CCSDS
The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

MISSION OPERATIONS— MESSAGE ABSTRACTION LAYER BINDING TO TCP/IP TRANSPORT AND SPLIT BINARY ENCODING

RECOMMENDED STANDARD

CCSDS 524.2-B-1

BLUE BOOK
November 2017

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Standards

MISSION OPERATIONS—MESSAGE ABSTRACTION LAYER BINDING TO TCP/IP TRANSPORT AND SPLIT BINARY ENCODING

RECOMMENDED STANDARD

CCSDS 524.2-B-1

BLUE BOOK

November 2017

AUTHORITY

Issue:	Recommended Standard, Issue 1
Date:	November 2017
Location:	Washington, DC, USA

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4), and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the e-mail address below.

This document is published and maintained by:

CCSDS Secretariat
National Aeronautics and Space Administration
Washington, DC, USA
E-mail: secretariat@mailman.ccsds.org

STATEMENT OF INTENT

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommended Standards** and are not considered binding on any Agency.

This **Recommended Standard** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommendation** is entirely voluntary. Endorsement, however, indicates the following understandings:

o Whenever a member establishes a CCSDS-related **standard**, this **standard** will be in accord with the relevant **Recommended Standard**. Establishing such a **standard** does not preclude other provisions which a member may develop.

o Whenever a member establishes a CCSDS-related **standard**, that member will provide other CCSDS members with the following information:

  -- The **standard** itself.

  -- The anticipated date of initial operational capability.

  -- The anticipated duration of operational service.

o Specific service arrangements shall be made via memoranda of agreement. Neither this **Recommended Standard** nor any ensuing **standard** is a substitute for a memorandum of agreement.

No later than five years from its date of issuance, this **Recommended Standard** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Standard** is issued, existing CCSDS-related member standards and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such standards or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new standards and implementations towards the later version of the Recommended Standard.

FOREWORD

The intended use for this document is to allow the implementation of a protocol layer that binds the Mission Operations (MO) service framework to the TCP/IP transport using the split binary encoding. This document assumes that the reader is familiar with the MO concepts, especially the Message Abstraction Layer (MAL).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CCSDS has processes for identifying patent issues and for securing from the patent holder agreement that all licensing policies are reasonable and non-discriminatory. However, CCSDS does not have a patent law staff, and CCSDS shall not be held responsible for identifying any or all such patent rights.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Standard is therefore subject to CCSDS document management and change control procedures, which are defined in *Organization and Processes for the Consultative Committee for Space Data Systems* (CCSDS A02.1-Y-4). Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be sent to the CCSDS Secretariat at the e-mail address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

Member Agencies

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d’Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People’s Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Federal Space Agency (FSA)/Russian Federation.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- UK Space Agency/United Kingdom.

Observer Agencies

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- Electronics and Telecommunications Research Institute (ETRI)/Korea.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- Mohammed Bin Rashid Space Centre (MBRSC)/United Arab Emirates.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- South African National Space Agency (SANSA)/Republic of South Africa.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- Swiss Space Office (SSO)/Switzerland.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 524.2-B-1	Mission Operations—Message Abstraction Layer Binding to TCP/IP Transport and Split Binary Encoding, Recommended Standard, Issue 1	November 2017	Original issue

CONTENTS

Section Page

CONTENTS (continued)

Section Page

CONTENTS (continued)

Figure Page

Table

1 INTRODUCTION

1.1 PURPOSE

This Recommended Standard defines two aspects of message exchange between MO service providers and consumers:

a) The binding between the Mission Operations (MO) Message Abstraction Layer (MAL) specified in reference [1] and the Transmission Control Protocol over Internet Protocol (TCP/IP) specified in references [4] and [2]. This binding allows MO services to use TCP/IP as messaging technology in all situations where it may be required.

b) A split binary encoding for MAL data types. The specified binary encoding is generic (i.e., independent of the MAL binding to TCP/IP protocol) and the resulting encoded MAL messages can be exchanged via any communication protocol, for which a binding to the MAL exists. Equally, it is not mandatory to use the binary encoding specified in this book for encoding of the body of the messages when using the MAL to TCP/IP binding. Any MAL encoding, specified in other books, can be used for encoding the body of the messages when adopting the MAL to TCP/IP binding, specified in this book.

1.2 SCOPE

The scope of this Recommended Standard is the specification of the binding in terms of technology mapping to TCP/IP of:

a) MAL message;

b) MAL Transport Interface.

The MAL Blue Book (reference [1]) specifies the MAL protocol in an abstract way, i.e., without defining the concrete Protocol Data Units (PDUs). The MAL Binding to TCP/IP Transport and Split Binary Encoding specifies a complete and unambiguous mapping of:

a) the MAL message to a binary PDU to be transmitted over TCP/IP;

b) the MAL transport interface to the TCP/IP interface;

c) the MAL data types to a binary encoding format (split binary encoding).

This Recommended Standard does not specify:

a) individual implementations or products;

b) the implementation of entities or interfaces within real systems;

c) recommendations or best practices for deploying systems with proxies and/or firewalls;

d) a direct mapping to the socket API, as the TCP protocol specification (reference [4]) is unambiguously used for the mapping.

In a concrete deployment, on-the-wire interoperability between Application Layer MO service consumer and provider will be achieved by encoding the abstract MAL messages in the concrete split binary encoding and transmitting them by means of TCP/IP PDUs, as defined in this Recommended Standard.

1.3 APPLICABILITY

This Recommended Standard specifies a mapping to a concrete communication protocol that enables different implementations of the MO service framework (see 2.2) to interoperate through TCP/IP communication protocol.

1.4 RATIONALE

CCSDS MO services are Application Layer services, which are specified in an abstract, implementation-and-communication-agnostic manner in terms of the MAL (Message Abstraction Layer).

In a concrete deployment scenario (instantiation of the abstract MO services in a concrete set of technologies) on-the-wire interoperability is achieved by agreeing on a concrete encoding and a concrete communication protocol for the exchange of the messages between the service provider and service consumer.

The goal of this Recommended Standard is to specify how to translate the abstract MAL message model in an unambiguous way into a concrete message exchange protocol based on TCP/IP.

This Recommended Standard also aims at defining a concrete split binary encoding format for the MAL data types that can be reused by any MAL binding to a communication protocol.

1.5 DOCUMENT STRUCTURE

This document is organized as follows:

a) section 1 presents the purpose, scope, applicability, and rationale, and lists the definitions, conventions, and references used throughout this Recommended Standard;

b) section 2 presents an overview of the MAL Binding to TCP/IP Transport and Split Binary Encoding in relation with the MO service framework;

c) section 3 specifies the MAL binding to TCP/IP transport protocol, by providing an unambiguous mapping of the MAL messages to the TCP/IP PDUs;

d) section 4 specifies the mapping of the MAL transport interface to the TCP/IP interface;

e) section 5 specifies the split binary encoding format for the MAL data types.

1.6 DEFINITIONS

For the purposes of this document, the following definitions apply.

binding: The access mechanism for a Service. Bindings are used to locate and access Service Interfaces. Services use bindings to describe the access mechanisms that consumers have to use to call the Service. The binding specifies unambiguously the protocol stack required to access a Service Interface. Bindings may be defined statically at compile time or they may use a variety of dynamic run-time mechanisms (DNS, ports, discovery).

MAL TCP/IP PDU: The protocol data unit, transmitted over TCP/IP, that holds the content of a MAL message.

MAL header: The header of the MAL message contains the meta-data and is mapped to the protocol specific header encodings. MAL messages are composed of two conceptual segments, the MAL header and the MAL body.

protocol: The set of rules and formats (semantic and syntactic) used to determine the communication behaviour of a protocol layer in the performance of the layer functions. The state machines that operate and the PDUs that are exchanged specify a protocol.

protocol layer: The implementation of a specific protocol. It provides a protocol service access point to layers above and uses the protocol service access point of the layer below.

service access point, SAP: The point at which one layer's functions are provided to the layer above. A layer may provide protocol services to one or more higher layers and use the protocol services of one or more lower layers.

1.7 NOMENCLATURE

1.8 NORMATIVE TEXT

The following conventions apply for the normative specifications in this Recommended Standard:

a) the words 'shall' and 'must' imply a binding and verifiable specification;

b) the word 'should' implies an optional, but desirable, specification;

c) the word 'may' implies an optional specification;

d) the words 'is', 'are', and 'will' imply statements of fact.

NOTE – These conventions do not imply constraints on diction in text that is clearly informative in nature.

1.8.1 INFORMATIVE TEXT

In the normative sections of this document, informative text is set off from the normative specifications either in notes or under one of the following subsection headings:

- Overview;
- Background;
- Rationale;
- Discussion.

1.9 BIT NUMBERING CONVENTION

In this document, the following convention is used to identify each bit in an *N*-bit field. The first bit in the field to be transmitted (i.e., the most left justified when drawing a figure) is defined to be 'Bit 0'; the bit following is defined to be 'Bit 1', and so on up to 'Bit *N*–1'. When the field is used to express a binary value (such as a counter), the Most Significant Bit (MSB) shall be the first transmitted bit of the field, i.e., 'Bit 0'. The bit numbering convention is represented in figure 1-1.

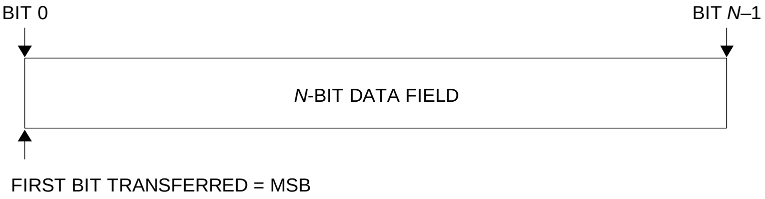

Figure 1-1: Bit Numbering Convention

In accordance with modern data communications practice, spacecraft data fields are often grouped into eight-bit 'words' which conform to the above convention. Throughout this Recommended Standard, the following nomenclature, represented in figure 1-2, is used to describe this grouping:

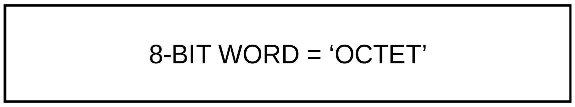

Figure 1-2: Octet Convention

By CCSDS convention, all 'spare' or 'unused' bits shall be permanently set to value 'zero'. The TCP protocol follows the big-endian order.

1.10 REFERENCES

The following publications contain provisions which, through reference in this text, constitute provisions of this document. At the time of publication, the editions indicated were valid. All publications are subject to revision, and users of this Recommended Standard are encouraged to investigate the possibility of applying the most recent editions of the publications indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS publications.

NOTE – A list of informative references is provided in annex G.

[1] *Mission Operations Message Abstraction Layer*. Issue 2. Recommendation for Space Data System Standards (Blue Book), CCSDS 521.0-B-2. Washington, D.C.: CCSDS, March 2013.

[2] J. Postel. *Internet Protocol*. STD 5. Reston, Virginia: ISOC, September 1981.

[3] S. Deering and R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification*. RFC 2460. Reston, Virginia: ISOC, December 1998.

[4] J. Postel. *Transmission Control Protocol*. STD 7. Reston, Virginia: ISOC, September 1981.

[5] *IEEE Standard for Floating-Point Arithmetic*. 2nd ed. IEEE Std. 754-2008. New York: IEEE, 2008.

[6] F. Yergeau. *UTF-8, a Transformation Format of ISO 10646*. STD 63. Reston, Virginia: ISOC, November 2003.

[7] *Time Code Formats*. Issue 4. Recommendation for Space Data System Standards (Blue Book), CCSDS 301.0-B-4. Washington, D.C.: CCSDS, November 2010.

[8] *Mission Operations—MAL Space Packet Transport Binding and Binary Encoding*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 524.1-B-1. Washington, D.C.: CCSDS, August 2015.

[9] R. Hinden and S. Deering. *IP Version 6 Addressing Architecture*. RFC 4291. Reston, Virginia: ISOC, February 2006.

2 OVERVIEW

2.1 GENERAL

This Recommended Standard allows MO services defined in terms of the MAL to interoperate across an end-to-end communication link using a normative binding of the MAL abstractions to TCP/IP for exchanging messages. This is of particular interest for MO services, for which the service provider and consumer are both deployed on the ground, for instance, when the MO service provider is located in a Mission Control Centre and the consumer in the Science Control Centre. Another scenario, for which the use of the TCP/IP communication protocol is of interest, is when a service provider on the ground acts as a proxy of a service in space towards consumers deployed on the ground. With the extension of the IP protocol to the space domain, this Recommended Standard may become also relevant for the space-to-ground and space-to-space communication scenarios. The messages that provider and consumer exchange to implement the MO services are encoded in PDUs and carried via TCP/IP, which acts as a Message Layer mapping. This can run directly over a datalink protocol, such as IEEE 802.3 (Ethernet) and IEEE 802.11 (Wireless LAN).

To achieve this goal, this Recommended Standard provides a mapping of the MAL transport interface, the MAL abstract message specification (reference [1]) to the TCP/IP protocol stack (references [2], [3], and [4]). It also provides a concrete split binary encoding, which can be used to encode the body of the MAL messages exchanged over the TCP/IP protocol.

The MAL Blue Book (reference [1]) defines an abstract transport interface as a set of request and indication primitives. The mapping to a concrete transport protocol specifies how these primitives are provided according to the rules and requirements of that particular messaging protocol.

The mapping of MAL to a concrete communication protocol translates the MAL message model into one or several protocol specific PDUs. MAL messages are composed of two conceptual segments, the MAL header and the MAL body. The header of the MAL message contains the metadata and is mapped to the protocol specific header encodings. The body of the MAL message can, however, be encoded, using an encoding of choice, which fits best the requirements of a particular deployment. To give a concrete example, when using the MAL to TCP/IP protocol binding, which is specified in this book, the body of the MAL messages can be encoded, using the split binary encoding specified in this book. For a different deployment using the very same MAL to TCP/IP protocol binding, the body of the messages can be encoded using a different encoding such as one of the binary encodings specified in reference [8] or any other encoding.

Full interoperability of services (the so called on-the-wire interoperability) is achieved if the same MAL to transport protocol binding and the same encoding for the body of the MAL messages are used by the service provider and the service consumer. Alternatively, a bridge must be used to translate from one binding/encoding to another (reference [G1]).

The diagram shown in figure 2-1 presents the set of standards documentation in support of the Mission Operations Services Concept. This MAL Binding to TCP/IP Transport and Split Binary Encoding book belongs to the technology mappings documentation.

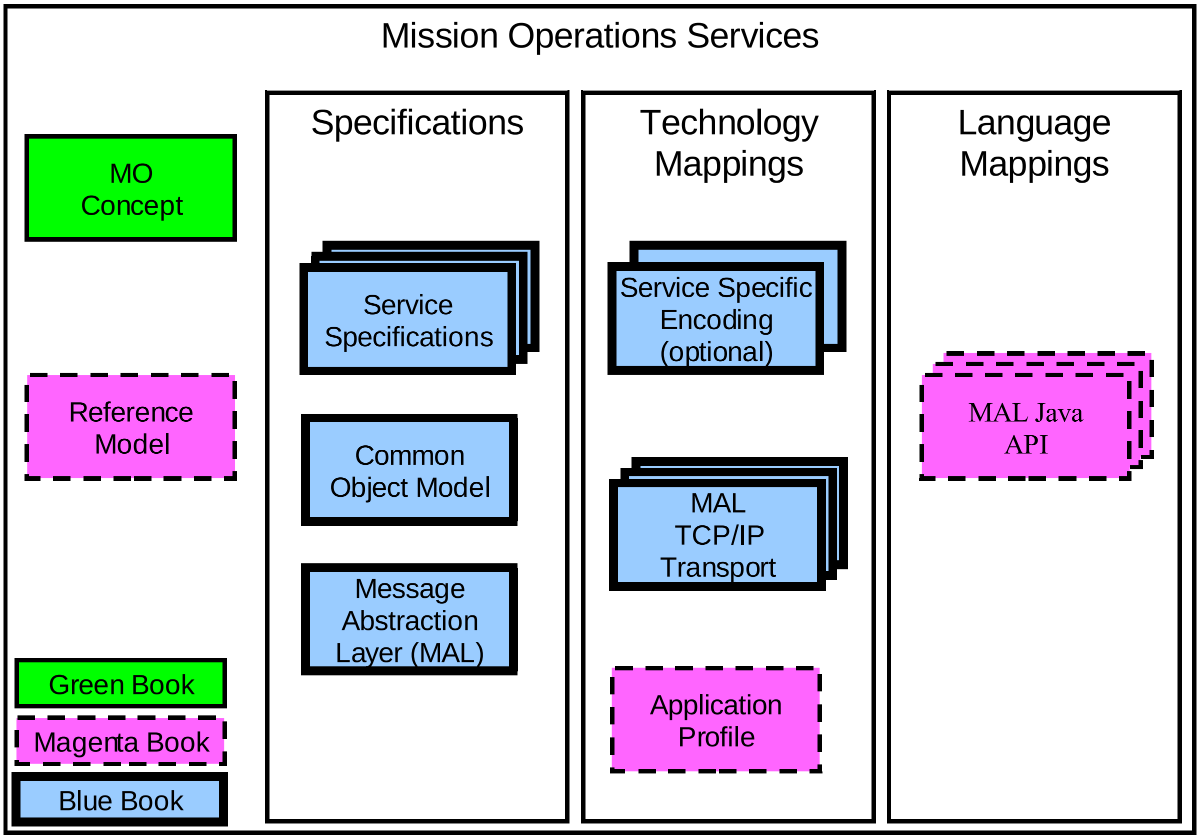

Figure 2-1: Mission Operations Services Concept Document Set

2.2 MO SERVICE FRAMEWORK OVER TCP/IP

The CCSDS Spacecraft Monitoring & Control (SM&C) working group has developed a concept for an MO service framework, which follows the principles of service-oriented architectures. The framework defines two important aspects: the first is a protocol for interaction between two separate entities; the second is a set of common services providing functionality shared by most of the MO services. An overview of this framework is presented in figure 2-2.

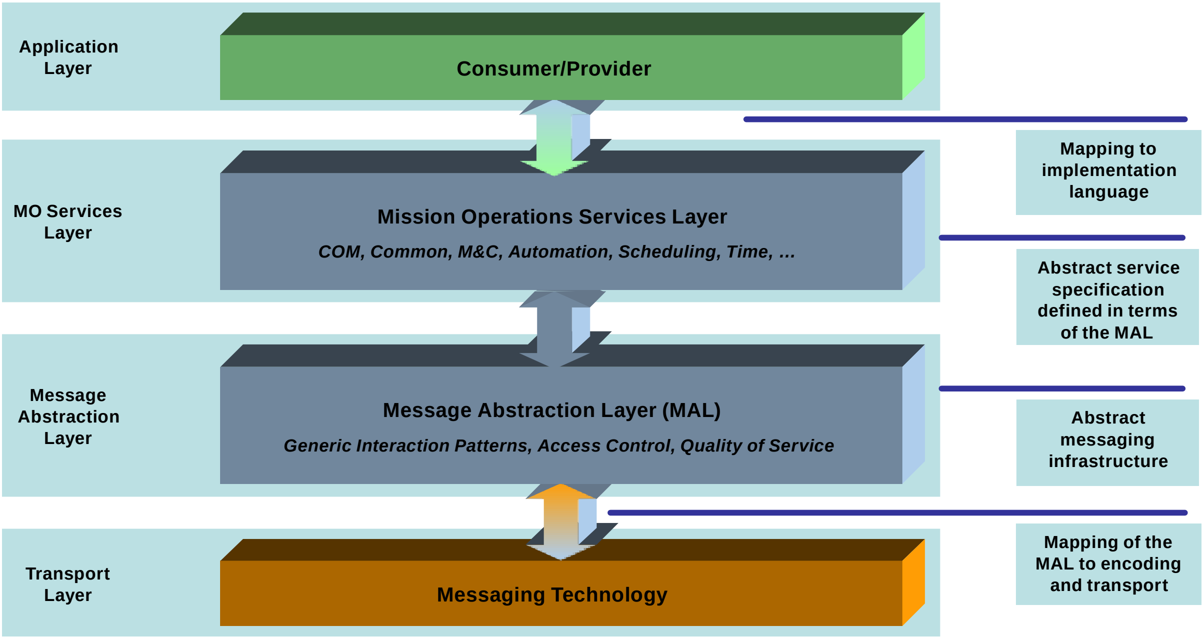
Application Layer
Consumer/Provider
MO Services Layer
Mission Operations Services Layer
COM, Common, M&C, Automation, Scheduling, Time, ...
Message Abstraction Layer
Message Abstraction Layer (MAL)
Generic Interaction Patterns, Access Control, Quality of Service
Transport Layer
Messaging Technology
Mapping to implementation language
Abstract service specification defined in terms of the MAL
Abstract messaging infrastructure
Mapping of the MAL to encoding and transport

Figure 2-2: Overview of the MO Service Framework

This Recommended Standard specifies:

a) how the specific technology shall be used;

b) how any transmission errors or issues shall be communicated to higher layers;

c) how all underlying Data Link or Network Layer issues shall be handled;

d) the physical representation of the MAL messages necessary to constitute the operation templates;

e) the mapping of the message structure rules for that technology;

f) the encoding of the MAL data types.

It does not specify:

a) individual application services, implementations, or products;

b) the implementation of entities or interfaces within real systems;

c) the methods or technologies required to acquire data;

d) the management activities required to schedule a service;

e) the representation of any service-specific PDUs (this is derived from the encoding format defined in this document in section 5).

The MAL Blue Book (reference [1]) groups all the interfaces to the Transport Layer in a single place called the MAL transport interface (subsection 3.7 of reference [1]). Thanks to this, only the MAL transport interface needs to be mapped to the TCP/IP protocol, without the need to map the entire MAL Blue Book.

Figure 2-3 expands the previous figure (figure 2-2) by presenting the MAL to TCP/IP transport protocol binding layer in the MO service framework stack and highlighting the various interfaces and their main primitives. It also shows that the mapping of the MAL transport interface to the TCP/IP layer requires the insertion of a layer in between. This layer is called the MAL TCP/IP Binding. It is responsible for the translation of the abstract MAL message to the MAL TCP/IP PDU transferred through concrete TCP/IP segments.

The protocol stack represented in figure 2-3 is conceptual. It can be implemented in various ways. For example, an implementation of the stack may, for performance reasons, merge the MAL layer and the MAL TCP/IP Transport Layer into a single layer called ‘MAL over TCP/IP’.

The names of the main interfaces used and implemented by each layer are given by figure 2-3. The main primitives are shown for each interface:

a) the primitives for every operation provided by an MO service;

b) the primitives for every interaction pattern provided by MAL;

c) the primitives for transmitting and receiving a single MAL message or multiple MAL messages;

d) the primitives for transmitting and receiving data from the TCP/IP channel.

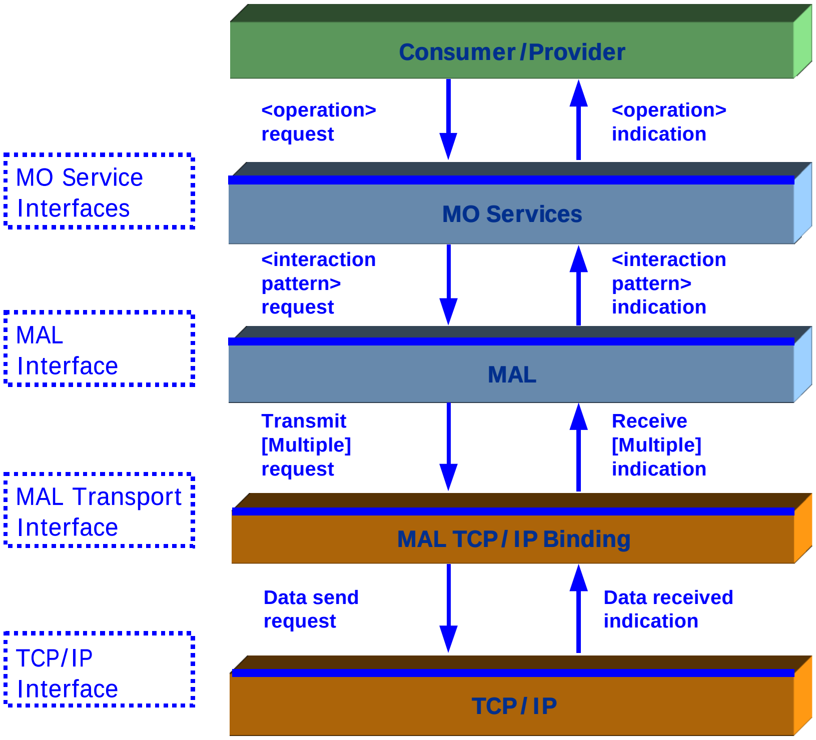

Figure 2-3: MO Service Framework above TCP/IP

2.3 TYPICAL USE

Possible uses of the MAL binding to TCP/IP transport protocol may be between MO entities (service consumer and provider) operating on ground, for example:

a) ground applications deployed on the same machine or interacting over a local area network using TCP/IP;

b) ground components interacting over a wide area network;

c) mobile applications consuming MAL services over wireless networks.

A typical deployment is illustrated in figure 2-4. In this example, the MO service framework is only used by the end nodes: a ground end node (e.g., in a mission control centre) and another ground end node (e.g., in a science/payload control centre).

Figure 2-4 shows how the abstract MO stack is implemented on both end nodes. More specifically the figure shows what components are deployed, how they are related to the abstract stack (the five layers in the background), and what API and SAP are used.

The first concrete PDU is produced at the binding level as the result of the mapping of the MAL message to TCP/IP.

The lower protocol layers are not represented.

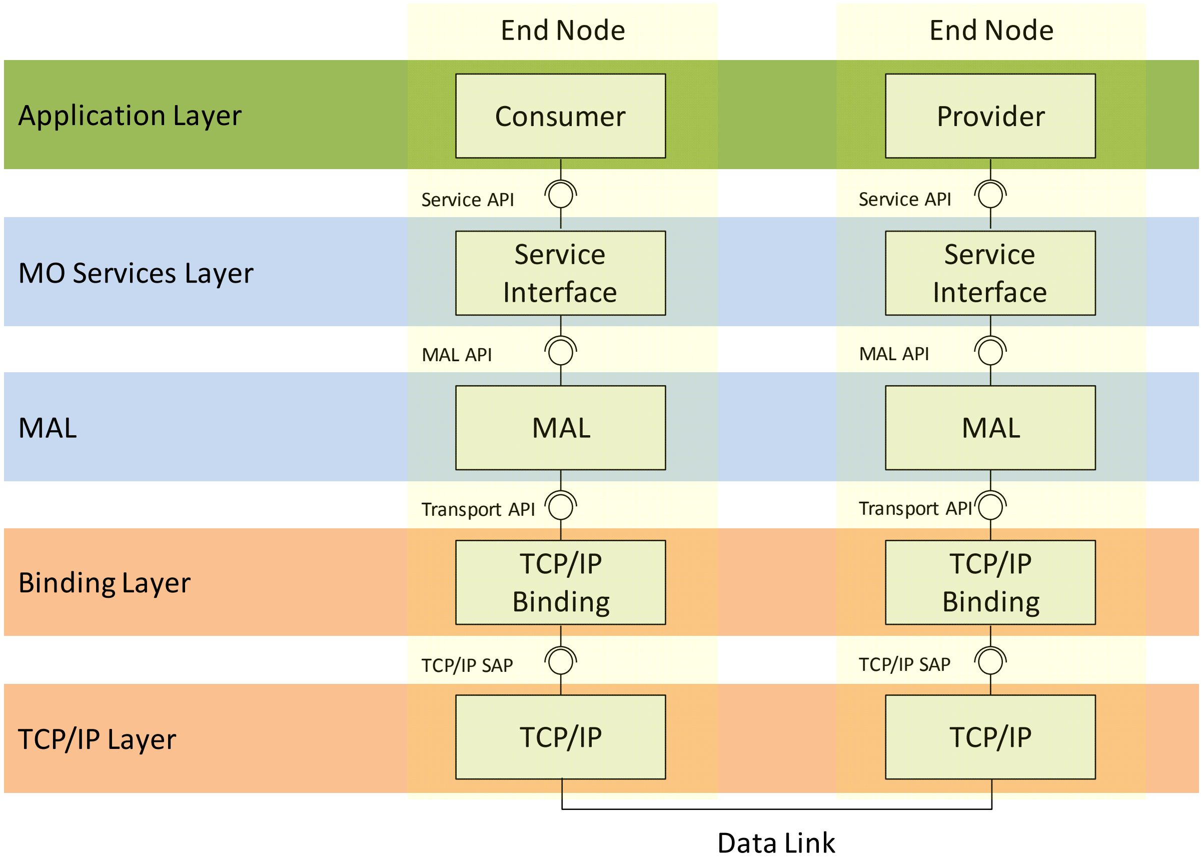

Figure 2-4: Typical Deployment of the MAL TCP/IP Transport

2.4 MAL MESSAGE MAPPING

2.4.1 MAPPING TO TCP/IP

TCP is a stream-oriented transport protocol, providing reliable, ordered, and error-checked delivery of a stream of octets between applications running on hosts communicating over an IP network. This Recommended Standard introduces a MAL TCP/IP PDU, which is delivered using TCP/IP. Therefore each field of the MAL message needs to map to a field of this binary PDU.

This Recommended Standard uses the IP header field ‘Source IP address’ and the TCP header field ‘Source port’ as an identifier of the sending application process. Similarly, it uses the IP header field ‘Destination IP address’ and the TCP header field ‘Destination port’ as an identifier of the receiving application process. Except for the addressing information of URIFrom and URITo, the rest of the MAL message header and the body is encapsulated as the payload of the TCP protocol.

Figure 2-5 illustrates the mapping of the MAL message to the MAL TCP/IP PDU transmitted over TCP/IP. Most of the MAL message fields are mapped according to a one-to-one equivalence. In this case the original MAL header field name is kept and the background colour is blue. However, the following fields require a more complex mapping:

a) The MAL header field ‘URI From’ is mapped to the IP header field ‘Source IP address’, to the TCP header field ‘Source port’, and to the ‘Source Id’ field, defined as part of this Recommended Standard; the background colour is yellow.

b) The MAL header field ‘URI To’ is mapped to the IP header field ‘Destination IP address’, to the TCP header field ‘Destination port’, and to the ‘Destination Id’ field, defined as part of this Recommended Standard; the background colour is yellow.

c) The MAL header fields ‘Interaction Type’ and ‘Interaction Stage’ are mapped to the field ‘SDU Type’; the background colour is purple.

The MAL header fields ‘Authentication Id’, ‘Timestamp’, ‘Priority’, ‘Domain’, ‘Network Zone’ and ‘Session Name’, as well as the ‘Source Id’ and ‘Destination Id’ parts of the MAL header fields ‘URI From’ and ‘URI To’, are optional: their presence in the header is specified by the QoS properties defined in annex C. These fields are mapped to two fields: a presence flag that indicates whether the value is encoded in the MAL TCP/IP PDU header or not, and a field that gives the value in case it is encoded; the background colour is green.

A field named ‘Version Number’ is introduced as first header field of the MAL TCP/IP PDU defined by this Recommended Standard: the purpose of this field is to allow future evolutions of the MAL TCP/IP PDU header as defined by this version of the Recommended Standard; the background colour is red.

In order to allow flexibility in the selection of the encoding formats to be used for MAL message body, this Recommended Standard does not prescribe a mandatory encoding, but introduces two additional fields in the MAL TCP/IP PDU header:

a) The header field ‘Encoding Id’ identifies which encoding format was used to encode the MAL message body; the background colour is red.

b) The header field ‘Variable Length’ reports the length in octets of the remainder of the message, including the variable part of the MAL TCP/IP PDU header and then the encoded MAL message body, which starts after the end of the field; the background colour is red.

The MAL header fields cannot be NULL, even in MAL messages whose interaction type is SEND. The MAL header fields are encoded using the fixed length encoding specified in 3.4 in order to allow direct offset-based references to the header fields.

Finally, the MAL message body field and its equivalent MAL TCP/IP PDU data field have a grey background.

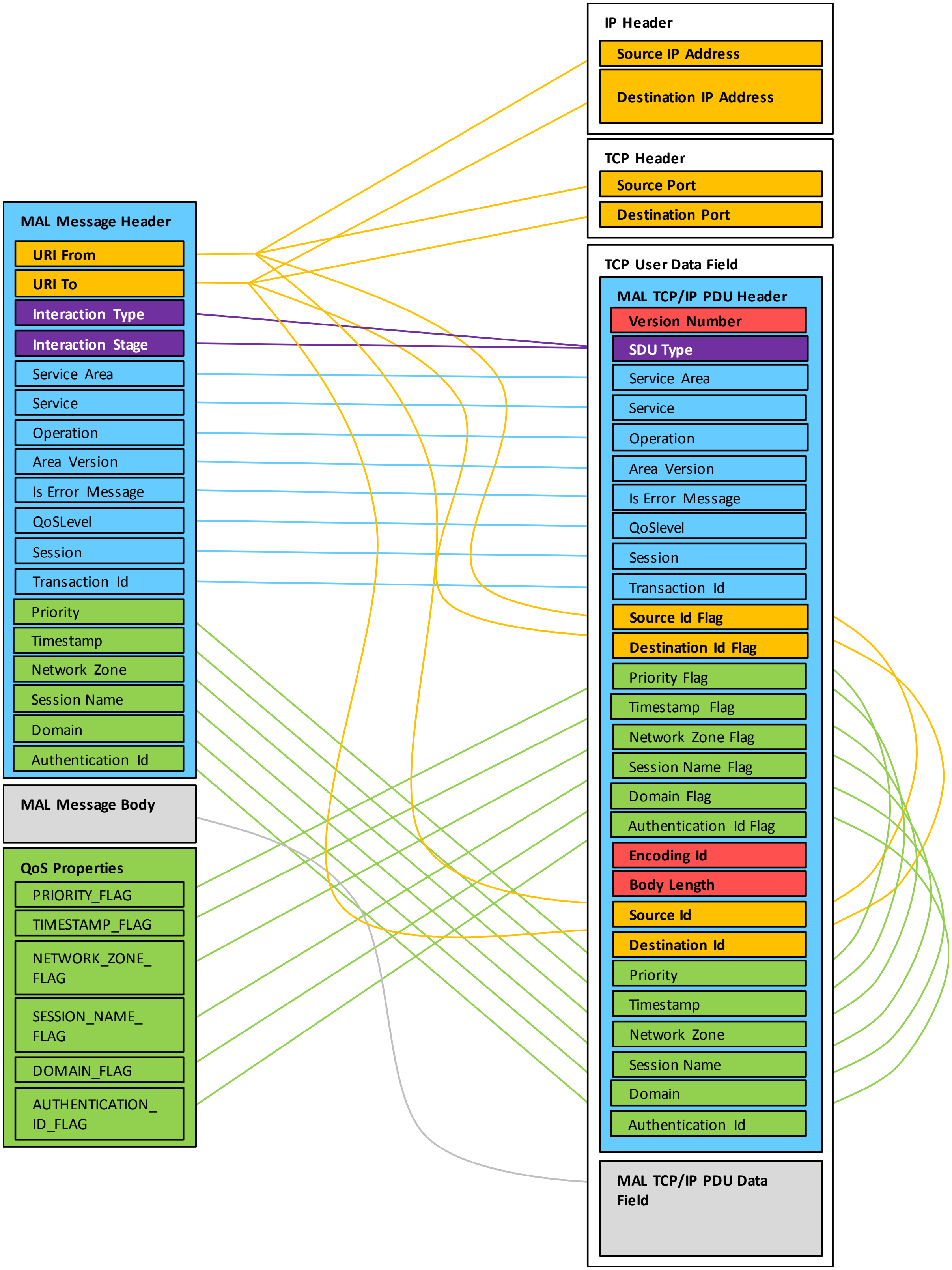

Figure 2-5: MAL Message Mapping to TCP/IP

If the MAL TCP/IP PDU is oversized in relation to the TCP/IP configuration, then TCP/IP might decide to segment the MAL TCP/IP PDU into several TCP/IP segments, as depicted in figure 2-6. The segmentation and reassembly of the TCP segments is done transparently from the point of view of the MAL Binding layer.

The generated TCP/IP segments resulting from the mapping and transmission of a MAL message are delivered according to the ‘Destination IP address’ and ‘Destination port’. The segments are reassembled and the resulting MAL TCP/IP PDU needs to be delivered to the right MAL recipient application, which is identified by the MAL header field ‘URI To’.

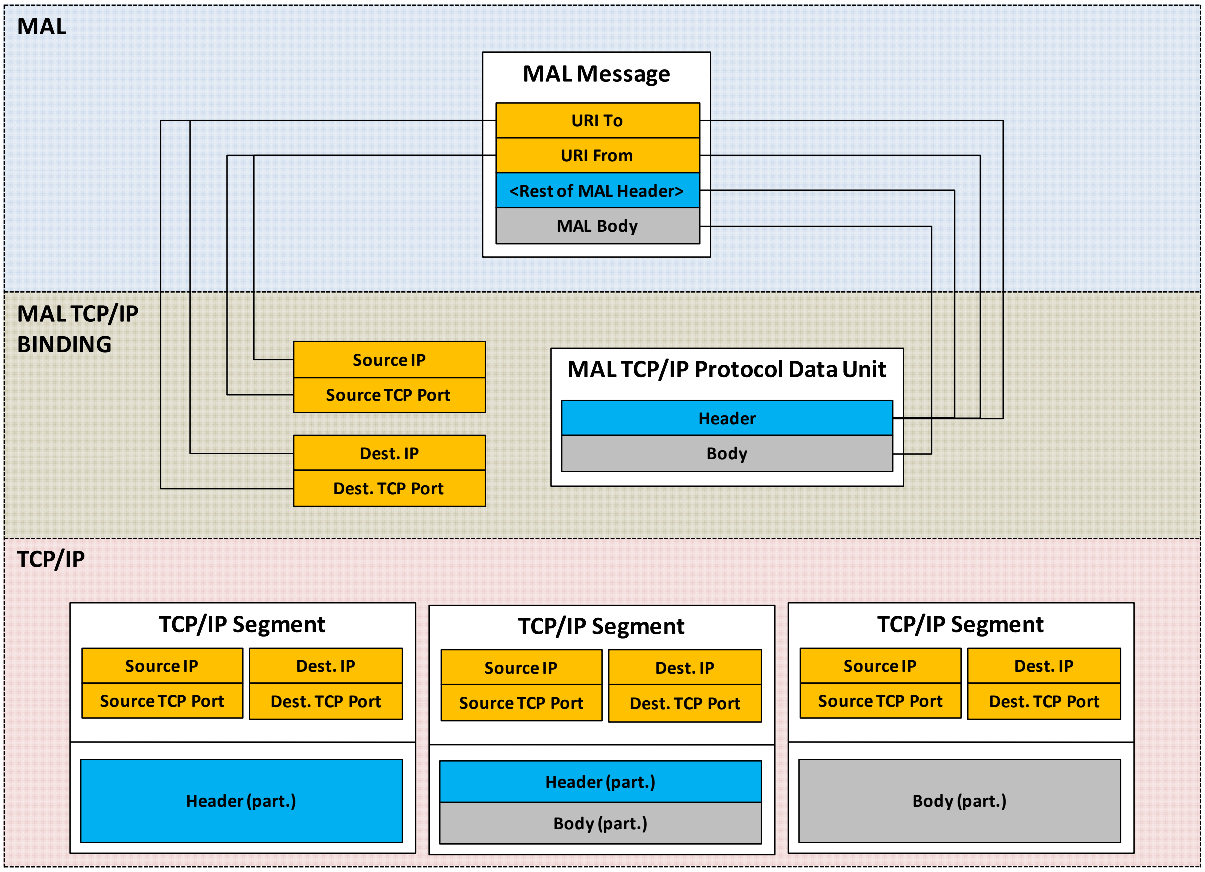

Figure 2-6: MAL TCP/IP PDU and TCP/IP Segmentation

MAL relies on an error control mechanism (e.g., a CRC field) that is handled by the TCP protocol and is hence outside the scope of this Recommended Standard. The Transmission Control Protocol provides checksum at TCP segment level, retransmission of lost TCP segments, and sequencing of out-of-order TCP segments, therefore providing ASSURED QoS implemented by default.

Further information on the rationale for the ordering of the MAL TCP/IP PDU is given in 3.1.

2.4.2 MAPPING CONFIGURATION PARAMETERS

This Recommended Standard defines parameters that allow configuring and optimizing the MAL message mapping and the format of the MAL TCP/IP PDU transmitted via TCP/IP. Those parameters are called ‘Mapping Configuration Parameters’ (MCP). They are defined in annex B.

The MCPs are either mission specific or application process specific. They are managed parameters, defined by some out-of-band agreement.

The MCPs are needed to fully specify the encoding format. These are managed parameters to avoid the cost of additional configuration fields that must be dynamically encoded in the MAL TCP/IP PDU and interpreted at decoding time. The following encoding configuration options can be customized:

a) time code formats;

b) default values of MAL header fields that are not encoded in the MAL TCP/IP PDU, but that are assigned at decoding time to the resulting MAL header.

MCPs must be exchanged out of band between the provider and the user as a separate exchange of configuration information. This could, for example, be done by email, or through a common registry like the Space Assigned Numbers Authority (SANA). The way MCPs are transmitted is not specified here, nor is there a recommendation or standard from CCSDS for this exchange.

2.4.3 MAPPING SPECIFICATION

The MAL Binding to TCP/IP Transport and Split Binary Encoding defines a generic encoding format for every MAL data type (see section 5). Therefore the MAL message can be encoded in a generic way.

A simple tabular notation is used to specify the format of the mapping result, i.e., the MAL TCP/IP PDU header, and the data field, in case the same encoding format are used to encode the MAL message body. This tabular notation is composed of three levels:

a) the name of each field;

b) the encoding format of each field, as defined in section 5:

   1) the length of the format is put in brackets, either in bits or octets;

   2) the length can be variable;

c) if the value to encode is directly given in binary format, then the encoding format is called ‘Binary value’;

d) the condition, or the number of times the field is encoded in, or the static value to be assigned to the field.

The variable length of a field can be caused by an encoding format that is statically defined for a given mission, e.g., the time code formats. A variable length can also result from an encoding format whose length is inherently variable, like:

a) List;

b) String;

c) Varint.

The List and String formats contain a length field as specified respectively in 5.5 and 5.21.

The Varint, as defined by the split binary encoding format, allows to encode an integer using a number of octets that depends on the integer value. Each encoded octet begins with a continuation bit (Bit ‘0’) allowing to know whether there are more octets to decode or not. The advantage is that the number of octets required to encode small integer values is reduced. The drawback is that big integer values require more octets to be encoded. Depending on its value, a four-octet integer can be encoded with one, two, three, four, or five octets. In case of signed integers, the split binary encoding adopts the so-called ‘zig-zag encoding’. In this technique negative numbers are mapped onto positive numbers so that values with a small absolute value have a small Varint encoded value. The signed and unsigned Varint formats are specified in 5.26 and 5.25.

2.4.4 COMPLETE MAPPING

The MAL message mapping completeness is ensured by the following conditions:

a) every MAL data type is mapped;

b) every MAL message field is mapped;

c) every mandatory TCP/IP field is assigned.

Moreover, the translation from a MAL message to its binary form is reversible. No information is lost in the translation from a MAL message to its binary form.

2.5 MAL TRANSPORT INTERFACE MAPPING

The mapping of the MAL transport interface requires specifying the expected behaviour for each of the MAL transport primitives. Three types of behaviour are defined:

a) a MAL transport request initiating a TCP/IP request by sending a MAL TCP/IP PDU split over a set of TCP/IP segments, and returning a reply;

b) a TCP/IP indication initiating a MAL transport indication when receiving a MAL TCP/IP PDU;

c) a MAL transport request returning a reply without calling the TCP/IP layer.

The MAL transport mapping is complete as all the primitives are mapped. Moreover, the behaviour of each primitive is fully specified.

3 MAL MESSAGE MAPPING

3.1 OVERVIEW

This section specifies how the MAL message header, body, and QoS properties are mapped to the MAL TCP/IP PDU transmitted over TCP/IP.

Table 3-1 is taken from reference [1] and provides the full list of fields in the MAL message header.

Table 3-1: MAL Message Header Fields

Field	Type	Value
URI From	URI	Message Source URI
Authentication Id	Blob	Source Authentication Identifier
URI To	URI	Message Destination URI
Timestamp	Time	Message generation timestamp
QoSlevel	QoSLevel	The QoS level of the message
Priority	UInteger	The QoS priority of the message
Domain	List<Identifier>	Domain of the message
Network Zone	Identifier	Network zone of the message
Session	SessionType	Type of session of the message
Session Name	Identifier	Name of the session of the message
Interaction Type	InteractionType	Interaction Pattern Type
Interaction Stage	UOctet	Interaction Pattern Stage
Transaction Id	Long	Unique to consumer
Service Area	UShort	Service Area Identifier
Service	UShort	Service Identifier
Operation	UShort	Service Operation Identifier
Area version	UOctet	Area version
Is Error Message	Boolean	'True' if this is an error message; else 'False'

The MAL message header is mapped to the IP header, TCP header, and MAL TCP/IP PDU header.

The IP header version 4 is specified by the Internet Protocol standard (reference [2]). Table 3-2 expands the IP header specification with the encoding format of each field, as defined in 3.4, and the static values to be assigned when using this standard. IP version 4 supports optional header options, which follow the 'Destination IP Address' header field; the specification and use of IP version 4 header options are not constrained by this Recommended Standard.

Table 3-2: Internet Protocol Version 4 Header Format

Version	Header Length	Type of Service	Total Length (fragment size)	Identification	Flags	Fragment Offset
Binary value (4 bits)	Unsigned 4-bit integer (4 bits)	Binary value (8 bits)	Unsigned 16-bit integer (16 bits)	Unsigned 16-bit integer (16 bits)	Binary value (3 bits)	Unsigned 13-bit integer (13 bits)
Always equal to ‘0100’						
Time To Live	Protocol	Header Checksum	Source IP Address	Destination IP Address		
Unsigned 8-bit integer (8 bits)	Unsigned 8-bit integer (8 bits)	Binary value (16 bits)	Binary value (32 bits)	Binary value (32 bits)		
	Always equal to ‘00000110’					

The IP header version 6 is specified by the Internet Protocol version 6 draft standard (reference [3]). Table 3-3 expands the IP version 6 header specification with the encoding format of each field, as defined in 3.4, and the static values to be assigned when using this standard. IP version 6 supports extension headers, which follow the ‘Destination Address’ header field. Each extension header contains a ‘Next Header’ field. In case extension headers are not used, the ‘Next Header’ field shall be set to ‘00000110’. If extension headers are used, the last extension header shall have the ‘Next Header’ field set to ‘00000110’.

Table 3-3: Internet Protocol Version 6 Header Format

Version	Traffic Class	Flow Label	Payload Length	Next Header	Hop Limit
Binary value (4 bits)	Binary value (8 bits)	Binary value (20 bits)	Unsigned 16-bit integer (16 bits)	Unsigned 8-bit integer (8 bits)	Unsigned 8-bit integer (8 bits)
Always equal to ‘0110’					
Source Address	Destination Address				
Binary value (128 bits)	Binary value (128 bits)				

The TCP header is specified by the Transmission Control Protocol standard (reference [4]). Table 3-4 expands the IP header specification with the encoding format of each field, as defined in 3.4, and the static values to be assigned when using this standard. TCP supports optional header options, which follow the ‘Urgent Pointer’ header field; the specification and use of TCP header options are not constrained by this Recommended Standard.

Table 3-4: Transmission Control Protocol Header Format

Source Port	Destination Port	Sequence Number	Acknowledgement Number	Data Offset	Reserved	Control Bits	Window Size	Checksum	Urgent Pointer
Unsigned 16-bit integer (16 bits)	Unsigned 16-bit integer (16 bits)	Unsigned 32-bit integer (32 bits)	Unsigned 32-bit integer (32 bits)	Unsigned 4-bit integer (4 bits)	Binary value (3 bits)	Binary value (9 bits)	Unsigned 16-bit integer (16 bits)	Binary value (16 bits)	Unsigned 16-bit integer (16 bits)
					Always equal to '000'				

The MAL TCP/IP PDU to be delivered over TCP/IP defined by this Recommended Standard is composed of a header and a data field.

The header is comprised of the fields 'Version Number', 'SDU Type', 'Service Area', 'Service', 'Operation', 'Area version', 'Is Error Message', 'QoSlevel', 'Session', 'Transaction Id', 'Source Id Flag', 'Destination Id Flag', 'Priority Flag', 'Timestamp Flag', 'Network Zone Flag', 'Session Name Flag', 'Domain Flag', 'Authentication Id Flag', 'Encoding Id', 'Variable Length', 'Source Id', 'Destination Id', 'Priority', 'Timestamp', 'Network Zone', 'Session Name', 'Domain', and 'Authentication Id'.

The ordering and structure of the MAL TCP/IP PDU header fields is justified as follows:

a) The field 'Version Number' needs to come first because it identifies the header as defined by this Recommended Standard.

b) The fields 'SDU Type', 'Service Area', 'Service', 'Operation', 'Area Version', and 'Is Error Message' identify the interaction that occurs between the source and destination peers, i.e., the interaction type, the current stage of the interaction, the service that is used, its version, and the operation that is invoked; this information should come first after the header field 'Version Number'.

c) The field 'SDU type' is inserted after the field 'Version Number' in order to reach an octet boundary.

d) The fields 'QoSlevel' and 'Session' are inserted after the field 'Is Error Message' and are encoded respectively using 3 bits and 4 bits, in order to reach an octet boundary (1 + 3 + 4 = 8).

e) The field 'Transaction Id' refines the previous information about the interaction: it identifies the current interaction occurrence.

f) The fields 'Source Id Flag', 'Destination Id Flag', 'Priority Flag', 'Timestamp Flag', 'Network Zone Flag', 'Session Name Flag', 'Domain Flag', and 'Authentication Id Flag' are grouped together in order to reach an octet boundary.

g) The fields ‘Encoding Id’ and ‘Variable Length’ are inserted after the set of presence flags.

h) The fields ‘Source Id’ and ‘Destination Id’ allow to refine the identity of the interacting peers in case there are several consumers or providers of the same service per IP address and TCP port.

i) The variable length fields ‘Source Id’, ‘Destination Id’, ‘Priority’, ‘Timestamp’, ‘Network Zone’, ‘Session Name’, ‘Domain’, and ‘Authentication Id’ are located at the end of the header, in order to fix the length of the initial part of the header.

The format of the MAL TCP/IP PDU header is shown in table 3-5. All the fields of the header are encoded using the encoding format specified in 3.4.

Optional fields are specified out of band, with a mapping configuration parameter as defined by annex B.

Variable length fields are delineated as follows:

a) the formats ‘Blob’, ‘Identifier’ and ‘List’ contain a length field as specified respectively in 3.4.8, 3.4.6, and 3.4.7;

b) the format ‘String’ contains a length field as specified in 3.4.3;

c) the format ‘UInteger’ is encoded as a 4-octet Unsigned Varint, as specified in 3.4.4;

d) the format ‘Time’ needs to be statically defined out of band, as specified in 3.4.5.

The QoS properties defined by annex C allow to set the values of the header flags, which cannot be deduced from the MAL header fields.

The mapping of the MAL message is composed of the following specifications:

a) the URI format to be applied to the MAL header fields ‘URI From’ and ‘URI To’;

b) the mapping of the MAL header to the header of the MAL TCP/IP PDU defined by this Recommended Standard;

c) the values to be assigned to the fields of the IP and TCP headers that are not the result of the MAL message header and body mapping;

d) the mapping of the MAL message body to the body of the MAL TCP/IP PDU defined by this Recommended Standard.

This Recommended Standard does not prescribe the encoding formats to be used for the encoding of the MAL message body. Subsection 3.6.3 and section 5 define the encoding format to be used in case the split binary encoding is selected.

Table 3-5: MAL TCP/IP PDU Header Format

Version Number	SDU Type	Service Area	Service	Operation	Area Version	Is Error Message	QoSlevel	Session	Transaction Id	Source Id Flag	Destination Id Flag	Priority Flag	Time-stamp Flag	Network Zone Flag	Session Name Flag
Binary value (3 bits)	Unsigned 5-bit Integer (5 bits)	Unsigned 16-bit Integer (16 bits)	Unsigned 16-bit Integer (16 bits)	Unsigned 16-bit Integer (16 bits)	Unsigned 8-bit Integer (8 bits)	Binary value (1 bit)	Unsigned 3-bit Integer (3 bits)	Unsigned 4-bit Integer (4 bits)	Unsigned 64-bit Integer (64 bits)	Binary value (1 bit)	Binary value (1 bit)	Binary value (1 bit)	Binary value (1 bit)	Binary value (1 bit)	Binary value (1 bit)

Domain Flag	Authen-tication Id Flag	Encoding Id	Variable Length	Source Id	Destination Id	Priority	Timestamp	Network Zone	Session Name	Domain	Authentication Id
Binary value (1 bit)	Binary value (1 bit)	Unsigned 8-bit Integer (8 bits)	Unsigned 32-bit Integer (32 bits)	String (var., mult. of octet)	String (var., mult. of octet)	UInteger (var. mult. of octet)	Time (var., mult. of octet)	Identifier (var., mult. of octet)	Identifier (var., mult. of octet)	List<Identifier> (var., mult. of octet)	Blob (var., mult. of octet)
				If ‘Source Id Flag’ is ‘1’	If ‘Destination Id Flag’ is ‘1’	If ‘Priority Flag’ is ‘1’	If ‘Timestamp Flag’ is ‘1’	If ‘Network Flag’ is ‘1’	If ‘Session Flag’ is ‘1’	If ‘Domain Flag’ is ‘1’	If ‘Authentication Id Flag’ is ‘1’

3.2 URI FORMAT

3.2.1 The format of the MAL header fields 'URI From' and 'URI To' shall comply with the following rules:

NOTE – The following statements are about the MAL abstraction called URI and not about how it is mapped to the MAL TCP/IP PDU.

a) The URI scheme name shall be 'maltcp'.

b) The scheme name shall be followed by a colon separator ':' and a double slash '//'.

c) The double slash shall be followed by the IP address, using a format depending on the selected Internet Protocol version.

d) If version 6 of the Internet Protocol is used, the IP address shall be represented using the textual representation specified in reference [9], subsection 2.2. The IP address shall be enclosed in square brackets '[' and ']'.

e) If version 4 of the Internet Protocol is used, the IP address shall be represented in dot-decimal notation.

f) The IP address shall be followed by a colon separator ':' and the TCP port number, an integer represented in decimal.

g) The TCP port number shall be a positive integer, excluding zero, strictly less than 65536.

h) The TCP port number may be followed by a slash separator '/' and a non-empty string which is called the 'Source Id' for the field 'URI From' and the 'Destination Id' for the field 'URI To'.

NOTES

1 The source or destination identifier is optional.

2 An example of URI using an Internet Protocol version 4 address is 'maltcp://192.168.0.1:2534/Service'. This URI references the source or destination 'Service' provided by the application accessible from the TCP port '2534' on the host located at address '192.168.0.1'.

3 An example of URI using an Internet Protocol version 6 address is 'maltcp://[2001:0db8:85a3:0000:0000:8a2e:0370:7334]:972/Service'. This URI references the source or destination 'Service' provided by the application accessible from the TCP port '972' on the host located at address '2001:0db8:85a3:0000:0000:8a2e:0370:7334'.

3.2.2 The IP address and TCP port number shall uniquely identify an application that implements the conceptual MAL layer of the MO stack. In a concrete deployment a single or multiple MO service provider/consumer applications may be deployed on top of the MAL layer. For optimization reasons, the conceptual MAL layer and the MO service provider/consumer Application Layers may be merged. In case of multiple MO service provider/consumer applications deployed over a single MAL layer application, the source/destination identifier header introduced by the MAL TCP/IP PDU must be used to uniquely address individual MO service provider/consumer applications.

3.2.3 The source or destination identifier shall be unique for a given MO service provider or consumer application, which is conceptually deployed on top of a MAL application that is uniquely identified by its IP address and TCP port number.

NOTE – A single application, which is identified by a single IP address and a single TCP port number, may represent several MO service provider/consumer entities. In order to uniquely address a single MO service provider/consumer entity, source and destination identifiers are used to refine the IP address and TCP port number.

3.2.4 The scheme name ‘maltcp’ shall be added to the SANA registry ‘MAL Binding URI Scheme Name’ and shall refer to the Mission Operations MAL Binding to TCP/IP Transport and Split Binary Encoding document ‘CCSDS 524.2-B-1’.

NOTE – This SANA registry is defined in D2.2.

3.3 MAL HEADER MAPPING

3.3.1 OVERVIEW

The following subsections provide the mapping of each field of the MAL message header to the MAL TCP/IP PDU.

The mapping configuration parameters are defined in annex B.

3.3.2 URI FROM

3.3.2.1 The MAL header field ‘URI From’ shall be mapped according to one of two available mappings. The generic mapping is defined in 3.3.2.2. The optimized mapping is defined in 3.3.2.3 through 3.3.2.7. The optimized mapping may be used only when the transport uses a connection where the TCP ‘Local port’ is the TCP port number of the MAL header field ‘URI From’ to transmit the message.

3.3.2.2 The MAL header field ‘URI From’ shall be assigned to the MAL TCP/IP Protocol Data Unit header field ‘Source Id’ and the MAL TCP/IP Protocol Data Unit header field ‘Source Id Flag’ shall be set to the value ‘1’.

3.3.2.3 If the MAL message is mapped to a MAL TCP/IP PDU delivered using IP version 4, the IP address part of the MAL header field ‘URI From’ shall be assigned to the IP version 4 header field ‘Source IP Address’.

3.3.2.4 If the MAL message is mapped to a MAL TCP/IP PDU delivered using IP version 6, the IP address of the MAL header field 'URI From' shall be assigned to the IP version 6 header field 'Source Address'.

3.3.2.5 The TCP port number of the MAL header field 'URI From' shall be assigned to the TCP header field 'Source Port'.

3.3.2.6 If the MAL header field 'URI From' contains a Source Id, then this identifier shall be assigned to the MAL TCP/IP PDU header field 'Source Id' and the MAL TCP/IP PDU header field 'Source Id Flag' shall be set to the value '1'.

3.3.2.7 If the MAL header field 'URI From' does not contain a Source Id, then the MAL TCP/IP PDU header field 'Source Id Flag' shall be set to the value '0' and the MAL TCP/IP PDU header field 'Source Id' shall be left out.

3.3.3 AUTHENTICATION ID

3.3.3.1 If the QoS property AUTHENTICATION_ID_FLAG is TRUE, or not passed with the MAL message, then the MAL header field 'Authentication Id' shall be assigned to the MAL TCP/IP PDU header field 'Authentication Id' and the 'Authentication Id Flag' shall be set to the value '1'.

3.3.3.2 If the QoS property AUTHENTICATION_ID_FLAG is FALSE, then the following rules shall be applied:

a) the MAL TCP/IP PDU header field 'Authentication Id Flag' shall be set to the value '0', and the MAL TCP/IP PDU header field 'Authentication Id' shall be left out;

b) if the mapping configuration parameter AUTHENTICATION_ID is defined, then its value shall be assigned to the MAL header field 'Authentication Id';

c) if the mapping configuration parameter AUTHENTICATION_ID is not defined, then an empty MAL::Blob shall be assigned to the MAL header field 'Authentication Id'.

3.3.4 URI TO

3.3.4.1 If the MAL message is mapped to a MAL TCP/IP PDU delivered using IP version 4, the IP address of the MAL header field 'URI To' shall be assigned to the IP version 4 header field 'Destination IP Address'.

3.3.4.2 If the MAL message is mapped to a MAL TCP/IP PDU delivered using IP version 6, the IP address of the MAL header field 'URI To' shall be assigned to the IP version 6 header field 'Destination Address'.

3.3.4.3 The TCP port number of the MAL header field 'URI To' shall be assigned to the TCP header field 'Destination Port'.

3.3.4.4 If the MAL header field 'URI To' contains a Destination Id, then this identifier shall be assigned to the MAL TCP/IP PDU header field 'Destination Id' and the MAL TCP/IP PDU header field 'Destination Id Flag' shall be set to the value '1'.

3.3.4.5 If the MAL header field 'URI To' does not contain a Destination Id, then the MAL TCP/IP PDU header field 'Destination Id Flag' shall be set to the value '0' and the MAL TCP/IP PDU header field 'Destination Id' shall be left out.

3.3.5 TIMESTAMP

3.3.5.1 If the QoS property TIMESTAMP_FLAG is TRUE, or not passed with the MAL message, then the MAL header field 'Timestamp' shall be assigned to the MAL TCP/IP PDU header field 'Timestamp' and the 'Timestamp Flag' shall be set to the value '1'.

3.3.5.2 If the QoS property TIMESTAMP_FLAG is FALSE, then the following rules shall be applied:

a) the MAL TCP/IP PDU header field 'Timestamp Flag' shall be set to the value '0' and the MAL TCP/IP PDU header field 'Timestamp' shall be left out;

b) the value '0' shall be assigned to the MAL header field 'Timestamp'.

3.3.6 QOSLEVEL

The value of the MAL header field 'QoSlevel' shall be assigned to the MAL TCP/IP PDU header field 'QoSlevel' as specified by table 3-6.

Table 3-6: QoSlevel Field Encoding

QoSlevel value	Encoded Value
BESTEFFORT	0
ASSURED	1
QUEUED	2
TIMELY	3

3.3.7 PRIORITY

3.3.7.1 If the QoS property PRIORITY_FLAG is TRUE, or not passed with the MAL message, then the MAL header field 'Priority' shall be assigned to the MAL TCP/IP PDU header field 'Priority' and the 'Priority Flag' shall be set to the value '1'.

3.3.7.2 If the QoS property PRIORITY_FLAG is FALSE, then the following rules shall be applied:

a) the MAL TCP/IP PDU header field ‘Priority Flag’ shall be set to the value ‘0’ and the MAL TCP/IP PDU header field ‘Priority’ shall be left out;

b) if the mapping configuration parameter PRIORITY is defined, then its value shall be assigned to the MAL header field ‘Priority’;

c) if the mapping configuration parameter PRIORITY is not defined, then the value ‘0’ shall be assigned to the MAL header field ‘Priority’.

3.3.8 DOMAIN

3.3.8.1 If the QoS property DOMAIN_FLAG is TRUE, or not passed with the MAL message, then the MAL header field ‘Domain’ shall be assigned to the MAL TCP/IP PDU header field ‘Domain’ and the ‘Domain Flag’ shall be set to the value ‘1’.

3.3.8.2 If the QoS property DOMAIN_FLAG is FALSE, then the following rules shall be applied:

a) the MAL TCP/IP PDU header field ‘Domain Flag’ shall be set to the value ‘0’ and the MAL TCP/IP PDU header field ‘Domain’ shall be left out;

b) if the mapping configuration parameter DOMAIN is defined, then its value shall be assigned to the MAL header field ‘Domain’;

c) if the mapping configuration parameter DOMAIN is not defined, then an empty List<MAL::Identifier> shall be assigned to the MAL header field ‘Domain’.

3.3.9 NETWORK ZONE

3.3.9.1 If the QoS property NETWORK_ZONE_FLAG is TRUE, or not passed with the MAL message, then the MAL header field ‘Network Zone’ shall be assigned to the MAL TCP/IP PDU header field ‘Network Zone’ and the ‘Network Flag’ shall be set to the value ‘1’.

3.3.9.2 If the QoS property NETWORK_ZONE_FLAG is FALSE, then the following rules shall be applied:

a) the MAL TCP/IP PDU header field ‘Network Zone Flag’ shall be set to the value ‘0’ and the MAL TCP/IP PDU header field ‘Network Zone’ shall be left out;

b) if the mapping configuration parameter NETWORK_ZONE is defined, then its value shall be assigned to the MAL header field ‘Network Zone’;

c) if the mapping configuration parameter NETWORK_ZONE is not defined, then an empty MAL::Identifier shall be assigned to the MAL header field ‘Network Zone’.

3.3.10 SESSION

The value of the MAL header field ‘Session’ shall be assigned to the MAL TCP/IP PDU header field ‘Session’ as specified by table 3-7.

Table 3-7: Session Field Encoding

Session value	Encoded Value
LIVE	0
SIMULATION	1
REPLAY	2

3.3.11 SESSION NAME

3.3.11.1 If the QoS property SESSION_NAME_FLAG is TRUE, or not passed with the MAL message, then the MAL header field ‘Session Name’ shall be assigned to the MAL TCP/IP PDU header field ‘Session Name’ and the ‘Session Flag’ shall be set to the value ‘1’.

3.3.11.2 If the QoS property SESSION_NAME_FLAG is FALSE, then the following rules shall be applied:

a) the MAL TCP/IP PDU header field ‘Session Name Flag’ shall be set to the value ‘0’ and the MAL TCP/IP PDU header field ‘Session Name’ shall be left out;

b) if the mapping configuration parameter SESSION_NAME is defined, then its value shall be assigned to the MAL header field ‘Session Name’;

c) if the mapping configuration parameter SESSION_NAME is not defined, then an empty MAL::Identifier shall be assigned to the MAL header field ‘Session Name’.

3.3.12 INTERACTION TYPE AND STAGE

The MAL header fields ‘Interaction Type’ and ‘Interaction Stage’ shall be mapped to the MAL TCP/IP PDU header field ‘SDU Type’ as defined by table 3-8, where the SDU type used to map an interaction stage raising an error shall be the same as the SDU type used by this interaction stage without an error.

Table 3-8: Interaction Type and Stage Mapping

Interaction Type	Interaction Stage	SDU Type (decimal)
SEND	SEND	0
SUBMIT	SUBMIT ACK ERROR	1 2 2
REQUEST	REQUEST RESPONSE ERROR	3 4 4
INVOKE	INVOKE ACK ACK_ERROR RESPONSE RESPONSE ERROR	5 6 6 7 7
PROGRESS	PROGRESS ACK ACK_ERROR UPDATE UPDATE_ERROR RESPONSE RESPONSE ERROR	8 9 9 10 10 11 11
PUBLISH-SUBSCRIBE	REGISTER REGISTER_ACK REGISTER_ERROR PUBLISH_REGISTER PUBLISH_REGISTER_ACK PUBLISH_REGISTER_ERROR PUBLISH PUBLISH_ERROR NOTIFY NOTIFY ERROR DEREGISTER DEREGISTER ACK PUBLISH_DEREGISTER PUBLISH_DEREGISTER_ACK	12 13 13 14 15 15 16 16 17 17 18 19 20 21

3.3.13 TRANSACTION ID

The value of the MAL header field ‘Transaction Id’ shall be assigned to the MAL TCP/IP PDU header field ‘Transaction Id’.

3.3.14 SERVICE AREA

The value of the MAL header field ‘Service Area’ shall be assigned to the MAL TCP/IP PDU header field ‘Service Area’.

3.3.15 SERVICE

The value of the MAL header field ‘Service’ shall be assigned to the MAL TCP/IP PDU header field ‘Service’.

3.3.16 OPERATION

The value of the MAL header field ‘Operation’ shall be assigned to the MAL TCP/IP PDU header field ‘Operation’.

3.3.17 AREA VERSION

The value of the MAL header field ‘Area Version’ shall be assigned to the MAL TCP/IP PDU header field ‘Area Version’.

3.3.18 IS ERROR MESSAGE

If the MAL header field ‘Is Error Message’ is TRUE, then the MAL TCP/IP PDU header field ‘Is Error Message’ shall be assigned with the value ‘1’; otherwise it shall be assigned with the value ‘0’.

3.4 HEADER FIELDS ENCODING RULES

3.4.1 OVERVIEW

The following subsections specify the encoding rules of each field type of the MAL message header to the MAL TCP/IP PDU. Only the encoding rules for the types used in the definition of the TCP/IP header and the MAL TCP/IP PDU header are specified.

3.4.2 UNSIGNED N-BIT INTEGER

3.4.2.1 An unsigned N-bit integer field shall be encoded on a given number of bits N.

3.4.2.2 The bit ‘Bit N–1’ shall designate the least significant bit of the unsigned integer.

3.4.2.3 Each bit of the unsigned integer shall be encoded from ‘Bit 0’ to ‘Bit N–1’.

3.4.3 STRING

A String field shall be encoded as specified in 5.21.

3.4.4 UINTEGER

A UInteger field shall be encoded as specified in 5.18.

3.4.5 TIME

A Time field shall be encoded as specified in 5.22.

3.4.6 IDENTIFIER

An Identifier field shall be encoded as specified in 5.12.

3.4.7 LIST

3.4.7.1 A List field shall be encoded as specified in 5.5, with the exceptions stated in 3.4.7.2 and 3.4.7.3.

3.4.7.2 The field 'Presence Field' of a Nullable Element shall be encoded as an Unsigned 8-bit Integer.

3.4.7.3 The field 'Presence Field' shall be set to '0' if the element is NULL; it shall be set to '1' if the element is not NULL.

3.4.8 BLOB

A Blob field shall be encoded as specified in 5.7.

3.5 MAL TCP/IP PROTOCOL DATA UNIT SPECIFIC FIELDS

3.5.1 OVERVIEW

The following subsections specify the values to be assigned to the MAL TCP/IP PDU header fields that are not the result of the MAL header mapping.

3.5.2 VERSION NUMBER

3.5.2.1 The field 'Version Number' shall identify the structure of the MAL TCP/IP PDU header as defined by this Recommended Standard.

3.5.2.2 The field 'Version Number' shall be assigned with the binary value '001'.

3.5.2.3 The version number '001' shall be added to the SANA registry 'MAL TCP/IP Binding Version Number' and shall refer to the Mission Operations MAL Binding to TCP/IP Transport and Split Binary Encoding document 'CCSDS 524.2-B-1'.

NOTE – This SANA registry is defined in D2.1.

3.5.3 ENCODING ID

3.5.3.1 The field 'Encoding Id' shall identify the encoding rules used to encode the MAL message body.

3.5.3.2 The 'Encoding Id' shall be an integer number greater or equal to 0 and strictly lower than 256.

3.5.3.3 The mapping between the value of the 'Encoding Id' header field and the encoding rules is not defined as part of this Recommended Standard.

3.5.3.4 Organizations implementing this Recommended Standard shall agree on the encoding rules and related identification using this field.

3.5.4 VARIABLE LENGTH

The length in octets of the variable header part and the encoded MAL message body shall be assigned to the MAL TCP/IP PDU header field 'Variable Length' encoded as Unsigned 32-bit Integer.

NOTE – The overall size of an encoded MAL message will be limited due to the use of 32-bit "Body Length" field in the MAL TCP/IP PDU.

3.6 MAL MESSAGE BODY MAPPING

3.6.1 OVERVIEW

The following subsections specify how the MAL message body is mapped to the MAL TCP/IP PDU.

Subsection 3.6.3 specifies an encoding format that can be used to encode the MAL message body.

NOTE – This Recommended Standard does not define the adoption of the encoding format specified in 3.6.3 as the mandatory encoding format to be adopted for the MAL TCP/IP transport binding; others may be used, such as reference [8].

3.6.2 BODY MAPPING

3.6.2.1 The MAL message body shall be encoded using the selected encoding format and assigned to the MAL TCP/IP PDU data field.

3.6.2.2 The length in octets of the encoded MAL message body shall be added to the MAL TCP/IP PDU header field 'Variable Length'.

3.6.2.3 The identifier of the selected encoding format for the MAL message body shall be assigned to the MAL TCP/IP PDU header field 'Encoding Id'.

3.6.3 BODY ENCODING

3.6.3.1 Overview

This subsection specifies how the MAL message body is encoded using the split binary encoding.

The mapping from the MAL Data Types to the encoding format is available in section 5.

NOTE – The body encoding format can be reused by a MAL binding to a messaging technology that is not TCP/IP.

The split binary encoding is an extension to the Varint encoding specified in reference [8]:

a) All Boolean values and presence flags that are part of the message are encoded into a bit field; the bit field is placed at the beginning of the encoded message.

b) All signed integer values are encoded using a technique called ZigZag encoded Varint (see 5.26): negative values are mapped to positive values and the resulting value is then encoded using the standard Varint algorithm. The mapping is performed so that values with a small absolute value have a small Varint encoded value. A signed integer value n is mapped using the following formulas:

$(n \geq 0) \rightarrow 2n$

$(n < 0) \rightarrow -2n - 1$

The bit field is split into two parts; the first part is a 32 bit unsigned Varint holding the size in octets of the bit field, followed by the bit field itself. The bit field is encoded as an unsigned octet array of an integer number of octets. The Least Significant Bit (LSB) of the first octet of the array holds the first Boolean value, the next bit the next Boolean value.

To increase efficiency of the encoded value, only the bits up to, and including, the most significant '1' are stored. Any bit after the most significant '1' are assumed to be '0'.

The Split Binary Encoding format is represented in table 3-9.

Table 3-9: Split Binary Encoding Format

Bit Field Length	Bit Field	Body
UInteger	Binary value	Binary value
(see 5.18)		(see 3.6.3.3 and 5)
(var. mult. of octet)	(var., mult. of octet)	(var., mult. of octet)

3.6.3.2 Body

3.6.3.2.1 The encoded MAL message body shall be composed by:

a) a length field, referred to as 'Bit Field Length';

b) a bit field, referred to as 'Bit Field';

c) the encoded elements of the MAL message body.

3.6.3.2.2 The field 'Bit Field Length' shall be placed at the beginning of the encoded message.

3.6.3.2.3 The field 'Bit Field' shall follow the 'Bit Field Length' field.

3.6.3.2.4 Each element of the MAL message body, with the exception of the MAL::Boolean values, shall be encoded as a Body Element in the same order as it is declared in the operation definition.

3.6.3.2.5 The encoded elements of the MAL message body shall follow the field 'Bit Field'.

3.6.3.2.6 Any Boolean value that is part of the MAL message body, including any Boolean presence flag used for Nullable Element types (see section 5) and Publish Update elements (see 3.6.3.5), shall be encoded into the field 'Bit Field'.

3.6.3.2.7 The field 'Bit Field Length' shall be assigned with the length in octets of the 'Bit Field' field.

3.6.3.2.8 The value of the 'Bit Field Length' field shall be encoded as an Unsigned Varint.

3.6.3.2.9 The field 'Bit Field' shall be encoded as an unsigned octet array of an integer number of octets.

3.6.3.2.10 Boolean values of the MAL message body shall be encoded as bits, starting from the LSB of the first octet in the same order they appear in the MAL message body.

3.6.3.2.11 If the Boolean value is equal to 'true', the corresponding bit in the bit field shall be set to '1'.

3.6.3.2.12 If the Boolean value is equal to ‘false’, the corresponding bit in the bit field shall be set to ‘0’.

3.6.3.2.13 The field ‘Bit Field’ shall store only the bits up to, and including, the most significant ‘1’. Any bit after the most significant ‘1’ shall be assumed to have value ‘0’.

3.6.3.2.14 Padding bits of the last octet of the field ‘Bit Field’ shall be set to ‘0’.

3.6.3.2.15 If the body declared by the operation is empty, then the encoded MAL message body shall be empty.

3.6.3.3 Body Element

3.6.3.3.1 If the MAL header fields ‘Interaction Type’ and ‘Interaction stage’ are respectively equal to PUBLISH-SUBSCRIBE and PUBLISH, and if the body element is typed ‘List< <<Update Value Type>> >’ (reference [1]), then the body element shall be encoded as follows:

Area	Service	Area Version	Type	Body Element
Unsigned Integer (16-bit)	Unsigned Integer (16-bit) (see 5.25) (variable, multiple of octet)	Unsigned Integer (8-bit)	Signed Integer (24-bit)	Publish Update List (variable, multiple of octet)
If <<Update Value Type>> is abstract				

3.6.3.3.2 The area number, service, area version, and type of the element shall be merged into a single Unsigned 64-bit Integer and encoded as an Unsigned Varint.

3.6.3.3.3 The area number of the element actual type shall be encoded as an Unsigned 16-bit Integer and assigned to the first 16 most significant bits of the Unsigned 64-bit Integer.

3.6.3.3.4 If the element actual type is defined by a service, then the service number of the element actual type shall be encoded as an Unsigned 16-bit Integer and assigned to the 16 most significant bits of the Unsigned 64-bit Integer that follows the encoded area number.

3.6.3.3.5 If the element actual type is not defined by a service, then the value ‘0’ shall be encoded as an Unsigned 16-bit Integer and assigned to the 16 most significant bits of the Unsigned 64-bit Integer that follows the encoded area number.

3.6.3.3.6 The area version of the element actual type shall be encoded as an Unsigned 8-bit Integer and assigned to the 8 most significant bits of the Unsigned 64-bit Integer that follows the encoded service number.

3.6.3.3.7 The short form part of the element actual type shall be encoded as a Signed 24-bit Integer and assigned to the 24 most significant bits of the Unsigned 64-bit Integer that follows the encoded area version.

3.6.3.3.8 If the MAL header fields ‘Interaction Type’ and ‘Interaction stage’ are respectively equal to PUBLISH-SUBSCRIBE and PUBLISH, and if the body element is typed ‘List<UpdateHeader>’, then the body element shall be encoded as an Element (see section 5).

3.6.3.3.9 If the MAL header fields ‘Interaction Type’ and ‘Interaction stage’ are respectively equal to PUBLISH-SUBSCRIBE and NOTIFY, and if the body element is typed ‘List< <<Update Value Type>> >’ (reference [1]), then the body element shall be encoded as an Element (see section 5).

3.6.3.3.10 If the MAL header fields ‘Interaction Type’ and ‘Interaction stage’ are respectively equal to PUBLISH-SUBSCRIBE and NOTIFY, and if the body element is typed either Identifier or ‘List<UpdateHeader>’, then the body element shall be encoded as an Element (see section 5).

3.6.3.3.11 If the MAL header field ‘Interaction Type’ is PUBLISH-SUBSCRIBE and if the ‘Interaction stage’ is neither PUBLISH or NOTIFY and if the MAL header field ‘Is Error Message’ is FALSE, then each body element shall be encoded as an Element (see section 5).

3.6.3.3.12 If the MAL header field ‘Is Error Message’ is TRUE, then the body elements shall be encoded as follows:

a) the first body element (typed ‘UInteger’) shall be encoded as a UInteger (see 5.18);

b) the second body element (typed ‘Element’) shall be encoded as a Nullable Element (see section 5).

3.6.3.3.13 If the MAL header field ‘Interaction Type’ is not PUBLISH-SUBSCRIBE and if the MAL header field ‘Is Error Message’ is FALSE, then each body element shall be encoded as a Nullable Element (see section 5).

3.6.3.4 Publish Update List

3.6.3.4.1 A Publish Update List shall be encoded as follows:

List Length	Update
UInteger (variable, multiple of octet)	Publish Update (variable, multiple of octet)
	Repeated for every update in the list

3.6.3.4.2 The field ‘List Length’ shall be assigned with the length of the list encoded as a UInteger (see 5.18).

3.6.3.4.3 The updates shall be encoded in the same order as in the list.

3.6.3.5 Publish Update

3.6.3.5.1 The following fields shall be encoded:

Presence Flag	Encoded Update Size	Update
Encoded in the bit field (see 3.6.3.2)	UInteger (variable, multiple of octet)	Element (see 5.2) (variable, multiple of octet)
	If Presence Flag is TRUE	

3.6.3.5.2 The field 'Presence Flag' shall be FALSE if the element is NULL; it shall be TRUE if the element is not NULL.

3.6.3.5.3 The field 'Encoded Update Size' shall be assigned with the number of octets used to encode the following field 'Update'.

3.6.3.5.4 The field 'Encoded Update Size' shall be encoded as a UInteger (see 5.18).

4 MAL TRANSPORT INTERFACE MAPPING

4.1 OVERVIEW

The MAL specification (reference [1]) ‘Transport Interface’ section defines the interface to be provided by the MAL TCP/IP transport binding layer. The following subsections specify the expected behaviour for each of the MAL transport interface request and indication primitives. If an indication is a response to a request, then the behaviour of the indication is specified in the same subsection as the request.

The following primitives are defined in the MAL transport interface and need to be provided by the MAL TCP/IP transport binding layer:

a) SUPPORTEDQOS request;

b) SUPPORTEDQOS RESPONSE indication;

c) SUPPORTEDIP request;

d) SUPPORTEDIP RESPONSE indication;

e) TRANSMIT request;

f) TRANSMIT ACK indication;

g) TRANSMIT ERROR indication;

h) TRANSMITMULTIPLE request;

i) TRANSMITMULTIPLE ACK indication;

j) TRANSMITMULTIPLE ERROR indication;

k) RECEIVE indication;

l) RECEIVEMULTIPLE indication.

The parameters are listed in table 4-1.

The following primitives defined by the Transmission Control Protocol (reference [4]) are used by the mapping:

a) ‘OPEN’;

b) ‘SEND’;

c) ‘RECEIVE’;

d) ‘CLOSE’.

The parameters are listed in table 4-2.

Table 4-1: MAL Transport Interface Primitives

Primitive	**Parameters**
SUPPORTEDQOS request	QoS Level
SUPPORTEDQOS RESPONSE indication	Boolean
SUPPORTEDIP request	Interaction Type
SUPPORTEDIP RESPONSE indication	Boolean
TRANSMIT request	MAL Message QoS Properties
TRANSMIT ACK indication	-
TRANSMIT ERROR indication	MAL Message Header Error Number Extra Information QoS Properties
TRANSMITMULTIPLE request	List of: – MAL Message – QoS Properties
TRANSMITMULTIPLE ACK indication	-
TRANSMITMULTIPLE ERROR indication	List of: – MAL Message Header – Error Number – Extra Information – QoS Properties
RECEIVE indication	MAL Message QoS Properties
RECEIVEMULTIPLE indication	List of: – MAL Message – QoS Properties

Table 4-2: TCP Interface Primitives

Primitive	**Parameters**
OPEN	Local port Foreign address Active/Passive Timeout (optional) Precedence (optional) Security/compartment (optional) Options (optional)
SEND	Local connection name Buffer address Buffer count PUSH flag URGENT flag Timeout (optional)
RECEIVE	Local connection name Buffer address Buffer count
CLOSE	Local connection name

Because of the stream-oriented nature of the TCP protocol, delivered MAL TCP/IP PDUs might be split across several calls to the TCP 'RECEIVE' primitive. A single MAL TCP/IP PDU is read using the following approach:

a) The initial 23 octets of the header are read; this part of the MAL TCP/IP PDU header is fixed: it contains the length of the remainder of the MAL TCP/IP PDU (variable header part plus body) in octets and the presence flags of each optional header field.

b) The remainder of the message may be read in one pass: the number of octets specified by the 'Variable Length' header field is retrieved by invoking the TCP 'RECEIVE' primitive.

c) As soon as a MAL application contains no more MO service consumer and no more MO service provider the TCP 'CLOSE' primitive shall be called for each connection with the local connection name passed to the 'CLOSE' primitive as 'Local connection name'. Additionally, implementation-specific calls to the 'CLOSE' primitive may be performed.

NOTE – The approach defined above is informative and does not prescribe any specific implementation of this Recommended Standard.

4.2 SUPPORTEDQOS REQUEST

4.2.1 The SUPPORTEDQOS request primitive shall be provided.

4.2.2 Support for the Quality of Service (QoS) levels defined by MAL shall depend on the capabilities of the underlying layer used to convey the TCP segments.

4.3 SUPPORTEDIP REQUEST

4.3.1 The SUPPORTEDIP request primitive shall be provided.

4.3.2 The SUPPORTEDIP request primitive shall return TRUE for the interaction patterns SEND, SUBMIT, REQUEST, INVOKE, and PROGRESS.

4.3.3 The SUPPORTEDIP request primitive shall return FALSE for the interaction pattern PUBLISH-SUBSCRIBE.

4.3.4 The MAL layer shall support PUBLISH-SUBSCRIBE itself.

NOTE – The MAL specification (reference [1]) requires that implementations of the MAL layer support the Publish-Subscribe pattern but that they can delegate this pattern to a transport that supports the pattern natively. The TCP/IP protocol stack does not support the Publish-Subscribe pattern natively; therefore a MAL implementation has to support this pattern itself.

4.4 TRANSMIT REQUEST

4.4.1 The TRANSMIT request primitive shall be provided in order to translate a MAL message into one MAL TCP/IP PDU and send it by calling the TCP primitive ‘SEND’.

4.4.2 If any of the MAL header fields is NULL, then the TRANSMIT ERROR primitive shall be called with the error number MAL::INTERNAL.

4.4.3 The MAL header field ‘Transaction Id’ shall not be NULL.

4.4.4 The MAL message header fields and body elements shall be mapped to the MAL TCP/IP PDU according to the specification given in section 3 of this Recommended Standard.

4.4.5 If either of the fields ‘URI From’ or ‘URI To’ is not compliant with the URI format defined in 3.2, then the TRANSMIT ERROR primitive shall be called with the error number MAL::INTERNAL.

4.4.6 When the MAL TCP/IP PDU must be delivered, then the following rules shall be applied:

a) If there is no existing connection to the MAL application identified by the IP destination address and TCP destination port, a TCP/IP connection shall be established using the TCP ‘OPEN’ primitive.

b) A TCP/IP connection shall be identifiable by its local connection name, returned by a successful invocation of the TCP ‘OPEN’ primitive.

c) The IP destination address of the MAL header field ‘URI To’ and the TCP destination port of the MAL header field ‘URI To’ shall be passed to the TCP ‘OPEN’ primitive as ‘Foreign address’.[1]

d) The TCP source port of the MAL header field ‘URI From’ may be passed to the TCP ‘OPEN’ primitive as ‘Local port’;[1] if it is passed, then the optimized mapping of the MAL header field ‘URI From’ defined in 3.3.2 may be used, otherwise the generic mapping shall be used.

e) The parameter ‘Active/Passive’ of the TCP ‘OPEN’ primitive shall be set to ‘Active’.

f) If the TCP ‘OPEN’ primitive fails, then the TRANSMIT ERROR primitive shall be called with the error number MAL::INTERNAL.

g) If the TCP ‘OPEN’ primitive successfully returns with the local connection name, the MAL TCP/IP PDU shall be sent using the TCP ‘SEND’ primitive.

[1] The rules defined in 4.4.6 c) and d) establish a prescriptive mapping between the port parts of the service URIs and the port parts of the TCP headers. This prescriptive mapping eases interoperability, and it works efficiently for single-hop service operations. The constraints introduced by this mapping in case of multi-hop service operations shall be assessed.

h) The local connection name of the connection established between the two MAL applications shall be passed to the TCP ‘SEND’ primitive as ‘Local connection name’.

i) The address to the first octet of the MAL TCP/IP PDU to transmit shall be passed to the TCP ‘SEND’ primitive as ‘Buffer address’.

j) The length in octets of the complete MAL TCP/IP PDU to transmit shall be passed to the TCP ‘SEND’ primitive as ‘Buffer count’.

k) The ‘PUSH’ parameter of the TCP ‘SEND’ primitive shall be set to ‘TRUE’.

l) The ‘URGENT’ parameter of the TCP ‘SEND’ primitive shall be set to ‘FALSE’.

m) The TCP segments that result from the segmentation of the MAL TCP/IP PDU mapped from the MAL message shall be delivered to the MAL application identified by the IP destination address and TCP destination port of the MAL header field ‘URI To’.

4.4.7 If an error is returned by the invocation of the TCP ‘SEND’ primitive, then the TRANSMIT ERROR primitive shall be called with the error number MAL::INTERNAL.

4.4.8 If the invocation of the TCP ‘SEND’ primitive successfully returns, then the TRANSMIT ACK primitive shall be called.

4.4.9 If the MAL TCP/IP PDU Data Field is larger than $2^{32}-1$ octets, then the TRANSMIT ERROR primitive shall be called with the error number MAL::INTERNAL.

4.5 TRANSMITMULTIPLE REQUEST

4.5.1 The TRANSMITMULTIPLE request primitive shall be provided by calling the TRANSMIT request primitive for every MAL message.

4.5.2 If the TRANSMIT ERROR indication is called for any of the MAL messages, the TRANSMIT ERROR indications should be collected, and the TRANSMITMULTIPLE ERROR indication should be called with the content of the collected TRANSMIT ERROR indications.

4.5.2.1 The individual TRANSMIT ERROR indications shall not be transmitted to MAL.

4.5.2.2 Only the TRANSMITMULTIPLE ERROR indication shall be called.

4.6 RECEIVE INDICATION

4.6.1 The RECEIVE indication primitive shall be provided in order to receive one MAL TCP/IP PDU and translate it into a MAL message.

4.6.2 The RECEIVE indication primitive shall be called once a complete MAL TCP/IP PDU is read by the underlying TCP/IP connection.

4.6.3 In order to read a whole MAL TCP/IP PDU, the TCP 'RECEIVE' primitive shall be called one or several times.

4.6.4 The TCP 'RECEIVE' primitive shall be called on an already established TCP/IP connection.

4.6.5 If the MAL application contains an MO service provider and a TCP/IP connection with the MO service consumer is not existing, the TCP 'OPEN' primitive shall be called before attempting any invocation of the TCP 'RECEIVE' primitive, using the following rules:

a) The TCP source port assigned to the MAL application shall be passed to the TCP 'OPEN' primitive as 'Local port'.

b) The 'Foreign address' shall be set as 'unspecified' (see subsection 3.8 of reference [4]).

c) The parameter 'Active/Passive' of the TCP 'OPEN' primitive shall be set to 'Passive'.

d) If the TCP 'OPEN' primitive fails, no MAL TCP/IP PDUs shall be received using the URI composed by the IP address and TCP port assigned to the MAL application.

e) If the TCP 'OPEN' primitive successfully returns with the local connection name, MAL TCP/IP PDUs shall be read through one or more invocations of the TCP 'RECEIVE' primitive.

f) The local connection name of the connection established between two MAL applications shall be passed to the TCP 'RECEIVE' primitive as 'Local connection name'.

4.6.6 The TCP 'RECEIVE' primitive shall be called until a complete MAL TCP/IP PDU is read.

4.6.7 The MAL message header fields and body elements shall be generated according to the specifications given in section 3 of this Recommended Standard, by using the following input data:

a) the MAL TCP/IP PDU;

b) the XML specification of the MO service (see section 6 of reference [1]) identified by the MAL header fields 'Service Area', 'Service', and 'Area Version'.

4.6.8 If the MAL TCP/IP PDU header field 'Source Id Flag' holds the value '1' and if the MAL TCP/IP PDU header field 'Source Id' holds a well formed URI, then the MAL TCP/IP PDU header field 'Source Id' shall be assigned to the MAL header field 'URI From'. Otherwise the MAL header field 'URI From' shall be decoded according to the optimized mapping defined in 3.3.2.

4.6.9 If the field 'URI To' is unknown, then the error MAL::DESTINATION_UNKNOWN shall be returned if the Interaction Pattern allows a MAL error message to be returned. The MAL header field 'URI From' of the returned error message shall be assigned with the 'URI To' field of the initial message, even if this URI is unknown.

4.7 RECEIVEMULTIPLE INDICATION

The RECEIVEMULTIPLE indication primitive shall not be provided.

5 MAL DATA ENCODING

5.1 OVERVIEW

This section specifies a complete and unambiguous mapping of the MAL data types to a binary encoding format.

Encoding is a function that translates a MAL::Element into a sequence of encoded fields. This translation is defined in a generic, modular, and ‘octet aligned’ way:

a) ‘generic’ means that the encoding format is not specifically defined for every data structure and every usage context but can be used to encode every data structure and every usage context generically;

b) ‘modular’ means that the encoding format is defined by isolating every MAL data type and declaration context and specifying the encoding format for each of them;

c) ‘octet aligned’ means that every encoded field starts on an octet boundary and contains one or more octets.

The type of an element can designate either the declared type of the field the element is assigned to or the actual type of the element. In order to avoid any ambiguity the word ‘type’ is always qualified as follows:

a) declared type: the type of the field the element is assigned to;

   1) if the field belongs to the MAL message header, then the declared type is given by table 3-1;

   2) if the field belongs to the MAL message body, then the declared type and more generally the declaration context is given by the XML specification of the service (see section 6 of reference [1]);

b) actual type: the type of the element (i.e. the actual type element cannot be abstract).

MAL only specifies non-abstract types that are final, i.e., that cannot be extended. As a consequence, if the declared type of an element is non-abstract then the actual type is the same as the declared type.

NOTE – This encoding format can be reused by a MAL binding to a messaging technology that is not the TCP/IP transport binding.

The following subsections specify the rules to be applied when encoding an element. These rules depend on the element declaration context (e.g., the declared type) and the element actual type.

5.2 ELEMENT

5.2.1 If the element is not an element of a list, and if the declared type of the element is MAL::Attribute, then the following field shall be encoded:

Attribute Tag
Unsigned 8-bit Integer (1 octet)

5.2.2 The field 'Attribute Tag' shall be assigned with the short form part of the attribute minus 1 so that the tag starts from zero.

5.2.3 If the element is not an element of a list, and if the declared type of the element is either MAL::Element, or MAL::Composite, or an abstract composite; or if the declared type of the element is either List<MAL::Attribute>, or List<MAL::Element>, or List<MAL::Composite>, or List<<abstract composite>>; then the following fields shall be encoded:

<table>
<tr><th colspan="2">Area</th><th>Service</th><th>Area Version</th><th>Type</th></tr>
<tr><td></td><td></td><td colspan="2">Unsigned Varint
(see 5.25)
(variable, multiple of octet)</td><td></td></tr>
</table>

NOTE – As specified by MAL (reference [1]), only the last element of a message body can be declared either abstract or as a list of an abstract type. Therefore the encoding format specified above cannot be used more than once per message body.

5.2.4 The area number, service, area version, and type of the element shall be merged into a single Unsigned 64-bit Integer and encoded as an Unsigned Varint.

5.2.5 The area number of the element actual type shall be encoded as an Unsigned 16-bit Integer and assigned to the first 16 most significant bits of the Unsigned 64-bit Integer.

5.2.6 If the element actual type is defined by a service, then the service number of the element actual type shall be encoded as an Unsigned 16-bit Integer and assigned to the 16 most significant bits of the Unsigned 64-bit Integer that follows the encoded area number.

5.2.7 If the element actual type is not defined by a service, then the value '0' shall be encoded as an Unsigned 16-bit Integer and assigned to the 16 most significant bits of the Unsigned 64-bit Integer that follows the encoded area number.

5.2.8 The area version of the element actual type shall be encoded as an Unsigned 8-bit Integer and assigned to the 8 most significant bits of the Unsigned 64-bit Integer that follows the encoded service number.

5.2.9 The short form part of the element actual type shall be encoded as a Signed 24-bit Integer and assigned to the 24 most significant bits of the Unsigned 64-bit Integer that follows the encoded area version.

NOTE – The short form part is required to be a signed 32-bit integer by MAL (reference [1]). Negative numbers identify the list types. Therefore using a signed 24-bit integer as type field prevents the mapping of data types whose short form part is strictly greater than $2^{23}-1$.

5.2.10 The element shall be encoded as follows:

a) if the actual type of the element is a MAL::Attribute, then the element shall be encoded as specified by its actual type;

b) if the actual type of the element is a MAL::Enumeration, then the element shall be encoded as an Enumeration;

c) if the actual type of the element is a MAL::List, then the element shall be encoded as a List;

d) if the actual type of the element is a MAL::Composite, then the element shall be encoded as a Composite.

5.3 ENUMERATION

NOTE – Each element in an enumeration is assigned with two integer values: the ordinal value and the numeric value. The ordinal value is a sequential counter, starting at zero for the first element, and incremented by one in the same order as the elements of the enumeration. The numeric value is defined in reference [1].

5.3.1 The ordinal value shall be encoded.

NOTE – The length of the field used to encode the ordinal value can be set according to the maximum ordinal value, as specified by the enumeration definition.

5.3.2 If the maximum ordinal value is strictly less than 256, then the ordinal value shall be encoded as an Unsigned 8-bit Integer.

5.3.3 If the maximum ordinal value is greater than or equal to 256 and strictly less than 2^{16}, then the ordinal value shall be encoded as a UShort (see 5.16).

5.3.4 If the maximum ordinal value is greater than or equal to 2^{16} and strictly less than 2^{32}, then the ordinal value shall be encoded as a UInteger (see 5.18).

NOTE – As specified in reference [1], the MAL enumeration size is limited by the MAL::UInteger range.

5.4 COMPOSITE

5.4.1 Each field of the Composite shall be encoded in the same order as it is declared in the Composite definition.

5.4.2 Each field of the Composite shall be encoded as follows:

a) if the value of the attribute ‘canBeNull’ in the field declaration is TRUE, then the field shall be encoded as a Nullable Element;

b) otherwise the field shall be encoded as an Element.

5.4.3 If the Composite inherits from another Composite, then the inherited Composite shall be encoded first.

5.5 LIST

5.5.1 A List shall be encoded as follows:

List Length	List Element
UInteger (variable, multiple of octet)	Nullable Element (variable, multiple of octet)
	Repeated for every element in the list

5.5.2 The field ‘List Length’ shall be assigned with the length of the list encoded as a UInteger (see 5.18).

NOTE – The type List is required to be unbounded by MAL (reference [1]). Using a UInteger as length field limits the list length to $2^{32}-1$ elements.

5.5.3 The error MAL::INTERNAL shall be raised if the length of the List is strictly greater than $2^{32}–1$.

5.5.4 The list elements shall be encoded in the same order as in the list.

5.6 NULLABLE ELEMENT

5.6.1 A Nullable Element shall be encoded as follows:

a) the field ‘Presence Flag’ shall be encoded as part of the bit field located at the beginning of the encoded message (see 3.6.3.2);

b) if the element is not NULL, the field ‘Element’ shall be encoded as an Element (see 5.2), and the field ‘Presence Flag’ shall be set to TRUE;

c) if the element is NULL, the field ‘Element’ shall not be encoded, and the field ‘Presence Flag’ shall be set to FALSE.

Presence Flag	Element
Encoded in the bit field (see 3.6.3.2)	Element (variable, multiple of octet)
	If Presence Flag is TRUE

5.7 BLOB

5.7.1 A MAL::Blob shall be encoded as follows:

Blob Length	Blob Octet
UInteger (variable, multiple of octet)	Unsigned 8-bit Integer (1 octet)
	Repeated for every octet in the Blob

5.7.2 The field 'Blob length' shall be assigned with the length of the Blob encoded as a UInteger (see 5.18).

NOTE – The type Blob is required to be unbounded by MAL (reference [1]). Using a UInteger as length field limits the Blob length to $2^{32}-1$ octets.

5.7.3 The error MAL::INTERNAL shall be raised if the length of the Blob is strictly greater than $2^{32}–1$.

5.7.4 The Blob octets shall be encoded in the same order as in the Blob.

5.8 BOOLEAN

A MAL::Boolean shall be encoded as a bit of the bit field as specified in 3.6.3.2.

5.9 DURATION

A MAL::Duration shall be encoded according to the binary interchange format of width 64 bits defined in reference [5].

NOTE – Using binary interchange format of width 64 bits will result in a 15 to 17 significant decimal digits precision.

5.10 FLOAT

A MAL::Float value shall be encoded according to the binary interchange format of width 32 bits defined in reference [5].

5.11 DOUBLE

A MAL::Double shall be encoded according to the binary interchange format of width 64 bits defined in reference [5].

5.12 IDENTIFIER

A MAL::Identifier shall be encoded as a String.

5.13 OCTET

A MAL::Octet shall be encoded as a Signed 8-bit Integer.

5.14 UOCTET

A MAL::UOctet shall be encoded as an Unsigned 8-bit Integer.

5.15 SHORT

A MAL::Short shall be encoded as a 2-octet Signed Varint (see 5.26).

5.16 USHORT

A MAL::UShort shall be encoded as a 2-octet Unsigned Varint (see 5.25).

5.17 INTEGER

A MAL::Integer shall be encoded as a 4-octet Signed Varint (see 5.26).

5.18 UINTEGER

A MAL::UInteger shall be encoded as a 4-octet Unsigned Varint (see 5.25).

5.19 LONG

A MAL::Long shall be encoded as a 8-octet Signed Varint (see 5.26).

5.20 ULONG

A MAL::ULong shall be encoded as a 8-octet Unsigned Varint (see 5.25).

5.21 STRING

5.21.1 A MAL::String shall be encoded as follows:

String Length	Character
UInteger (variable, multiple of octet)	UTF-8 (variable, multiple of octet)
	Repeated for every character in the String

5.21.2 The field ‘String Length’ shall be assigned with the number of octets required to encode the characters of the string.

5.21.3 The field ‘String Length’ shall be encoded as a UInteger (see 5.18).

NOTE – The types Identifier, String, and URI are required to be unbounded by MAL (reference [1]). Using a UInteger as length field limits the encoded string length to $2^{32}-1$ octets.

5.21.4 The error MAL::INTERNAL shall be raised if the length of the String is strictly greater than $2^{32}–1$.

5.21.5 The field ‘Character’ shall be encoded according to the UTF-8 format (reference [6]).

5.21.6 The String characters shall be encoded in the same order as in the String.

5.22 TIME

5.22.1 A MAL::Time shall be encoded according to the CCSDS Time Code Format (reference [7]).

5.22.2 The Time Code Format P-Field shall be defined as CCSDS Day Segmented Time Code (CDS), no extension flag, with the epoch set as 1958 January 1, 16-bit day segment length and no submillisegment segment. The P-Field binary representation is ‘01000000’.

5.23 FINETIME

5.23.1 A MAL::FineTime shall be encoded according to the CCSDS Time Code Format (reference [7]).

5.23.2 The Time Code Format P-Field shall be defined as CCSDS Day Segmented Time Code (CDS), no extension flag, with the epoch set as 1958 January 1, 16-bit day segment length and 32-bit length of submillisecond segment. The P-Field binary representation is ‘01000010’.

5.24 URI

A MAL::URI shall be encoded as a String.

5.25 UNSIGNED VARINT

5.25.1 The length of the unsigned integer to encode shall be either 2, 4, or 8 octets.

5.25.2 The unsigned integer to encode shall be divided in groups of 7 bits as specified by table 5-1.

Table 5-1: Unsigned Integer 7-Bit Groups

		Unsigned Integer Length (octets)		
		2	**4**	**8**
Group 1	Begin	Bit N-7	Bit N-7	Bit N-7
	End	Bit N-1	Bit N-1	Bit N-1
Group 2	Begin	Bit N-14	Bit N-14	Bit N-14
	End	Bit N-8	Bit N-8	Bit N-8
Group 3	Begin	5 zeros and Bit 0	Bit N-21	Bit N-21
	End	Bit N-15	Bit N-15	Bit N-15
Group 4	Begin	-	Bit N-28	Bit N-28
	End	-	Bit N-22	Bit N-22
Group 5	Begin	-	3 zeros and Bit 0	Bit N-35
	End	-	Bit N-29	Bit N-29
Group 6	Begin	-	-	Bit N-42
	End	-	-	Bit N-36
Group 7	Begin	-	-	Bit N-49
	End	-	-	Bit N-43
Group 8	Begin	-	-	Bit N-56
	End	-	-	Bit N-50
Group 9	Begin	-	-	Bit N-63
	End	-	-	Bit N-57
Group 10	Begin	-	-	6 zeros and Bit 0
	End	-	-	Bit 0

NOTE – The 7-bit groups are ordered from the least significant group to the most significant group. The bit ordering is not changed.

5.25.3 If the most significant 7-bit group contains only zeros, and if it is not the first group, then it should be discarded; this prescription shall be repeated until either the most significant 7-bit group does not contain any zero, or it is the first group.

5.25.4 The 7-bit groups shall be encoded from the least significant group to the most significant group.

5.25.5 Every 7-bit group shall be encoded in an octet as shown below:

Continuation bit	7-bit Group
Binary value (1 bit)	Binary value (7 bits)

5.25.6 The field 'Continuation bit' shall be assigned with the value '1' if there are further 7-bit groups to come, otherwise it shall be assigned with the value '0'.

5.26 SIGNED VARINT

5.26.1 The length of the signed integer to encode shall be either 2, 4, or 8 octets.

5.26.2 The signed integer shall be translated to an unsigned integer according to the bit shifting formula specified in table 5-2 where the following symbols are used:

a) the variable 'n' is the two's complement representation of the signed integer;

b) the operator '<<' shifts the left-hand operand to the left by a number of positions given by the right-hand operand; a zero is shifted into the rightmost position;

c) the operator '>>' shifts the left-hand operand to the right by a number of positions given by the right-hand operand, and the sign bit is shifted into the leftmost position;

d) the operator '^' performs a bitwise exclusive OR operation.

Table 5-2: Signed Integer Bit Shifting

Signed Integer Length (octets)	Bit Shifting Formula
2	(n << 1) ^ (n >> 15)
4	(n << 1) ^ (n >> 31)
8	(n << 1) ^ (n >> 63)

NOTE – The bit shifting translation, also called a 'zig-zag translation', is applied in order that small negative values (-1, -2, etc.) use the same number of octets as their opposite positive values. Otherwise these small negative values would use one more octet than the integer size (e.g., 5 octets for a 32-bit integer) and only the big negative values would benefit from the size reduction.

5.26.3 The resulting unsigned integer shall be encoded as an Unsigned Varint (see 5.25).

ANNEX A

PROTOCOL IMPLEMENTATION CONFORMANCE STATEMENT PROFORMA

(NORMATIVE)

A1 INTRODUCTION

A1.1 OVERVIEW

This annex provides the Protocol Implementation Conformance Statement (PICS) Requirements List (RL) for an implementation of the Mission Operations MAL Binding to TCP/IP Transport and Split Binary Encoding standard. The PICS for an implementation is generated by completing the RL in accordance with the instructions below. An implementation claiming conformance must satisfy the mandatory requirements referenced in the RL.

An implementation's completed RL is called the PICS. The PICS states which protocol features have been implemented. The following entities can use the PICS:

– the protocol implementer, as a checklist to reduce the risk of failure to conform to the standard through oversight;

– the supplier and acquirer or potential acquirer of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;

– the user or potential user of the implementation, as a basis for initially checking the possibility of interworking with another implementation (while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICSes);

– a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A1.2 NOTATION

A1.2.1 Status Column Symbols

The following are used in the RL to indicate the status of features:

Symbol	Meaning
M	Mandatory
O	Optional

A1.2.2 Support Column Symbols

The support of every item as claimed by the implementer is stated by entering the appropriate answer (Y, N, or N/A) in the support column.

Symbol	Meaning
Y	Yes, supported by the implementation
N	No, not supported by the implementation
N/A	Not applicable

A1.3 GENERAL INFORMATION

A1.3.1 IDENTIFICATION OF PICS

Ref	Question	Response
1	Date of Statement (DD/MM/YYYY)	
2	CCSDS document number containing the PICS	
3	Date of CCSDS document containing the PICS	

A1.3.2 IDENTIFICATION OF IMPLEMENTATION UNDER TEST (IUT)

Ref	Question	Response
1	Implementation name	
2	Implementation version	
3	Machine name	
4	Machine version	
5	Operating System name	
6	Operating System version	
7	Special Configuration	
8	Other Information	

A1.3.3 USER IDENTIFICATION

Supplier	
Contact Point for Queries	
Implementation name(s) and Versions	
Other Information Necessary for full identification —e.g., name(s) and version(s) for machines and/or operating systems; System Name(s)	

A1.4 INSTRUCTIONS FOR COMPLETING THE RL

An implementer shows the extent of compliance to the protocol by completing the RL; the resulting completed RL is called a PICS.

A1.5 MO TCP/IP TRANSPORT AND SPLIT BINARY ENCODING PICS

A1.5.1 Message Abstraction Layer

Item	Protocol Feature	Reference	Status	Support
1-1	Transaction Handling	[1] subsection 3.2	M	
1-2	State Transitions	[1] subsection 3.3	M	
1-3	Message Composition	[1] subsection 3.4	M	
1-4	MAL Service Interface	[1] subsection 3.5	M	
1-5	Access Control Interface	[1] subsection 3.6	M	
1-6	Transport Interface	[1] subsection 3.7	M	
1-7	MAL Data Type Specification	[1] section 4	M	
1-8	MAL Errors	[1] section 5	M	

A1.5.2 MAL Message Mapping

Item	Protocol Feature	Reference	Status	Support
2-1	URI Format	3.2	M	
2-2	MAL Header Mapping	3.3	M	
2-3	Field ‘Timestamp’	3.3.5	O	
2-4	Fields ‘Priority’, ‘Domain’, ‘Network Zone’, ‘Session Name’	3.3.7 3.3.8 3.3.9 3.3.11	O	
2-5	Field ‘Authentication Id’	3.3.3	O	
2-6	MAL TCP/IP PDU Specific Fields	3.5	M	
2-7	MAL Message Body Mapping	3.6	M	

A1.5.3 MAL Transport Interface Mapping

Item	Protocol Feature	Reference	Status	Support
3-1	SupportedQoS Request	4.2	M	
3-2	SupportedIP Request	4.3	M	
3-3	Transmit Request	4.4	M	
3-4	TransmitMultiple Request	4.5	M	
3-5	Receive Indication	4.6	M	
3-6	ReceiveMultiple Indication	4.7	M	

A1.5.4 MAL Data Encoding

Item	Protocol Feature	Reference	Status	Support
4-1	MAL Data Encoding	section 5	M	

ANNEX B

MAPPING CONFIGURATION PARAMETERS

(NORMATIVE)

This annex defines the parameters that are provided by the MAL TCP/IP transport protocol in order to configure and optimize the MAL message mapping and the format of the MAL TCP/IP PDU transmitted over TCP/IP.

The mapping configuration parameters are managed parameters, defined by some out-of-band agreement. Those parameters can be exchanged, for example, by email, or through a common registry like the Space Assigned Numbers Authority (SANA). Ideally, those parameters would be retrieved by using the Mission Operations directory service.

Table B-1 lists the mapping configuration parameters.

Table B-1: Mapping Configuration Parameters

Parameter Name	Type	Description
AUTHENTICATION_ID	MAL:Blob	Value to be assigned to the MAL header field ‘Authentication Id’ if the QoS property AUTHENTICATION_ID_FLAG is FALSE
DOMAIN	List<MAL::Identifier>	Value to be assigned to the MAL header field ‘Domain’ if the QoS property DOMAIN_FLAG is FALSE
NETWORK_ZONE	MAL::Identifier	Value to be assigned to the MAL header field ‘Network Zone’ if NETWORK_ZONE_FLAG is FALSE
PRIORITY	MAL::UInteger	Value to be assigned to the MAL header field ‘Priority’ if PRIORITY_FLAG is FALSE
SESSION_NAME	MAL::Identifier	Value to be assigned to the MAL header field ‘Session Name’ if SESSION_NAME_FLAG is FALSE
SOURCE_ID	MAL::String	Value to be assigned to the MAL header field ‘Source Id’ if SOURCE_ID_FLAG is FALSE
DESTINATION_ID	MAL::String	Value to be assigned to the MAL header field ‘Destination Id’ if DESTINATION_ID_FLAG is FALSE

ANNEX C

QOS PROPERTIES

(NORMATIVE)

This annex defines the QoS properties that are provided by the MAL TCP/IP transport protocol. QoS properties are set on a per-message basis as specified by MAL (reference [1]).

Table C-1 lists the QoS properties.

Table C-1: QoS Properties

QoS Property Name	Type	Description
AUTHENTICATION_ID_FLAG	MAL::Boolean	Value to be assigned to the MAL TCP/IP PDU Header field ‘Authentication Id Flag’
DOMAIN_FLAG	MAL::Boolean	Value to be assigned to the MAL TCP/IP PDU Header field ‘Domain Flag’
NETWORK_ZONE_FLAG	MAL::Boolean	Value to be assigned to the MAL TCP/IP PDU Header field ‘Network Zone Flag’
PRIORITY_FLAG	MAL::Boolean	Value to be assigned to the MAL TCP/IP PDU Header field ‘Priority Flag’
SESSION_NAME_FLAG	MAL::Boolean	Value to be assigned to the MAL TCP/IP PDU Header field ‘Session Name Flag’
TIMESTAMP_FLAG	MAL::Boolean	Value to be assigned to the MAL TCP/IP PDU Header field ‘Timestamp Flag’
SOURCE_ID_FLAG	MAL::Boolean	Value to be assigned to the MAL TCP/IP PDU Header field ‘Source Id Flag’
DESTINATION_ID_FLAG	MAL::Boolean	Value to be assigned to the MAL TCP/IP PDU Header field ‘Destination Id Flag’

ANNEX D

SECURITY, SANA, AND PATENT CONSIDERATIONS

(INFORMATIVE)

D1 SECURITY CONSIDERATIONS

D1.1 OVERVIEW

This annex subsection discusses various aspects of security with respect to the MAL TCP/IP transport protocol.

D1.2 SECURITY BACKGROUND

The following security aspects are typically separated:

a) data and data origin authentication: corroboration of the source of information that is contained in a message;

b) authorization: conveyance, to another entity, of official sanction to do or be something;

c) confidentiality: keeping information secret from all but those who are authorized to see it;

d) integrity: detecting that information has not been altered by unauthorized or unknown means.

The MAL TCP/IP transport protocol is not responsible for ensuring all these security aspects; however, it has to fulfil the security criteria expected by the MAL layer from every transport binding. These criteria are:

a) the Transport Layer is responsible for the transmission of the authentication identifier assigned by the MAL layer to every consumer;

b) the Transport Layer has to provide authentication, confidentiality, and integrity of the transmitted messages.

D1.3 SECURITY CONCERNS WITH RESPECT TO THE CCSDS DOCUMENT

D1.3.1 Data Privacy

It is assumed that message authentication and confidentiality are provided beneath the TCP layer and are transparent to the TCP/IP transport binding and above. As a consequence, once a message rises above the TCP protocol layer, the message has been authenticated and all encryption has been removed.

D1.3.2 Data Integrity

Integrity is ensured by the protocol that conveys the TCP segments, in addition to the checksum functions specified by TCP and IP (see references [2], [3], and [4]).

D1.3.3 Authentication of Communicating Entities

Authentication of the consumers is done above the MAL layer through a specific service that enables a consumer to get an authentication identifier. The meaning of that authentication identifier is dependent on the security system used for the deployment. This identifier must allow the MAL access control implementation to perform a lookup for authorization purposes.

The authentication identifier is transmitted by the MAL TCP/IP transport protocol in the parameter 'Authentication Id' of the MAL TCP/IP PDU header; however, this parameter may be omitted as it is optional.

The MAL authentication identifier is an implementation- and technology-specific security credential created at a higher layer by MAL access control. The TCP/IP transport protocol allows this implementation specific security credential to be transferred from TCP/IP source to destination. It is possible for this security credential to be used by the protocol below TCP for authentication or even confidentiality purposes (e.g., IPsec), but that is not specified here.

D1.3.4 Control of Access to Resources

Authorization is done by the MAL access control that performs any required authorization checks and converts the consumer identifier into technology-dependent security credentials.

D1.4 POTENTIAL THREATS AND ATTACK SCENARIOS

Potential threats and attack scenarios depend on the layer that is beneath the MAL TCP/IP transport protocol because this is the layer that defines the security algorithms ensuring authentication, confidentiality, and integrity.

D1.5 CONSEQUENCES OF NOT APPLYING SECURITY TO THE TECHNOLOGY

The only security aspect that may not be applied is the transmission of the authentication identifier in the TCP/IP MAL header. If the authentication identifier is not transmitted by the MAL TCP/IP transport protocol, then delivered messages may be rejected by the MAL access control.

D2 SANA CONSIDERATIONS

D2.1 VERSION NUMBER

The recommendations of this document request SANA to create the registry defined as follows:

a) the registry named ‘MAL TCP/IP Transport Version Number’ consists of a table of parameters:

   1) Version Number: a string of text specifying the three bits to be assigned to the MAL TCP/IP PDU Header field ‘Version Number’;

   2) Reference: a string of text referencing the CCSDS document that specifies the version of the MAL TCP/IP transport;

b) the initial registry should be filled with the values in table D-1.

Table D-1: MAL TCP/IP Transport Version Number Initial Values

Version Number	Reference
000	CCSDS 524.2-B-1

D2.2 URI SCHEME NAME

The recommendations of this document request SANA to create the registry defined as follows:

a) the registry named ‘MAL TCP/IP Transport URI Scheme Name’ consists of a table of parameters:

   1) Scheme Name: a string of text specifying the name of the URI scheme defined by the MAL binding;

   2) Reference: a string of text referencing the CCSDS document that specifies the MAL binding;

b) the initial registry should be filled with the values in table D-2.

Table D-2: MAL TCP/IP Transport Binding URI Scheme Name Initial Values

Scheme Name	Reference
maltcp	CCSDS 524.2-B-1

D2.3 MAL ENCODING IDS

The recommendations of this document request SANA to create the registry defined as follows:

a) The registry named ‘MAL Encoding Ids’ consists of a table of parameters:

   1) Encoding Id: an unsigned integer between 0 and 255 specifying the encoding identifier to be used by the Encoding Id field available on MAL Bindings; values greater than 127 are to be reserved for non-standard encodings.

   2) Encoding: a string of text describing the encoding name.

   3) Reference: a string of text referencing the CCSDS document that specifies the book and respective chapter (if applicable) where the Encoding is defined.

b) The initial registry should be filled with the values in table D-3.

Table D-3: MAL Encoding IDs

Encoding Id	Encoding	Reference
0	Fixed Binary	CCSDS 524.1-B-1 – Chapter 5
1	Variable Length Binary	CCSDS 524.1-B-1 – Chapter 5
2	Split Binary	CCSDS 524.2-B-1 – Chapter 5

D3 PATENT CONSIDERATIONS

No patents are known to apply to this Recommended Standard.

ANNEX E

ENCODING EFFICIENCY

(INFORMATIVE)

E1 INTRODUCTION

This subsection lists the potential overhead costs caused by the encoding format.

E2 BANDWIDTH OVERHEAD

E2.1 MAL TCP HEADER

The overhead caused by the secondary header depends on whether the optional fields are inserted or not in the MAL TCP/IP PDU header: ‘Priority’, ‘Timestamp’, ‘Network Zone’, ‘Session Name’, ‘Domain’, ‘Authentication Id’, ‘Source Id’ and ‘Destination Id’.

The minimum overhead is 23 octets. It is obtained with all optional MAL header fields not passed (presence flags set to FALSE) and with the ‘Source Id’ and ‘Destination Id’ header fields not passed (presence flags set to FALSE). The additional overheads are given by table E-1 in terms of:

a) the name of a MAL TCP/IP PDU header field;

b) the encoding format of the field (as specified by section 5).

Table E-1: MAL TCP/IP PDU Header Additional Overheads

Field Name	Encoding Format
Source Id	String
Destination Id	String
Priority	UInteger
Timestamp	Time
Network Zone	Identifier
Session Name	Identifier
Domain	List<Identifier>
Authentication Id	Blob

E2.2 PRESENCE FLAG

An overhead of one bit is added for every field that can be NULL in a composite data structure. Depending on the location of the nullable field in the message, the bit might be omitted (see 3.6.3.2.13).

An overhead up to seven bits might be added, to pad the bit field in order to reach an octet boundary.

E2.3 POLYMORPHISM

Type information is only encoded where absolutely necessary, i.e., when polymorphism is employed. Otherwise, type information is directly taken from the service specification.

The last element of a MAL body may be abstract. In this case, a type information needs to be added depending on the last element type. If the last element is typed MAL::Element, MAL::Composite, or an abstract composite, then the overhead is 8 octets. If the last element is MAL::Attribute, then the overhead is 1 octet. This overhead is added at most once per MAL message body.

An overhead of 1 octet is also added for every field typed MAL::Attribute declared in a composite structure.

E2.4 EFFORT REDUCTION

Missions benefit from the generic nature of the encoding format. Mission-specific solutions of mapping MAL messages to TCP/IP-delivered messages are avoided, thus allowing easier cross support, reuse of components for several missions, and less personnel training. The effort required by the definition of a specific encoding format is reduced to zero. There is no dependency on the spacecraft database anymore.

Besides, a mission has still the ability to define compact data structures dedicated to its needs and to use its preferred encoding format for the MAL message body.

ANNEX F

ACRONYMS

(INFORMATIVE)

This annex lists the acronyms used in this Recommended Standard.

API	application programming interface
CCSDS	Consultative Committee for Space Data Systems
CUC	CCSDS Unsegmented Time Code
IP	Internet Protocol or interaction pattern
LSB	least significant bit
MAL	Message Abstraction Layer
MO	mission operations
MSB	most significant bit
PICS	Protocol Implementation Conformance Statement
PDU	PROTOCOL DATA UNIT
QoS	QUALITY OF SERVICE
RL	REQUIREMENTS LIST
SANA	Space Assigned Number Authority
SDU	service data unit
SM&C	CCSDS Spacecraft Monitoring and Control
TAI	International Atomic Time
TCP	Transmission Control Protocol
URI	Universal Resource Identifier
UTC	Universal Coordinated Time

ANNEX G

INFORMATIVE REFERENCES

(INFORMATIVE)

[G1] *Mission Operations Services Concept*. Issue 3. Report Concerning Space Data System Standards (Green Book), CCSDS 520.0-G-3. Washington, D.C.: CCSDS, December 2010.