EUROCONTROL

ATNP/WG3/WP **16-11**
12 May 1999

AERONAUTICAL TELECOMMUNICATION NETWORK PANEL

WORKING GROUP 3 (APPLICATIONS AND UPPER LAYERS)

Naples, Italy, 18 - 21 May 1999

Agenda Item 6.4: ULCS Post Package 1 work

Draft Guidance Material for Sub-Volume 4 Enhancements

Prepared by: Tony Kerr
Presented by: Steve Van Trees

SUMMARY

This paper provides draft Guidance Material for the enhancements to Sub-Volume 4 (Upper Layer Communication Service) of ICAO Doc. 9705 which are proposed for approval at ATNP/3.

TABLE OF CONTENTS

1. Introduction 1
    1.1. References 1

2. Upper Layer Naming and Addressing 2
    2.1. Introduction 2
    2.2. General Description 2
    2.3. Discussion 3
        2.3.1. Context Management constraints 3
    2.4. Naming Tree Extensions 4
    2.5. Format and encoding of the Sys-id 4
    2.6. Format of Peer Identifier parameters 4
    2.7. How will this work in practice? 5
        2.7.1. CM Logon data 5
        2.7.2. System Management Agent addressing 5
        2.7.3. Airborne application addressing 5
        2.7.4. Ground application addressing at ICAO designated facilities 6
        2.7.5. Ground application addressing at non-ICAO designated facilities 6
        2.7.6. Called PSAP Address known a priori 6
    2.8. Registration Issues 6
        2.8.1. Application Type Registration 6
        2.8.2. Sys-ID Registration 6
        2.8.3. Facility Designator Registration 6
    2.9. Compatibility Considerations 7
    2.10. Multiple System Management Agents 7
    2.11. Redundant Systems 8
    2.12. Other Considerations 9

3. Connectionless Dialogue Service 10
    3.1. Connectionless Upper Layers Encoding Example 10

4. ULCS Security Guidance Material 12

5. GACS Guidance Material 13
    5.1. Introduction 13
    5.2. Background 13
    5.3. Development Approach for GACS 14

5.4. Aeronautical Data Interchange Requirements 14
5.5. GACS Service Overview 15
    5.5.1. G-TRANSFER Service 15
    5.5.2. G-TRANSFER-CONFIRMED Service 16
    5.5.3. G-END Service 17
5.6. GACS Protocol Stacks 17
    5.6.1. CO Protocol Stack 17
    5.6.2. CL Protocol Stack 17
5.7. Conclusion 18
5.8. GACS General Description 18
5.9. GACS Service Definition 20
5.10. The G-TRANSFER service 21
5.11. The G-TRANSFER-CONFIRMED service 22
5.12. The G-END service 23
5.13. The G-MULTICAST service 23

1. INTRODUCTION

This paper provides draft Guidance Material for the enhancements to Sub-Volume 4 (Upper Layer Communication Service) of ICAO Doc. 9705 which are proposed for approval at ATNP/3.

1.1. References

[1]	ISO/IEC 10035-1:1195/Amd.1:1997	Connectionless ACSE Amendment
[2]	ISO/IEC 9548-1:1996/Amd.1:1998 (submitted for fast-track)	Efficiency Enhanced Connectionless Session Protocol
[3]	ISO/IEC 9576-1:1995/Amd.1:1998 (submitted for fast-track)	Efficiency Enhanced Connectionless Presentation Protocol
[4]	ATNP/WG3/WP 14-11	Proposed ATN Upper Layer Naming and Addressing Extensions

2. UPPER LAYER NAMING AND ADDRESSING

2.1. Introduction

The ongoing development of ATN application concepts and implementation programmes have highlighted a number of restrictions in the upper layer naming and addressing provisions in ICAO Doc. 9705 edition 1, which may have caused problems in future ATN applications and implementation architectures.

The AE Title (AET) was previously defined as:

{iso.identified-organisation.icao.atn-end-system-air [or ground]. <end-system-id>. operational. <ae-qualifier>}

In general, if there were different instances of the same application on the same end system, then this could be catered for by using Invocation Identifiers in the addressing. However, if there are multiple system management agents in an ATN end system, with each responsible for a different set of MOs, then arguably they are not 'the same application' and would need distinguished addresses. But we should not expect the ground system to know the systems management configuration of the aircraft. There could for example be a single Agent acting as a proxy for ALL airborne management information.

It might have been possible to extend the ATN UL naming for systems management by allocating additional AE qualifiers for SMA (currently only the single value 5 is allocated). But this would not solve the general problem.

This more general problem, was that it is not possible to address explicitly multiple instances of ANY CNS/ATM-1 application in an ATN end system. There may be requirements in Package 2 for multiple CM applications (say) to exist in an aircraft.

Also, it is inherent in the CM protocol that there is only one address per application type, and that sub-arcs below AEQualifier in the naming hierarchy are not catered for. If a CM-Logon is performed to exchange further addresses, then previous addresses are overwritten.

Further, ATN Routers may have identifiers taken from alternative name spaces. In such cases the name-address mapping specified in the ULCS SARPs will break down when trying to communicate with SM Agents in Routers.

To summarise, the following issues needed to be solved:

a) The ATN naming and addressing in Doc 9705 edition 1 does not handle multiple instances of the same application type;

b) The CM application does not allow for naming arcs below AEQualifier to be exchanged in a Logon;

c) Routers can have names from different naming trees;

d) How to register additional AE types, either ICAO or external (e.g. CTS, GACS).

2.2. General Description

The Dialogue Service as specified in Doc 9705, first edition, requires that the peer communicating end systems be identified by either an ICAO facility designator or a 24-bit aircraft identifier. There are some cases where the entity to be addressed does not fit into this scheme.

As a consequence it is proposed to enhance the ATN upper layer naming requirements:

a) to add an additional arc to the ATN naming hierarchy to further qualify a given application type at a given location with the addition of a system identifier,

b) to add a new optional parameter to the D-START service to allow the new Sys-ID parameter to be specified,

c) to extend the addressing parameters in D-START primitives to allow a full presentation address to be optionally specified.

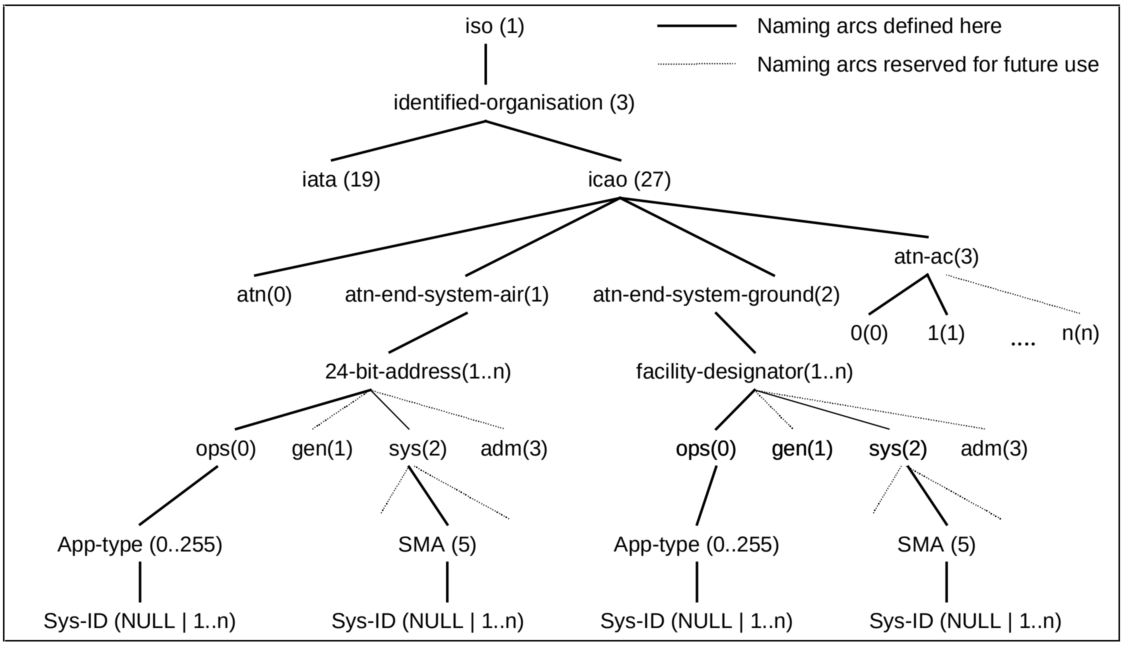

Figure 2-1. ATN Naming Hierarchy

2.3. Discussion

The OSI standards allow for different instances of the same application on the same end system, by means of Invocation Identifiers in the addressing. However, if there were multiple system management agents in an ATN end system, with each responsible for a different set of MOs, then arguably they are <u>not</u> 'the same application' and would need distinguished addresses.

Alternatively, it might be possible to extend the ATN UL naming for systems management by allocating additional AE qualifiers for SMA (currently only the single value 5 is allocated).

But we should not expect the ground system to know the systems management configuration of the aircraft. There could for example be a single Agent acting as a proxy for ALL airborne management information.

It might be possible to extend the syntax of the AE-qualifier, for example to redefine it as a sequence of INTEGER. However, ACSE requires the AE-qualifier to be either an X.500 Relative Distinguished Name, or a single unconstrained INTEGER (and for ATN, only the latter form is currently valid).

2.3.1. Context Management constraints

It is inherent in the CM application protocol that there is only one TSAP (and hence PSAP, as session and presentation selectors are not used) address per application type. Sub-arcs below AEQualifier in the naming hierarchy are not catered for. If, after the initial CM-Logon exchange, a further CM-Logon were performed to exchange additional addresses, then previous addresses are overwritten.

The CM protocol restricts the AE-qualifier to an integer in the range (0 .. 255), and this is not extensible (i.e. there is no extensions marker in the ASN.1 definition). (The type is called AEQualifier in the CM technical provisions - APName in earlier drafts).

2.4. Naming Tree Extensions

An arc has been added to the ATN naming tree (Figure 2-1), subordinate to what was previously the AE-qualifier field. This additional arc may be either NULL (i.e. absent) for backwards compatibility, or an unambiguous End System identifier, called ”Sys-id” in the following discussion.

In effect, this re-defines the Application Process (AP) to be what was previously the Application Entity (AE), so that now an AP Title identifies a given application type in a given location, rather than just identifying the location. The AE Title is then redefined such that it now identifies a given application type on a specific End System within that location.

Use is now made of the “sys” arc of the naming tree for system management applications.

This has the effect of further qualifying the AET for a given facility or aircraft. This is illustrated in Figure 2-1. Here, the “app-type” arc is the former AE-qualifier (i.e. ads (0), cma (1), cpc (2), etc.). The new arc is “Sys-id”, which can either be NULL, for backwards compatibility, or a System Identifier, an INTEGER in the range 1 to some undefined upper limit.

2.5. Format and encoding of the Sys-id

The Sys-id described above is the concatenation of the Location (LOC) and System Identifier (SYS) used in the ATN NSAP address.

The LOC field is a 2-octet value whose purpose is to distinguish routing areas within the same routing domain (RD). If more than one RD is located on a single aircraft, it distinguishes each such RD and the routing areas contained within them. LOC values are assigned by the addressing authority for the RD containing the identified routing area.

The SYS field is a 6-octet value which is used to uniquely identify an ATN ES or IS within a given routing area. It is assigned by the addressing authority responsible for the Network Addressing Domain that corresponds with the Routing Area in which the identified system is located. For example, if the System is attached to an IEEE 802 Local Area Network (such as an Ethernet), then a common approach is to use the 48-bit LAN address as the value of the SYS field.

The 8-octet Sys-id will be optionally passed to ACSE as the Calling AE Qualifier of A-ASSOCIATE primitives, and will appear as calling-AE-qualifier in AARQ APDUs. ACSE requires that this field be either an INTEGER or a Relative Distinguished Name. For backwards compatibility, the INTEGER form must be used.

Thus, the 8-octet value of the LOC+SYS is encoded as a (large) ASN.1 INTEGER when required.

2.6. Format of Peer Identifier parameters

With the first edition Dialogue Service specification, a dialogue can only be established with a peer application which has a known (registered somewhere) 24-bit aircraft address or 8 character ICAO ground facility designator. This is a restriction in the Called-Peer-ID parameter of the D-START service.

The syntax of this parameter is extended to allow a called PSAP address to be specified in place of the Called-Peer-ID, to cater for cases where:

a) an ATN system does not comply with the ICAO naming tree, and/or

b) the called PSAP address is known a priori.

2.7. How will this work in practice?

2.7.1. CM Logon data

The CM Logon exchange is used to convey address components. The “AE-qualifier” values exchanged in the CM protocol are actually AP type identifiers, and to avoid confusion, the AEQualifier in the CM technical provisions could be re-named APType.

For ATS applications (AP types CM, ADS, CPDLC, FIS) the addresses conveyed in the CM primitives shall be deemed to be the addresses of current operational ATS invocation of applications only (i.e. NULL beneath the AE-qualifier arc in the naming tree).

For the System Management application, the CM-User can communicate via the CM-Logon the addresses of all system management Agents at that location.

2.7.2. System Management Agent addressing

For the system management application, the AE Title structure is defined as:

> {iso (1).identified-organisation (3).icao (27).atn-end-system-air (1)[or ground (2)].<end-system-id>.sys (2).SMA (5).Sys-id}

That is, the Sys-id becomes the AE-qualifier, and is optionally conveyed as such in the Calling AE Qualifier field of the ACSE A-ASSOCIATE service.

At the Dialogue Service boundary, the SMASE will specify the called end-system-id in the D-START request, as at present. In addition, the DS-User will be able to specify a Sys-id, to disambiguate the addressed SMA in cases where there is more than one SMA in an aircraft or ground facility.

2.7.3. Airborne application addressing

To identify an application in an airborne system, the AE Title structure is defined as:

> {iso (1).identified-organisation (3).icao (27).atn-end-system-air (1). <end-system-id>.ops (0).App-type.Sys-id}

That is, the Sys-id becomes the AE-qualifier, and is optionally conveyed as such in the Calling AE Qualifier field of the ACSE A-ASSOCIATE service.

The DS-User will specify the called end-system-id in the D-START request, as at present. In addition, the DS-User will be able to specify a Sys-id, to disambiguate the addressed application in cases where there is more than one instance of that application type in an aircraft.

Where the sender does not specify a Sys-id, then the address resolution mechanism assumes that the current active ATS invocation of the application is being addressed (currently, this defaults to the only invocation of the application).

2.7.4. Ground application addressing at ICAO designated facilities

To identify an application in a ground system which has a registered ICAO facility designator, the AE Title structure is defined as:

> {iso (1).identified-organisation (3).icao (27).atn-end-system-ground (2).<end-system-id>.ops (0).App-type.Sys-id}

That is, the Sys-id becomes the AE-qualifier, and is optionally conveyed as such in the Calling AE Qualifier field of the ACSE A-ASSOCIATE service.

The DS-User will specify the called end-system-id in the D-START request, as at present. In addition, the DS-User will be able to specify a Sys-id, to disambiguate the addressed application in cases where there is more than one instance of that application type in a ground facility.

Where the sender does not specify a Sys-id, then the address resolution mechanism assumes that the current active ATS invocation of the application is being addressed (currently, this defaults to the only invocation of the application).

2.7.5. Ground application addressing at non-ICAO designated facilities

If an application requires to start a dialogue with a peer application on a system at a ground location which does not have a registered ICAO Facility Designator, then it is not able to use the “Called Peer Id” parameter at the Dialogue Service boundary. Instead, the calling application will have to obtain the PSAP address of the peer by some local means, and proceed as defined in the following subsection.

2.7.6. Called PSAP Address known a priori

If the calling application has prior knowledge of the Presentation Address of a destination application, then the name-address mapping mechanism of the Dialogue Service can be by-passed by allowing the address to be specified directly in the D-START request.

2.8. Registration Issues

2.8.1. Application Type Registration

Additional AE types, either ICAO-defined or external (e.g. CTS, GACS) may currently be registered only by proposing a modification to ICAO Doc. 9705, Table 4.3-2.

Registration is only a strict requirement for ATS applications which make use of the CM application to exchange address and version information over the air-ground data link. Such applications are specified by ATNP WG3, so this should not be an issue.

2.8.2. Sys-ID Registration

The Sys-ID proposed in this paper is composed of LOC and SYS components, both of which are registered by the ATN Network Addressing (Sub-) Domain Authority which contains the parent Routing Domain, as defined in Sub-Volume 5. The registration mechanisms are outside the scope of ATNP WG3.

2.8.3. Facility Designator Registration

An alternative solution for handling ground systems which have no registered ICAO facility designator, is to register them, according to the provisions of ICAO Docs. 7910 “Location

Indicators" and 8585 "Designators for Aircraft Operating Agencies, Aeronautical Authorities and Services". This is beyond the scope of ATNP.

2.9. Compatibility Considerations

It is a key requirement of these proposed upper layer naming and addressing extensions that backwards compatibility with the first edition of ICAO Doc. 9705 shall be maintained.

For convenience in this section, the term "Package 1" indicates the first edition naming and addressing provisions, and the term "Package 2" indicates the extended naming and addressing provisions.

When establishing an association between a Package 1 application and a Package 2 application, there will be one fewer component in the Application Entity Title of the former. However, this will not cause any interworking problems, as the Called AE Title is not conveyed to the peer system.

If a Package 2 DS-User addresses a Package 1 application, and uses the Calling Peer ID parameter, then Package 1 implementations will receive one more component than expected in the Calling AP Title parameter of ACSE. Also, the Calling AE Qualifier parameter value will not have one of the expected values. Thus, a Package 1 CF implementation may have problems if it performs rigorous checking of these parameters. There is a risk that less rigorous implementations may incorrectly decode the Calling AP Title and thus present an invalid Calling Peer ID value to the receiving DS-User. This may in turn cause interoperability problems if there are cases where the Calling Peer ID is validated to check that the caller is a known 24-bit address or ICAO facility designator.

If a Package 1 system addresses a Package 2 system which has more than one invocation of the addressed application type, then there could be some ambiguity as to which invocation should respond. There would need to be a pre-defined default responder.

2.10. Multiple System Management Agents

One of the goals of the UL naming and addressing enhancements was to allow multiple System Management Agent ASEs at the same physical location to be unambiguously addressed. The situation is illustrated in the following diagram.

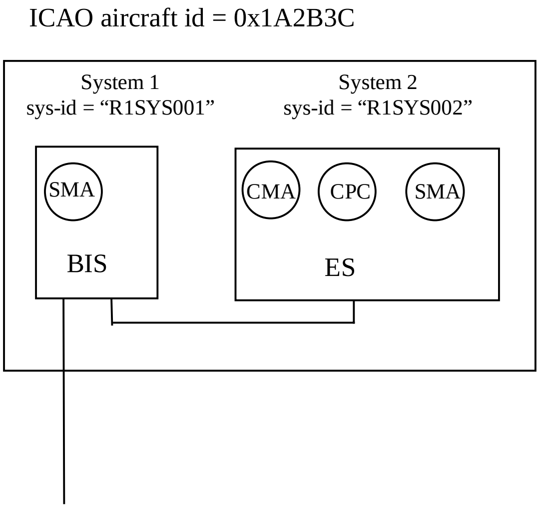

Figure 2-2. Two System Management Agents at the same Facility/ Aircraft

Here, two ATN systems are present at the airborne location identified by its ICAO 24-bit aircraft-id (Hex 1A2B3C); an ATN Router (BIS) and an ATN End System (ES). Both contain a System Management Agent (SMA) ASE and the ES also contains CM (CMA) and CPDLC (CPC) ASEs.

When the aircraft comes within range of a ground station, it needs to exchange application addressing and version information with the ground, which it does via the CM Logon service. In this case the CM Logon request would include the following information:

aircraftFlightIdentification = flight identifier (text string 2 - 8 characters)

cMLongTSAP = local TSAP address (RDP, ARS, LOC, SYS, NSEL, TSEL), LOC+SYS = “R1SYS0002”

groundInitiatedApplications:

“AEQualifier” (= app-type)	**Version**	**Address (longTSAP or shortTSAP)**
CPC (2)	1	(LOC+SYS = “R1SYS002”)
SMA (5)	1	(LOC+SYS = “R1SYS002”)
SMA (5)	1	(LOC+SYS = “R1SYS001”)

The ground-based manager application could then interrogate the resulting addressing database entries and individually address the two SMAs by specifying the (LOC+SYS) information at the Dialogue Service boundary.

2.11. Redundant Systems

In this scenario, an ES has been implemented using a dual redundant architecture for resilience. In normal operation, a single address is used for both systems, and the address is dynamically assigned to the "active” system, the other system being in “standby” mode.

It is possible that a management application may wish to explicitly communicate with one or the other system, for example to determine the system status. In this case, the management application, unlike the operational ATM applications, would have to use a unique address for each of the redundant systems.

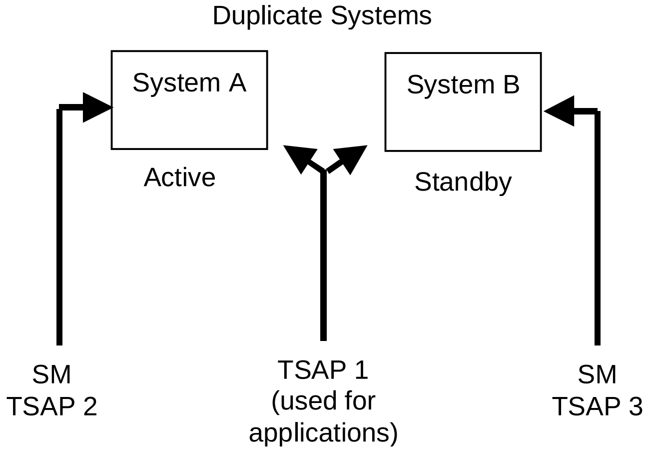

Figure 2-3. Redundant Systems

In this case the CM Logon request could include the following information:

aircraftFlightIdentification = flight identifier (text string 2 - 8 characters)

cMLongTSAP = TSAP 1 address (RDP, ARS, LOC, SYS, NSEL, TSEL)

groundInitiatedApplications:

"AEQualifier" (= app-type)	**Version**	**Address (longTSAP or shortTSAP)**
CPC (2)	1	"generic" TSAP 1 address
SMA (5)	1	TSAP 2 (LOC+SYS = "R1SYS00A")
SMA (5)	1	TSAP 3 (LOC+SYS = "R1SYS00B")

In order to address the current active CPC or SMA application, the ground system could omit the Sys-ID field in the Dialogue Service parameters, and use TSAP 1. To specifically address a given physical system, a ground management application could interrogate the resulting addressing database entries and individually address the two SMAs by specifying the (LOC+SYS) information at the Dialogue Service boundary.

2.12. Other Considerations

If it is required to use CM to exchange addressing information for other classes of application (such as AOC for example), then it is a pre-requisite that their app-type identifiers are registered in the global register which is currently contained in Sub-Volume 4.

3. CONNECTIONLESS DIALOGUE SERVICE

3.1. Connectionless Upper Layers Encoding Example

The following is an annotated example of the bits that might be received / sent as User Data when using the T-UNIT-DATA service.

The total overhead of the CL Session, Presentation and ACSE protocols in this example is some 29 octets.

Note that, if the optional Calling Peer ID and Calling Sys-ID parameters are not specified (as is the case with the GACS protocol), a saving of some 16.5 octets is achieved, giving an overhead of 12.5 octets.

1100 0000	Session Short Unit Data (SUD) SPDU
0000 0010	Presentation SHORT-UNIT-DATA PPDU (unaligned PER)
0	Extension bit: no extension values present in AUDT-apdu
0000 0110 0000	Bitmap for 12 OPTIONAL fields indicates presence of calling-AP-title, calling-AE-qualifier
	application-context-name ::= OBJECT IDENTIFIER
0000 0100	length of application context name = 4 octets
0010 1011	AC name = {1.3.27.3.1} : DS-User Version No = 1
0001 1011	
0000 0011	
0000 0001	
	calling-AP-title (OPTIONAL)
0	Extension bit: no extensions present
1	AP-title-form2 (OID form) is used
0000 0110	Length of Calling AP-Title = 6 octets
0010 1011	Calling AP Title = {1.3.27.1.500.0.12 } : air end system, 24-bit address =0000 0000 0000 0001 1111 0100, operational, GACS
0001 1011	
0000 0001	
1000 0011	
0111 0100	
0000 0000	
0000 1100	
	calling-AE-qualifier (OPTIONAL)
0	Extension bit: no extensions present
1	AE-qualifier-form2 (INTEGER form) is used
0000 1000	Length of Calling AE-Qualifier = 8 octets
xxxx xxxx	Value of Calling AE-Qualifier = LOC+SYS = xxxx xxxx xxxx xxxx + yyyy yyyy yyyy yyyy yyyy yyyy yyyy yyyy yyyy yyyy yyyy yyyy
xxxx xxxx	
yyyy yyyy	
yyyy yyyy	
yyyy yyyy	
yyyy yyyy	
yyyy yyyy	
yyyy yyyy	
	user-information
0	Extension bit: no extension values present in user-information, so SEQUENCE OF is exactly 1 in length.
000	Bitmap indicates no OPTIONAL fields present in EXTERNAL type
10	Choice 2 = BIT STRING encoding

10xxxxxx xxxxxxxx	Length determinant for bit string (2 octets, assuming length is greater than 127 but less than 16K bits)
xxxxxxxxxxxxxxxxxxxxxxxx xxxxxxxxxx	D-UNIT-DATA User data

If the D-UNIT-DATA has the maximum permissible length of 63488 octets, minus session presentation and ACSE overheads (Transport service constraint), then the length determinant of the User Data bitstring would be encoded as follows:

1100 0100	Length determinant for bit string fragment
xxxx .. xxxx	64K bits of User Data (= 8192 octets)
1100 0100	Length determinant for bit string fragment
xxxx .. xxxx	64K bits of User Data (= 8192 octets)
1100 0100	Length determinant for bit string fragment
xxxx .. xxxx	64K bits of User Data (= 8192 octets)
1100 0100	Length determinant for bit string fragment
xxxx .. xxxx	64K bits of User Data (= 8192 octets)
1100 0100	Length determinant for bit string fragment
xxxx .. xxxx	64K bits of User Data (= 8192 octets)
1100 0100	Length determinant for bit string fragment
xxxx .. xxxx	64K bits of User Data (= 8192 octets)
1100 0100	Length determinant for bit string fragment
xxxx .. xxxx	64K bits of User Data (= 8192 octets)
1100 0010	Length determinant for bit string fragment
xxxx .. xxxx	32K bits of User Data (= 4096 octets)
1011 1110 1101 1000	Length determinant for bit string fragment 3ed8H = 16088 bits
xxxx .. xxxx	16088 bits of User Data (= 2011 octets)

Thus, the maximum permissible length of D-UNIT-DATA User Data is some (8192 x 7) + 4096 + 2011 = 63451 octets.

4. ULCS SECURITY GUIDANCE MATERIAL

Note.— To be included in a future draft of this document. Draft Guidance can currently be found in ATN WG3 Working Papers, for example WG3/WP15-42. However, this must be treated with caution, as the underlying security concept is not yet completely stable.

5. GACS GUIDANCE MATERIAL

5.1. Introduction

This document describes the services and procedures of a Generic ATN Communication Service (GACS), 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 ATN infrastructure component, developed in the framework of the OSI architecture, which provides an Application Layer data transfer service suitable for future air-ground and ground-ground aeronautical message applications, as well as enabling the migration of pre-existing datalink applications to the ATN environment. It provides a basic data 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.

5.2. Background

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.

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

Each of these services has distinct information interchange 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 clearances 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 data exchanges vary in type, length, transfer time requirements, frequency of transmission, response requirements and other communications requirements.

The need exists, therefore, for a flexible communication service which would support the requirements of various applications, while accommodating the transfer characteristics of the application interchanges.

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

The intent of the GACS specification is to define a standard which would accommodate most future aeronautical datalink applications while allowing pre-existing non-ATN applications to utilise the ATN communication services.

Any proposed solution for a generic real-time end-to-end information interchange standard within the ATN environment should consider the communication requirements of most aeronautical applications.

The GACS specification therefore, recommends a common service that could be used to exchange multiple message types with various formats and bit encodings.

5.3. Development Approach for GACS

The basic approach used to develop the GACS service was to define the interchange services and procedures separate from the bit encodings used to exchange the Protocol Control Information (PCI) and the user application-specific information. 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 application data types (message classes) and encoding formats. Specific aeronautical message formats and encodings are not defined here. The GACS services are designed specifically to allow the definition of user data formats outside of the protocol by external organisations.

The GACS approach was restricted to end-to-end real-time interactions. Store-and-forward messaging is not within the scope of GACS, and is covered by the AMHS services for ground-ground communication.

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

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

b) Define a generic aeronautical protocol 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 defined abstract syntax which takes into account the limited bandwidth of the underlying air/ground sub-networks.

5.4. Aeronautical Data Interchange Requirements

Early studies determined a generic set of data communication requirements for aeronautical applications which could be used as a basis for determining the services of a standard ATN upper layer communication service. Those requirements may be summarised as:

a) Ability to send multiple data formats (for both pre-ATN and future applications).

b) Ability to convey security information.

c) Ability to convey application message priorities.

d) Ability to send various types of transfer 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 header (i.e. protocol and encoding) overhead.

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

5.5. 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 datalink applications. The services have been designed to meet the aeronautical data interchange requirements listed previously.

The GACS service provides an enhanced Dialogue Service which can be exploited by future ATN application service elements (ASEs). It also allows existing formats (e.g. ACARS character-oriented protocol) to be encapsulated for transmission over the ATN.

The basic services offered by the GACS specification are:

- G-TRANSFER
- G-TRANSFER-CONFIRMED
- G-END

These services are described in the following subsections.

5.5.1. G-TRANSFER Service

This service allows the GACS-User to request a class of communication which will result in the selection of either a connection-oriented communication stack, or (if available) a connectionless stack.

When a connectionless communications provider is selected, the G-TRANSFER request primitive results in the transfer of the GACS-User’s data as User Data of the connectionless service, i.e. a datagram is transferred.

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.

5.5.2. G-TRANSFER-CONFIRMED Service

This is similar to the G-TRANSFER service, but in addition it offers end-to-end confirmation that the user data has been delivered to the receiving application.

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.

5.5.3. G-END Service

G-END is an optional service which allows a GACS-User to request the termination of a communication relationship with a peer GACS-User.

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.

5.6. GACS Protocol Stacks

Figure 5-1 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.

5.6.1. CO Protocol Stack

Stack 1 depicts the GACS/CO protocol suite which includes connection oriented services for the upper layers of the OSI reference model including Transport. This protocol suite would be used by applications which require association 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 data items. Error recovery and reliable transport services by the underlying protocols are provided.

5.6.2. CL Protocol Stack

Stack 2 depicts the GACS/CL protocol suite which includes connectionless services for the upper layers of the OSI reference model including Transport. This protocol configuration would be used by applications which require independent datagram transfers where individual data items 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 and where subsequent interchanges are infrequent and unrelated, and where error recovery outside the AP is not required.

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

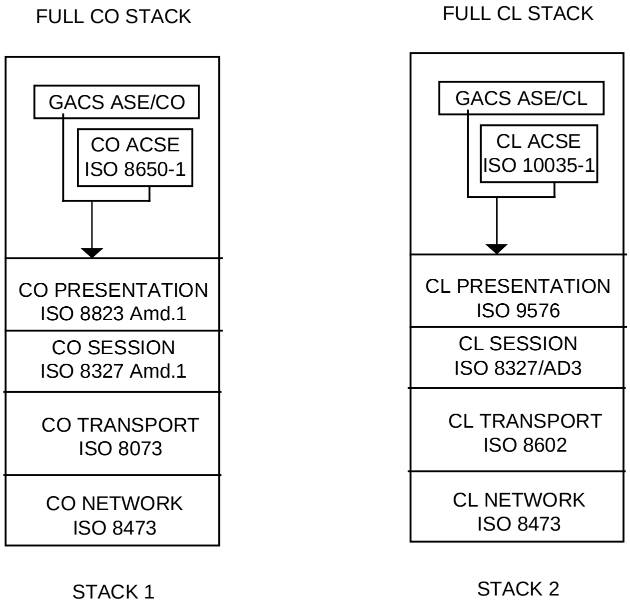

Figure 5-1: GACS Protocol Stacks

5.7. Conclusion

The GACS specification defines a generic service that may be used by aeronautical datalink applications to exchange various types of information across the ATN.

The **GACS** service provides a basic data transfer capability which allows pre-ATN 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 protocol, developed in the framework of the OSI architecture, which provides flexible data interchange services suitable for pre-ATN datalink applications and future air-ground and/or ground-ground aeronautical datalink ASEs.

5.8. GACS General Description

The Generic ATN Communication service (GACS) 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.

The GACS specification is designed to optimise the use of communications bandwidth, and consequently uses the Dialogue Service defined in Sub-Volume 4 of ICAO Doc 9705, including extensions for unit data and presentation address handling services. The Dialogue Service in turn uses the ATN Transport service defined in Sub-Volume 5 of ICAO Doc 9705.

The GACS 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.

The GACS service provision as specified in [ULCS] 4.9 can be realised alternatively as an "Application Layer message protocol" or as a "simple generic service". The two approaches as illustrated in Figure 5-2 are very different and have different fields of applicability.

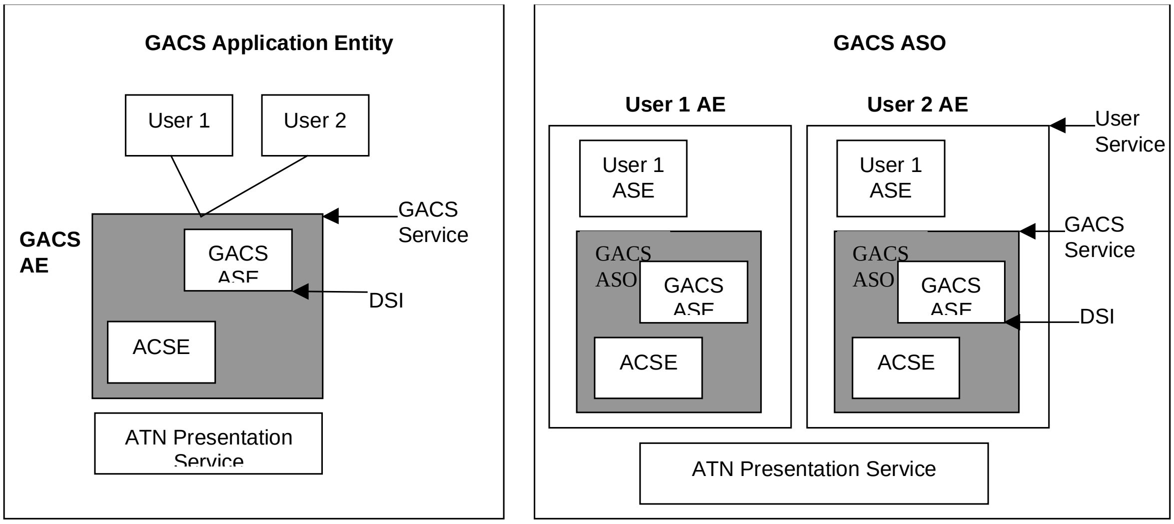

Figure 5-2. GACS Application versus GACS ASO

For the GACS implementation project, the **ATN Application Entity** (AE) approach is being adopted, as illustrated in the left-hand side of Figure 5-2. This provides an ATN access point to existing (e.g. ACARS-based) and future applications which are not specified as ASEs within the defined ATN upper layer architecture.

However, the GACS software will be designed to be modular, so that it could be adapted in the future to realise the alternative **ATN Application Service Object** (ASO) or "service" approach, as illustrated in the right-hand side of Figure 5-2. This would provide effectively an enhanced Dialogue Service to future air-ground and ground-ground ATN applications which are specified as ASEs (ATC and AOC).

The GACS AE approach is appropriate for the migration of existing applications. The GACS ASO (enhanced dialogue service) approach would be preferred for any new ATC or AOC application.

The GACS Application Entity will be a distinct ATN application installed in aircraft and ground systems acting as a point of access to the ATN. An ATN address will be allocated to the GACS AE. The existing CM Application can be used without modification to exchange the address and version number of the GACS application in the air-ground environment. The GACS application is identified by an OBJECT IDENTIFIER.

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 identifier.

The GACS service will be exposed via an application programming interface (API), providing a communications interface to user applications.

Several GACS-Users will be able simultaneously to use services from the same GACS Application. The GACS Application multiplexes data supplied by GACS-Users over the same dialogue when the intended recipient and the requested communication characteristics are identical.

The GACS Application itself does not know anything about the user 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.

5.9. GACS Service Definition

The GACS software will provide the services listed in Table 5-1.

Table 5-1. 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.

For the basic G-TRANSFER and G-TRANSFER-CONFIRMED services, a number of options are defined, and these can be selected via the “Level Of Service” parameter:

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. 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.

5.10. The G-TRANSFER service

The G-TRANSFER service enables the transparent transmission of data between GACS-Users.

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). G-TRANSFER request and indication service primitives are defined, as illustrated in Figure 5-3.

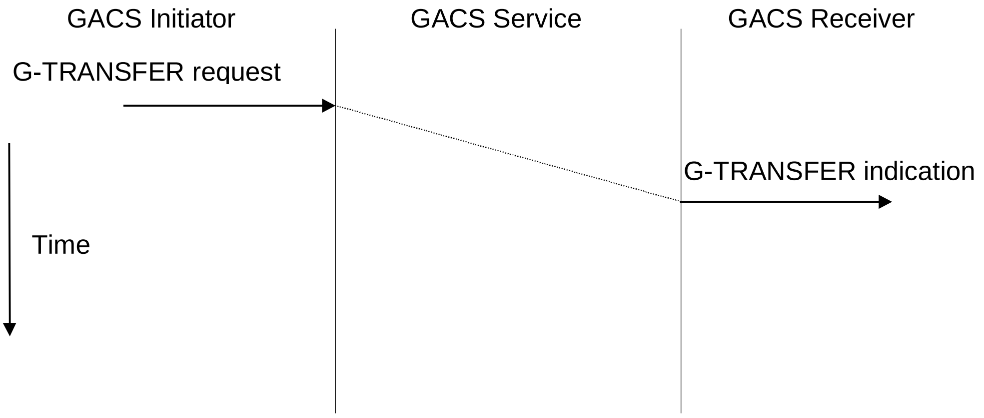

Figure 5-3. G-TRANSFER sequence diagram

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 shown in Table 5-2.

Table 5-2. 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(=)*

5.11. The G-TRANSFER-CONFIRMED service

The G-TRANSFER-CONFIRMED enables the transparent transmission of data between GACS-Users, with confirmation of data delivery to the recipient system(s).

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). G-TRANSFER-CONFIRMED request, indication and confirmation service primitives are defined, as illustrated in Figure 5-4.

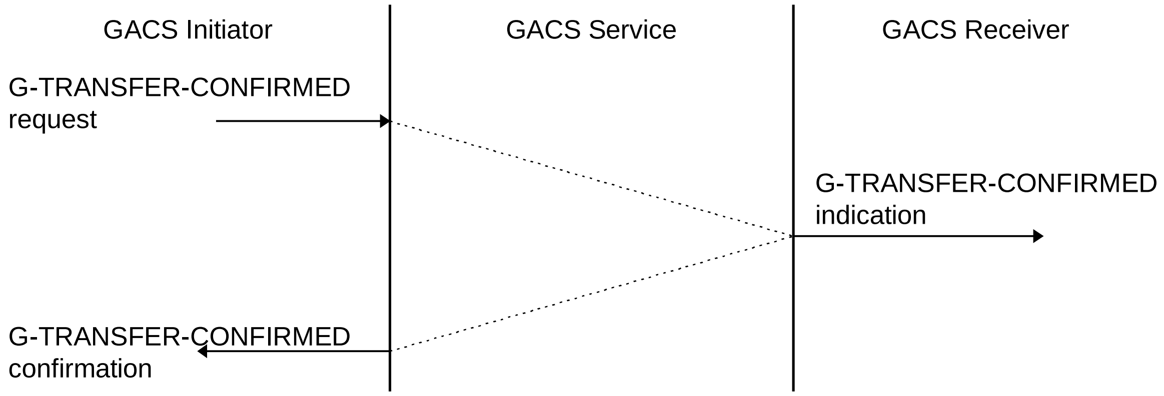

Figure 5-4. G-TRANSFER-CONFIRMED sequence diagram

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 shown in Table 5-3.

Table 5-3. 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(=)*	

5.12. The G-END service

The G-END service enables the orderly termination of a communications relationship between GACS-Users.

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). G-END request and indication service primitives are defined, as illustrated in Figure 5-5.

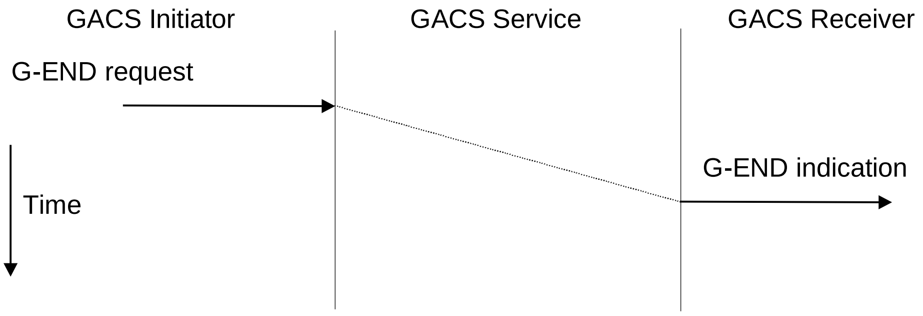

Figure 5-5. G-END sequence diagram

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 shown in Table 5-4.

Table 5-4. G-END parameters

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

5.13. The G-MULTICAST service

The G-MULTICAST will enable or disable the receipt of messages addressed to a group address, when this is supported by the ATN lower layers.

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.

Only a G-MULTICAST request primitive is defined, as illustrated in Figure 5-6.

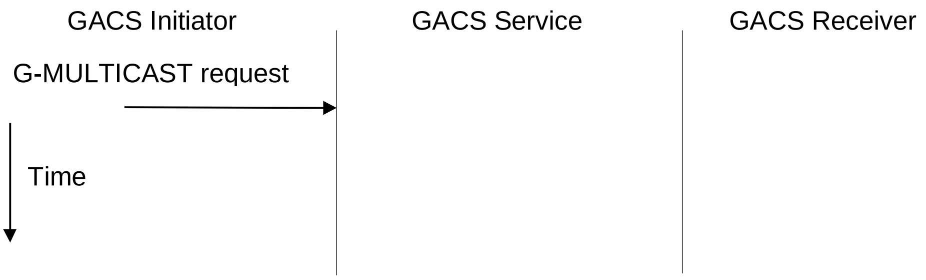

Figure 5-6. G-MULTICAST sequence diagram

The initiating GACS-User issues a G-MULTICAST request primitive at any time. The parameters of the G- MULTICAST primitives are shown in Table 5-5.

Table 5-5. G- MULTICAST parameters

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