contains a RLRE APDU, with the reason field set to “not-finished”.

Abbreviated name	Target	Description
P-DATA req (ABRT)		P-DATA Request primitive issued. The User Data parameter contains an ABRT APDU, with a non-empty user-information field.
P-DATA req (User)		P-DATA Request primitive issued. The User Data parameter contains an ATN-App ASE APDU (e.g. an ADS-ASE protocol data unit)
P-U-ABORT req		P-U-ABORT Request primitive issued.

4.3.3.2 Services Invoked by the Application User

Note 1.— The actions that result from inputs generated by the user of this ATN-App AE (see Figure 4.3-1) are defined here.

Note 2.— The service primitives available to the Application User are specific to the ATN application. This service is detailed in the individual application specifications.

4.3.3.2.1 When Invoked

4.3.3.2.1.1 Invocations of Application User Request and Response primitives by the Application-user shall be allowed when the CF is in any valid state.

4.3.3.2.2 Action Upon Invocation

4.3.3.2.2.1 When the Application User Request or Response primitive is issued, the CF shall:

a) Invoke the equivalent primitive of the ATN-App ASE service, with a one-to-one mapping of parameters; and

b) Remain in its current state.

4.3.3.3 Services Invoked by ATN-App ASE

4.3.3.3.1 ATN-App ASE Indication and Confirmation primitives

4.3.3.3.1.1 When Invoked

4.3.3.3.1.1.1 Invocations of ATN-App ASE Indication and Confirmation primitives by the ATN-App ASE shall be allowed when the CF is in any valid state.

4.3.3.3.1.2 Action Upon Invocation

4.3.3.3.1.2.1 When the ATN-App ASE Indication or Confirmation primitive is issued, the CF shall:

a) Invoke the equivalent primitive of the Application-user service with a one-to-one mapping of parameters; and

b) Remain in its current state.

4.3.3.3.2 D-START Request primitive

4.3.3.3.2.1 When Invoked

4.3.3.3.2.1.1 When the D-START Request primitive is invoked by the ATN-App ASE, a new instance of communication shall be created, with its CF initially in the NULL state.

4.3.3.3.2.2 Action Upon Invocation

4.3.3.3.2.2.1 When the D-START Request is validly invoked, the CF shall:

a) Retrieve the AE-qualifier as defined for the ATN-App AE,

b) Construct the Application Context name, with the value of the final arc set equal to the DS-User Version Number parameter if provided, and set to zero otherwise.

c) Retrieve the calling Presentation address

d) Look up the called Presentation address from the Called Peer Id parameter.

e) If the Calling Peer Id parameter is present, then retrieve the Calling AP Title and Calling AE-qualifier. If it is not present, then Calling AP Title and Calling AE-qualifier are not used in the A-ASSOCIATE request (and they will not then be included in the resulting A-ASSOCIATE-REQUEST (AARQ) APDU).

Note.— The way that the Calling AP Title and the Calling AE-Qualifier are retrieved is a local implementation matter.

f) If the Security Requirements parameter is not present, make no use of the A-ASSOCIATE parameter "ACSE Requirements". If the Security Requirements parameter is present, set the ACSE Requirements parameter to the symbolic value "authentication"; and map the Security Requirements value to the A-ASSOCIATE Authentication-value parameter.

g) Construct an A-ASSOCIATE Request primitive with the following parameters:

Table 4.3-8.

A-ASSOCIATE Request parameter	**ISO Status**	**ATN value**
Mode	U	Not used (default value)
Application Context Name	M	As derived in b) above
Application Context Name List	C	Not used
Calling AP Title	U	As derived in e) above
Calling AE Qualifier	U	As derived in e) above
Calling AP Invocation-identifier	U	Not used
Calling AE Invocation-identifier	U	Not used

A-ASSOCIATE Request parameter	**ISO Status**	**ATN value**
Called AP Title	U	Not used
Called AE Qualifier	U	Not used
Called AP Invocation-identifier	U	Not used
Called AE Invocation-identifier	U	Not used
ACSE Requirements	U	As derived in f) above
Authentication-mechanism Name	U	Not used
Authentication-value	U	As derived in f) above
User Information	U	D-START User Data parameter
Calling Presentation Address	M	Derived as in c) above
Called Presentation Address	M	Derived as in d) above
Presentation Context Definition List	U	Not used
Default Presentation Context Name	U	Not used
Quality of Service	M	See following subsection
Presentation Requirements	U	Not used (default value)
Session Requirements	M	No Orderly Release (NOR), Duplex
Initial Synchronization Point Serial No	C	Not used
Initial Assignment of Tokens	C	Not used
Session-connection Identifier	U	Not used

h) Invoke the A-ASSOCIATE Request primitive

i) Enter the ASSOCIATION PENDING state as an initiator CF.

4.3.3.3.2.3 Quality of Service parameter mappings

Note.— The following paragraphs specify how the Quality of Service parameters in D-START Request and Response primitives are conveyed to the ATN Internet.

4.3.3.3.2.3.1 The Routing Class component of the quality of service parameter in D-START Request and Response primitives shall be conveyed to the ATN Internet and mapped to ATN Security Label by local means, using the values for Security Tag Value specified in Table 5.6-1.

Note.— 5.2.7.3.1 states that the mechanism by which the connection initiator provides the appropriate ATN Security Label is a local matter. For example, it may be identified by an extension to the transport service interface, be implicit in the choice of a given Transport Service Access Point (TSAP), or be identified using a Systems Management function.

4.3.3.3.2.3.2 If no value for Routing Class is specified in the D-START Request primitive, then a default value shall be assigned as follows:

a) If the ATN-App AE is one of the ATS applications specified in 2.1 - 2.4, the value corresponding to “ATSC: No Traffic Type Policy Preference” is assigned;

b) otherwise, the traffic type defaults to General Communications, and no Security Tag Value is conveyed.

4.3.3.3.2.3.3 The Routing Class value conveyed to the ATN Internet when the D-START Response primitive is invoked shall be the same as that which was passed to the DS-User in the D-START Indication primitive.

4.3.3.3.2.3.4 The Priority component of the quality of service parameter in D-START Request and Response primitives shall be provided to the TS-Provider, by implementation-specific means, using the values for “Transport Layer Priority” specified in Table 1-2.

Note.— Although transport priority and network priority are semantically independent of each other, 5.5.1.2 requires that the Transport Service (TS)-user specifies the Application Service Priority, which in turn is mapped into the resulting Connectionless Network Protocol (CLNP) PDUs according to Table 1-2, which defines the fixed relationship between transport priority and the network priority.

4.3.3.3.2.3.5 If no value for Priority is specified in the D-START Request primitive, then the value corresponding to “Network/systems administration” shall be used.

4.3.3.3.2.3.6 The Priority value conveyed when the D-START Response primitive is invoked shall be the same as that which was passed to the DS-User in the D-START Indication primitive.

4.3.3.3.2.3.7 The residual error rate (RER) component of the quality of service parameter in D-START Request and Response primitives shall map to the residual error rate component of the A-ASSOCIATE Quality of Service parameter, and is used to convey requests for the use or non-use of transport checksum to the TS-Provider.

Note.— 5.5.1.2 requires that the TS-User specifies the required residual error rate to determine whether or not the transport checksum is required.

4.3.3.3.2.3.8 If no valid value for RER is specified in the D-START Response primitive, then the value shall be taken to be the same as that which was passed to the DS-User in the D-START Indication primitive.

Note.— If the RER value in the D-START Indication was “high”, then valid values in the response are “low” and “high”. If the RER value in the D-START Indication was “low”, then the only valid value in the response is “low”.

4.3.3.3.3 D-START Response primitive

4.3.3.3.3.1 When Invoked

4.3.3.3.3.1.1 The D-START Response primitive may be validly invoked by the ATN-App ASE when the CF is the responder CF (see 4.3.3.6.1.2.1) and is in the ASSOCIATION PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.3.3.2 Action Upon Invocation

4.3.3.3.3.2.1 When a D-START Response primitive is validly invoked, the CF shall :

a) Construct the Application Context name, with the value of the final arc set equal to the DS-User Version Number parameter if provided, and set to zero otherwise.

b) Retrieve the responding Presentation address

c) If the Security Requirements parameter is not present, make no use of the A-ASSOCIATE parameter “ACSE Requirements”. If the Security Requirements parameter is present, set the ACSE Requirements parameter to the symbolic value “authentication”; and map the Security Requirements value to the A-ASSOCIATE Authentication-value parameter.

d) Construct an A-ASSOCIATE Response primitive with the following parameters:

Table 4.3-9.

A-ASSOCIATE Response parameter	**ISO Status**	**ATN Value**
Application Context Name	M	As derived in a) above
Application Context Name List	C	Not used
Responding AP Title	U	Not used
Responding AE Qualifier	U	Not used
Responding AP Invocation-identifier	U	Not used
Responding AE Invocation-identifier	U	Not used
ACSE Requirements	C	As derived in c) above
Authentication-mechanism Name	U	Not used
Authentication-value	U	As derived in c) above
User Information	U	D-START User Data parameter
Result	M	D-START Result parameter
Diagnostic	U	Not used
Responding Presentation Address	M	Derived as in b) above
Presentation Context Definition Result List	C	Not used

A-ASSOCIATE Response parameter	**ISO Status**	**ATN Value**
Default Presentation Context Result	C	Not used
Quality of Service	M	As for D-START Request (see preceding section)
Presentation Requirements	U	Not used (default value)
Session Requirements	M	No Orderly Release (NOR), Duplex
Initial Synchronization Point Serial No	C	Not used
Initial Assignment of Tokens	C	Not used
Session-connection Identifier	U	Not used

e) If the D-START Response *Result* parameter has the abstract value "accepted", invoke an A-ASSOCIATE Response primitive with the Result parameter set to "accepted", and remain in the ASSOCIATION PENDING state.

f) If the D-START Response *Result* parameter has the abstract value "rejected (permanent)" or "rejected (transient)", invoke an A-ASSOCIATE Response primitive with the Result parameter set to the same abstract value, and remain in the ASSOCIATION PENDING state.

4.3.3.3.4 D-END Request primitive

4.3.3.3.4.1 When Invoked

4.3.3.3.4.1.1 The D-END Request primitive may be validly invoked by the ATN-App ASE when the CF is in the DATA TRANSFER state; if it is in any other state then appropriate error recovery action shall be taken.

Note.— For example, if the CF is in the RELEASE PENDING state, then the D-END Request is rejected locally, with an appropriate result code.

4.3.3.3.4.2 Action Upon Invocation

4.3.3.3.4.2.1 When a D-END Request primitive is validly invoked, the CF shall :

a) Construct an A-RELEASE Request primitive with the following parameter values:

Table 4.3-10.

A- RELEASE Request parameter	**ISO Status**	**ATN Value**
Reason	U	"normal"
User Information	U	D-END User Data parameter

b) Invoke the A-RELEASE Request primitive; and

c) Enter the RELEASE PENDING state as the Release Initiator CF.

4.3.3.3.5 D-END Response primitive

4.3.3.3.5.1 When Invoked

4.3.3.3.5.1.1 The D-END Response primitive may be validly invoked by the ATN-App ASE when the CF is the Release Responder CF and is in the RELEASE PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.3.5.2 Action Upon Invocation

4.3.3.3.5.2.1 When a D-END Response primitive is validly invoked and the *Result* parameter has the value "accepted", the CF shall:

a) Construct an A-RELEASE Response primitive with parameter values as follows:

Table 4.3-11.

A- RELEASE Response parameter	**ISO Status**	**ATN Value**
Reason	U	"normal"
User Information	U	D-END User Data parameter
Result	M	"affirmative"

b) Invoke the A-RELEASE Response primitive

c) Remain in the RELEASE PENDING state.

4.3.3.3.5.2.2 When a D-END Response primitive is validly invoked and the *Result* parameter has the abstract value "rejected" the CF shall:

a) Construct an A-RELEASE Response primitive with parameter values as follows:

Table 4.3-12.

A- RELEASE Response parameter	**ISO Status**	**ATN Value**
Reason	U	"not finished"
User Information	U	D-END User Data parameter
Result	M	"negative"

b) Invoke the A-RELEASE Response primitive; and

c) Remain in the RELEASE PENDING state.

4.3.3.3.6 D-DATA Request primitive

4.3.3.3.6.1 When Invoked

4.3.3.3.6.1.1 The D-DATA Request primitive may be validly invoked by the ATN-App ASE when the CF is in the DATA TRANSFER state, or (if it is the Release Responder) in the RELEASE PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.3.6.2 Action Upon Invocation

4.3.3.3.6.2.1 When a D-DATA Request primitive is validly invoked, the CF shall :

a) Using the definition of presentation-user-data in 4.3.2.6, encode the D-DATA Request User Data parameter with presentation-context-identifier value corresponding to "user-ase-apdu";

b) Invoke a P-DATA Request primitive with the resulting encoding as User Data; and

c) Remain in the same state.

4.3.3.3.7 D-ABORT Request primitive

4.3.3.3.7.1 When Invoked

4.3.3.3.7.1.1 Invocations of the D-ABORT Request primitive by the ATN-App ASE shall be allowed when the CF is in any valid state, except the NULL state; if an invocation occurs when the CF is in the NULL state then an error has occurred (see 4.3.3.1.2.4).

4.3.3.3.7.2 Action Upon Invocation

4.3.3.3.7.2.1 When a D-ABORT Request primitive is validly invoked, the CF shall:

a) If the Originator parameter of the D-ABORT has the symbolic value "User", then set Diagnostic to "No reason given". If the Originator parameter is absent or has any symbolic value other than "User", then set Diagnostic to "Protocol error".

b) Construct an A-ABORT Request primitive with the following parameter values:

Table 4.3-13.

A-ABORT Request parameter	**ISO Status**	**ATN Value**
Diagnostic	U	derived as in a) above
User Information	U	D-ABORT User Data parameter, if present and not empty.

c) Invoke the A-ABORT Request primitive; and

d) Remain in the same state.

4.3.3.4 ACSE Services delivered to the CF

Note.— Events which occur at the upper service boundary of ACSE, i.e. Indication and Confirmation primitives which are generated by the ACPM and which require handling by the CF, are defined here.

4.3.3.4.1 A-ASSOCIATE Indication primitive

4.3.3.4.1.1 When Invoked

4.3.3.4.1.1.1 The A-ASSOCIATE Indication primitive may be validly invoked by the ACSE Protocol Machine (ACPM) when the CF is in the ASSOCIATION PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.4.1.2 Action Upon Invocation

4.3.3.4.1.2.1 When an A-ASSOCIATE Indication primitive is validly invoked, the CF shall:

a) If the final component of the Application Context Name parameter is non-zero, then use it as the DS-User Version Number in the D-START Indication primitive. If it has the value zero, then omit the DS-User Version Number parameter in the D-START Indication.

b) If the Calling AP Title parameter is present, extract the Calling Peer Id from it.

c) If the ACSE Requirements parameter is present, and it indicates that the authentication functional unit is requested, then extract the Authentication-value parameter.

d) Construct a D-START Indication primitive, with the following parameter values:

Table 4.3-14.

D-START Indication parameter	**Value**
Calling Peer ID	Derived as in b) above
DS-User Version Number	Derived as in a) above
Security Requirements	Derived as in c) above
Quality Of Service	See following subsection
User Data	A-ASSOCIATE User Information parameter

e) Invoke the D-START Indication primitive; and

f) Remain in the ASSOCIATION PENDING state.

4.3.3.4.1.3 Quality of Service parameter mappings

Note.— The following paragraphs specify how the Quality of Service parameters in A-ASSOCIATE Indication and Confirmation primitives are conveyed to the DS-User as parameters of the D-START Indication and Confirmation primitives.

4.3.3.4.1.3.1 The Routing Class component of the quality of service parameter in D-START indication and confirmation primitives shall be obtained from the ATN Internet by local means, using the abstract values for Security Tag Values as specified in Table 5.6-1.

4.3.3.4.1.3.2 The Priority component of the quality of service parameter in D-START indication and confirmation primitives shall be taken from information provided by the TS-Provider, by implementation-specific means, using the abstract values for "Transport Layer Priority" specified in Table 1-2.

4.3.3.4.1.3.3 The RER component of the quality of service parameter in D-START indication and confirmation primitives shall be taken from the residual error rate component of the A-ASSOCIATE Quality of Service parameter.

4.3.3.4.2 A-ASSOCIATE Confirmation primitive

4.3.3.4.2.1 When Invoked

4.3.3.4.2.1.1 The A-ASSOCIATE Confirmation primitive may be validly invoked by the ACPM when the CF is in the ASSOCIATION PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.4.2.2 Action Upon Invocation

4.3.3.4.2.2.1 When an A-ASSOCIATE Confirmation primitive is validly invoked, and the *Result* parameter has the abstract value "accepted" the CF shall:

a) If the final component of the Application Context Name parameter is non-zero, then use it as the DS-User Version Number in the D-START Confirmation primitive. If it has the value zero, then omit the DS-User Version Number parameter in the D-START Confirmation.

b) If the ACSE Requirements parameter is present, and it indicates that the authentication functional unit is selected, then extract the Authentication-value parameter.

c) Construct a D-START Confirmation primitive, with parameter values as follows:

Table 4.3-15.

D-START Confirmation parameter	**Value**
DS-User Version Number	Derived as in a) above
Security Requirements	Derived as in b) above
Quality Of Service	As for A-ASSOCIATE Indication (see preceding section)
Result	"accepted"
Reject Source	Not used
User Data	A-ASSOCIATE User Information parameter

d) Invoke the D-START Confirmation primitive

e) Enter the DATA TRANSFER state as the initiator CF.

4.3.3.4.2.2.2 When an A-ASSOCIATE Confirmation primitive is validly invoked, and the *Result* parameter has the abstract value "rejected (permanent)" or "rejected (transient)" the CF shall:

a) If the final component of the Application Context Name parameter is non-zero, then use it as the DS-User Version Number in the D-START Confirmation primitive. If it has the value zero, then omit the DS-User Version Number parameter in the D-START Confirmation primitive.

b) If the ACSE Requirements parameter is present, and it indicates that the authentication functional unit is selected, then extract the Authentication-value parameter.

c) If the A-ASSOCIATE Confirmation *Result Source* parameter has the abstract value "ACSE service-user" form a Reject Source parameter with value "DS user". If the A-ASSOCIATE Confirmation *Result Source* parameter has the abstract value "ACSE service-provider" or "presentation service-provider" form a Reject Source parameter with value "DS provider".

d) Construct a D-START Confirmation primitive, with parameter values as follows:

Table 4.3-16.

D-START Confirmation parameter	**Value**
DS-User Version Number	Derived as in a) above
Security Requirements	Derived as in b) above
Quality Of Service	As for A-ASSOCIATE Indication (see preceding section)
Result	"rejected (permanent)" or "rejected (transient)", from the A-ASSOCIATE Result parameter
Reject Source	Derived as in c) above
User Data	A-ASSOCIATE User Information parameter

e) Invoke the D-START Confirmation primitive

f) Enter the NULL state.

4.3.3.4.3 A-RELEASE Indication primitive

4.3.3.4.3.1 When Invoked

4.3.3.4.3.1.1 The A-RELEASE Indication primitive may be validly invoked by the ACPM when the CF is in the RELEASE PENDING or the RELEASE COLLISION state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.4.3.2 Action Upon Invocation

4.3.3.4.3.2.1 When an A-RELEASE Indication primitive is validly invoked, and the CF is in the RELEASE PENDING state, it shall:

a) Construct a D-END Indication primitive, with the User Data parameter set equal to the value of the User Information parameter of the A-RELEASE Indication primitive.

b) Invoke the D-END Indication

c) Remain in the RELEASE PENDING state.

4.3.3.4.3.2.2 When an A-RELEASE Indication primitive is validly invoked, and the CF is in the RELEASE COLLISION state, and it is the Initiator CF, it shall:

a) Construct a D-END Confirmation primitive, with the User Data parameter set equal to the value of the User Information parameter of the A-RELEASE Indication primitive, if present.

Note.— The D-END Confirmation is not issued to the DS-User until the orderly release procedure is complete, and an A-RELEASE Confirmation is received.

b) Construct an A-RELEASE response primitive with parameter values as follows:

Table 4.3-17.

A- RELEASE Response parameter	**ISO Status**	**ATN Value**
Reason	U	"normal"
User Information	U	Not present
Result	M	"affirmative"

c) Invoke the A-RELEASE Response primitive; and

d) Remain in the RELEASE COLLISION state.

4.3.3.4.3.2.3 When an A-RELEASE Indication primitive is validly invoked, and the CF is in the RELEASE COLLISION state, and it is the Responder CF, it shall:

a) Construct a D-END Confirmation primitive, with the User Data parameter set equal to the value of the User Information parameter of the A-RELEASE Indication primitive, if present.

Note.— The D-END Confirmation is not issued to the DS-User until the orderly release procedure is complete, and an A-RELEASE Confirmation is received.

b) Remain in the RELEASE COLLISION state.

4.3.3.4.4 A-RELEASE Confirmation primitive

4.3.3.4.4.1 When Invoked

4.3.3.4.4.1.1 The A-RELEASE Confirmation primitive may be invoked by the ACPM when the CF is in the RELEASE PENDING or RELEASE COLLISION state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.4.4.2 Action Upon Invocation

4.3.3.4.4.2.1 When an A-RELEASE Confirmation primitive is validly invoked, and the CF is in the RELEASE PENDING state, and the *Result* parameter has the abstract value "affirmative" the CF shall:

a) Construct a D-END Confirmation primitive with the following parameter values.

Table 4.3-18.

D-END Confirmation parameter	**Value**
Result	"affirmative"
User Data	User Information parameter from the A-RELEASE Confirmation, if present

b) Invoke the D-END Confirmation primitive.

c) Issue a P-U-ABORT request primitive, with no parameters.

Note.— This will cause the release of the underlying transport connection.

d) Enter the NULL state.

4.3.3.4.4.2.2 When an A-RELEASE Confirmation primitive is validly invoked, and the CF is in the RELEASE PENDING state, and the *Result* parameter has the abstract value "negative" the CF shall:

a) Construct a D-END Confirmation primitive with the following parameter values.

Table 4.3-19.

D-END Confirmation parameter	**Value**
Result	"rejected"
User Data	User Information parameter from the A-RELEASE Confirmation, if present

b) Invoke the D-END Confirmation primitive.

c) Enter the DATA TRANSFER state.

4.3.3.4.4.2.3 When an A-RELEASE Confirmation primitive is validly invoked, and the *Result* parameter has the abstract value "affirmative", and the CF is in the RELEASE COLLISION state, and it is the Initiator CF, it shall:

a) Issue the D-END Confirmation primitive, which was previously formed in response to the reception of an A-RELEASE Indication primitive, to the DS-User.

b) Issue a P-U-ABORT request primitive, with no parameters.

Note.— This will cause the release of the underlying transport connection.

c) Enter the NULL state.

4.3.3.4.4.2.4 When an A-RELEASE Confirmation primitive is validly invoked, and the *Result* parameter has the abstract value "affirmative", and the CF is in the RELEASE COLLISION state, and it is the Responder CF, it shall:

a) Issue the D-END Confirmation primitive, which was previously formed in response to the reception of an A-RELEASE Indication primitive, to the DS-User.

b) Construct an A-RELEASE Response primitive, with the *Result* parameter set to "affirmative".

c) Invoke the A-RELEASE Response.

d) Remain in the RELEASE COLLISION state.

4.3.3.4.5 A-ABORT Indication primitive

4.3.3.4.5.1 When Invoked

4.3.3.4.5.1.1 Invocations of the A-ABORT Indication primitive by the ACPM shall be allowed when the CF is in any valid state, except the NULL state; if an invocation occurs when the CF is in the NULL state then an error has occurred (see 4.3.3.1.2.4).

4.3.3.4.5.2 Action Upon Invocation

4.3.3.4.5.2.1 When an A-ABORT Indication primitive is validly invoked, the CF shall:

a) If the *Abort Source* parameter of the A-ABORT Indication is set to "ACSE service-user" and the Diagnostic parameter is set to "No reason given", issue a D-ABORT Indication primitive to the DS-User, with the Originator parameter set to "User" and the User Data parameter set equal to the User Information parameter in the A-ABORT Indication, if present.

b) If the *Abort Source* parameter of the A-ABORT Indication is set to "ACSE service-user" and the Diagnostic parameter is absent or is set to any value other than "No reason given", then issue a D-ABORT Indication primitive to the DS-User, with the Originator parameter set to "Provider" and the User Data parameter set equal to the User Information parameter in the A-ABORT Indication, if present.

c) If the *Abort Source* parameter of the A-ABORT Indication has the abstract value "ACSE service-provider", then issue a D-ABORT Indication primitive to the DS-User, with the Originator parameter set to the abstract value "Provider", and the User Data parameter set equal to the User Information parameter in the A-ABORT Indication, if present.

d) Enter the NULL state.

4.3.3.4.6 A-P-ABORT Indication primitive

4.3.3.4.6.1 When Invoked

4.3.3.4.6.1.1 Invocations of the A-P-ABORT Indication primitive by the ACPM shall be allowed when the CF is in any valid state, except the NULL state; if an invocation occurs when the CF is in the NULL state then an error has occurred (see 4.3.3.1.2.4).

4.3.3.4.6.2 Action Upon Invocation

4.3.3.4.6.2.1 When an A-P-ABORT Indication primitive is validly invoked, the CF shall:

a) issue a D-P-ABORT Indication primitive to the DS-User, and discard any Provider Reason parameter in the A-P-ABORT Indication; and

b) Enter the NULL state.

4.3.3.5 Services used by ACSE

Note.— ACSE, edition 2 mandates the mapping of ACSE APDUs to the underlying presentation service provider. However, when the efficient encoding options of Session and Presentation protocols are used, the full Presentation service is no longer available. Therefore, invocations of presentation service primitives by ACSE are "intercepted" by the CF and re-mapped to the "actual" presentation service as appropriate.

4.3.3.5.1 P-CONNECT Request primitive

4.3.3.5.1.1 When Invoked

4.3.3.5.1.1.1 The P-CONNECT Request primitive may be validly invoked by the ACPM when the CF is in the ASSOCIATION PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.5.1.2 Action Upon Invocation

4.3.3.5.1.2.1 When a P-CONNECT Request primitive is validly invoked, the CF shall transparently invoke the equivalent presentation service primitive and remain in the same state.

4.3.3.5.2 P-CONNECT Response primitive

4.3.3.5.2.1 When Invoked

4.3.3.5.2.1.1 The P-CONNECT Response primitive may be validly invoked by the ACPM when the CF is in the ASSOCIATION PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.5.2.2 Action Upon Invocation

4.3.3.5.2.2.1 When the P-CONNECT Response primitive is validly invoked, the CF shall:

a) transparently invoke the equivalent presentation service primitive.

b) If the P-CONNECT Response Result parameter has the abstract value “acceptance” then enter the DATA TRANSFER state, otherwise enter the NULL state.

4.3.3.5.3 P-U-ABORT Request primitive

4.3.3.5.3.1 When Invoked

4.3.3.5.3.1.1 Invocations of the P-U-ABORT Request primitive by the ACPM shall be allowed when the CF is in any valid state.

4.3.3.5.3.2 Action Upon Invocation

4.3.3.5.3.2.1 When a P-U-ABORT Request primitive is validly invoked, the CF shall:

a) If the P-U-ABORT request user data parameter is present, and the CF is in the DATA TRANSFER state:

1) Encode the presentation user data as indicated in 4.3.2 with the P-U-ABORT user data parameter (an ABRT APDU) as the presentation data value and presentation context identifier value corresponding to “acse-apdu”.

2) Invoke a P-DATA Request primitive with the resulting encoding as User Data.

b) Otherwise, invoke a P-U-ABORT Request primitive with no parameters.

Note.— This will cause the underlying transport connection to be disconnected.

c) Enter the NULL state.

4.3.3.5.4 P-RELEASE Request primitive

Note.— ACSE, edition 2 mandates the mapping of A-RELEASE APDUs (RLRQ and RLRE) to the P-RELEASE service. However, when the efficient encoding options of Session and Presentation protocols are used, the Session No-Orderly Release (NOR) functional unit is selected, and no mapping for the P-RELEASE service is available. In order to provide an orderly release service, the CF re-maps invocations of the P-RELEASE service at the lower service boundary of ACSE to invocations of the P-DATA service, with the release APDUs transferred as user information.

4.3.3.5.4.1 When Invoked

4.3.3.5.4.1.1 The P-RELEASE Request primitive may be validly invoked by the ACPM when the CF is in the RELEASE PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.5.4.2 Action Upon Invocation

4.3.3.5.4.2.1 When a P-RELEASE Request primitive is validly invoked, the CF shall:

a) Encode the presentation user data as indicated in 4.3.2.6 with the P-RELEASE user data parameter (a RLRQ APDU) as the presentation data value and presentation context identifier corresponding to “acse-apdu”.

b) Invoke a P-DATA Request primitive with the resulting encoding as User Data; and

c) Remain in the RELEASE PENDING state.

4.3.3.5.5 P-RELEASE Response primitive

4.3.3.5.5.1 When Invoked

4.3.3.5.5.1.1 The P-RELEASE Response primitive may be validly invoked by the ACPM when the CF is in the RELEASE PENDING or RELEASE COLLISION state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.5.5.2 Action Upon Invocation

4.3.3.5.5.2.1 When a P-RELEASE Response primitive is validly invoked, and the CF is in the RELEASE PENDING state, and the *Result* parameter has the abstract value “affirmative” the CF shall:

a) encode the presentation user data as indicated in 4.3.2 with the P-RELEASE user data parameter (a RLRE APDU) as the presentation data value and presentation context identifier corresponding to “acse-apdu”.

b) Invoke a P-DATA Request primitive with the resulting encoding as User Data.

c) Enter the NULL state.

Note.— The peer AEI is now expected to issue a P-U-ABORT request, which will cause the release of the underlying connection.

4.3.3.5.5.2.2 When a P-RELEASE Response primitive is validly invoked, and the CF is in the RELEASE PENDING state, and the *Result* parameter has the abstract value "negative" the CF shall:

a) Encode the presentation user data as indicated in 4.3.2 with the P-RELEASE user data parameter (a RLRE APDU) as the presentation data value and presentation context identifier corresponding to "acse-apdu";

b) Invoke a P-DATA Request primitive with the resulting encoding as User Data; and

c) Enter the DATA TRANSFER state.

4.3.3.5.5.2.3 When a P-RELEASE Response primitive is validly invoked, and the CF is in the RELEASE COLLISION state, and it is the Initiator CF, it shall:

a) Encode the presentation user data as indicated in 4.3.2 with the P-RELEASE user data parameter (a RLRE APDU) as the presentation data value and presentation context identifier corresponding to "acse-apdu";

b) Invoke a P-DATA Request primitive with the resulting encoding as User Data; and

c) Remain in the RELEASE COLLISION state.

4.3.3.5.5.2.4 When a P-RELEASE Response primitive is validly invoked, and the CF is in the RELEASE COLLISION state, and it is the Responder CF, it shall:

a) Encode the presentation user data as indicated in 4.3.2 with the P-RELEASE user data parameter (a RLRE APDU) as the presentation data value and presentation context identifier corresponding to "acse-apdu";

b) Invoke a P-DATA Request primitive with the resulting encoding as User Data; and

c) Enter the NULL state.

Note.— The peer AEI is now expected to issue a P-U-ABORT request, which will cause the release of the underlying connection.

4.3.3.5.5.2.5 **Recommendation. -** *After entering the NULL state, implementations should release the underlying connection (e.g., by issuing the P-U-ABORT request) if the communication peer does not cause the connection to be released as expected, after a period of time not less than twice the anticipated end-to-end transit time.*

4.3.3.6 Supporting Services delivered to the CF

Note 1.— The mapping by the CF of presentation service indication and confirmation primitives, which are invoked by the presentation service provider, is defined in the following paragraphs.

Note 2.— The following provisions describe the behaviour to be exhibited by the ATN-App AE when the supporting communications service exhibits behaviour modelled by the passing of indication or confirmation primitives to the application layer.

4.3.3.6.1 P-CONNECT Indication primitive

4.3.3.6.1.1 When Invoked

4.3.3.6.1.1.1 When the P-CONNECT Indication primitive is invoked by the supporting service, a new instance of communication shall be created, with its CF initially in the NULL state.

4.3.3.6.1.2 Action Upon Invocation

4.3.3.6.1.2.1 When a P-CONNECT Indication primitive is validly invoked, the CF shall:

a) transparently invoke the equivalent presentation service primitive at the lower ACSE service boundary; and

b) enter the ASSOCIATION PENDING state as the responder CF.

4.3.3.6.2 P-CONNECT Confirmation primitive

4.3.3.6.2.1 When Invoked

4.3.3.6.2.1.1 The P-CONNECT Confirmation primitive may be validly invoked by the supporting service when the CF is in the ASSOCIATION PENDING state; if it is in any other state then appropriate error recovery action shall be taken.

4.3.3.6.2.2 Action Upon Invocation

4.3.3.6.2.2.1 When a P-CONNECT Confirmation primitive is validly invoked, the CF shall :

a) transparently invoke the equivalent presentation service primitive at the lower ACSE service boundary; and

b) Remain in the ASSOCIATION PENDING state.

4.3.3.6.3 P-U-ABORT Indication primitive

4.3.3.6.3.1 When Invoked

4.3.3.6.3.1.1 Invocations of the P-U-ABORT Indication primitive by the supporting service shall be allowed when the CF is in any valid state.

4.3.3.6.3.2 Action Upon Invocation

4.3.3.6.3.2.1 When a P-U-ABORT Indication primitive is validly invoked, the CF shall

a) if the CF is in the NULL state, take no action;

b) otherwise, transparently invoke the equivalent presentation service primitive at the lower ACSE service boundary, and remain in the same state.

4.3.3.6.4 P-P-ABORT Indication primitive

4.3.3.6.4.1 When Invoked

4.3.3.6.4.1.1 Invocations of the P-P-ABORT Indication primitive by the supporting service shall be allowed when the CF is in any valid state.

4.3.3.6.4.2 Action Upon Invocation

4.3.3.6.4.2.1 When a P-P-ABORT Indication primitive is validly invoked, the CF shall:

a) if the CF is in the NULL state, then take no action;

b) otherwise, transparently invoke the corresponding presentation service primitive at the lower ACSE service boundary; and

c) remain in the same state.

4.3.3.6.5 P-DATA Indication primitive

4.3.3.6.5.1 When Invoked

4.3.3.6.5.1.1 Invocations of the P-DATA Indication primitive by the supporting service shall be allowed when the CF is in a valid state to receive the decoded APDU, as listed in 4.3.3.6.5.2; if an invocation occurs when the CF is not in a valid state then an error has occurred (see 4.3.3.1.2.4).

4.3.3.6.5.2 Action Upon Invocation

4.3.3.6.5.2.1 When a P-DATA Indication primitive is validly invoked, the CF shall decode the presentation user data as indicated in 4.3.2 to determine the destination ASE of the APDU, and extract the presentation data value.

Note.— The destination ASE is determined from the value of the presentation-context-identifier in the received User-data. Valid values are acse-apdu and user-ase-apdu, which correspond to destination ASEs of ACSE and ATN-App ASE, respectively.

4.3.3.6.5.2.2 ACSE APDU Received

4.3.3.6.5.2.2.1 If the destination ASE is ACSE then the CF shall determine the type of ACSE APDU present in the extracted presentation data value.

Note.— ACSE APDUs which may validly be received in a P-DATA indication are A-Release-Request (RLRQ), A-Release-Response (RLRE), and A-Abort (ABRT) APDUs.

4.3.3.6.5.2.2.2 If the received APDU is RLRQ, the CF shall:

a) if in the DATA TRANSFER state, then invoke a P-RELEASE Indication primitive at the ACSE lower service boundary, with the RLRQ as User Data, and enter the RELEASE PENDING state as the Release Responder CF;

b) if in the RELEASE PENDING state, and the CF is the Release Initiator, then invoke a P-RELEASE Indication primitive at the ACSE lower service boundary with the RLRQ as User Data, and enter the RELEASE COLLISION state;

c) if in the NULL state, and this CF has previously issued an ABRT APDU and is awaiting disconnection by the peer, then take no action and remain in the NULL state;

d) if none of the conditions a) to c) is satisfied, then take error handling action as described in 4.3.3.6.5.2.4.

4.3.3.6.5.2.2.3 If the received APDU is RLRE, the CF shall:

a) if the Reason field in the RLRE has the value “not-finished”, and the CF is in the RELEASE PENDING state, then invoke a P-RELEASE Confirmation primitive at the ACSE lower service boundary, with the result parameter set to “negative”, and the RLRE as User Data; remain in the RELEASE PENDING state;

b) if the Reason field in the RLRE has the value “normal”, and the CF is in the RELEASE PENDING or RELEASE COLLISION state, then invoke a P-RELEASE Confirmation primitive at the ACSE lower service boundary, with the result parameter set to “affirmative”, and the RLRE as User Data; remain in the same state;

c) if the CF is in the NULL state, and this CF has previously issued an ABRT APDU and is awaiting disconnection by the peer, then take no action and remain in the NULL state;

d) if none of the conditions a) to c) is satisfied, then take error handling action in 4.3.3.6.5.2.4.

4.3.3.6.5.2.2.4 If the received APDU is ABRT, the CF shall:

a) if the CF is in the state DATA TRANSFER, or RELEASE PENDING, or RELEASE COLLISION, then invoke a P-U-ABORT Indication primitive at the ACSE lower service boundary, with the ABRT as User Data, and issue a P-U-ABORT request with no parameters to the underlying service; remain in the same state;

b) if the CF is in the NULL state, then take no action unless this CF has previously issued an ABRT APDU and is awaiting disconnection by the peer, in which case issue a P-U-ABORT request to the underlying service; remain in the same state;

c) if neither of the conditions a) and b) is satisfied, then take error handling action as described in 4.3.3.6.5.2.4.

4.3.3.6.5.2.3 ATN-App APDU Received

4.3.3.6.5.2.3.1 If the destination ASE is ATN-App ASE, then the CF shall:

a) if the CF is in the DATA TRANSFER state, or the CF is in the RELEASE PENDING state and is the Release Initiator CF, then issue a D-DATA Indication primitive to the DS-User, with the received presentation data value as the user data parameter, and remain in the same state;

b) if the CF is in the NULL state, and this CF has previously issued an ABRT APDU and is awaiting disconnection by the peer, then take no action and remain in the same state;

c) if neither of the conditions a) and b) is satisfied, then take error handling action as described in 4.3.3.6.5.2.4.

4.3.3.6.5.2.4 Error conditions

4.3.3.6.5.2.4.1 If the destination ASE is invalid (i.e. neither ACSE nor ATN-App ASE), or an unrecognised APDU is received, or a valid APDU is received when the CF is not in the correct state (as defined in 4.3.3.6.5.2.2 and 4.3.3.6.5.2.3), then the CF shall:

a) if not in the NULL state then issue a P-U-ABORT request with no parameters to the supporting service: and

b) regardless of CF state behave as if a P-U-ABORT indication had been received.

4.4 SESSION LAYER REQUIREMENTS

Note.— The session layer requirements are described in many cases by means of completed protocol implementation conformance statement (PICS) proforma tables. In such tables, the "Ref." column contains a reference to the relevant section in the session layer PICS proforma, ISO/IEC 8327-2.

4.4.1 Protocol versions implemented

4.4.1.1 Session protocol versions shall be supported as specified in Table 4.4-1.

Table 4.4-1. Session Protocol Versions Supported

Ref.	Version	ISO Status	ATN Support
S.A.3/1	Version 1	O.1	-
S.A.3/2	Version 2	O.1	M

O.1: The ISO PICS requires that the implementation of one, and only one, version of the protocol is described.

4.4.2 Session Functional units

4.4.2.1 Session functional units (S-FUs) shall be selected as specified in Table 4.4-2.

Table 4.4-2. Selection of Session functional units

Ref.	Functional unit	ISO Status	ATN Support
S.A.6.1/1	Kernel	M	M
S.A.6.1/2	Negotiated release	O	X
S.A.6.1/3	Half Duplex (HD)	O.2	X
S.A.6.1/4	Duplex	O.2	M
S.A.6.1/5	Expedited Data (EX)	O	X
S.A.6.1/6	Typed Data	O	X
S.A.6.1/7	Capability Data Exchange	C1	X
S.A.6.1/8	Minor Synchronize (SY)	O	X
S.A.6.1/9	Symmetric Synchronize (SS)	O	X
S.A.6.1/10	Data Separation	C2	X
S.A.6.1/11	Major Synchronize	O	X
S.A.6.1/12	Resynchronise	O	X
S.A.6.1/13	Exceptions	C3	X
S.A.6.1/14	Activity Management (ACT)	O	X
See note	No-orderly release (NOR)	O	M
See note	Special User-data	O	X

Note.— Functional units added by efficiency enhancement amendment.

O.2: The ISO standard requires at least one of the functional units Duplex and Half Duplex to be implemented.

C1:if [S-FU(ACT)] then O else N/A

C2:if [S-FU(SY) or S-FU(SS)] then O else N/A

C3:if [S-FU(HD)] then O else N/A

4.4.3 Protocol mechanisms

4.4.3.1 Session protocol mechanisms shall be supported as specified in Table 4.4-3.

Table 4.4-3. Session Protocol Mechanisms Supported

Ref.	Mechanism	ISO Status	ATN Support	Associated mnemonic
S.A.6.2/1	Use of transport expedited data (Extended control Quality of Service)	C4	X	S-EXP_T
S.A.6.2/2	Reuse of transport connection	O	O	S-REUSE_T
S.A.6.2/3	Basic concatenation	M	M (Note 2)	
S.A.6.2/4	Extended concatenation (sending)	O	X	
S.A.6.2/5	Extended concatenation (receiving)	O	X	S-XCONC_RCV
S.A.6.2/6	Segmenting (sending)	O	X	S-SEG_SDR
S.A.6.2/7	Segmenting (receiving)	O	X	S-SEG_RCV
S.A.6.2/8	Max. size of SS-user-data (S-CONNECT) > 512	O	O	S-MAXSIZE_512
S.A.6.2/9	Max. size of SS-user-data (S-CONNECT) > 10240	O	O	S-MAXSIZE_10240
S.A.6.2/10	Max. size of SS-user-data (S-ABORT) >9	O	X	S-MAXSIZE_9
See note 1	Null-encoding protocol option	-	M	
See note 1	Short-connect protocol option	-	M	
See note 1	Short-encoding protocol option	-	X	

Note 1.— Protocol options added by efficiency enhancement amendment.

Note 2.— Only Category 1 SPDUs are used for this ATN profile. By definition, these are never concatenated. Therefore, Basic concatenation is not applicable to this specification, but is supported to the extent necessary for compliance with the ISO PICS.

C4:if [S-FU(EX)] then M else O

4.4.3.2 The session protocol shall implement the efficiency enhancements in ISO/IEC 8327-1:1996/Amd. 1:1997 as specified, together with all approved amendments and defect report resolutions.

4.4.3.3 If the null encoding protocol option is offered by the initiating Session Protocol machine (SPM), the responding SPM shall select only the kernel, full-duplex and no-orderly release functional units for use on this connection.

4.4.3.4 Session Protocol Data Units (SPDUs) associated with the Short-connect protocol option (i.e. Short Connect (SCN), Short Accept (SAC), Short Accept Continue (SACC), Short Refuse (SRF) and Short Refuse Continue (SRFC)) shall be transferred as User-data on the Transport layer T-CONNECT primitives, where possible.

Note.— This is only possible if the complete SPDUs, including any User-data, meet any size restrictions of the T-CONNECT User-data.

4.4.4 Supported Roles

4.4.4.1 Session Connection

4.4.4.1.1 The roles for Session Connection shall be supported as specified in Table 4.4-4.

Table 4.4-4. Session Connection Roles Supported

Ref.	Role	ISO Status	ATN Support	Mnemonic
S.A.7.1.1.1/1	Connection initiator	O.3	M	S-CON_initiator
S.A.7.1.1.1/2	Connection responder	O.3	M	S-CON_responder

O.3: The ISO standard requires a conforming implementation to support at least one of these roles as required by the implementation.

4.4.4.1.2 When a connection establishment request is accepted, the SHORT-CPA PPDU in the SS-User-data of the positive S-CONNECT response/confirmation primitive shall map to the User-data parameter of a SAC SPDU.

4.4.4.1.3 When a connection establishment request is refused, the SHORT-CPR PPDU in the SS-User-data of the negative S-CONNECT response/confirmation primitive shall map to the User-data parameter of a SRF SPDU.

4.4.4.2 Orderly release

4.4.4.2.1 The roles for Session Orderly Release shall be supported as specified in Table 4.4-5.

Table 4.4-5. Session Orderly Release Roles Supported

Ref.	Role	ISO Status	ATN Support	Mnemonic
S.A.7.1.1.2/1	Requestor	O.4	N/A (See note)	S-REL_requestor
S.A.7.1.1.2/2	Acceptor	O.4	N/A (See note)	S-REL_acceptor

O.4: The ISO standard requires a conforming implementation to support at least one of these roles as part of the Kernel functional unit. However, selection of the No Orderly Release functional unit removes this requirement.

Note.— Not applicable, as the No Orderly Release functional unit is selected. For ATN applications, orderly release is provided by the CF as described in 4.3.

4.4.4.3 Normal Data Transfer

4.4.4.3.1 The roles for Session Normal Data Transfer shall be supported as specified in Table 4.4-6.

Table 4.4-6. Session Normal Data Transfer Roles Supported

Ref.	Role	ISO Status	ATN Support	Mnemonic
S.A.7.1.1.3/1	Requestor	O.5	M	S-DATA_requestor
S.A.7.1.1.3/2	Acceptor	O.5	M	S-DATA_acceptor

O.5: The ISO standard requires a conforming implementation to support at least one of these roles.

4.4.5 Supported SPDUs

Note.— This section specifies the SPDUs associated with the supported Session functional units. There are no additional SPDUs associated with the Duplex functional unit, or with the No Orderly Release functional unit.

4.4.5.1 Support for the SPDUs associated with the Kernel functional unit

4.4.5.1.1 Support for SPDUs shall be as specified in Table 4.4-7.

Table 4.4-7. Supported Session Protocol Data Units

		Sender		**Receiver**		
Ref.	**SPDU**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**	**Mnemonics**
S.A.7.1.2/1	Connect (CN)	C5	N/A (Note 4)	C6	N/A (Note 4)	
S.A.7.1.2/2	Overflow Accept (OA)	C7	N/A (Note 4)	C8	N/A (Note 4)	S-OA_SDR / S-OA_RCV
S.A.7.1.2/3	Connect Data Overflow (CDO)	C9	N/A (Note 4)	C10	N/A (Note 4)	S-CDO_SDR / S-CDO_RCV
S.A.7.1.2/4	Accept (AC)	C6	N/A (Note 4)	C5	N/A (Note 4)	
S.A.7.1.2/5	Refuse (RF)	C6	N/A (Note 4)	C5	N/A (Note 4)	
S.A.7.1.2/6	Finish (FN)	C11	N/A (Note 2)	C12	N/A (Note 2)	
S.A.7.1.2/7	Disconnect (DN)	C12	N/A (Note 2)	C11	N/A (Note 2)	
S.A.7.1.2/8	Abort	M	N/A (Note 3)	M	N/A (Note 3)	
S.A.7.1.2/9	Abort Accept (AA)	O	N/A (Note 3)	M	N/A (Note 3)	
S.A.7.1.2/1 0	Data Transfer (DT)	C13	N/A (Note 3)	C14	N/A (Note 3)	
S.A.7.1.2/1 1	Prepare (PR)	C15	X	C15	X	S-PR_SDR / S-PR_RCV
See note 1	Short Connect (SCN)	C17	M	C17	M	
See note 1	Short Accept (SAC)	C17	M	C17	M	
See note 1	Short Refuse (SRF)	C17	M	C17	M	

		Sender		Receiver		
Ref.	**SPDU**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**	**Mnemonics**
See note 1	Null (NL)	C18	M	C18	M	
See note 1	Short Connect Continue (SCNC)	C16	N/A	C16	N/A	
See note 1	Short Accept Continue (SACC)	C17	M	C17	M	
See note 1	Short Refuse Continue (SRFC)	C17	M	C17	M	
See note 1	Short Finish (SFN)	C16	N/A	C16	N/A	
See note 1	Short Disconnect (SDN)	C16	N/A	C16	N/A	
See note 1	Short Data Transfer (SDT)	C16	N/A	C16	N/A	
See note 1	Short Abort (SAB)	C16	N/A	C16	N/A	

Note 1.— PDUs defined in efficiency enhancement amendment.

Note 2.— Not applicable, as the no-orderly-release functional unit is selected.

Note 3.— Not applicable, as the null-encoding protocol option is selected.

Note 4.— Not applicable, as the short-connect protocol option is selected.

C5: if [S-CON_initiator] then M else N/A
C6: if [S-CON_responder] then M else N/A
C7: if [S-V1 or (NOT S-CON_responder)] then N/A else if [S-MAXSIZE_10240] then M else O
C8: if [NOT S-V1 and S-CON_responder and S-MAXSIZE_10240] then M else N/A
C9: if [S-V1 or (NOT S-CON_initiator)] then N/A else if [S-MAXSIZE_10240] then M else O
C10: if [NOT S-V1 and S-CON_initiator and S-MAXSIZE_10240] then M else N/A
C11: if [S-REL_requestor] then M else N/A
C12: if [S-REL_acceptor] then M else N/A
C13: if [S-DATA_requestor] then M else N/A
C14: if [S-DATA_acceptor] then M else N/A
C15: if [NOT S-V1 and S-MAXSIZE_9 and S-EXP_T] then M else N/A
C16: used only if the short-encoding protocol option is selected.
C17: used if short-encoding or null-encoding is used.
C18: used only if the null-encoding protocol option is supported.

4.4.5.1.2 SCN, SAC, SRF, SACC and SRFC SPDUs shall be encoded such that the parameter bit of the SI&P octet is set to the value 0, indicating that all following octets are User-information (i.e. no SPDU parameters are present).

Note.— This is a requirement of the null-encoding protocol option.

4.4.5.2 Support for the SPDUs associated with Token Exchange

4.4.5.2.1 Support for Session protocol data units associated with Token exchange shall be as specified in Table 4.4-8.

Table 4.4-8. SPDUs associated with Token Exchange

		Sender		**Receiver**	
Ref.	**SPDU**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**
S.A.7.1.3/1	Give Tokens (GT)	M	- (See note 2)	M	- (See note 2)
S.A.7.1.3/2	Please Tokens (PT)	M	- (See note 2)	M	- (See note 2)

Note 1.— The ISO PICS states that these two SPDUs are used for Token Exchange, but they are also used as category 0 SPDUs in basic concatenation. Therefore, their implementation is mandatory even if no token is supported (reference ISO/IEC 8327-1 clauses 7.16 and 7.17). However, if the null-encoding protocol option is selected, their encoding will be null, i.e. not present.

Note 2.— Not applicable, as the null-encoding protocol option is selected.

4.4.6 Use of null-encoding and short-connect protocol options

4.4.6.1 The null-encoding and short-connect session protocol options shall be selected for use, with the requirements as specified in Table 4.4-9.

Table 4.4-9. Use of the null encoding and short-connect Session protocol options

Ref.	Requirement	Base Status	ATN Requirement
a	The calling and called session selectors are null	C1	M
b	The session-requirements parameter in the S-CONNECT service includes the kernel, full-duplex and no-orderly-release functional units only.	C1	M

C1: The SPMs may use the short-connect protocol option to establish a session connection using the null-encoding option. The null-encoding protocol option is available for use on an established connection only if the conditions a and b in Table 4.4-9 are both true.

4.4.7 Mapping to the ATN Internet Transport Service

4.4.7.1 The use of the connection-oriented transport service provided by the ATN Internet shall be as specified in Clause 6 of ISO/IEC 8327-1, except as stated in this section.

4.4.7.2 The called and calling Transport Service Access Point (TSAP) address shall be provided to the TS-Provider on a per Transport Connection basis, using the called and calling Presentation Service Access Point (PSAP) addresses as provided to ACSE in the A-ASSOCIATE request, with null presentation and session selectors.

4.4.7.3 The TS-user shall indicate in all T-CONNECT requests that the transport expedited flow is not required.

4.4.7.4 Information on the use or non-use of the transport checksum shall be conveyed between the TS-User and TS-Provider via the "residual error rate" component of the T-CONNECT quality of service parameter.

Note 1.— 5.5.1.2 requires that the TS-user specifies the required residual error rate to determine whether or not the transport checksum is required. In the ATN, the Quality of Service provided to applications is maintained using capacity planning techniques that are outside of the scope of this specification. Network administrators are responsible for designing and implementing a network that will meet the QOS requirements of the CNS/ATM applications that use it.

Note 2.— If the TS-User requests the use of checksum (RER = "low") in the request primitive, the peer can only accept the use of checksum for this Transport Connection. If the TS-User proposes non-use of checksum (RER = "high") in the request primitive, the peer can either accept the non-use of checksum or force the use of checksum for this Transport Connection.

4.4.7.5 The use or non-use of the transport checksum shall be negotiated by the TS-provider on a per Transport Connection basis, based on TS-User requests in the T-CONNECT request and response primitives, as follows:

a) If the required residual error rate in the T-CONNECT request has the abstract value "low", then the TS-provider uses best endeavours to obtain the lowest available residual error rate, including the use of the transport checksum in all Transport Protocol Data Units (TPDUs). The residual error rate in the T-CONNECT indication is set to the abstract value "low", and the responder can only accept this value in the T-CONNECT response.

b) If the required residual error rate in the T-CONNECT request has the abstract value "high", then the TS-provider proposes non-use of the transport checksum. The residual error rate in the T-CONNECT indication is set to the abstract value "high", and the responder can either accept this value, or request "low" in the T-CONNECT response. In the former case, transport checksum is not used, and in the latter case the TS-provider uses the transport checksum for all TPDUs.

4.4.7.6 The Application Service Priority shall be provided to the TS-Provider on a per Transport Connection basis, by implementation-specific means, and using the values for "Transport Layer Priority" specified in Table 1-2.

Note.— Although transport priority and network priority are semantically independent of each other, it is required (in 5.5.1.2), that the TS-user specifies the Application Service Priority, which in turn is mapped into the resulting CLNP PDUs according to Table 1-2, which defines the fixed relationship between transport priority and the network priority.

4.4.7.7 The ATN Security Label shall be provided to the TS-Provider on a per Transport Connection basis.

4.4.7.8 The ATN Security Label value shall be encoded according to 5.6.2.2.2.2 b), and passed between TS-User and TS-Provider by implementation-specific means.

4.4.7.9 The QOS parameter "Routing Class" shall be conveyed as the Security Tag field of the security tag set for Traffic Type and Associated Routing Policies within the ATN Security Label.

Note 1.— 5.2.7.3.1 states: "The mechanism by which the [transport] connection initiator provides the appropriate ATN Security Label is a local matter. For example, it may be identified by an extension to the transport service interface, be implicit in the choice of a given TSAP, or be identified using a Systems Management function."

Note 2.— 5.5.1.2 requires the TS-User to provide the ATN Security Label as specified in Figure 5.6-1 and 5.6.2.2.2.2 b). The encoding of the ATN Security Label is summarised below. The D-START QOS parameter "Routing Class" maps to the field labelled "Traffic Type & category".

ATN Security Label field	***Value (Hex)***	***Length (Octets)***
Security Registration ID Length	*6*	*1*
Security Registration ID = OID {1.3.27.0.0}	*06, 04, 2B, 1B, 00, 00*	*6*
Security Information Length	*4*	*1*
Security information:		
Tag Set Name Length	*1*	*1*
Tag Set Name = "Traffic Type & Associated Routing Policies"	*0F*	*1*
Tag Set Length	*1*	*1*
Security Tag Value = Traffic Type & category (from Table 5.6-1)	*01 (for example)*	*1*
	Total:	*12 Octets*

4.4.7.10 No Transport Service quality of service parameters other than those specified in the preceding subsections shall be specified when establishing a transport connection.

4.5 PRESENTATION LAYER REQUIREMENTS

Note.— The presentation layer requirements are described in many cases by means of completed protocol implementation conformance statement (PICS) proforma tables. In such tables, the "Ref." column contains a reference to the relevant section in the presentation layer PICS proforma ISO/IEC 8823-2.

4.5.1 Protocol mechanisms

4.5.1.1 The Presentation protocol mechanisms supported shall be as specified in Table 4.5-1.

Table 4.5-1. Presentation Protocol Mechanisms Supported

Ref.	Protocol Mechanism	ISO Status	ATN Support	Mnemonic
P.A.6.1/2	Normal mode	O.1	M	
P.A.6.1/1	X.410-1984 mode	O.1	X	
See note	Nominated context	O	N/A	
See note	Short encoding	O	N/A	
See note	Packed encoding rules	O	N/A	
See note	Short-connect	O	M	
See note	Null-encoding	O	M	

Note.— Optional protocol mechanisms defined in efficiency enhancement amendment.

O.1: The ISO standard requires that either Normal mode or X.410 (1984) mode or both be supported.

4.5.1.2 The presentation protocol shall implement the efficiency enhancements in ISO/IEC 8823-1: 1994/Amd. 1: 1997 as specified, together with all approved amendments and defect report resolutions.

4.5.2 Use of null-encoding and short-connect protocol options

4.5.2.1 The null-encoding and short-connect presentation protocol options shall be selected for use, with the requirements as specified in Table 4.5-2.

Table 4.5-2. Use of the null encoding and short-connect Presentation protocol options

Ref.	Requirement	Base Status	ATN Requirement
a	The presentation context definition list contains precisely one item in which the abstract syntax is known to the responding Presentation Protocol Machine (PPM) by bilateral agreement.	C1	N/A
b	The presentation context definition list is empty and the default context is known by bilateral agreement	C1	M
c	The presentation context definition list is empty and the abstract syntax of the default context is known to the responding PPM by bilateral agreement and is specified in ASN.1	C1	M
d	The calling and called presentation selectors are null	C2	M
e	The presentation-requirements parameter in the P-CONNECT service includes the kernel functional unit only.	C2	M

C1: The null-encoding protocol option is available for use on an established connection only if at least one of the conditions a, b and c in Table 4.5-2 is true.

C2: The short-connect protocol option is used only in connection establishment to establish a connection on which the null-encoding option will be used; it can only be used if both of the conditions d and e in Table 4.5-2 is true.

4.5.3 Mapping of Presentation Primitives to the Null Encoding option

Note.— When the null-encoding presentation protocol option is selected, no presentation protocol control information is present once the connection has been established. Thus, no presentation PDUs are supported. The presentation connection is only terminated by the termination of the supporting session and transport connections.

4.5.3.1 The user of the presentation service shall not issue any presentation primitives other than P-CONNECT request, P-CONNECT response, P-DATA request and P-U-ABORT request.

4.5.3.2 When it is required to release the presentation connection, the presentation service user shall issue a P-U-ABORT request.

4.5.3.3 Any user data in a P-U-ABORT request shall be ignored by the presentation service provider.

4.5.4 Functional units

4.5.4.1 The Presentation functional units selected shall be as specified in Table 4.5-3.

Table 4.5-3. Selection of Presentation functional units

Ref.	Presentation functional unit	ISO Status	ATN Support	Mnemonic
P.A.6.2/1	Kernel	M	M	
P.A.6.2/2	Presentation Context Management	O	X	P-FU(CM)
P.A.6.2/3	Presentation Context Restoration	C1	X	P-FU(CR)

C1: if Presentation Context Management (2) is supported then O else N/A

4.5.4.2 The Presentation pass-through functional units selected shall be as specified in Table 4.5-4.

Table 4.5-4. Selection of Presentation pass-through functional units

Ref.	Pass-through to Session functional units	ISO Status	ATN Support	Mnemonic
P.A.6.2/4	Negotiated release	O	X	S-FU(NR)
P.A.6.2/5	Half Duplex	O.2	X	S-FU(HD)
P.A.6.2/6	Duplex	O.2	M	S-FU(FD)
P.A.6.2/7	Expedited Data	O	X	S-FU(EX)
P.A.6.2/8	Typed Data	O	X	S-FU(TD)
P.A.6.2/9	Capability Data Exchange	C1	X	S-FU(CD)
P.A.6.2/10	Minor Synchronize	O	X	S-FU(SY)
P.A.6.2/11	Symmetric Synchronize	O	X	S-FU(SS)
P.A.6.2/12	Data Separation	O	X	S-FU(DS)
P.A.6.2/13	Major Synchronize	O	X	S-FU(MA)

Ref.	Pass-through to Session functional units	ISO Status	ATN Support	Mnemonic
P.A.6.2/14	Resynchronise	O	X	S-FU(RESYNC)
P.A.6.2/15	Exceptions	C2	X	S-FU(EXCEP)
P.A.6.2/16	Activity Management	O	X	S-FU(ACT)
See note	No-orderly release (NOR)	-	M	S-FU(NOR)

Note.— The NOR Session functional unit is defined in the ISO Session service efficiency enhancement amendment.

O.2: The ISO standard requires that pass-through for at least one of the Session functional units Duplex and Half Duplex be supported.

C1: if [S-FU(ACT) then O else N/A

C2: if [S-FU(HD) then O else N/A

4.5.5 Elements of procedure

4.5.5.1 Supported roles

4.5.5.1.1 Presentation Connection

4.5.5.1.1.1 The supported roles for establishing Presentation connections shall be as specified in Table 4.5-5.

Table 4.5-5. Presentation Connection roles

Ref.	Role	ISO Status	ATN Support	Mnemonic
P.A.7.1.1.1/1	Initiator	O.3	M	P-CON_initiator
P.A.7.1.1.1/2	Responder	O.3	M	P-CON_responder

O.3: The ISO standard requires a conforming implementation to support at least one of these roles.

4.5.5.1.1.2 When a connection establishment request is accepted, the AARE (accepted) in the User-data of the positive P-CONNECT response/confirmation primitive shall map to the User-data parameter of a SHORT-CPA PPDU.

4.5.5.1.1.3 When a connection establishment request is refused, the AARE (rejected) in the User-data of the negative P-CONNECT response/confirmation primitive shall map to the User-data parameter of a SHORT-CPR PPDU.

4.5.5.1.2 Orderly release

4.5.5.1.2.1 The supported roles for the orderly release of Presentation connections shall be as specified in Table 4.5-6.

Table 4.5-6. Presentation Connection orderly release roles

Ref.	Role	ISO Status	ATN Support	Mnemonic
P.A.7.1.1.3/1	Requestor	O	N/A	P-REL_requestor
P.A.7.1.1.3/2	Acceptor	O	N/A	P-REL_acceptor

4.5.5.1.3 Normal Data

4.5.5.1.3.1 The supported roles for Normal Data shall be as specified in Table 4.5-7.

Table 4.5-7. Presentation Normal Data roles

Ref.	Role	ISO Status	ATN Support	Mnemonic
P.A.7.1.1.2/1	Requestor	O	M	P-DATA_requestor
P.A.7.1.1.2/2	Acceptor	O	M	P-DATA_acceptor

4.5.6 Supported Presentation Protocol Data Units (PPDUs)

Note.— This section specifies the PPDUs associated with the supported Presentation functional units. There are no additional PPDUs or additional pass-through functionality associated with the supported Session functional units.

4.5.6.1 Supported PPDUs associated with the Kernel services

The Presentation Protocol Data Units supported shall be as specified in Table 4.5-8.

Table 4.5-8. Supported Presentation Protocol Data Units

		Sender		**Receiver**		
Ref.	**PPDU**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**	**Mnemonics**
P.A.7.1.2/1	Connect presentation (CP)	C3	N/A (Note 2)	C4	N/A (Note 2)	
P.A.7.1.2/2	Connect presentation accept (CPA)	C4	N/A (Note 2)	C3	N/A (Note 2)	S-OA_SDR / S-OA_RCV
P.A.7.1.2/3	Connect presentation reject (CPR)	C4	N/A (Note 2)	C3	N/A (Note 2)	S-CDO_SDR / S-CDO_RCV
P.A.7.1.2/4	Abnormal release provider (ARP)	M	N/A (Note 2)	M	N/A (Note 2)	
P.A.7.1.2/5	Abnormal release user (ARU)	O	N/A (Note 2)	M	N/A (Note 2)	
P.A.7.1.2/6	Presentation Data (TD)	C5	N/A (Note 2)	C6	N/A (Note 2)	
Note 1	Short Connect (SHORT-CP)	O	M	O	M	
Note 1	Short Connect Accept (SHORT-CPA)	O	M	O	M	
Note 1	Short Connect Reject (SHORT-CPR)	O	M	O	M	

Note 1.— PDUs defined in efficiency enhancement amendment.

Note 2.— PPDUs not applicable, as the short-connect and null-encoding protocol options are selected.

C3: if [P-CON_initiator] then M else N/A
C4: if [P-CON_responder] then M else N/A
C5: if [P-DATA_requestor] then M else N/A
C6: if [P-DATA_acceptor] then M else N/A

4.5.6.2 Structure and encoding of PPDUs

4.5.6.2.1 The SHORT-CP, SHORT-CPA and SHORT-CPR PPDUs shall have the encoding-choice bit-field set to “unaligned PER”.

4.6 ACSE SPECIFICATION

Note.— The ACSE requirements are described in many cases by means of completed protocol implementation conformance statement (PICS) proforma tables. In such tables, the "Ref." column contains a reference to the relevant section in the ACSE PICS proforma ISO/IEC 8650-2.

4.6.1 Protocol details

4.6.1.1 The specification of the ACSE protocol supported shall be as defined in Table 4.6-1.

Table 4.6-1. Identification of ACSE Protocol Specification

Identification of Protocol Specification	ATN Support	Comments
ISO/IEC 8650-1:1995	M	See note

Note.— This is the second edition of the ACSE protocol specification.

4.6.2 Protocol versions

4.6.2.1 The version of the ACSE protocol supported shall be as specified in Table 4.6-2.

Table 4.6-2. Identification of ACSE Protocol version

Ref.		**ISO Status**	**ATN Support**	**Mnemonic**
A.A.4.2/1	Version 1	O.1	M	A-V1
A.A.4.2/2	Version 2	O.1		

O.1: The ISO PICS requires support of the implementation of only one version of the protocol to be described.

4.6.3 Supported roles

4.6.3.1 Association establishment

4.6.3.1.1 The supported roles for Association Establishment shall be as specified in Table 4.6-3.

Table 4.6-3. ACSE Roles for Association Establishment

Ref.	Capability	ISO Status	ATN Support	Mnemonic
A.A.6.1/1	Association initiator	O.2	See text	A-CON_initiator
A.A.6.1/2	Association responder	O.2	See text	A-CON_responder

O.2: The ISO standard requires a conforming implementation to support at least one of the roles.

4.6.3.1.2 Either one or both of the ACSE roles “Association initiator” or “Association responder” shall be supported.

4.6.3.2 Normal Release procedure

4.6.3.2.1 The supported roles for the Normal Release procedure shall be as specified in Table 4.6-4.

Table 4.6-4. ACSE Roles for Normal Release

Ref.	Role	ISO Status	ATN Support	Mnemonic
A.A.6.2/1	Initiator	O	See text	A-REL_requestor
A.A.6.2/2	Responder	O	See text	A-REL_acceptor

4.6.3.2.2 Either one or both of the ACSE Normal Release roles “Initiator” or “Responder” shall be supported.

4.6.3.2.3 The ACSE Release Responder shall be allowed to give a negative response, despite the fact that the session Negotiated Release functional unit is not selected for the association.

Note.— The above provision waives the ISO/IEC 8649 requirement that the Responder may give a negative response only if session Negotiated Release is selected. This is possible because, for ATN, the ACSE release PDUs do not map directly to the Presentation release service; they are re-mapped by the CF to P-DATA.

4.6.3.3 Abnormal Release procedure

4.6.3.3.1 The supported roles for the Abnormal Release procedure shall be as specified in Table 4.6-5.

Table 4.6-5. ACSE Roles for Abnormal Release

Ref.	Role	ISO Status	ATN Support	Mnemonic
A.A.6.3/1	Initiator	M	M	
A.A.6.3/2	Responder	M	M	

4.6.4 Protocol mechanisms

4.6.4.1 The ACSE protocol mechanisms supported shall be as specified in Table 4.6-6.

Table 4.6-6. ACSE Protocol Mechanisms Supported

Ref.	Protocol Mechanism	ISO Status	ATN Support	Mnemonic
A.A.7/1	Normal mode	O.4	M	
A.A.7/2	X.410-1984 mode	O.4	X	
A.A.7/2	Rules for extensibility	M	M	
A.A.7/4	Supports operation of Session version 2	O	M	S-O-SESS-V2

O.4: The ISO standard requires either Normal mode or X.410-1984 mode or both to be supported.

4.6.4.2 Extensibility and Encoding

4.6.4.2.1 For the purposes of this specification, the abstract syntax module defined in clause 9 of the ACSE protocol specification shall be augmented with the ASN.1 extensibility notation, as specified in ISO/IEC 8650-1: 1996/Amd. 1: 1997

4.6.4.2.2 The system shall support that encoding which results from applying the ASN.1 packed encoding rules (basic, unaligned variant), as specified in ISO/IEC 8825-2, to the abstract syntax module specified in 4.6.4.2.1.

4.6.4.2.3 Packed encoding (basic, unaligned) shall be used for encoding all ACSE Protocol Control Information (PCI) for interchange.

4.6.5 ACSE Functional units

4.6.5.1 The ACSE functional units selected shall be as specified in Table 4.6-7.

Table 4.6-7. Selection of ACSE Functional Units

Ref.	Role	ISO Status	ATN Support	Mnemonic
A.A.8/1	Normal mode	M	M	
A.A.8/2	Authentication	O	C1	A-FU(AU)

C1: If the Dialogue Service user requires the use of the Security Requirements parameter of the D-START primitives, then M, else O.

4.6.6 Supported APDUs

4.6.6.1 The ACSE Protocol data units supported shall be as specified in Table 4.6-8.

Table 4.6-8 Supported ACSE Protocol Data Units

Ref.	APDU	Sender		Receiver		Comment
		ISO Status	ATN Support	ISO Status	ATN Support	
A.A.9/ 1	AARQ	C1	M	C2	M	
A.A.9/ 2	AARE	C2	M	C1	M	
A.A.9/ 3	RLRQ	C3	M	C4	M	
A.A.9/ 4	RLRE	C4	M	C3	M	
A.A.9/ 5	ABRT	C5	M	C5	M	

C1: if [A-CON_initiator] then M else N/A
C2: if [A-CON_responder] then M else N/A
C3: if [A-REL_requestor] then M else N/A
C4: if [A-REL_acceptor] then M else N/A
C5: if [S-O-SESS-V2] then M else N/A

4.6.6.2 Supported APDU parameters

4.6.6.2.1 A-Associate-request (AARQ)

4.6.6.2.1.1 The parameters in the AARQ APDU shall be supported as specified in Table 4.6-9.

Table 4.6-9. Supported AARQ Parameters

		Sender		Receiver	
Ref.	Parameter	ISO Status	ATN Support	ISO Status	ATN Support
A.A.10.1/1	Protocol Version	C6	X	C2	M
A.A.10.1/2	Application Context Name	C1	M	C2	M
A.A.10.1/3	Calling AP title	C6	M	C2	M
A.A.10.1/4	Calling AE qualifier	C6	M	C2	M
A.A.10.1/5	Calling AP invocation-identifier	C6	X	C2	M
A.A.10.1/6	Calling AE invocation-identifier	C6	X	C2	M

		Sender		Receiver	
Ref.	**Parameter**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**
A.A.10.1/7	Called AP title	C6	X	C2	M
A.A.10.1/8	Called AE qualifier	C6	X	C2	M
A.A.10.1/9	Called AP invocation-identifier	C6	X	C2	See text
A.A.10.1/10	Called AE invocation-identifier	C6	X	C2	See text
A.A.10.1/11	ACSE-requirements	C8	See text	C9	M
A.A.10.1/12	Authentication-mechanism name	C8	X	C9	N/A
A.A.10.1/13	Authentication-value	C8	See text	C9	M
A.A.10.1/14	Implementation information	C6	X	C7	O
A.A.10.1/15	User information	C6	M	C7	M

C1: if [A-CON_initiator] then M else N/A
C2: if [A-CON_responder] then M else N/A
C6: if [A-CON_initiator] then O else N/A
C7: if [A-CON_responder] then O else N/A
C8: if [A-CON_initiator and A-FU(AU)] then M else N/A
C9: if [A-CON_responder and A-FU(AU)] then M else N/A

4.6.6.2.1.2 The AARQ parameters “ACSE-Requirements” and “Authentication-value” shall be supported, for sending, only if the connection initiator role (A-CON_initiator) and the Authentication functional unit (A-FU(AU)) are supported.

Note.— The ATN specification is non-conformant to the ISO PICS proforma, in that the “Authentication-mechanism-name” parameter is not supported for sending.

4.6.6.2.1.3 The AARQ parameters “ACSE-Requirements” and “Authentication-value” shall be supported for receiving if the connection responder role (A-CON_responder) is supported, but are ignored if the Authentication functional unit (A-FU(AU)) is not supported by the responder.

Note.— The ATN specification is non-conformant to the ISO PICS proforma, in that the “Authentication-mechanism-name” parameter is “N/A” for receiving, if the Authentication functional unit is selected, and “ACSE-requirements” and “Authentication-value” are “M” for receiving, even if the Authentication functional unit is not supported.

4.6.6.2.1.4 The AARQ parameters “Called AP invocation-identifier” and “Called AE invocation-identifier” shall be supported, for receiving, if the Association Responder role is supported.

4.6.6.2.2 A-Associate-response (AARE)

4.6.6.2.2.1 The parameters in the AARE APDU shall be supported as specified in Table 4.6-10.

Table 4.6-10. Supported AARE Parameters

		Sender		Receiver	
Ref.	**Parameter**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**
A.A.10.2/1	Protocol Version	C7	X	C1	M
A.A.10.2/2	Application Context Name	C2	M	C1	M
A.A.10.2/3	Responding AP title	C7	X	C1	M
A.A.10.2/4	Responding AE qualifier	C7	X	C1	M
A.A.10.2/5	Responding AP invocation-identifier	C7	X	C1	M
A.A.10.2/6	Responding AE invocation-identifier	C7	X	C1	M
A.A.10.2/7	Result	C2	M	C1	M
A.A.10.2/8	Result source - diagnostic	C10	M	C11	M
A.A.10.2/9	ACSE-requirements	C9	See text	C8	See text
A.A.10.2/10	Authentication-mechanism name	C9	X	C8	N/A
A.A.10.2/11	Authentication-value	C9	See text	C8	See text
A.A.10.2/12	Implementation information	C7	X	C6	O
A.A.10.2/13	User information	C7	M	C6	M

C1: if [A-CON_initiator] then M else N/A

C2: if [A-CON_responder] then M else N/A

C6: if [A-CON_initiator] then O else N/A

C7: if [A-CON_responder] then O else N/A

C8: if [A-CON_initiator and A-FU(AU)] then M else N/A

C9: if [A-CON_responder and A-FU(AU)] then M else N/A

C10: if [A-CON_responder] then (if [A-FU(AU)] then M (with a value range of 0 to 14) else M (with a value range of 0 to 10) else N/A

C11: if [A-CON_initiator] then (if [A-FU(AU)] then M (with a value range of 0 to 14) else M (with a value range of 0 to 10) else N/A

4.6.6.2.2.2 The AARE parameters “ACSE-Requirements”, Authentication-mechanism-name” and “Authentication-value” shall be supported, for sending, only if the connection responder role (A-CON_responder) and the Authentication functional unit (A-FU(AU)) are supported.

4.6.6.2.2.3 The AARE parameters "ACSE-Requirements" and "Authentication-value" shall be supported, for receiving, only if the connection initiator role (A-CON_initiator) and the Authentication functional unit (A-FU(AU)) are supported.

4.6.6.2.3 A-Release-request (RLRQ)

4.6.6.2.3.1 The parameters in the RLRQ APDU shall be supported as specified in Table 4.6-11.

Table 4.6-11. Supported RLRQ Parameters

		Sender		Receiver	
Ref.	**Parameter**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**
A.A.10.3/1	Reason	C12	M	C4	M
A.A.10.3/2	User information	C12	M	C4	M

C4: if [A-REL_acceptor] then M else N/A
C12: if [A-REL_requestor] then O else N/A

4.6.6.2.4 A-Release-response (RLRE)

4.6.6.2.4.1 The parameters in the RLRE APDU shall be supported as specified in Table 4.6-12.

Table 4.6-12 Supported RLRE Parameters

		Sender		Receiver	
Ref.	**Parameter**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**
A.A.10.4/1	Reason	C13	M	C3	M
A.A.10.4/2	User information	C13	M	C3	M

C3: if [A-REL_requestor] then M else N/A
C13: if [A-REL_acceptor] then O else N/A

4.6.6.2.5 A-Abort (ABRT)

4.6.6.2.5.1 The parameters in the ABRT APDU shall be supported as specified in Table 4.6-13.

Table 4.6-13. Supported ABRT Parameters

		Sender		Receiver	
Ref.	**Parameter**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**
A.A.10.5/1	Abort source	M	M	M	M
A.A.10.5/2	Diagnostic	C14	M	C14	M
A.A.10.5/3	User information	O	M	M	M

C14: if [A-FU(AU)] then M else N/A

4.6.6.3 Supported parameter forms

4.6.6.3.1 AE title name form

4.6.6.3.1.1 The Application Entity Title parameter shall be supported in the forms specified in Table 4.6-14.

Table 4.6-14. AE Title Name Form

		Sender		Receiver	
Ref.	**Syntax form**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**
A.A.11.1/1	Form 1 (Directory name)	O.5	X	M	O
A.A.11.1/2	Form 2 (Object identifier and integer)	O.5	M	M	M

O.5: The ISO standard requires a conforming implementation to support at least one of the forms.

4.6.6.3.2 Authentication value form

4.6.6.3.2.1 The Authentication value parameter shall be supported in the forms specified in Table 4.6-15.

Table 4.6-15. Authentication Value Form

Prerequisite: A-FU(AU)

		Sender		Receiver	
Ref.	**Authentication value form**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**
A.A.11.2/1	GraphicString	O.6	See text	C14	M
A.A.11.2/2	BIT STRING	O.6	See text	C14	M
A.A.11.3/3	EXTERNAL	O.6	See text	C14	M

		Sender		Receiver	
Ref.	**Authentication value form**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**
A.A.11.4/4	Other	O.6	X	C14	N/A

O.6: The ISO standard requires a conforming implementation to support at least one of the forms.
C14: if [A-FU(AU)] then M else N/A

4.6.6.3.2.2 If the authentication functional unit is supported, at least one of the Authentication-value forms listed in Table 4.6-15 shall be implemented for sending.

4.6.6.3.3 User information form

4.6.6.3.3.1 User information reference

4.6.6.3.3.1.1 The User information parameter shall use the forms of reference specified in Table 4.6-16.

Table 4.6-16. User information reference

		Sender		Receiver	
Ref.	**Parameter**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**
	direct-reference	O	X	M	N/A
	indirect-reference	O	O (See Note)	M	M
	data-value-descriptor	O	X	M	N/A

Note.— Indirect-reference contains a presentation-context-id value as specified in Table 4.3-3 when the single-ASN-1-type encoding form is used, and is absent otherwise.

4.6.6.3.3.2 User information encoding type

4.6.6.3.3.2.1 The User information parameter encoding choice shall be as specified in Table 4.6-17.

Table 4.6-17. User information encoding choice

		Sender		Receiver	
Ref.	**Parameter**	**ISO Status**	**ATN Support**	**ISO Status**	**ATN Support**
	single-ASN1-type	O	O	M	M
	octet-aligned	O	X	M	N/A
	arbitrary	O	O	M	M

4.6.7 Mapping to the Presentation Service

4.6.7.1 The mapping of ACSE APDUs and parameters to presentation service primitives shall be performed by the CF as specified in 4.3, which takes precedence over the direct mapping defined in clause 8 of ISO/IEC 8650-1.