EUROCONTROL

ATN PROJECT

ATN End Systems

(COM-4)

Specification for Generic ATN Communication Service

Abstract

This document contains the Specification and draft Guidance Material for a Generic ATN Communication Service (GACS), which provides a transparent end-to-end data transfer service between two users. Quality of Service and recipient addressing parameters are specified on a per-message basis, and the underlying communications service can be either connection-oriented or connectionless.

File Ref. : DED6/TC6/T07/DEL/D05V0_F.DOC

Author. : A J Kerr

Version No. : 0.F

Date of Issue : 28-Oct-1998

COPYRIGHT NOTICE

APPROVAL

Authority	Date	Signature
Name Title		
Name Title		
Name Title		

DOCUMENT CONTROL LOG

Section(s)	Date	Amendment Number	Reason for change and/or Change Note Number.
All	12/01/98	0.A	Initial draft
All	21/01/98	0.B	Second draft
All	26/02/98	0.C	Revised after review. SAM-SEND renamed SAM-TRANSFER. Message type parameter added. PDU definitions added. Chapter 2 added.
All	12/05/98	0.D	SAM renamed GACS. Specification restructured after review by ATNP/WG3/SG3. ASO approach added in addition to AE approach.
All	15/09/98	0.E	Specification updated after review in WG3/SG3, taking STNA comments into account. Former 1.3 (GACS Realisation) moved to 1.1 (Introduction). Detailed protocol description added.
1.3	28/10/98	0.F	Post ATNP/WG3 Bordeaux. Details of mapping to PDU fields tabulated and completed in protocol description.

Version	Originating Company	Author	Contract	Task
0.A, 0.B, 0.C, 0.D, 0.E, 0.F	Level-7/ECsoft	A J Kerr	TC6	T07

KEYWORDS

ATN, SAM, GACS

DISTRIBUTION LIST

Copy Number	Document Holder	Location
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		

TABLE OF CONTENTS

1. GENERIC ATN COMMUNICATION SERVICE SPECIFICATION

1.1 Introduction

1.1.1 Scope and Structure of Specification

The Generic ATN Communication service (GACS) defined here allows a user of the service to transfer data transparently across the ATN to another user (or to multiple users). The user is able to specify the required quality of service (QoS) and recipient addressing parameters on a per-message basis.

This specification is designed to optimise the use of communications bandwidth, and consequently uses the Dialogue Service defined in [ULCS] 4.2, including extensions for unit data and presentation address handling services. The Dialogue Service in turn uses the ATN Transport service defined in [ICS] 5.5.

The user is able to select the required level of service, which in turn results in the use of either a connection-oriented (CO) or connectionless (CL) supporting protocol stack.

Note.— This specification is structured as follows:

1.1: INTRODUCTION contains the purpose and structure of the GACS Specification, and a background to the functionality defined herein. Two possible architectures for implementing the GACS protocol are outlined.

1.2: GACS SERVICE DEFINITION describes the abstract service provided by the GACS specification.

1.3: PROTOCOL DEFINITION contains the formal definition of messages exchanged by GACS-ASEs using Abstract Syntax Notation One (ASN.1), and describes the exchanges of messages allowed by the GACS protocol.

1.4: COMMUNICATION REQUIREMENTS contains the requirements that the GACS ASE application imposes on the underlying communication system.

1.5: USER REQUIREMENTS outlines the requirements that a user of a GACS ASE must meet.

1.6: SUBSETTING RULES contains the conformance requirements which all implementations of the GACS protocol obey.

1.1.2 Conventions

Note 1.— For a given primitive, the presence of each parameter is described by one of the following values in the parameter tables of the service primitive descriptions:

a) blank not present;

b) C conditional upon some predicate explained in the text;

c) C(=) conditional upon the value of the parameter to the immediate left being present, and equal to that value;

d) M mandatory;

e) M(=) mandatory, and equal to the value of the parameter to the immediate left;

f) U user option.

Note 2.— The following abbreviations are used in this document:

a) Req request; an invocation of a service primitive initiated from a GACS user to the GACS service;

b) Ind indication; an invocation of a service primitive delivered from the GACS service to a GACS user;

c) Cnf confirmation; an invocation of a service primitive delivered from the GACS service to a GACS user, which confirms that a previous request primitive from that user has been acted upon by the GACS service;

Note 3.— An unconfirmed service results in one message being transmitted, in one direction.

Note 4.— A confirmed service provides end-to-end confirmation that a message sent by one user was received by its peer user.

1.1.3 Generic ATN Communication Service Overview

Note 1.— The service defined in this specification provides a generic data transfer capability. To allow maximum flexibility and optimal use of data link bandwidth, the following basic services are defined;

a) G-TRANSFER,

b) G-TRANSFER-CONFIRMED,

c) G-END,

d) G-MULTICAST.

Note 2.— The G-TRANSFER service

The G-TRANSFER service allows a user to send a single message to a specified destination (or multiple destinations) via the ATN internet. A number of options are defined:

a) Connectionless Mode. If the user does not require a resilient communications service (e.g. because the message is not mission-critical, or because the user application itself implements an error recovery protocol) then this can be requested per message. In this case, a connectionless (CL) protocol stack, if available, will be used to transfer the message, provided the size constraints of the CL stack are not exceeded.

b) Connection-Oriented Mode. If the user does require a resilient communications service (e.g. because the message is mission-critical, and the user application itself does not implement an error recovery protocol) then this can be requested per message. In this case, a connection-oriented (CO) protocol stack will be used to transfer the message.

c) Multi-shot Option. If the user intends to send multiple messages with the same Quality of Service (QoS) requirements to the same destination(s), then it can optionally request a "multi-shot" mode. This establishes and maintains a communication relationship with the specified peer(s), and provides an optimised use of the communications link, using a CO protocol stack.

Note 3.— The G-TRANSFER-CONFIRMED service

The G-TRANSFER-CONFIRMED service allows a user to send a single message to a specified destination (or multiple destinations) and to receive confirmation that the message was delivered to the remote user application(s).

The confirmed service supports the same single message and multi-shot options as the G-TRANSFER service.

Note 4.— The G-END service

The G-END service is an optional service which allows a user of the multi-shot option to inform the GACS service that a communications relationship with the specified peer(s) is no longer required to be maintained. This allows an orderly freeing of resources, and an assurance that there are no messages in transit to or from that particular peer(s).

If G-END is not used, then any established communications relationship between two peers will automatically be ended by the GACS service on expiry of a configurable inactivity timer.

Note 5.— The G-MULTICAST service

The G-MULTICAST service is an optional local service which provides users with access to the multicast capabilities of the connectionless transport service, so that a user can receive messages addressed to a group address.

Note 6.— In the G-TRANSFER and G-TRANSFER-CONFIRMED services, messages can optionally be sent to a specified Presentation address. This provision is only supported if the appropriate Upper Layer naming and addressing enhancements to the Dialogue service are also supported.

1.1.4 GACS Realisation

The GACS service provision can be realised alternatively as an "Application Layer message protocol" or as a "simple generic service". The two approaches are very different and have different fields of applicability.

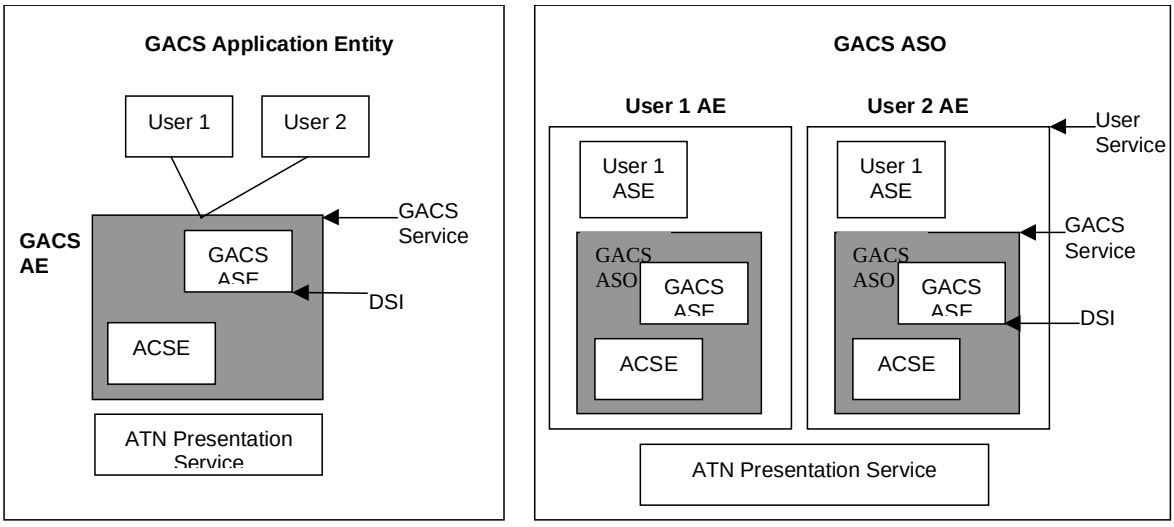

Figure 1.1-1. GACS Application versus GACS ASO

The two architectures, as illustrated in Figure 1.1-1 are:

a) GACS can be realised as an **ATN Application Entity** (AE) providing an ATN access point to existing (e.g. ACARS-based) and future applications which are not specified to use the defined ATN upper layer architecture.

b) GACS can be realised as an **ATN Application Service Object** (ASO) or "service" providing an enhanced dialogue service in the ATN Upper Layers Architecture to future air-ground and ground-ground ATN applications (ATC and AOC).

The GACS AE approach is appropriate for the migration of existing applications. However, this approach is not the only approach for applications to use the ATN. The GACS ASO (enhanced dialogue service) approach would be preferred for any new ATC or AOC application.

1.1.5 The GACS Application Entity

This would be a distinct ATN application installed in aircraft and ground systems acting as a point of access to the ATN. An ATN address is allocated to the GACS AE. The GACS application is identified by an OBJECT IDENTIFIER such as:

"iso(1) identified-organisation (3) icao (27) atn-end-system-air (1) 24-bit-address (x) ops (0) gacs (y)".

It would be natural to implement the GACS service as an application programming interface (API), providing a communications interface to user applications.

Several GACS-Users could use services from the same GACS Application. The GACS Application therefore multiplexes data supplied by GACS-Users over the same dialogue when the intended recipient and the requested communication characteristics are identical. The CM Application is used to exchange the address and version number of the GACS application in the air-ground environment.

GACS-Users in this approach are not considered as fully integrated ATN applications; they have no distinct ATN names and no ATN addresses. CM is not used to negotiate version numbers for the users. Specific mechanisms need to be implemented to switch the incoming data to the relevant GACS-User, based on the message-type field.

The GACS Application itself does not know anything about the message contents, or the encoding rules for these messages. Typical communication functions, such as sequence numbering and request/reply correlation are entirely the responsibility of the GACS-User applications.

1.1.6 The GACS Application Service Object

The GACS ASO is defined to provide new ASEs with an enhanced service within the AE. New ASEs can be developed over either the Dialogue service or the GACS service, depending upon their requirements.

In this framework, any new application is considered as a fully integrated ATN application identified by a specific ATN address and an OBJECT IDENTIFIER such as:

"iso(1) identified-organisation (3) icao (27) atn-end-system-air (1) 24-bit-address (x) ops (0) aoc1 (y)".

GACS itself is not identified as an ATN application in this configuration. CM is used to assess the application capability of the peer system and to exchange dynamically the addressing information of these applications in the air-ground environment.

This architecture is completely in line with the ATN ULA.

ASEs can be privately defined. The ASE protocol and the format of the data exchanged by AOC ASEs for example does not need to be standardised by ICAO or disclosed externally. These ASEs are "black boxes" for the ATN (the same way the existing ASEs such as ADS, CPDLC, etc. are considered as "black boxes" by the ULCS architecture).

1.2 GACS Service Definition

1.2.1 Service Primitives

1.2.1.1 Implementations which claim to support the GACS functionality shall exhibit the behaviour defined by the service primitives in Table 1.2-1.

Note.— There is no requirement to implement the GACS service interface in any implementation; however, it is necessary to implement the end system in such a way that it will be impossible to detect (from the peer system) whether or not an interface has been built.

Table 1.2-1. Summary of GACS Service primitives

Service	Description
G-TRANSFER	This is an unconfirmed service used to transfer User-Data between communicating GACS-Users.
G-TRANSFER-CONFIRMED	This is a confirmed service used to transfer User-Data between communicating GACS-Users, and to provide the sender with confirmation that the data was received at the remote peer system(s).
G-END	This is an unconfirmed service used optionally to terminate an established communications relationship between communicating GACS-Users.
G-MULTICAST	This is an unconfirmed service used optionally to indicate whether a user wishes to receive messages sent to a particular group address.

1.2.2 Sequence of Primitives

1.2.2.1 Implementations which claim to support the GACS functionality shall exhibit behaviour allowing communicating users, consistent with the appropriate use of the corresponding service primitives, to:

a) send and receive user data;

b) optionally, establish a communications relationship between peer users for exchanging multiple messages;

c) optionally, terminate an established communications relationship.

1.2.2.2 The service shall permit a user to send and receive data at any time by using the G-TRANSFER or G-TRANSFER -CONFIRMED service.

1.2.2.3 The service shall permit a user to invoke the G-END service at any time after the multi-shot mode has been selected, and before the multi-shot inactivity timer has expired.

1.2.2.4 If a G-END request is invoked when there is no corresponding established communications relationship, an error indication shall be returned to the invoker.

Note.— The G-END service may only be used successfully after a multi-shot communications relationship has been established by using the appropriate option of the G-TRANSFER (-CONFIRMED) service.

1.2.2.5 The service shall permit a user to invoke the G-MULTICAST service at any time.

1.2.2.6 If a G-MULTICAST request is invoked when there is no supporting connectionless multicast service available, an error indication shall be returned to the invoker.

1.2.3 The G-TRANSFER service

1.2.3.1 The behaviour defined by the G-TRANSFER service shall be provided to enable the transparent transmission of data between GACS-Users.

Note 1.— G-TRANSFER is an unconfirmed service which is invoked by one GACS-User (the initiator) to send data to a peer GACS-User (or multiple peer users). If more than one recipient is specified, this is treated as multiple sequential single-recipient invocations of the service. G-TRANSFER request and indication service primitives are defined, as illustrated in Figure 1.2-1.

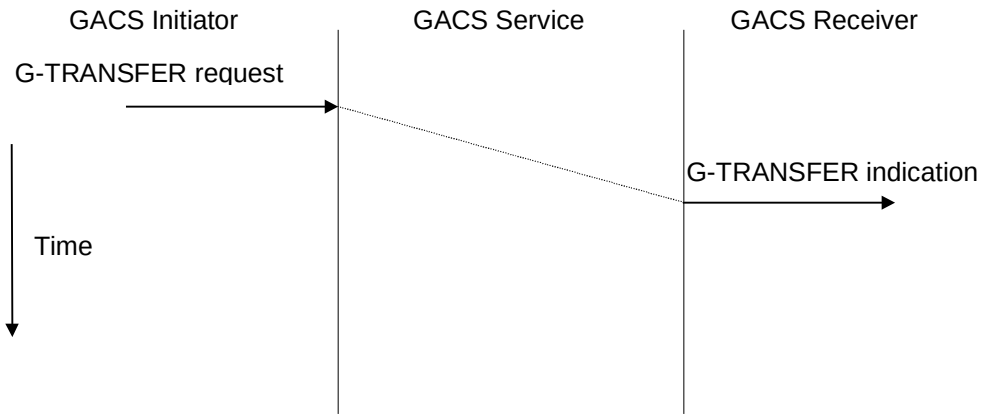

Figure 1.2-1. G-TRANSFER sequence diagram

Note 2.— The initiating GACS-User issues a G-TRANSFER request primitive. When the receiving GACS-User receives the G-TRANSFER indication primitive, the User Data is presented transparently to that user. It is a local matter to decide whether or not any reply is needed. Either GACS-User may issue a G-TRANSFER request at any time. Any sequencing constraints must be enforced by the GACS-Users themselves. The parameters of the G-TRANSFER primitives are specified in Table 1.2-2.

Table 1.2-2. G-TRANSFER parameters

Parameter Name	*Req*	*Ind*
Recipient List	*M*	
Sender	*U*	*C(=)*
Message Type	*U*	*C(=)*
Message Identifier	*U*	*C(=)*
Message Reference	*U*	*C(=)*
GACS-User Version Number	*U*	*C(=)*
Security Requirements	*U*	*C(=)*
Class of Communication	*M*	*M(=)*
Priority	*M*	*M(=)*
RER	*U*	*C(=)*
Requested Level of Service	*M*	*M(=)*
User Data	*U*	*C(=)*

Note 3.— The Recipient List parameter is used to specify with the maximum flexibility the name or address of the location of the intended peer GACS-User(s). It is a list of one or more elements. Each element takes an abstract value corresponding to either a 24-bit ICAO aircraft-id (for an airborne location), an ICAO facility designator (for a registered ground location), or a PSAP address (for any location).

Note 4.— The Sender parameter is optionally used to request that the recipient(s) be informed of the location of the initiating GACS-User. It takes an abstract value corresponding to either a 24-bit ICAO aircraft-id (for an airborne location), an ICAO facility designator (for a registered ground location), or a PSAP address (for any location). Its presence in the indication primitive is conditional upon it being specified by the GACS-User in the request primitive.

Note 5.— The Message Type parameter allows peer GACS-Users to refer unambiguously to a defined set of messages. For example, it could be used to indicate that the User Data has a pre-defined format corresponding to a message set defined by an external organisation. The Message Type identifies the message set, and is similar to a protocol identifier. It could be used for routing messages of a given type to an appropriate element of the overall GACS-User functionality. The use and permitted values of this parameter are out of the scope of the GACS service specification, and will be defined in GACS-User specifications. Its presence in the indication primitive is conditional upon it being specified by the GACS-User in the request primitive.

Note 6.— The Message Identifier parameter allows a GACS-User to assign an identifier to this particular message. The parameter is optional in the request primitive. Its presence in the indication primitive is conditional upon it being specified by the initiating GACS-User in the request primitive.

Note 7.— The Message Reference parameter allows peer GACS-Users to refer unambiguously to a message that was previously sent or received, by setting it equal to the Message Identifier of that message. The parameter is optional in the request primitive. Its presence in the indication primitive is conditional upon it being specified by the initiating User in the request primitive.

Note 8.— The Class of Communication parameter requires the initiating GACS-User to specify the traffic type and routing requirements for this message. Valid abstract values are:

ATS: No Traffic Type Policy Preference

ATS: Traffic preference for Class A ATSC route(s)

ATS: Traffic preference for Class B ATSC route(s)

ATS: Traffic preference for Class C ATSC route(s)

ATS: Traffic preference for Class D ATSC route(s)

ATS: Traffic preference for Class E ATSC route(s)

ATS: Traffic preference for Class F ATSC route(s)

ATS: Traffic preference for Class G ATSC route(s)

ATS: Traffic preference for Class H ATSC route(s)

AOC: No Traffic Type Policy Preference

AOC: Route traffic only via Gatelink

AOC: Route traffic only via VHF Data Link

AOC: Route traffic only via Satellite Data Link

AOC: Route traffic only via HF Data Link

AOC: Route traffic only via Mode S Data Link

AOC: Route traffic using an ordered preference of Gatelink first, then VHF Data Link

AOC: Route traffic using an ordered preference of Gatelink first, then VHF Data Link, then Satellite

AOC: Route traffic using an ordered preference of Gatelink first, then VHF Data Link, then HF Data Link, then Satellite Data Link.

ATN Administrative Communications

General Communications

ATN Systems Management Communications

Note 9.— The Priority parameter requires the initiating GACS-User to specify in the request primitive its requirements for the priority of this message. Valid abstract values are:

Network / Systems Management

Distress Communications

Urgent Communications

High Priority Flight Safety Messages

Normal Priority Flight Safety Messages

Meteorological Communications

Flight Regularity Communications

Aeronautical Information Service Messages

Network / Systems Administration

Aeronautical Administrative Messages

Urgent Priority Administrative and U.N. Charter Communications

High Priority Administrative and State / Government Communications

Normal Priority Administrative

Low Priority Administrative

Note 10.— The Requested Level of Service parameter allows the Initiating GACS-User to specify its requirements for the integrity and style of use of the communications channel. If the GACS-User intends to send multiple messages with the same Quality of Service (QoS) requirements to the same destination(s), then it can optionally specify a "Multi-shot" mode. Valid abstract values are:

Single shot, no error recovery, unconfirmed service

Single shot, error recovery, unconfirmed service

Multi shot, error recovery, unconfirmed service

Note 11.— The User Data parameter allows the Initiating-GACS-User to send User Data transparently to the Receiving GACS-User(s). Its presence in the indication primitive is conditional upon it being specified by the GACS-User in the request primitive. The maximum length of the User Data will typically be an implementation constraint and must be defined in the specification of the applications which use the GACS service.

Note 12.— The GACS-User Version Number, RER and Security Requirements parameters are exactly as defined for the Dialogue Service in [ULCS] 4.2.

1.2.4 The G-TRANSFER-CONFIRMED service

1.2.4.1 The behaviour defined by the G-TRANSFER-CONFIRMED service shall be provided to enable the transparent transmission of data between GACS-Users, with confirmation of data delivery to the recipient system(s).

Note 1.— G-TRANSFER-CONFIRMED is a confirmed service which is invoked by one GACS-User (the initiator) to send data to one or more peer GACS-User(s). If more than one recipient is specified, this is treated as multiple sequential single-recipient invocations of the service. G-TRANSFER-CONFIRMED request, indication and confirmation service primitives are defined, as illustrated in Figure 1.2-2.

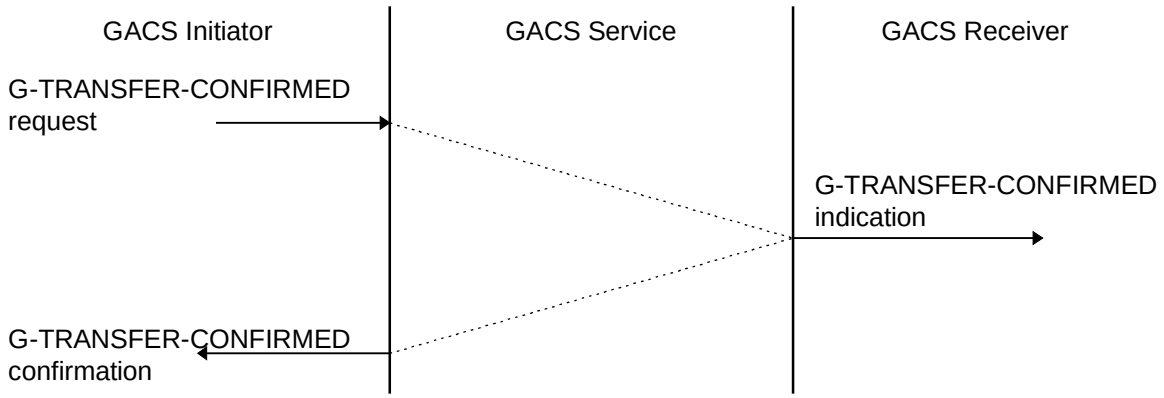

Figure 1.2-2. G-TRANSFER-CONFIRMED sequence diagram

Note 2.— The initiating GACS-User issues a G-TRANSFER-CONFIRMED request primitive. When the receiving GACS-User receives the G-TRANSFER-CONFIRMED indication primitive, the User Data is presented transparently to that user, and a G-TRANSFER-CONFIRMED confirmation primitive is automatically returned to the initiator. It is a local matter to decide whether or not any user reply to the indication primitive is needed. Either GACS-User may issue a G-TRANSFER-CONFIRMED request at any time. Any sequencing constraints must be enforced by the GACS-Users themselves. The parameters of the G-TRANSFER-CONFIRMED primitives are specified in Table 1.2-3.

Table 1.2-3. G-TRANSFER-CONFIRMED primitive parameters

Parameter Name	*Req*	*Ind*	*Cnf*
Recipient List	*M*		
Sender	*U*	*C(=)*	*M*
Message Type	*U*	*C(=)*	*C(=)*
Message Identifier	*M*	*M(=)*	
Message Reference	*U*	*C(=)*	*M*
GACS-User Version Number	*U*	*C(=)*	*C*
Security Requirements	*U*	*C(=)*	
Class of Communication	*M*	*M(=)*	
Priority	*M*	*M(=)*	
RER	*U*	*C(=)*	
Requested Level of Service	*M*	*M(=)*	
Result			*M*
User Data	*U*	*C(=)*	

Note 3.— The Recipient List parameter is identical to the G-TRANSFER service.

Note 4.— The Sender parameter is optionally used to request that the recipient(s) be informed of the location of the initiating GACS-User, and is also used to inform the initiator of the location which invoked the confirmation primitive. It takes an abstract value corresponding to either a 24-bit ICAO aircraft-id (for an airborne location), an ICAO facility designator (for a registered ground location), or a PSAP address (for any location). Its presence in the indication primitive is conditional upon it being specified by the GACS-User in the request primitive. Its presence in the confirmation primitive is mandatory.

Note 5.— The Message Type parameter is identical to the G-TRANSFER service except that its presence in the confirmation primitive is conditional upon it being specified by the GACS-User in the request primitive.

Note 6.— The Message Identifier parameter is used to allow initiating GACS-Users to correlate confirmation primitives with previously invoked request primitives. The parameter is therefore mandatory in the request primitive. The value in the indication primitive is equal to the value set by the initiating GACS-User in the request primitive.

Note 7.— The Message Reference parameter allows peer GACS-Users to refer unambiguously to a message that was previously sent or received, by setting it equal to the Message Identifier of that message. The parameter is optional in the request primitive. Its presence in the indication primitive is conditional upon it being specified by the initiating GACS-User in the request primitive. It is mandatory in the confirmation primitive, where its value is equal to the Message Identifier value of the message which is being confirmed.

Note 8.— The Class of Communication parameter is identical to the G-TRANSFER service.

Note 9.— The Priority parameter is identical to the G-TRANSFER service.

Note 10.— The Requested Level of Service parameter allows the Initiating User to specify its requirements for the integrity of the communications channel. Valid abstract values are:

Single shot, no error recovery, confirmed service

Single shot, error recovery, confirmed service

Multi-shot, error recovery, confirmed service

Note 11.— The GACS-User Version Number is as defined for the Dialogue Service in [ULCS] 4.2. Its presence in the indication and confirmation primitives is conditional upon it being specified by the initiating GACS-User in the request primitive. The value in the confirmation primitive is set by the GACS-Provider in the peer system, if possible, based on local knowledge of the peer GACS-User(s).

Note 12.— The RER and Security Requirements parameters are exactly as defined for the Dialogue Service in [ULCS] 4.2.

Note 13.— The Result parameter informs the Initiating GACS-User of the outcome of the G-TRANSFER-CONFIRMED request. Valid abstract values are:

Successful delivery

Delivery failed

Service problem - delivery status uncertain

Note 14.— The User Data parameter is identical to the G-TRANSFER service.

1.2.5 The G-END service

1.2.5.1 The behaviour defined by the G-END service shall be provided to enable the orderly termination of a communications relationship between GACS-Users.

Note 1.— G-END is an unconfirmed service which is optionally invoked by one GACS-User (who is then the initiator) to terminate a communications relationship with one or more peer GACS-User(s). If more than one recipient is specified, this is treated as multiple sequential single-recipient invocations of the service. G-END request and indication service primitives are defined, as illustrated in Figure 1.2-3.

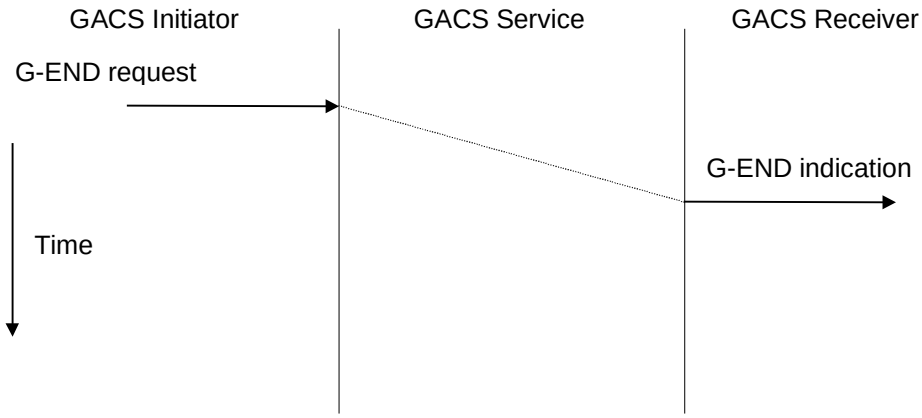

Figure 1.2-3. G-END sequence diagram

Note 2.— The initiating GACS-User issues a G-END request primitive at any time after using the G-TRANSFER or G-TRANSFER-CONFIRMED service with a multi-shot Level of Service. When the receiving GACS-User receives the G-END indication primitive, it knows that the current communications relationship with the peer is over. A new relationship may be established at any time. It is a local matter to decide whether or not any user reply is needed. Any sequencing constraints must be enforced by the GACS-Users themselves. The parameters of the G-END primitives are specified in Table 1.2-4.

Table 1.2-4. G-END parameters

Parameter Name	*Req*	*Ind*
Recipient List	*M*	
Sender		*M*
Message Type	*U*	*C(=)*
Message Reference	*U*	*C(=)*
User Data	*U*	*C(=)*

Note 3.— Possible collisions of the G-END primitives are handled internally by the Dialogue Service, and are not visible to GACS-Users.

Note 4.— The Recipient List parameter is identical to the G-TRANSFER service.

Note 5.— The Sender parameter is used to inform the recipient of the location which invoked the D-END request primitive. This is required in cases where a GACS-User has set up multi-shot connections to multiple peers, and it needs to identify which peer has invoked the D-END service. Its presence in the indication primitive is therefore mandatory. It takes an abstract value corresponding to either a 24-bit ICAO aircraft-id (for an airborne location), an ICAO facility designator (for a registered ground location), or a PSAP address (for any location).

Note 6.— The Message Type parameter is identical to the G-TRANSFER service.

Note 7.— The Message Reference parameter is identical to the G-TRANSFER service .

Note 8.— The User Data parameter is identical to the G-TRANSFER service.

1.2.6 The G-MULTICAST service

1.2.6.1 The behaviour defined by the G-MULTICAST service shall be provided to enable or disable the receipt of messages addressed to a group address.

*Note 1.— G-*MULTICAST *is an unconfirmed service which is optionally invoked by a GACS-User to inform the local communications system whether that user wishes to receive messages sent to a particular group address. This service is only available when supported by a connectionless communications provider.*

Note 2.— A G-MULTICAST request primitive is defined, as illustrated in Figure 1.2-4.

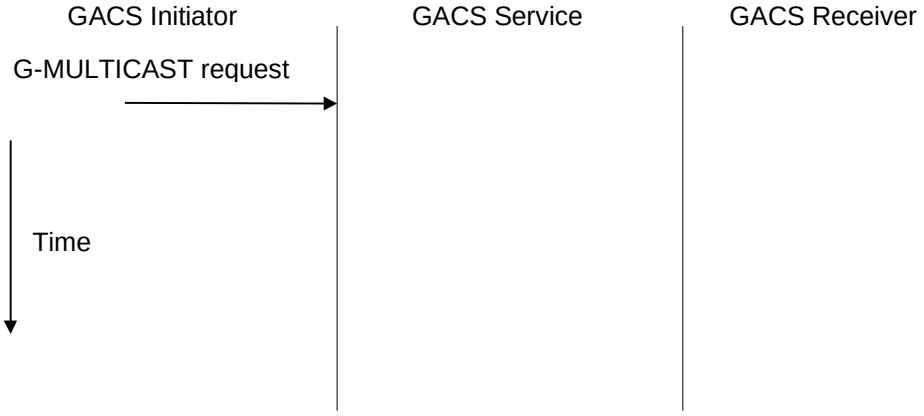

Figure 1.2-4. G-MULTICAST sequence diagram

Note 3.— The initiating GACS-User issues a G-MULTICAST request primitive at any time. The parameters of the G- MULTICAST primitives are specified in Table 1.2-5.

Table 1.2-5. G- MULTICAST parameters

Parameter Name	*Req*
Group Address	*M*
Toggle	*U*

Note 4.— The Group Address parameter specifies a PSAP address which includes in the supplied transport address a Group NSAP address.

Note 5.— The Toggle parameter is used to enable or disable the receipt of messages addressed to the group address. The default is to enable such receipt. Valid abstract values are:

Enable Multicast Receipt

Disable Multicast Receipt

1.3 Protocol Definition

Note.— This section describes the format and sequencing of messages allowed by the GACS protocol.

1.3.1 Model

Note 1.— The GACS service is provided by a protocol entity which is modelled as a finite state machine whose specification is given in this section. The GACS protocol entity communicates with its service-user by means of the GACS service primitives defined in section 1.2. It communicates with its Dialogue Service provider by means of the D- service primitives defined in [ULCS] 4.2.

Note 2.— The GACS protocol entity is driven by the receipt of input events from its GACS service-user, from its Dialogue Service provider that supports the communication relationships, and from an internal inactivity timer. The input events from the GACS service-user are GACS request primitives. The input events from the Dialogue Service provider are D-service indication and confirmation primitives.

1.3.1.1 A new invocation of a GACS protocol entity shall be employed upon the receipt of a D-START indication primitive or a GACS primitive which requires a new dialogue to be established (i.e. any single-shot G-TRANSFER or G-TRANSFER-CONFIRMED request or a multi-shot G-TRANSFER or G-TRANSFER-CONFIRMED request where no suitable communications relationship currently exists).

1.3.2 Sequence Rules

1.3.2.1 The GACS protocol entity shall provide a service such that a user is able to invoke any GACS service request at any time when the protocol entity is operational.

1.3.2.2 The GACS protocol entity shall be capable of receiving and processing any valid GACS APDU at any time when the protocol entity is operational.

1.3.3 Timers

1.3.3.1 The GACS entity shall implement an inactivity timer T_inact for each communications relationship that is established using the “multi-shot” option of the G-TRANSFER service or G-TRANSFER-CONFIRMED service.

1.3.3.2 The threshold value of T_inact, referred to as T_inact_max, shall be configurable.

1.3.3.3 T_inact shall be reset to zero and started whenever a positive D-START confirmation primitive is received from the Dialogue Service Provider.

1.3.3.4 T_inact shall be reset to zero each time that user data is transferred to or from the Dialogue Service Provider.

1.3.3.5 If the value of T_inact reaches the threshold T_inact_max, then the GACS entity shall invoke the D-END service to end the relationship.

1.3.3.6 ***Recommendation****.— A configurable timer should be implemented so that dialogue establishment can be aborted by the GACS-Provider if no response is received after issuing a D-START request.*

1.3.3.7 ***Recommendation.—*** *A configurable timer should be implemented so that orderly dialogue termination can be aborted by the GACS-Provider if no response is received after issuing a D-END request.*

1.3.4 Elements of procedure

1.3.4.1 Handling of Level of Service

1.3.4.1.1 The GACS service shall map to either a connectionless provider or a connection-oriented provider with or without a logical acknowledgement, depending upon the value of the Requested Level of Service parameter, as shown in Table 1.3-1.

Table 1.3-1. Level of Service mappings

Requested Level of Service parameter	**Mapping to communications provider**
Single shot, no error recovery, unconfirmed service	Map to connectionless service, if available. User Data sent via D-UNIT-DATA service. Otherwise treat as "Single shot, error recovery, unconfirmed service."
Single shot, error recovery, unconfirmed service	Map to connection-oriented service. User Data sent via D-START service with negative D-START response.
Single shot, no error recovery, confirmed service	Map to connectionless service, if available. User Data and confirmation from GACS service sent via separate D-UNIT-DATA service invocations. Otherwise treat as "Single shot, error recovery, confirmed service."
Single shot, error recovery, confirmed service	Map to connection oriented service. User Data sent via D-START service with negative D-START response. Result passed to Initiator.
Multi shot, error recovery, unconfirmed service	Map to connection-oriented service. Existing connection used, or new connection established using D-START (see note). User Data sent via D-DATA service.
Multi-shot, error recovery, confirmed service	Map to connection oriented service. Existing connection used, or new connection established using D-START (see note). User Data sent via D-DATA service. Confirmation returned via D-DATA service invoked by GACS entity.

Note.— When using multi-shot mode, User Data is always sent using the D-DATA service. If necessary, a connection is first established independently of the User Data transfer. This allows the T-CONNECT optimisation to be realised, so that upper layer connections are established simultaneously with transport connections, and also allows a clean separation of User Data from connection set-up information. This can be beneficial when it is desired to negotiate a secure dialogue before sending any User Data.

1.3.4.2 Multiple Recipients

1.3.4.2.1 If more than one recipient is specified by the GACS-User in a service primitive invocation, this shall be treated as multiple sequential single-recipient invocations of the service.

1.3.4.3 Collisions

1.3.4.3.1 Collision of G-TRANSFER and/or G-TRANSFER-CONFIRMED services invoked by two peers shall result in the establishment of two independent communication links.

1.3.4.3.2 In the case of collisions between the G-TRANSFER and/or G-TRANSFER-CONFIRMED service and G-END service invoked by two peers, the transfer shall be permitted to succeed. The transfer shall either occur over the existing link, or over a new communication link, depending upon the exact timing of the service invocations at the peer GACS entities.

Note.— Collision of the G-END service invoked by two peers will be internally resolved by the Dialogue service.

1.3.4.4 Disruptions

1.3.4.4.1 The services which are supported by connection-mode protocols may be disrupted by Abort events, either from the peer system or from the communication provider, and shall be able to handle such events.

1.3.4.4.2 If a GACS-User is waiting for confirmation of a previous message transfer, a provider abort event shall trigger such confirmation, with a status of confirmedUnknown, as it is not known whether the peer received the message before the abort.

1.3.4.5 Rules for extensibility

1.3.4.5.1 The protocol defined in this specification shall be designated Version 1 of the GACS protocol, and is identified as such by the absence of any extension fields in the abstract syntax.

1.3.4.5.2 When processing an incoming GACS APDU, the accepting protocol entity shall log the presence of any extension fields that are not defined in the abstract syntax of this protocol specification, and process the APDU as if the extensions were not present.

1.3.4.6 Exception handling

1.3.4.6.1 If a GACS PDU is received that cannot be decoded, then that PDU is considered an invalid PDU and exception handling procedures described in this section shall apply.

1.3.4.6.2 If a GACS PDU is received which is not an expected PDU at that time, then that PDU is considered an invalid PDU and exception handling procedures described in this section shall apply.

1.3.4.6.3 The error handling shall result in the dialogue being aborted, if one exists, and a notification being given to the GACS user, if possible.

1.3.5 GACS Protocol Data Unit Description

Note.— The GACS protocol uses a single PDU type to support all defined GACS services.

1.3.5.1 The fields of the GACS protocol data unit shall be used to support the GACS service as defined in this section.

1.3.5.2 The GACS protocol data unit shall consist of a header field and a User Data field.

1.3.5.3 The User Data field shall be used to convey the data passed to the GACS service via the User Data parameter transparently to the addressed recipient(s).

1.3.5.4 messageType

1.3.5.4.1 The messageType field of the GACS header shall be used to convey the Message Type parameter, passed to the GACS service by the user, transparently to the addressed recipient(s).

1.3.5.4.2 In confirmations generated by the GACS service provider, the messageType field shall have the same value as that in the GACS PDU whose receipt is being confirmed.

1.3.5.5 messageIdentifier

1.3.5.5.1 The messageIdentifier field of the GACS header shall be used to convey the Message Identifier parameter, passed to the GACS service by the user, transparently to the addressed recipient(s).

1.3.5.6 messageReference

1.3.5.6.1 The messageReference field of the GACS header shall be used to convey the Message Reference parameter, passed to the GACS service by the user, transparently to the addressed recipient(s).

1.3.5.6.2 In confirmations generated by the GACS service provider, the messageReference field shall have the same value as the messageIdentifier field in the GACS PDU whose receipt is being confirmed.

1.3.5.7 confirmation

1.3.5.7.1 The confirmation field of the GACS header shall be used to convey whether the GACS service user requested the G-TRANSFER service or the G-TRANSFER-CONFIRMED service.

1.3.5.7.2 In confirmation PDUs generated by the GACS service provider, the confirmation field shall be used to identify the PDU as a confirmation, and to state whether the confirmed PDU was passed to the receiving user without problems.

1.3.5.8 multiShot

1.3.5.8.1 The multiShot field of the GACS header shall be used to convey whether the GACS service user requested the “multi-shot” option of the CO G-TRANSFER service or G-TRANSFER-CONFIRMED service.

1.3.5.8.2 The multiShot field of the GACS header shall be also used to convey whether the GACS service user invoked the G-END service.

1.3.5.8.3 If a GACS user requests the multi-shot option, and the receiving (remote) GACS provider is able to support the option for this instance of communication, then a connection shall be maintained

between the two GACS users until it is terminated either by one of the GACS users invoking the G-END service or by a T_inact timeout.

1.3.5.9 sender

1.3.5.9.1 The sender field of the GACS header shall be used to convey the identity of the sender when this has been requested by the user of the G-TRANSFER service or G-TRANSFER-CONFIRMED service, and always in confirmation messages generated by the GACS service itself.

1.3.6 Formal Definition Of Messages

Note.— This section contains the formal definition of messages exchanged by GACS protocol entities, using Abstract Syntax Notation One (ASN.1). The notation is defined in ISO/IEC 8824-1.

1.3.6.1 The abstract syntax of the GACS protocol data units shall comply with the description contained in the ASN.1 module GACSProtocolVersion1, as defined in this section.

```
GACSProtocolVersion1 DEFINITIONS ::=
BEGIN
-- EXPORTS
-- everything
GACSpdu ::= SEQUENCE {
   gacsHeader  [0]   GACSHeaderType,
   userData    [1]   OCTET STRING OPTIONAL
}
GACSHeaderType ::= SEQUENCE {
   messageType       [0]   GACSMessageId OPTIONAL,
   messageIdentifier [1]   GACSMessageId OPTIONAL,
   messageReference  [2]   GACSMessageId OPTIONAL,
   confirmation      [3]   GACSConfirmation DEFAULT unconfirmedReq,
   multiShot         [4]   GACSMultiShot DEFAULT single,
   sender            [5]   GACSNameOrAddress OPTIONAL,
   ...
}
GACSMessageId ::= CHOICE {
   globalForm  [0] OBJECT IDENTIFIER,
   localForm   [1] INTEGER (0..255, ...),
   ...
}
GACSConfirmation ::= ENUMERATED {
   unconfirmedReq    (0),
   confirmedReq      (1),
   confirmedOk       (2),
   confirmedNotOk    (3),
```

```
   confirmedUnknown  (4),
   ...
}
GACSMultiShot ::= ENUMERATED {
   single            (0),
   maintained        (1),
   end               (2),
   ...
}
GACSNameOrAddress ::= CHOICE {
   name              [1] ULCSPeerId,
   pSAPAddress       [2] OCTET STRING
}
ULCSPeerId ::= SEQUENCE {
   locationID  [1] ULCSLocationType,
   sysID       [2] INTEGER  OPTIONAL,
   ...
}
ULCSLocationType ::= CHOICE {
   aircraft          [1] BIT STRING (SIZE(24)), -- 24-bit address
   groundFacility    [2] IA5String (SIZE(4..8)), -- ICAO designator
   ...
END -- of GACSProtocolVersion1
```

1.3.7 Encoding rules

1.3.7.1 The GACS protocol data units shall be encoded for transfer across the Dialogue Service boundary using the ASN.1 Packed Encoding Rules (PER) specified in ISO/IEC 8825-2, using the basic, unaligned variant.

1.3.8 GACS State Table

1.3.8.1 The GACS protocol entity shall behave as if it can exist only in one of the states defined in Table 1.3-2 for each instance of communication between two peer GACS-Users.

Table 1.3-2. States of the GACS Protocol Entity

State	Short name	Description
STA 0	Idle	No instance of communication; protocol machine is unallocated.
STA 1	Start-single	A D-START request has been submitted and a negative D-START confirmation is expected.
STA 2	Start-multi	A D-START request has been submitted and a positive D-START confirmation is expected.
STA 3	Ending	A D-END request has been submitted and a D-END confirmation is expected.
STA 4	Associated	A dialogue has been fully established with the peer

1.3.8.2 The GACS protocol entity shall be capable of detecting and processing each of the input events listed in Table 1.3-3.

Table 1.3-3. GACS Input Events

Event	Description
Unit Data req	G-TRANSFER request with Requested Level of Service set to (Single shot, no error recovery, unconfirmed service), or G-TRANSFER-CONFIRMED request with Requested Level of Service set to (Single shot, no error recovery, confirmed service) submitted by GACS-User
Single Shot req	G-TRANSFER request with Requested Level of Service set to (Single shot, error recovery, unconfirmed service), or G-TRANSFER-CONFIRMED request with Requested Level of Service set to (Single shot, error recovery, confirmed service) submitted by GACS-User
Multi-Shot req	G-TRANSFER request with Requested Level of Service set to (Multi shot, error recovery, unconfirmed service), or G-TRANSFER-CONFIRMED request with Requested Level of Service set to (Multi shot, error recovery, confirmed service) submitted by GACS-User
G-END req	G-END request submitted by GACS-User
D-START ind ()	D-START indication with no User Data delivered from Dialogue Service provider.
D-START ind (UD)	D-START indication, with User Data present delivered from Dialogue Service provider.
D-START cnf+	D-START confirmation from Dialogue Service provider, with Result = accepted
D-START cnf-	D-START confirmation with Result = rejected (transient) or rejected (permanent) delivered from Dialogue Service provider
D-DATA ind(cnf)	GACS PDU received via D-DATA indication with confirmation = confirmedOk delivered from Dialogue Service provider
D-DATA ind(UD)	GACS PDU received via D-DATA indication with confirmation = unconfirmedReq or confirmedReq delivered from Dialogue Service provider
D-UNIT-DATA ind(UD)	GACS PDU received via D-UNIT-DATA indication with confirmation = unconfirmedReq or confirmedReq delivered from Dialogue Service provider
D-UNIT-DATA ind(cnf)	GACS PDU received via D-UNIT-DATA indication with confirmation = confirmedOk delivered from Dialogue Service provider
D-END cnf+	D-END confirmation with Result = accepted delivered from Dialogue Service provider
D-END cnf-	D-END confirmation with Result = rejected delivered from Dialogue Service provider
D-END ind	D-END indication delivered from Dialogue Service provider
D-ABORT ind	D-ABORT indication delivered from Dialogue Service provider
D-P-ABORT ind	D-P-ABORT indication delivered from Dialogue Service provider
T-inact	Internal inactivity timer trigger

1.3.8.3 The GACS protocol entity shall exhibit external behaviour in accordance with the state table specified in Table 1.3-6, which shows diagrammatically the state transitions and actions performed by the protocol in response to incoming events, as follows:

a) Incoming events are shown in the first column of the state table, and are enumerated in Table 1.3-3. Valid protocol states are shown in the header row of the state table, and are described in table 1.3-2.

b) When an input event occurs, the state transition and any action to be taken are indicated by the cell of the state table which is the intersection of the incoming event name and the current protocol state.

c) Each cell in the state table shows:

1) the new state that the CF enters after the action has been performed, indicated by "STA I", where I is an integer

2) 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).

3) the action, if any, which is to be performed. The possible actions are described in Table 1.3-5.

d) Blank cells indicate error conditions.

e) Any of the input events which result in a new dialogue being established may occur at any time, and will result in a new instance of the state table being created, initially in the Idle state (STA 0).

1.3.8.4 For the purpose of this specification, the state table shall be treated as atomic, such that when an input event is invoked, that event is processed to completion within the same logical processing thread.

Note. — This provision does not imply any particular implementation architecture.

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

a) The occurrence of an input event other than those listed in Table 1.3-3; or

b) A combination of input event and current state which corresponds to a blank cell in Table 1.3-6.

1.3.8.6 In the event of a conflict between the actions implied by the state table and the text elsewhere in this specification, the text shall take precedence.

Table 1.3-4. Predicates

Predicate	Description
p1	Sender requested GACS-TRANSFER-CONFIRMED service (confirmation parameter = confirmedReq)
~p1	Sender requested GACS-TRANSFER (unconfirmed) service (confirmation parameter = unconfirmedReq)

Table 1.3-5. Output Actions

Action	Description
D-UNIT-DATAreq	Issue GACS PDU via D-UNIT-DATA request to Dialogue Service provider, with confirmation = unconfirmedReq or confirmedReq, as requested by GACS-User
D-UNIT-DATAreq(cnf)	Invoke D-UNIT-DATA request with GACS PDU as User Data to Dialogue Service provider, with confirmation = confirmedOk
D-STARTreq()	Invoke D-START request with no User Data to Dialogue Service provider
D-STARTreq (UD)	Invoke D-START request with GACS PDU as User Data to Dialogue Service provider, with confirmation = unconfirmedReq or confirmedReq, as requested by GACS-User
D-DATAreq(UD)	Invoke D-DATA request with GACS PDU as User Data to Dialogue Service provider, with confirmation = unconfirmedReq or confirmedReq, as requested by GACS-User
D-DATAreq(cnf)	Invoke D-DATA request with GACS PDU as User Data to Dialogue Service provider, with confirmation = confirmedOk

D-ABORTreq	Invoke D-ABORT request with no User Data to Dialogue Service provider
D-END req	Invoke D-END request with GACS PDU, if any, as User Data to Dialogue Service provider
D-STARTrsp+	Invoke D-START response with no User Data and Result = accepted to Dialogue Service provider
D-STARTrsp-(cnf)	Invoke D-START response with GACS PDU as User Data with confirmation = confirmedOk and Result = rejected (transient)
D-STARTrsp-	Invoke D-START response with no User Data and Result = rejected (transient) to Dialogue Service provider
D-ENDrsp+	Invoke D-END response with no User Data and Result = accepted to Dialogue Service provider
G-TFRind	Deliver G-TRANSFER indication to local GACS-User
G-TFR-CNFind	Deliver G-TRANSFER-CONFIRMED indication to local GACS-User
G-TFR-CNFcnf	Deliver G-TRANSFER-CONFIRMED confirmation to local GACS-User, with confirmation set to the value in the GACS PDU if present. If no PDU is present, set confirmation to confirmedNotOk. If P-P-ABORT was received, set confirmation to confirmedUnknown.
G-END ind	Deliver G-END indication to local GACS-User
t_inact(S)	Start the inactivity timer (from value zero)
t_inact(R)	Reset the inactivity timer to value zero
t_inact(O)	Stop the inactivity timer

Table 1.3-6. State Table

	STA 0 Idle	STA 1 Start-single	STA 2 Start-multi	STA 3 Ending	STA 4 Associated
Unit Data req	STA 0 D-UNIT-DATAreq				
Single Shot req	STA 1 D-STARTreq (UD)				
Multi-Shot req	STA 2 D-STARTreq()				STA 4 D-DATAreq (UD) t_inact(R)
G-END req			STA 0 D-ABORTreq		STA 3 D-END req t_inact(R)
D-START ind ()	STA 4 D-STARTrsp+				
D-START ind (UD)	STA 0 ~p1: D-START rsp-, G-TFRind p1: D-STARTrsp-(cnf), G-TFR-CNFind				
D-START cnf+			STA 4 D-DATAreq(UD) t_inact(S)		
D-START cnf-		STA 0 p1: G-TFR-CNFcnf	STA 0 p1: G-TFR-CNFcnf		

	STA 0 Idle	STA 1 Start-single	STA 2 Start-multi	STA 3 Ending	STA 4 Associated
D-DATA ind(cnf)					STA 4 t_inact(R) p1:G-TFR-CNFcnf
D-DATA ind(UD)				STA 3 t_inact(R) ~p1:G-TFRind p1:G-TFR-CNFind	STA 4 t_inact(R) ~p1:G-TFRind p1:G-TFR-CNFind, D-DATAreq(cnf)
D-END cnf+				STA 0 t_inact(O)	
D-END cnf-					
D-END ind					STA 0 D-ENDrsp+ G-ENDind t_inact(O)
D-UNIT-DATA ind(cnf)	STA 0 G-TFR-CNFcnf				
D-UNIT-DATA ind(UD)	STA 0 ~p1:G-TFRind p1:G-TFR-CNFind, D-UNIT-DATAreq(cnf)				
D-ABORT ind		STA 0 p1: G-TFR-CNFcnf	STA 0 p1: G-TFR-CNFcnf	STA 0 t_inact(O)	STA 0 G-ENDind t_inact(O)
D-P-ABORT ind		STA 0 p1: G-TFR-CNFcnf	STA 0 p1: G-TFR-CNFcnf	STA 0 t_inact(O)	STA 0 G-ENDind t_inact(O)
T-inact				STA 0 D-ABORTreq t_inact(O)	STA 3 D-END req t_inact(R)

1.3.9 GACS Protocol Description

1.3.9.1 On initiation, the GACS protocol entity shall be in the Idle state.

1.3.9.2 G-TRANSFER Request primitive invoked by GACS-User

1.3.9.2.1 When invoked by the GACS-User, if the Recipient List parameter of the G-TRANSFER Request contains more than one element then the GACS protocol entity shall select each recipient in the list in turn and, for each recipient, behave as if the G-TRANSFER Request primitive had been invoked with only that element present in the Recipient List parameter.

1.3.9.2.2 If a G-TRANSFER request primitive is invoked and the Requested Level of Service parameter indicates “multi-shot” mode, and a dialogue already exists with the identified peer entity, with the same quality of service parameters, then the primitive shall be processed by the protocol entity which is associated with the existing dialogue and is in the “Associated” state (STA 4). Otherwise it shall be processed by a new invocation of the protocol entity.

1.3.9.2.3 If a G-TRANSFER request primitive is invoked with the Requested Level of Service parameter containing the abstract value “Single shot, no error recovery, unconfirmed service,” and the connectionless dialogue service is available, and the GACS entity is in the Idle state (STA 0), then the GACS entity shall:

a) Create a GACSpdu APDU with field values defined in Table 1.3-7.

b) At the supporting Dialogue Service boundary, invoke a D-UNIT-DATA request primitive with parameters as defined in Table 1.3-8.

c) Remain in the Idle state.

Table 1.3-7

GACSpdu field	**Value**
messageType	Message Type from the G-TRANSFER request primitive, if provided. Absent otherwise.
messageIdentifier	Message Identifier from the G-TRANSFER request primitive, if provided. Absent otherwise.
messageReference	Message Reference from the G-TRANSFER request primitive, if provided. Absent otherwise.
confirmation	Absent (default abstract value “unconfirmedReq”)
multiShot	Absent (default abstract value “single”)
sender	Sender identity if requested in the G-TRANSFER request. Absent otherwise.
userData	User Data from the G-TRANSFER request primitive, if provided. Absent otherwise.

Table 1.3-8

D-UNIT-DATA / D-START request parameter	**Status**	**Value**
Called Peer ID	M	One item of the Recipient List parameter value from the G-TRANSFER request.
Calling Peer ID	U	Not used.
DS-User Version Number	U	GACS-User Version Number from the G-TRANSFER request, if provided. Absent otherwise.
Security Requirements	U	Security Requirements from the G-TRANSFER request, if provided. Absent otherwise.
QOS: Routing Class	M	*Class of Communication* from the G-TRANSFER request.
QOS: Priority	M	*Priority* from the G-TRANSFER request.
QOS: Residual Error Rate	M	*RER* from the G-TRANSFER request, if provided. “low” otherwise.
User Data	M / U	The GACSpdu APDU created according to Table 1.3-7.

1.3.9.2.4 If a G-TRANSFER request primitive is invoked with the Requested Level of Service parameter containing the abstract value “Single shot, no error recovery, unconfirmed service” and the connectionless dialogue service is not available, or with the Requested Level of Service parameter containing the abstract value “Single shot, error recovery, unconfirmed service”, and the GACS entity is in the Idle state (STA 0), then the GACS entity shall:

a) Create a GACSpdu APDU with field values defined in Table 1.3-7.

b) At the supporting Dialogue Service boundary, invoke a D-START request primitive with parameters as defined in Table 1.3-8.

c) Enter the Start-single state (STA 1).

1.3.9.2.5 If a G-TRANSFER request primitive is invoked with the Requested Level of Service parameter containing the abstract value “Multi shot, error recovery, unconfirmed service”, and the GACS entity is in the Idle state (STA 0), then the GACS entity shall:

a) Create a GACSpdu APDU with field values defined in Table 1.3-9 and store it until a dialogue with the specified peer has been successfully established.

b) At the supporting Dialogue Service boundary, invoke a D-START request primitive with parameters as defined in Table 1.3-10.

c) Enter the Start-multi state (STA 2).

Table 1.3-9

GACSpdu field	**Value**
messageType	Message Type from the G-TRANSFER request primitive, if provided. Absent otherwise.
messageIdentifier	Message Identifier from the G-TRANSFER request primitive, if provided. Absent otherwise.
messageReference	Message Reference from the G-TRANSFER request primitive, if provided. Absent otherwise.
confirmation	Absent (default abstract value “unconfirmedReq”)
multiShot	“maintained”
sender	Sender identity if requested in the G-TRANSFER request. Absent otherwise.
userData	User Data from the G-TRANSFER request primitive, if provided. Absent otherwise.

Table 1.3-10

D-START request parameter	**Status**	**Value**
Called Peer ID	M	One item of the Recipient List parameter value from the G-TRANSFER request.
Calling Peer ID	U	Not used.
DS-User Version Number	U	GACS-User Version Number from the G-TRANSFER request, if provided. Absent otherwise.
Security Requirements	U	Security Requirements from the G-TRANSFER

D-START request parameter	Status	Value
		request, if provided. Absent otherwise.
QOS: Routing Class	M	*Class of Communication* from the G-TRANSFER request.
QOS: Priority	M	*Priority* from the G-TRANSFER request.
QOS: Residual Error Rate	M	*RER* from the G-TRANSFER request, if provided. "low" otherwise.
User Data	U	Absent

1.3.9.2.6 If a G-TRANSFER request primitive is invoked with the Requested Level of Service parameter containing the abstract value "Multi shot, error recovery, unconfirmed service", and the GACS entity is in the Associated state (STA 4), then the GACS entity shall:

a) Reset timer t_inact,

b) Create a GACSpdu APDU with field values defined in Table 1.3-9,

c) At the supporting Dialogue Service boundary, invoke a D-DATA request with the GACSpdu APDU as the D-DATA *User Data* parameter value, and

d) Remain in the Associated state (STA 4).

1.3.9.3 G-TRANSFER-CONFIRMED Request primitive invoked by GACS-User

1.3.9.3.1 When invoked by the GACS-User, if the Recipient List parameter of the G-TRANSFER-CONFIRMED Request contains more than one element then the GACS protocol entity shall select each recipient in the list in turn and, for each recipient, behave as if the G-TRANSFER-CONFIRMED Request primitive had been invoked with only that element present in the Recipient List parameter.

1.3.9.3.2 If a G-TRANSFER-CONFIRMED request primitive is invoked and the Requested Level of Service parameter indicates "multi-shot" mode, and a dialogue already exists with the identified peer entity, with the same quality of service parameters, then the primitive shall be processed by the protocol entity which is associated with the existing dialogue and is in the "Associated" state (STA 4). Otherwise it shall be processed by a new invocation of the protocol entity.

1.3.9.3.3 If a G-TRANSFER-CONFIRMED request primitive is invoked with the Requested Level of Service parameter containing the abstract value "Single shot, no error recovery, confirmed service" and the connectionless dialogue service is available, and the GACS entity is in the Idle state (STA 0), then the GACS entity shall:

a) Create a GACSpdu APDU with field values defined in Table 1.3-11.

b) At the supporting Dialogue Service boundary, invoke a D-UNIT-DATA request primitive with parameters as defined in Table 1.3-12.

c) Remain in the Idle state.

Table 1.3-11

GACSpdu field	Value
messageType	Message Type from the G-TRANSFER-CONFIRMED request primitive, if provided. Absent otherwise.

GACSpdu field	Value
messageIdentifier	Message Identifier from the G-TRANSFER-CONFIRMED request primitive.
messageReference	Message Reference from the G-TRANSFER-CONFIRMED request primitive, if provided. Absent otherwise.
confirmation	"confirmedReq"
multiShot	Absent (default abstract value "single")
sender	Sender identity if requested in the G-TRANSFER-CONFIRMED request. Absent otherwise.
userData	User Data from the G-TRANSFER-CONFIRMED request primitive, if provided. Absent otherwise.

Table 1.3-12

D-UNIT-DATA / D-START request parameter	Status	Value
Called Peer ID	M	One item of the Recipient List parameter value from the G-TRANSFER-CONFIRMED request.
Calling Peer ID	U	Not used.
DS-User Version Number	U	GACS-User Version Number from the G-TRANSFER-CONFIRMED request, if provided. Absent otherwise.
Security Requirements	U	Security Requirements from the G-TRANSFER-CONFIRMED request, if provided. Absent otherwise.
QOS: Routing Class	M	*Class of Communication* from the G-TRANSFER-CONFIRMED request.
QOS: Priority	M	*Priority* from the G-TRANSFER-CONFIRMED request.
QOS: Residual Error Rate	M	*RER* from the G-TRANSFER-CONFIRMED request, if provided. "low" otherwise.
User Data	M / U	The GACSpdu APDU created according to Table 1.3-11.

1.3.9.3.4 If a G-TRANSFER-CONFIRMED request primitive is invoked with the Requested Level of Service parameter containing the abstract value "Single shot, no error recovery, confirmed service" and the connectionless dialogue service is not available, or with the Requested Level of Service parameter containing the abstract value "Single shot, error recovery, confirmed service", and the GACS entity is in the Idle state (STA 0), then the GACS entity shall:

a) Create a GACSpdu APDU with field values defined in Table 1.3-11

b) At the supporting Dialogue Service boundary, invoke a D-START request primitive with parameters as defined in Table 1.3-12.

c) Enter the Start-single state (STA 1).

1.3.9.3.5 If a G-TRANSFER-CONFIRMED request primitive is invoked with the Requested Level of Service parameter containing the abstract value “Multi shot, error recovery, confirmed service”, and the GACS entity is in the Idle state (STA 0), then the GACS entity shall:

a) Create a GACSpdu APDU with field values defined in Table 1.3-13 and store it until a dialogue with the specified peer has been successfully established.

b) At the supporting Dialogue Service boundary, invoke a D-START request primitive with parameters as defined in Table 1.3-14.

c) Enter the Start-multi state (STA 2).

Table 1.3-13

GACSpdu field	**Value**
messageType	Message Type from the G-TRANSFER-CONFIRMED request primitive, if provided. Absent otherwise.
messageIdentifier	Message Identifier from the G-TRANSFER-CONFIRMED request primitive.
messageReference	Message Reference from the G-TRANSFER-CONFIRMED request primitive, if provided. Absent otherwise.
confirmation	“confirmedReq”
multiShot	“maintained”
sender	Sender identity if requested in the G-TRANSFER-CONFIRMED request. Absent otherwise.
userData	User Data from the G-TRANSFER-CONFIRMED request primitive, if provided. Absent otherwise.

Table 1.3-14

D-START request parameter	**Status**	**Value**
Called Peer ID	M	One item of the Recipient List parameter value from the G-TRANSFER-CONFIRMED request.
Calling Peer ID	U	Not used.
DS-User Version Number	U	GACS-User Version Number from the G-TRANSFER-CONFIRMED request, if provided. Absent otherwise.
Security Requirements	U	Security Requirements from the G-TRANSFER-CONFIRMED request, if provided. Absent otherwise.
QOS: Routing Class	M	*Class of Communication* from the G-TRANSFER-CONFIRMED request.
QOS: Priority	M	*Priority* from the G-TRANSFER-CONFIRMED request.
QOS: Residual Error Rate	M	*RER* from the G-TRANSFER-CONFIRMED request, if provided. “low” otherwise.

D-START request parameter	Status	Value
User Data	U	Absent

1.3.9.3.6 If a G-TRANSFER-CONFIRMED request primitive is invoked with the Requested Level of Service parameter containing the abstract value “Multi shot, error recovery, confirmed service”, and the GACS entity is in the Associated state (STA 4), then the GACS entity shall:

a) Reset timer t_inact,

b) Create a GACSpdu APDU with field values defined in Table 1.3-13.

c) At the supporting Dialogue Service boundary, invoke a D-DATA request with the GACSpdu APDU as the D-DATA *User Data* parameter value, and

d) Remain in the Associated state (STA 4).

1.3.9.4 G-END Request primitive invoked by GACS-User

1.3.9.4.1 When invoked by the GACS-User, if the Recipient List parameter of the G-END Request contains more than one element then the GACS protocol entity shall select each recipient in the list in turn and, for each recipient, behave as if the G-END Request primitive had been invoked with only that element present in the Recipient List parameter.

1.3.9.4.2 If a G-END request primitive is invoked and the GACS entity for the Dialogue which corresponds to the Recipient List entry is in the Associated state (STA 4), then the GACS entity shall:

a) Reset timer t_inact to value zero,

b) Create a GACSpdu APDU with field values defined in Table 1.3-15,

c) At the supporting Dialogue Service boundary, invoke a D-END request for the dialogue which corresponds to the Recipient List entry, with the GACSpdu APDU as the D-END *User Data* parameter value, and

d) Enter the Ending state (STA 3).

Table 1.3-15

GACSpdu field	Value
messageType	Message Type from the G-END request primitive, if provided. Absent otherwise.
messageIdentifier	Absent.
messageReference	Message Reference from the G-END request primitive, if provided. Absent otherwise.
confirmation	Absent (default value “unconfirmedReq”).
multiShot	“end”
sender	Sender identity.
userData	User Data from the G-END request primitive, if provided. Absent otherwise.

1.3.9.4.3 If a G-END request primitive is invoked and the GACS entity for the Dialogue which corresponds to the Recipient List entry is in the Start-multi state (STA 2), then the GACS entity shall:

a) Invoke a D-ABORT request for the Dialogue which corresponds to the Recipient List entry, with the D-ABORT Originator parameter set to the abstract value “user”, and no User Data, and

b) Enter the Idle state (STA 0).

1.3.9.5 D-UNIT-DATA Indication primitive invoked by supporting service

1.3.9.5.1 Upon receipt of a D-UNIT-DATA indication primitive, if the GACS entity is in the Idle state (STA 0), and the APDU contained in the D-UNIT-DATA User Data parameter is a valid GACSpdu APDU with the confirmation APDU-element set to the abstract value “unconfirmedReq,” and the multishot APDU-element set to the abstract value “single”, then the GACS entity shall:

a) Deliver a G-TRANSFER indication to the local GACS-User, with parameters as defined in Table 1.3-16.

b) Remain in the Idle state (STA 0).

Table 1.3-16

G-TRANSFER indication parameter	**Status**	**Value**
Sender	C(=)	The GACSpdu sender parameter value, if provided. Absent otherwise.
Message Type	C(=)	The GACSpdu messageType parameter value, if provided. Absent otherwise.
Message Identifier	C(=)	The GACSpdu messageIdentifier parameter value, if provided. Absent otherwise.
Message Reference	C(=)	The GACSpdu messageReference parameter value, if provided. Absent otherwise.
GACS-User Version Number	C(=)	DS-User Version Number from the D-UNIT-DATA indication, if provided. Absent otherwise.
Security Requirements	C(=)	Security Requirements from the D-UNIT-DATA indication, if provided. Absent otherwise.
Class of Communication	M(=)	The Routing Class field from the Quality of Service parameter of the D-UNIT-DATA indication.
Priority	M(=)	The Priority field from the Quality of Service parameter of the D-UNIT-DATA indication.
RER	C(=)	The Residual Error Rate field from the Quality of Service parameter of the D-UNIT-DATA indication.
Requested Level of Service	M(=)	“Single shot, no error recovery, unconfirmed service”
User Data	C(=)	userData field from the GACSpdu if provided. Absent otherwise.

1.3.9.5.2 Upon receipt of a D-UNIT-DATA indication primitive, if the GACS entity is in the Idle state (STA 0), and the APDU contained in the D-UNIT-DATA User Data parameter is a valid GACSpdu APDU with the confirmation APDU-element set to the abstract value “confirmedReq,” and the multishot APDU-element set to the abstract value “single”, then the GACS entity shall:

a) Deliver a G-TRANSFER-CONFIRMED indication to the local GACS-User, with parameters as defined in Table 1.3-17.

b) Create a confirmation GACSpdu APDU with field values defined in Table 1.3-18,

c) At the supporting Dialogue Service boundary, invoke a D-UNIT-DATA request primitive with parameters as defined in Table 1.3-19.

d) Remain in the Idle state (STA 0).

Table 1.3-17

G-TRANSFER-CONFIRMED indication parameter	**Status**	**Value**
Sender	C(=)	The GACSpdu sender parameter value, if provided. Absent otherwise.
Message Type	C(=)	The GACSpdu messageType parameter value, if provided. Absent otherwise.
Message Identifier	M(=)	The GACSpdu messageIdentifier parameter value.
Message Reference	C(=)	The GACSpdu messageReference parameter value, if provided. Absent otherwise.
GACS-User Version Number	C(=)	DS-User Version Number from the D-UNIT-DATA indication, if provided. Absent otherwise.
Security Requirements	C(=)	Security Requirements from the D-UNIT-DATA indication, if provided. Absent otherwise.
Class of Communication	M(=)	The Routing Class field from the Quality of Service parameter of the D-UNIT-DATA indication.
Priority	M(=)	The Priority field from the Quality of Service parameter of the D-UNIT-DATA indication.
RER	C(=)	The Residual Error Rate field from the Quality of Service parameter of the D-UNIT-DATA indication.
Requested Level of Service	M(=)	“Single shot, no error recovery, confirmed service”
User Data	C(=)	userData field from the GACSpdu if provided. Absent otherwise.

Table 1.3-18

GACSpdu field	**Value**
messageType	The messageType parameter value in the received GACSpdu, if provided. Absent otherwise.
messageIdentifier	Absent.
messageReference	Message Identifier from the received GACSpdu.
confirmation	If the G-TRANSFER-CONFIRMED indication is capable of being delivered then “confirmedOk”, otherwise “confirmedNotOk”
multiShot	Absent. (Default value “single”).
sender	Sender identity.
userData	Absent.

Table 1.3-19

D-UNIT-DATA request parameter	Status	Value
Called Peer ID	M	The location of the sender of the received D-UNIT-DATA indication.
Calling Peer ID	U	Not used.
DS-User Version Number	U	As for the GACS-User Version Number in the G-TRANSFER-CONFIRMED indication (Table 1.3-17).
Security Requirements	U	As for the Security Requirements in the G-TRANSFER-CONFIRMED indication (Table 1.3-17).
QOS: Routing Class	M	As for the Class of Communication in the G-TRANSFER-CONFIRMED indication (Table 1.3-17).
QOS: Priority	M	As for Priority in the G-TRANSFER-CONFIRMED indication (Table 1.3-17).
QOS: Residual Error Rate	M	As for RER in the G-TRANSFER-CONFIRMED indication (Table 1.3-17).
User Data	M	The GACSpdu APDU created according to Table 1.3-18.

1.3.9.5.3 Upon receipt of a D-UNIT-DATA indication primitive, if the GACS entity is in the Idle state (STA 0), and the APDU contained in the D-UNIT-DATA User Data parameter is a valid GACSpdu APDU with the confirmation APDU-element set to one of the abstract values (“confirmedOk”, “confirmedNotOk”, or “confirmedUnknown”) and no userData APDU-element present, then the GACS entity shall:

a) Deliver a G-TRANSFER-CONFIRMED confirmation to the local GACS-User, with parameters as defined in Table 1.3-20, and

b) Remain in the Idle state (STA 0).

Table 1.3-20

G-TRANSFER-CONFIRMED confirmation parameter	Status	Value
Sender	M	The GACSpdu sender parameter value.
Message Type	C(=)	The GACSpdu messageType parameter value, if provided. Absent otherwise.
Message Reference	M	The GACSpdu messageReference parameter value.
GACS-User Version Number	C	The DS-User Version Number from the D-UNIT-DATA indication, if provided. Absent otherwise.
Result	M	“Successful delivery”, “Delivery failed”, or “Service problem - delivery status uncertain”, for values of the GACSpdu confirmation parameter of “confirmedOk”, “confirmedNotOk”, or “confirmedUnknown”, respectively.

1.3.9.6 D-START Indication primitive delivered by supporting service

1.3.9.6.1 Upon receipt of a D-START indication primitive, if the GACS entity is in the Idle state (STA 0), and there is no D-START User Data parameter present, then the GACS entity shall:

a) At the supporting Dialogue Service boundary, invoke a D-START response primitive with parameters as defined in Table 1.3-21, and

b) If the Result parameter is "accepted" then enter the Associated state (STA 4), otherwise remain in the Idle state (STA 0).

Table 1.3-21

D-START response parameter	**Status**	**Value**
DS-User Version Number	U	The local GACS-User Version Number, if available. Absent otherwise.
Security Requirements	U	The local Security Requirements, if available. Absent otherwise.
QOS: Routing Class	U	*Routing Class* from the Quality of Service parameter of the received D-START indication
QOS: Priority	U	*Priority* from the Quality of Service parameter of the received D-START indication
QOS: Residual Error Rate	U	*Residual Error Rate* from the Quality of Service parameter of the received D-START indication.
Result	M	If the local GACS entity is able to accept the dialogue, then "accepted". If, for local reasons, the GACS entity is unable to accept the dialogue, then "rejected (transient)" or "rejected (permanent)", depending upon severity.
User Data	U	Absent

1.3.9.6.2 Upon receipt of a D-START indication primitive, if the GACS entity is in the Idle state (STA 0), and the APDU contained in the D-START User Data parameter is a valid GACSpdu APDU with the confirmation APDU-element set to the abstract value "unconfirmedReq," then the GACS entity shall:

a) Deliver a G-TRANSFER indication to the local GACS-User, with parameters as defined in Table 1.3-22,

b) At the supporting Dialogue Service boundary, invoke a D-START response primitive with parameters as defined in Table 1.3-23, and

c) Remain in the Idle state (STA 0).

Table 1.3-22

G-TRANSFER indication parameter	**Status**	**Value**
Sender	C(=)	The GACSpdu sender parameter value, if provided. Absent otherwise.
Message Type	C(=)	The GACSpdu messageType parameter value, if provided. Absent otherwise.
Message Identifier	C(=)	The GACSpdu messageIdentifier parameter value, if provided. Absent otherwise.

G-TRANSFER indication parameter	Status	Value
Message Reference	C(=)	The GACSpdu messageReference parameter value, if provided. Absent otherwise.
GACS-User Version Number	C(=)	DS-User Version Number from the received D-START indication, if provided. Absent otherwise.
Security Requirements	C(=)	Security Requirements from the received D-START indication, if provided. Absent otherwise.
Class of Communication	M(=)	The Routing Class field from the Quality of Service parameter of the received D-START indication.
Priority	M(=)	The Priority field from the Quality of Service parameter of the received D-START indication.
RER	C(=)	The Residual Error Rate field from the Quality of Service parameter of the received D-START indication.
Requested Level of Service	M(=)	"single shot, error recovery, unconfirmed service"
User Data	C(=)	userData field from the GACSpdu if provided. Absent otherwise.

Table 1.3-23

D-START response parameter	Status	Value
DS-User Version Number	U	The local GACS-User Version Number, if available. Absent otherwise.
Security Requirements	U	The local Security Requirements, if available. Absent otherwise.
QOS: Routing Class	U	*Routing Class* from the Quality of Service parameter of the received D-START indication
QOS: Priority	U	*Priority* from the Quality of Service parameter of the received D-START indication
QOS: Residual Error Rate	U	*Residual Error Rate* from the Quality of Service parameter of the received D-START indication.
Result	M	"rejected (transient)"
User Data	U	Absent

1.3.9.6.3 Upon receipt of a D-START indication primitive, if the GACS entity is in the Idle state (STA 0), and the APDU contained in the D-START User Data parameter is a valid GACSpdu APDU with the confirmation APDU-element set to the abstract value "confirmedReq," then the GACS entity shall:

a) Deliver a G-TRANSFER-CONFIRMED indication to the local GACS-User, with parameters as defined in Table 1.3-24,

b) Create a confirmation GACSpdu APDU with field values defined in Table 1.3-25,

c) At the supporting Dialogue Service boundary, invoke a D-START response primitive with parameters as defined in Table 1.3-26, and

d) Remain in the Idle state (STA 0).

Table 1.3-24

G-TRANSFER-CONFIRMED indication parameter	Status	Value
Sender	C(=)	The GACSpdu sender parameter value, if provided. Absent otherwise.
Message Type	C(=)	The GACSpdu messageType parameter value, if provided. Absent otherwise.
Message Identifier	M(=)	The GACSpdu messageIdentifier parameter value.
Message Reference	C(=)	The GACSpdu messageReference parameter value, if provided. Absent otherwise.
GACS-User Version Number	C(=)	DS-User Version Number from the received D-START indication, if provided. Absent otherwise.
Security Requirements	C(=)	Security Requirements from the received D-START indication, if provided. Absent otherwise.
Class of Communication	M(=)	The Routing Class field from the Quality of Service parameter of the received D-START indication.
Priority	M(=)	The Priority field from the Quality of Service parameter of the received D-START indication.
RER	C(=)	The Residual Error Rate field from the Quality of Service parameter of the received D-START indication.
Requested Level of Service	M(=)	“single shot, error recovery, confirmed service”
User Data	C(=)	userData field from the GACSpdu if provided. Absent otherwise.

Table 1.3-25

GACSpdu field	Value
messageType	As for the Message Type in the G-TRANSFER-CONFIRMED indication (Table 1.3-24).
messageIdentifier	Absent.
messageReference	Message Identifier from the received GACSpdu.
confirmation	If the G-TRANSFER-CONFIRMED indication is capable of being delivered then “confirmedOk”, otherwise “confirmedNotOk”.
multiShot	Absent. (Default value “single”).
sender	Sender identity.
userData	Absent.

Table 1.3-26

D-START response parameter	Status	Value
DS-User Version Number	U	The local GACS-User Version Number, if available. Absent otherwise.
Security Requirements	U	The local Security Requirements, if available. Absent otherwise.
QOS: Routing Class	U	*Routing Class* from the Quality of Service parameter of the received D-START indication
QOS: Priority	U	*Priority* from the Quality of Service parameter of the received D-START indication
QOS: Residual Error Rate	U	*Residual Error Rate* from the Quality of Service parameter of the received D-START indication.
Result	M	"rejected (transient)"
User Data	U	The GACSpdu APDU (Table 1.3-25)

1.3.9.7 D-START Confirmation primitive delivered by supporting service

1.3.9.7.1 Upon receipt of a D-START Confirmation primitive, if the GACS entity is in the Start-multi state (STA 2), and the D-START Result parameter has the abstract value "accepted," and the D-START User Data parameter is absent, then the GACS entity shall:

a) At the supporting Dialogue Service boundary, invoke a D-DATA request primitive with the previously-formed and stored GACSpdu APDU (see 1.3.9.2.5 or 1.3.9.3.5) as the User Data parameter.

b) Start the T_inact inactivity timer from value zero,

c) Enter the Associated state (STA 4).

1.3.9.7.2 Upon receipt of a D-START Confirmation primitive, if the GACS entity is in the Start-single state (STA 1) or the Start-multi state (STA 2), and the D-START Result parameter has the abstract value "rejected (transient)" or "rejected (permanent)," then the GACS entity shall:

a) If the local GACS-User is waiting for confirmation of a previous G-TRANSFER-CONFIRMED request, and the User Data parameter of the received D-START Confirmation contains a valid GACSpdu APDU, then deliver a G-TRANSFER-CONFIRMED confirmation to the local GACS-User, with parameters as defined in Table 1.3-27.

b) If the local GACS-User is waiting for confirmation of a previous G-TRANSFER-CONFIRMED request, and the User Data parameter of the received D-START Confirmation does not contain a valid GACSpdu APDU, then deliver a G-TRANSFER-CONFIRMED confirmation to the local GACS-User, with parameters as defined in Table 1.3-28.

c) Enter the Idle state (STA 0).

Table 1.3-27

G-TRANSFER-CONFIRMED confirmation parameter	**Status**	**Value**
Sender	M	The GACSpdu sender parameter value.
Message Type	C(=)	The GACSpdu messageType parameter value, if provided. Absent otherwise.
Message Reference	M	The GACSpdu messageReference parameter value.
GACS-User Version Number	C	DS-User Version Number from the received D-START confirmation, if provided. Absent otherwise.
Result	M	"Successful delivery" if the confirmation field of the received GACSpdu has value "confirmedOk". "Delivery failed" otherwise.

Table 1.3-28

G-TRANSFER-CONFIRMED confirmation parameter	**Status**	**Value**
Sender	M	The local location identifier
Message Type	C(=)	Absent
Message Reference	M	A local default value.
GACS-User Version Number	C	DS-User Version Number from the received D-START confirmation, if provided. Absent otherwise.
Result	M	"Service problem - delivery status uncertain""

1.3.9.8 D-DATA Indication primitive delivered by supporting service

1.3.9.8.1 Upon receipt of a D-DATA Indication primitive, if the GACS entity is in the Associated state (STA 4), and the D-DATA User Data parameter contains a valid GACSpdu APDU with the confirmation APDU-element set one of the values ("confirmedOk", "confirmedNotOk" or "confimedUnknown") then the GACS entity shall:

a) Reset the T_inact inactivity timer to value zero,

b) If the local GACS-User had previously issued a G-TRANSFER-CONFIRMED request, and the messageReference field of the received GACS APDU matches the Message Identifier parameter of that request, then deliver a G-TRANSFER-CONFIRMED confirmation to the local GACS-User, with parameters as defined in Table 1.3-29.

c) Remain in the Associated state (STA 4).

Table 1.3-29

G-TRANSFER-CONFIRMED confirmation parameter	**Status**	**Value**
Sender	M	The GACSpdu sender parameter value.

G-TRANSFER-CONFIRMED confirmation parameter	**Status**	**Value**
Message Type	C(=)	The GACSpdu messageType parameter value, if provided. Absent otherwise.
Message Reference	M	The GACSpdu messageReference parameter value.
GACS-User Version Number	C	The peer DS-User Version Number obtained when the dialogue was established, if known. Absent otherwise.
Result	M	"Successful delivery" if the confirmation field of the received GACSpdu has value "confirmedOk". "Delivery failed" otherwise.

1.3.9.8.2 Upon receipt of a D-DATA Indication primitive, if the GACS entity is in the Associated state (STA 4) or the Ending state (STA 3), and the D-DATA User Data parameter contains a valid GACSpdu APDU with the confirmation APDU-element set to the abstract value "unconfirmedReq," then the GACS entity shall:

a) Reset the T_inact inactivity timer to value zero,

b) Deliver a G-TRANSFER indication to the local GACS-User, with parameters as defined in Table 1.3-30, and

c) Remain in the same state.

Table 1.3-30

G-TRANSFER indication parameter	**Status**	**Value**
Sender	C(=)	The GACSpdu sender parameter value, if provided. Absent otherwise.
Message Type	C(=)	The GACSpdu messageType parameter value, if provided. Absent otherwise.
Message Identifier	C(=)	The GACSpdu messageIdentifier parameter value, if provided. Absent otherwise.
Message Reference	C(=)	The GACSpdu messageReference parameter value, if provided. Absent otherwise.
GACS-User Version Number	C(=)	If provided when the dialogue was established, the peer DS-User Version Number. Absent otherwise.
Security Requirements	C(=)	If provided when the dialogue was established, the peer Security Requirements. Absent otherwise.
Class of Communication	M(=)	The Routing Class field from the Quality of Service parameter when the dialogue was established.
Priority	M(=)	The Priority field from the Quality of Service parameter when the dialogue was established.
RER	C(=)	The Residual Error Rate field from the Quality of Service parameter when the dialogue was established.

G-TRANSFER indication parameter	Status	Value
Requested Level of Service	M(=)	"multi shot, error recovery, unconfirmed service"
User Data	C(=)	userData field from the GACSpdu if provided. Absent otherwise.

1.3.9.8.3 Upon receipt of a D-DATA Indication primitive, if the GACS entity is in the Associated state (STA 4) or the Ending state (STA 3), and the D-DATA User Data parameter contains a valid GACSpdu APDU with the confirmation APDU-element set to the abstract value "confirmedReq," then the GACS entity shall:

a) Reset the T_inact inactivity timer to value zero,

b) Deliver a G-TRANSFER-CONFIRMED indication to the local GACS-User, with parameters as defined in Table 1.3-31, and

c) If in the Associated state (STA 4):

1) create a confirmation GACSpdu APDU with field values defined in Table 1.3-18, and

2) at the supporting Dialogue Service boundary, invoke a D-DATA request primitive with the newly created GACSpdu APDU as the User Data parameter.

d) Remain in the same state.

Table 1.3-31

G-TRANSFER-CONFIRMED indication parameter	Status	Value
Sender	C(=)	The GACSpdu sender parameter value, if provided. Absent otherwise.
Message Type	C(=)	The GACSpdu messageType parameter value, if provided. Absent otherwise.
Message Identifier	M(=)	The GACSpdu messageIdentifier parameter value.
Message Reference	C(=)	The GACSpdu messageReference parameter value, if provided. Absent otherwise.
GACS-User Version Number	C(=)	If provided when the dialogue was established, the peer DS-User Version Number. Absent otherwise.
Security Requirements	C(=)	If provided when the dialogue was established, the peer Security Requirements. Absent otherwise.
Class of Communication	M(=)	The Routing Class field from the Quality of Service parameter when the dialogue was established.
Priority	M(=)	The Priority field from the Quality of Service parameter when the dialogue was established.
RER	C(=)	The Residual Error Rate field from the Quality of Service parameter when the dialogue was established.
Requested Level of Service	M(=)	"multi shot, error recovery, confirmed service"

G-TRANSFER-CONFIRMED indication parameter	Status	Value
User Data	C(=)	userData field from the GACSpdu if provided. Absent otherwise.

1.3.9.9 D-END Indication primitive delivered by supporting service

1.3.9.9.1 Upon receipt of a D-END Indication primitive, if the GACS entity is in the Associated state (STA 4), then the GACS entity shall:

a) Stop the T_inact inactivity timer,

b) Deliver a G-END indication to the local GACS-User, with parameters as defined in Table 1.3-32,

c) At the supporting Dialogue Service boundary, invoke a D-END response primitive with the Result parameter set to “accepted” and no User Data parameter, and

d) Enter the Idle state (STA 0).

Table 1.3-32

G-END indication parameter	Status	Value
Sender	M	If the D-END User Data contains a GACSpdu, the value of the GACSpdu *sender* parameter, if present. Otherwise the assumed sender based on local context information.
Message Type	C(=)	If the D-END User Data contains a GACSpdu, the value of the GACSpdu *messageType* parameter, if present. Absent otherwise.
Message Reference	C(=)	If the D-END User Data contains a GACSpdu, the value of the GACSpdu *messageReference* parameter, if present. Absent otherwise.
User Data	C(=)	If the D-END User Data contains a GACSpdu, the value of the GACSpdu *userData* parameter, if present. Absent otherwise.

1.3.9.10 D-END Confirmation primitive delivered by supporting service

1.3.9.10.1 Upon receipt of a D-END Confirmation primitive with the result parameter having value “accepted”, if the GACS entity is in the Ending state (STA 3), then the GACS entity shall:

a) Stop the T_inact inactivity timer,

b) Enter the Idle state (STA 0).

1.3.9.11 D-ABORT Indication or D-P-ABORT primitive delivered by supporting service

1.3.9.11.1 Upon receipt of a D-ABORT Indication or D-P-ABORT Indication primitive, for a dialogue which is not in the Idle state (STA 0), the GACS entity shall:

a) Stop the T_inact inactivity timer if it is running,

b) If in state Start-single (STA 1) or Start-multi (STA 2), and there is an outstanding G-TRANSFER-CONFIRMED confirmation, then deliver a G-TRANSFER-CONFIRMED confirmation to the local GACS-User, with parameters as defined in Table 1.3-33.

c) If in state Associated (STA 4), then deliver a G-END indication to the local GACS-User, with parameters as defined in Table 1.3-34,

d) Enter the Idle state (STA 0).

Table 1.3-33

G-TRANSFER-CONFIRMED confirmation parameter	**Status**	**Value**
Sender	M	The location of the peer GACS entity based on local context information.
Message Type	C(=)	Absent.
Message Reference	M	Local value indicating abnormal termination.
GACS-User Version Number	C	Absent.
Result	M	“Service problem - delivery status uncertain”

Table 1.3-34

G-END indication parameter	**Status**	**Value**
Sender	M	The location of the peer GACS entity based on local context information.
Message Type	C(=)	Absent.
Message Reference	C(=)	Absent.
User Data	C(=)	Absent.

1.3.9.12 Timer T_inact expiry

1.3.9.12.1 If the inactivity timer T_inact expires for a given dialogue, the GACS entity shall:

a) If in state Associated (STA 4), then

1) Reset T_inact to value zero,

2) Create a GACSpdu APDU with field values defined in Table 1.3-35,

3) At the supporting Dialogue Service boundary, invoke a D-END request with the GACSpdu APDU as the D-END *User Data* parameter value, and

4) Enter the Ending state (STA 3).

b) If in state Ending (STA 3), then

1) Stop T_inact,

2) At the supporting Dialogue Service boundary, invoke a D-ABORT request primitive with the originator parameter set to “provider” and no User Data parameter.

3) Enter the Idle state (STA 0).

Table 1.3-35

GACSpdu field	**Value**
messageType	Absent.
messageIdentifier	Absent.
messageReference	Absent.
confirmation	Absent (default value “unconfirmedReq”).
multiShot	“end”
sender	Sender identity.
userData	Absent.

1.4 Communication Requirements

Note 1.— This section contains the requirements that the GACS ASE application imposes on the underlying communication system.

Note 2.— The GACS specification makes use of the Dialogue Service (DS) as defined in [ULCS] 4.2. This is the abstract service which ATN Application ASEs use to interact with the UL communications service. That is, the DS is the combination of specific internal primitives made available by the Control Function (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.

1.4.1 Use of Connectionless Provider

1.4.1.1 When a connectionless communications provider is available and selected, GACS service primitives shall map onto D-UNIT-DATA service primitives.

Note.— There are constraints on the connectionless service data length. The Transport CL user data is limited to 63,488 octets per TSDU, and this limit will be reflected in the GACS service. If a GACSpdu APDU exceeds the size constraints of the D-UNIT-DATA service, then it will be necessary to use a connection-oriented communications provider, if available. Otherwise, it is the responsibility of the GACS-User to ensure that size constraints are respected.

1.4.2 Use of Connection-Oriented Provider

1.4.2.1 When a connection-oriented communications provider is available and selected, GACS service primitives shall map onto the connection-oriented Dialogue Service primitives (D-START, D-DATA, D-END, D-ABORT and D-P-ABORT).

1.4.3 Mapping to Dialogue Service Parameters

1.4.3.1 The GACS parameter Class of Communication shall map to the D-START or D-UNIT-DATA Routing Class QoS parameter as specified in Table 1.4-1.

Table 1.4-1. Routing Class Values

Abstract Class of Communication	**Routing Class Value (Hex)**
ATS: No Traffic Type Policy Preference	01
ATS: Traffic preference for Class A ATSC route(s)	10
ATS: Traffic preference for Class B ATSC route(s)	11
ATS: Traffic preference for Class C ATSC route(s)	12
ATS: Traffic preference for Class D ATSC route(s)	13
ATS: Traffic preference for Class E ATSC route(s)	14
ATS: Traffic preference for Class F ATSC route(s)	15
ATS: Traffic preference for Class G ATSC route(s)	16

ATS: Traffic preference for Class H ATSC route(s)	17
AOC: No Traffic Type Policy Preference	21
AOC: Route traffic only via Gatelink	22
AOC: Route traffic only via VHF Data Link	23
AOC: Route traffic only via Satellite Data Link	24
AOC: Route traffic only via HF Data Link	25
AOC: Route traffic only via Mode S Data Link	26
AOC: Route traffic using an ordered preference of Gatelink first, then VHF Data Link	27
AOC: Route traffic using an ordered preference of Gatelink first, then VHF Data Link, then Satellite	28
AOC: Route traffic using an ordered preference of Gatelink first, then VHF Data Link, then HF Data Link, then Satellite Data Link.	29
ATN Administrative Communications	30
General Communications	(absent)
ATN Systems Management Communications	60

1.4.3.2 The GACS parameter Priority shall map to the D-START or D-UNIT-DATA Priority QoS parameter as specified in Table 1.4-2.

Table 1.4-2. Priority parameter values

Abstract Priority Value	**QoS Priority Value (Decimal)**
Network / Systems Management	14
Distress Communications	13
Urgent Communications	12
High Priority Flight Safety Messages	11
Normal Priority Flight Safety Messages	10
Meteorological Communications	9
Flight Regularity Communications	8
Aeronautical Information Service Messages	7
Network / Systems Administration	6
Aeronautical Administrative Messages	5
Urgent Priority Administrative and U.N. Charter Communications	3
High Priority Administrative and State / Government Communications	2
Normal Priority Administrative	1
Low Priority Administrative	0

1.4.3.3 Each element of the GACS parameter Recipient List shall map to different invocations of the D-START or D-UNIT-DATA Called Peer ID parameter.

1.4.3.4 The GACS parameter Sender, if present, shall map to the D-START or D-UNIT-DATA Calling Peer ID parameter.

1.4.3.5 The GACS parameter User Version Number shall map to the D-START or D-UNIT-DATA DS-User Version Number parameter. If no User Version Number is specified, then DS-User Version Number shall be set to the value 1, indicating that this is the first version of the GACS protocol.

1.4.3.6 The GACS parameter Security Requirements shall map to the D-START or D-UNIT-DATA Security Requirements parameter.

1.4.3.7 The GACS parameter RER shall map to the D-START or D-UNIT-DATA RER QoS parameter. If no RER is specified, then the DS RER QoS parameter shall be set to the value low.

1.4.3.8 The GACS PDU, as defined in section 1.3.5 and 1.3.6 shall map to the D-DATA, D-START, D-END or D-UNIT-DATA User Data parameter.

1.5 User Requirements

Note.— This section outlines the requirements that a user of a GACS ASE must meet.

1.5.1 ***Recommendation**.— A configurable timer should be implemented so that the GACS-User can take appropriate action if no confirmation is received after issuing a D-TRANSFER-CONFIRMED request.*

1.6 Subsetting Rules

Note.— This section specifies conformance requirements which all implementations of the GACS protocol obey.

1.6.1 An implementation of the GACS service claiming conformance to the protocol specified in this document shall be realised either as a GACS-AE or a GACS-ASO as shown in Table 1.6-1.

Table 1.6-1. Conformant realisations

Conformant Configuration Ref.	**Functionality Description**
GACS-AE	GACS is a complete addressable ATN application accessible by user applications via an exposed interface.
GACS-ASO	GACS is an element of the ATN upper layers which is not directly addressable. It provides an enhanced dialogue service to user ASEs.

1.6.2 An implementation of the GACS service claiming conformance to the protocol specified in this document shall support the GACS protocol features as shown in Table 1.6-2.

Table 1.6-2. Conformant communication subsets

Conformant Configuration Ref.	**Functionality Description**
I	Both connection-oriented and connectionless underlying stacks supported.
II	Only connection-oriented underlying stack supported.
III	Only connectionless underlying stack supported.

Note 1.— The GACS service user is not made aware which of the subsets is implemented.

Note 2.— Configurations I and III give more efficient handling of the "single shot, no error recovery" level of service.

Note 3.— Implementations of Configuration I will not be able to interwork with implementations of Configuration II in cases where the "single shot, no error recovery" level of service is invoked by the GACS service user. Implementations of Configuration II will always be able to interwork with implementations of Configuration I.

Note 4.— Implementations of Configuration III will only be able to support the "single shot, no error recovery" level of service, and will not be able to interwork with implementations of Configuration II. Implementations of Configuration III will always be able to interwork with implementations of Configuration I.

2. GENERIC ATN COMMUNICATION SERVICE GUIDANCE AND RATIONALE

2.1 Introduction

This document contains a description of the GACS services and procedures which may be used to support a wide range of Air Traffic Service (ATS), Aeronautical Operational Control (AOC), and Aeronautical Administrative Communications (AAC) applications. It also includes a description of the supporting protocol stacks.

GACS is an Application Layer message protocol, developed in the framework of the OSI architecture, which provides a flexible message service suitable for existing and future air-ground and ground-ground aeronautical message applications. It provides a basic message transfer capability which allows existing message headers and formats to be exchanged with a minimum amount of protocol overhead in a standardised manner. This capability allows existing applications to transition to an OSI environment.

The Aeronautical Telecommunication Network (ATN) protocol services are based on the Open Systems Interconnection (OSI) architecture. Network related protocols and Transport protocols, which include layers 1 through 4 of the OSI reference model, comprise the internet communication service (ICS) of the ATN. These protocols include both connection-oriented (CO) and connectionless (CL) services. The type and operation of the network protocols (or lower layer protocols) should be transparent to applications which operate over them.

Within the framework of the OSI architecture, all end systems must support peer-to-peer (i.e. the same) protocols from the Transport Layer and above (or upper layer protocols).

The Session, Presentation, and Application Layer services directly support the application processes for dialogue, information syntax requirements (both abstract and transfer), and association requirements. These protocols must, therefore, be selected to accommodate the aeronautical application process data exchange requirements.

2.2 Background

End Systems participating in the ATN support applications which require the automated exchange of messages relating to aeronautical services such as ATS, AOC, AAC, and other possible types of aeronautical services.

Each of these services has distinct message exchange requirements which are based on the operational objectives of that service. ATS services may, for example, require the fast, reliable exchange and confirmation of tactical altitude change messages from a ground end system to an aircraft end system. An airline administrative service, however, may require the routine exchange of unconfirmed messages with many aircraft. Consequently, the ATN message exchanges vary in type, length, transfer time requirements, frequency of transmission, response requirements and other communications requirements.

The need exists, therefore, for a message transfer service which would support the requirements of various applications, regardless of the transfer characteristics of the messages.

Ideally, one message transfer standard which accommodates all message transfer requirements would be desirable to help facilitate interoperability in a cost effective manner.

The intent of this document is to define a message transfer standard which would accommodate most aeronautical message applications.

Any proposed solution for a generic message transfer standard within the ATN environment should consider the message transfer requirements of most aeronautical applications.

This document therefore, recommends one common message transfer service that could be used to exchange multiple message types with various formats and bit encodings.

2.3 Development Approach for GACS

The basic approach used to develop the GACS service was to define the message services and procedures separate from the bit encodings used to exchange the Protocol Control Information (PCI) and the message text. The definition of the services and procedures do, however, take into account the need to minimise the amount of encoded PCI exchanged over the air/ground sub-networks.

The services and procedures are designed to accommodate multiple message types and text formats. Specific aeronautical message text formats are not defined here. The GACS services are designed specifically to allow the definition of message text formats outside of the protocol by external organisations.

The GACS development approach can be summarised into the following set of objectives and guidelines:

a) Examine existing and future aeronautical operational scenarios and message exchange characteristics, and define a set of services and procedures that would accommodate most aeronautical message requirements.

b) Define a generic aeronautical message header that would accommodate existing as well as future message types and formats and allow these messages to operate within a common OSI framework.

c) Minimise the amount of communications functions within the application process.

d) Use existing standardised protocols and protocol implementation agreements when possible.

e) Define the protocol services and procedures separate from the encoding rules used to generate the transfer syntax (bit patterns).

f) Define the protocol services, procedures, and abstract syntax using standard definition tools.

g) Identify and/or define one or more set(s) of encoding rules which may be used with the message service abstract syntax which takes into account the limited bandwidth of the underlying air/ground sub-networks.

2.4 Aeronautical Messaging Requirements

Previous studies have determined a generic set of message transfer requirements for aeronautical applications which could be used as a basis for determining the services of a standard message protocol for the ATN. Those requirements are summarised below:

a) Ability to send multiple message formats (for both existing and future applications).

b) Ability to convey security information.

c) Ability to convey application message priorities.

d) Ability to send various types of message requests and receive a response for the request.

e) Ability to send messages with variable sizes.

f) Ability to exchange various message types with different data syntaxes and encoding.

g) Ability to convey quality of service requirements (e.g. transfer time constraints) to the communications facility.

h) Ability to exchange a message in a real-time application environment (i.e. facilitate the immediate exchange of data to peer applications as opposed to store/forward types of environments).

i) Ability to minimise message header (i.e. protocol and encoding) overhead.

j) Ability to exchange data in a CO or CL mode of application association.

2.5 GACS Service Overview

The Generic ATN Communication Service (GACS) is defined within the framework of the OSI architecture. It is an Application Layer service that is intended for use by aeronautical messaging applications. The services have been designed to meet the aeronautical message transfer requirements listed in a previous section of this paper.

The following are examples of categories of existing formats which may be exchanged:

- ACARS/COP
- ICM/TYPE A
- ICM/TYPE B
- FAA NAS
- FAA WX
- FAA NADIN MSN
- ICAO/AFTN IA-5
- ICAO/AFTAN ITA-2
- WMO

2.5.1 G-TRANSFER Service

When a connectionless communications provider is selected, the G-TRANSFER request primitive results in the transfer of the GACS User Data as User Data of the connectionless service.

When a connection-oriented communications provider is selected, and a single-shot Level of Service is requested, the G-TRANSFER request primitive results in the following sequence of events:

a) A connection is established using the D-START service. A GACS PDU is formed and mapped to the D-START request User Data.

b) On receiving a D-START indication containing User Data, the GACS service maps this to a G-TRANSFER indication primitive, which it passes to the GACS-User, and also invokes a negative D-START response, which will close the connection.

c) On receiving a negative D-START confirmation, the Initiator takes no further action.

When a connection-oriented communications provider is selected, and a multi-shot Level of Service is requested, the G-TRANSFER request primitive results in the following sequence of events:

a) The GACS entity forms a GACS PDU from the parameters of the G-TRANSFER request, including the G-TRANSFER User Data

b) If a connection to the addressed peer with the requested Quality of Service already exists, then the GACS PDU is mapped to the User Data of a D-DATA request primitive to send it over that connection.

c) If no connection to the addressed peer with the requested Quality of Service already exists, then a connection is first established using the D-START service, with no User Data.

d) On receiving a D-START indication containing no User Data, the GACS service invokes a positive D-START response, which will open the connection.

e) On receiving a positive D-START confirmation, the GACS service maps the GACS PDU to the User Data of a D-DATA request primitive.

f) On receiving a D-DATA indication, the GACS service maps this to a G-TRANSFER indication primitive, which it passes to the GACS-User.

2.5.2 G-TRANSFER-CONFIRMED Service

When a connection-oriented communications provider is selected, and a single-shot Level of Service is requested, the G-TRANSFER-CONFIRMED request primitive results in the following sequence of events:

a) A connection is established using the D-START service. A GACS PDU is formed and mapped to the D-START request User Data.

b) On receiving a D-START indication containing User Data, the GACS service maps this to a G-TRANSFER-CONFIRMED indication primitive, which it passes to the GACS-User, and also invokes a negative D-START response containing a GACS PDU with a positive confirmation as User Data, which will close the connection.

c) On receiving a negative D-START confirmation, the GACS service maps this to a G-TRANSFER-CONFIRMED confirmation primitive, which it passes to the GACS-User.

When a connection-oriented communications provider is selected, and a multi-shot Level of Service is requested, the G-TRANSFER-CONFIRMED request primitive results in the following sequence of events:

a) The GACS entity forms a GACS PDU from the parameters of the G-TRANSFER-CONFIRMED request, including the G-TRANSFER User Data

b) If a connection to the addressed peer with the requested Quality of Service already exists, then the GACS PDU is mapped to the User Data of a D-DATA request primitive to send it over that connection.

c) If no connection to the addressed peer with the requested Quality of Service already exists, then a connection is established using the D-START service, with no User Data

d) On receiving a D-START indication containing no User Data, the GACS service invokes a positive D-START response, which will open the connection.

e) On receiving a positive D-START confirmation, the GACS service maps the GACS PDU to the User Data of a D-DATA request primitive.

f) On receiving a D-DATA indication, the GACS service maps this to a G-TRANSFER-CONFIRMED indication primitive, which it passes to the GACS-User. The GACS service also invokes a D-DATA request with a GACS PDU containing a confirmation of receipt as User Data..

g) On receiving a D-DATA indication containing a confirmation of receipt, the GACS service maps this to a G-TRANSFER-CONFIRMED confirmation primitive, which it passes to the GACS-User.

2.5.3 G-END Service

When a G-END request primitive is invoked, the GACS service causes the following sequence of events:

a) If any of the optional parameters *Message Type*, *Message Reference* or *User Data* are present, a GACS PDU is formed.

b) A D-END request primitive is invoked, with the GACS PDU, if any, as User Data.

c) On receiving a D-END indication, the GACS service maps this to a G-END indication primitive, which it passes to the GACS-User, and also invokes a positive D-END response, which will close the connection.

d) On receiving a positive D-END confirmation, the GACS service takes no further action.

2.6 GACS Protocol Stacks

Figure 2.6-1 characterises two OSI protocol stack configurations within which the GACS service can operate. The Association Control Service Element (ACSE) is used to convey application context information.

2.6.1 CO Protocol Stack

Stack 1 depicts the GACS/CO protocol suite which includes connection oriented services for the upper layers of the model including Transport. This protocol suite would be used by applications which require association and session dialogue services where data exchanged over the association may be related and error recovery services are required.

This suite would be beneficial for applications which maintain lengthy or consecutive sessions, and exchange multiple related messages. Error recovery and reliable transport services by the underlying protocols are provided.

2.6.2 CL Protocol Stack

Stack 2 depicts the GACS/CL protocol suite which includes connectionless services for the upper layers of the model including Transport. This protocol configuration would be used by applications which require independent message transfers where messages are not related and error recovery services by the underlying protocols are not required.

This suite might be useful for applications which exchange messages periodically where subsequent messages are infrequent and unrelated, and where error recovery outside the AP is not required.

Acknowledgements for messages transferred over this service would be a separate connectionless application level message.

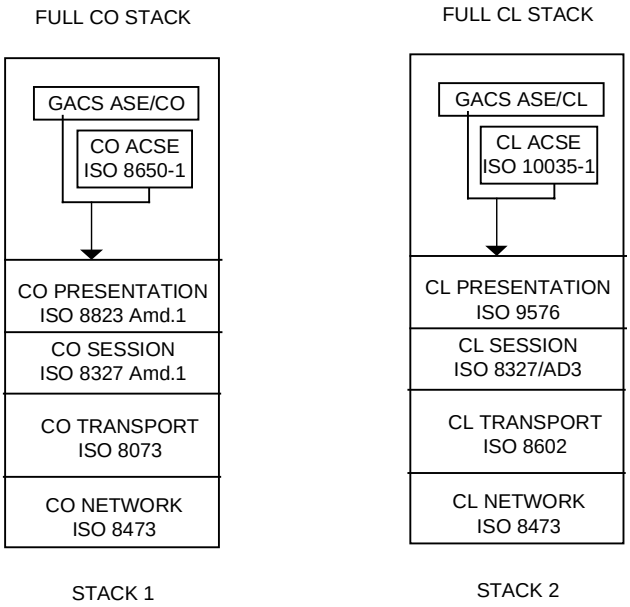

Figure 2.6-1: GACS Protocol Stacks

2.7 Conclusion

This document defines a message transfer service that may be used by aeronautical message applications to exchange various types of messages within the ATN.

The **GACS** service provides a basic message transfer capability which allows existing message headers and formats to be exchanged with a minimum amount of protocol overhead in a standardised manner. This capability allows existing applications to transition to an OSI environment.

The **GACS** services may be used in a connectionless or connection-oriented OSI environment. Applications may choose from different protocol options depending upon the operational requirements.

The **GACS** service allows any application encoding rules, including ASN.1 Basic Encoding Rules (BER) or text-oriented syntaxes, to be encapsulated within the abstract syntax of the PER-encoded **GACS** PDUs so that optimal bit encoding may be generated.

Finally, **GACS** is a message protocol, developed in the framework of the OSI architecture, which provides flexible message services suitable for existing and future air/ground and ground/air aeronautical message applications.