INTERNATIONAL STANDARD

ISO

21082

First edition
2016-07-01

Mission operations — MAL space packet transport binding and binary encoding

Opérations de mission - transport de paquets de l'espace MAL reliés et codage binaire

COPYRIGHT PROTECTED DOCUMENT

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2. www.iso.org/directives

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received. www.iso.org/patents

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

ISO 21082 was prepared by the Consultative Committee for Space Data Systems (CCSDS) (as CCSDS 524.1-B-1, August 2015) and was adopted (without modifications except those stated in clause 2 of this International Standard) by Technical Committee ISO/TC 20, *Aircraft and space vehicles*, Subcommittee SC 13, *Space data and information transfer systems*.

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

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.

CCSDS RECOMMENDED STANDARD FOR MISSION OPERATIONS—MAL SPACE PACKET
TRANSPORT BINDING AND BINARY ENCODING

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

<u>Member Agencies</u>

– 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.

<u>Observer Agencies</u>

– 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.
– KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
– Korea Aerospace Research Institute (KARI)/Korea.
– Ministry of Communications (MOC)/Israel.
– 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.
– 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 524.1-B-1	Mission Operations—MAL Space Packet Transport Binding and Binary Encoding, Recommended Standard, Issue 1	August 2015	Original issue

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

1 INTRODUCTION

1.1 PURPOSE

This Recommended Standard defines the binding between the Mission Operations (MO) Message Abstraction Layer (MAL) specified in reference [2] and the Space Packet Protocol specified in reference [1]. This binding allows MO Services to use the Space Packet Protocol as messaging technology in all situations where this may be required.

1.2 SCOPE

The scope of this Recommended Standard is the specification of the binding in terms of technology mapping to the Space Packet Protocol of:

a) MAL message;

b) MAL Transport Interface.

The MAL Blue Book (reference [2]) specifies the MAL protocol in an abstract way, i.e., without defining the concrete protocol data units. The MAL Space Packet Transport Binding and Binary Encoding specifies:

a) a complete and unambiguous mapping of the MAL message to the Space Packet;

b) a complete and unambiguous mapping of the MAL transport interface to the Space Packet Protocol interface;

c) a complete and unambiguous mapping of the MAL data types to fixed and variable length binary encoding formats.

This Recommended Standard does not specify:

a) individual implementations or products;

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

MO services defined in terms of MAL using the Space Packet Transport Binding as defined in this Recommended Standard are fully interoperable.

1.3 APPLICABILITY

This Recommended Standard specifies a technology mapping that enables different implementations of the MO service framework (see 2.2) to interoperate through the Space Packet Protocol.

1.4   RATIONALE

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, namely the Space Packet Protocol.

This Recommended Standard defines a binary encoding format for the MAL data types that may be re-used by a MAL binding to a messaging technology that is not the Space Packet Protocol.

1.5   DOCUMENT STRUCTURE

This Recommended Standard is organized as follows:

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

   b) section 2 presents an overview of the MAL Space Packet Transport Binding in relation with the MO service framework;

   c) section 3 specifies the mapping of the MAL message to the Space Packet;

   d) section 4 specifies the mapping of the MAL transport interface to the Space Packet Protocol interface;

   e) section 5 specifies a generic encoding format for the MAL data types.

1.6   DEFINITIONS

The following definitions are from the MO Reference Model (reference [8]).

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 protocol data units 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.

Protocol 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

1.7.1 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.

1.7.2 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.

1.8 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'.

BIT 0   BIT N–1

N-BIT DATA FIELD

FIRST BIT TRANSFERRED = MSB
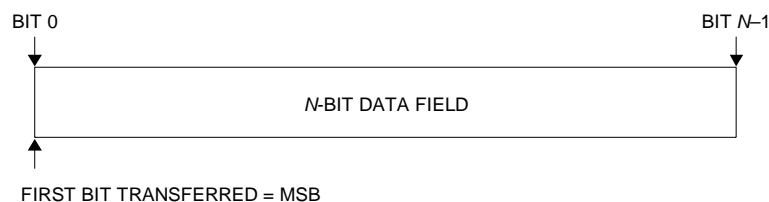
Figure 1-1: Bit Numbering Convention

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 is used to describe this grouping:

8-BIT WORD = 'OCTET'
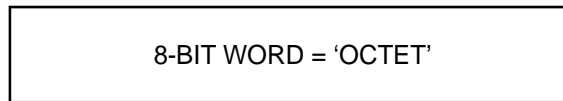
Figure 1-2: Octet Convention

By CCSDS convention, all 'spare' or 'unused' bits shall be permanently set to value 'zero'.

1.9   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 document 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.

[1]   *Space Packet Protocol*. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 133.0-B-1. Washington, D.C.: CCSDS, September 2003.

[2]   *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.

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

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

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

[6]   *Data Elements and Interchange Formats—Information Interchange—Representation of Dates and Times*. 3rd ed. International Standard, ISO 8601:2004. Geneva: ISO, 2004.

[7]   "Packet Version Number." Space Assigned Numbers Authority. http://sanaregistry.org/r/packet_version_number/.

[8]   *Mission Operations Reference Model*. Issue 1. Recommendation for Space Data System Practices (Magenta Book), CCSDS 520.1-M-1. Washington, D.C.: CCSDS, July 2010.

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 the Space Packet Protocol for exchanging messages. This is of particular interest for MO services that are deployed across the space-ground link, for instance when the MO service provider is located onboard a spacecraft and the consumer on the ground. The messages that the provider and consumer exchange to implement the MO services are encoded in Space Packets and carried via the Space Packet Protocol, which acts as a Message Layer mapping. This can run directly over a space Data Link Layer or over a true Transport Layer such as TCP, UDP, or BP.

To achieve this, this Recommended Standard provides a technology mapping of the MAL transport interface, the MAL abstract message, and the MAL data types specification (reference [2]) to the Space Packet Protocol (reference [1]). This technology mapping requires the specification of a concrete encoding format that is compliant with the Space Packet Protocol and that maps MAL header fields to SPP primary and secondary header fields.

The MAL Blue Book (reference [2]) defines an abstract transport interface as a set of request and indication primitives. A technology mapping specifies how these primitives are provided according to the messaging technology.

In particular, a technology mapping translates the MAL message model into one or several specific PDUs in compliance with the protocol used by the messaging technology. Moreover, a technology mapping casts the MAL data types into a specific encoding format. In this binding specification a binary encoding is used. Other encodings, such as XML, are also possible.

Full interoperability of services is achieved only once a binding has been created to a specific messaging technology. Because all MO service specifications are defined in terms of MAL using a formal service description language, a technology mapping must be used to bind the abstract service specifications to a specific messaging technology. This document specifies the mapping of the MAL abstractions to the CCSDS SPP.

The diagram shown in figure 2-1 presents the set of standards documentation in support of the Mission Operations Services Concept. This book belongs to the technology mappings documentation.

CCSDS RECOMMENDED STANDARD FOR MISSION OPERATIONS—MAL SPACE PACKET
TRANSPORT BINDING AND BINARY ENCODING
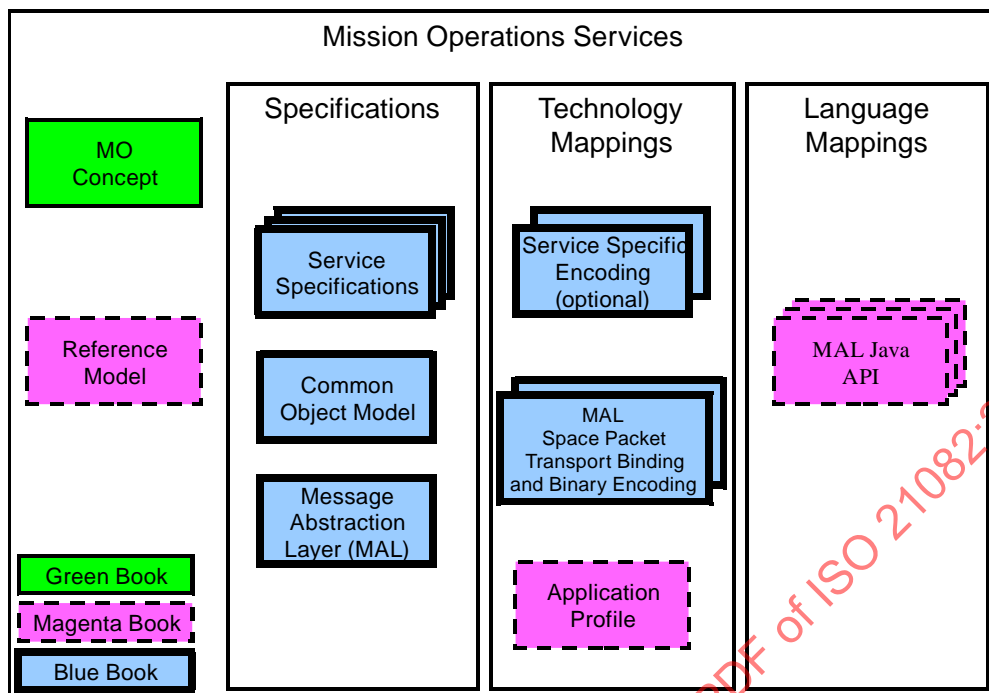
Figure 2-1: Mission Operations Services Concept Document Set

2.2   MO SERVICE FRAMEWORK OVER SPACE PACKET PROTOCOL

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 framework of common services providing functionality common to most uses of the service framework. An overview of this framework is presented in figure 2-2.
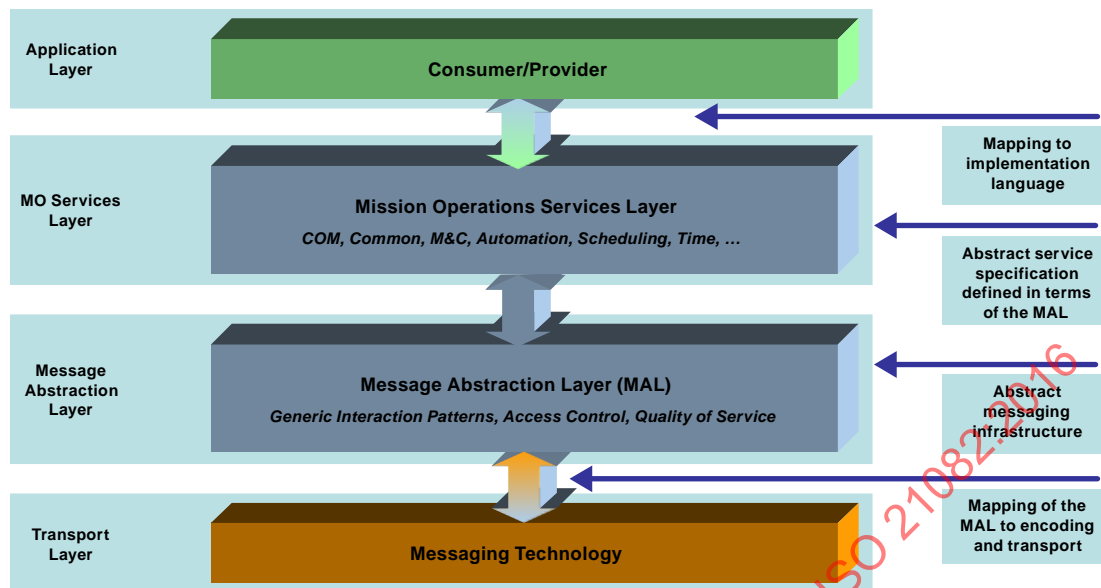
Figure 2-2: Overview of the MO Service Framework

This Recommended Standard defines how the MAL layer is mapped to the specific transport layer messaging technology called Space Packet Protocol. More specifically, it 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 rules defined in this document in section 5).

The MAL Blue Book (reference [2]) groups all the interfaces to the Transport Layer in a single specification called the MAL transport interface (subsection 3.7 of reference [2]).

CCSDS RECOMMENDED STANDARD FOR MISSION OPERATIONS—MAL SPACE PACKET
TRANSPORT BINDING AND BINARY ENCODING

Thanks to this, only the MAL transport interface needs to be mapped to the Space Packet 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 Space Packet Transport Binding layer in the MO service framework stack and highlighting the various interfaces and their main primitives.

Figure 2-3 shows that the mapping of the MAL transport interface to the Space Packet Protocol layer requires the insertion of a layer in between. This layer is called the MAL Space Packet Transport Binding. It is responsible for the translation of the abstract MAL message to the concrete Space Packet.

The protocol stack represented in figure 2-3 is conceptual. It can be implemented in various ways. For example, an onboard implementation of the stack may, for performance reasons, merge the MAL layer and the MAL Space Packet Transport Binding layer into a single layer called 'MAL over Space Packet Protocol'.

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 a Space Packet.
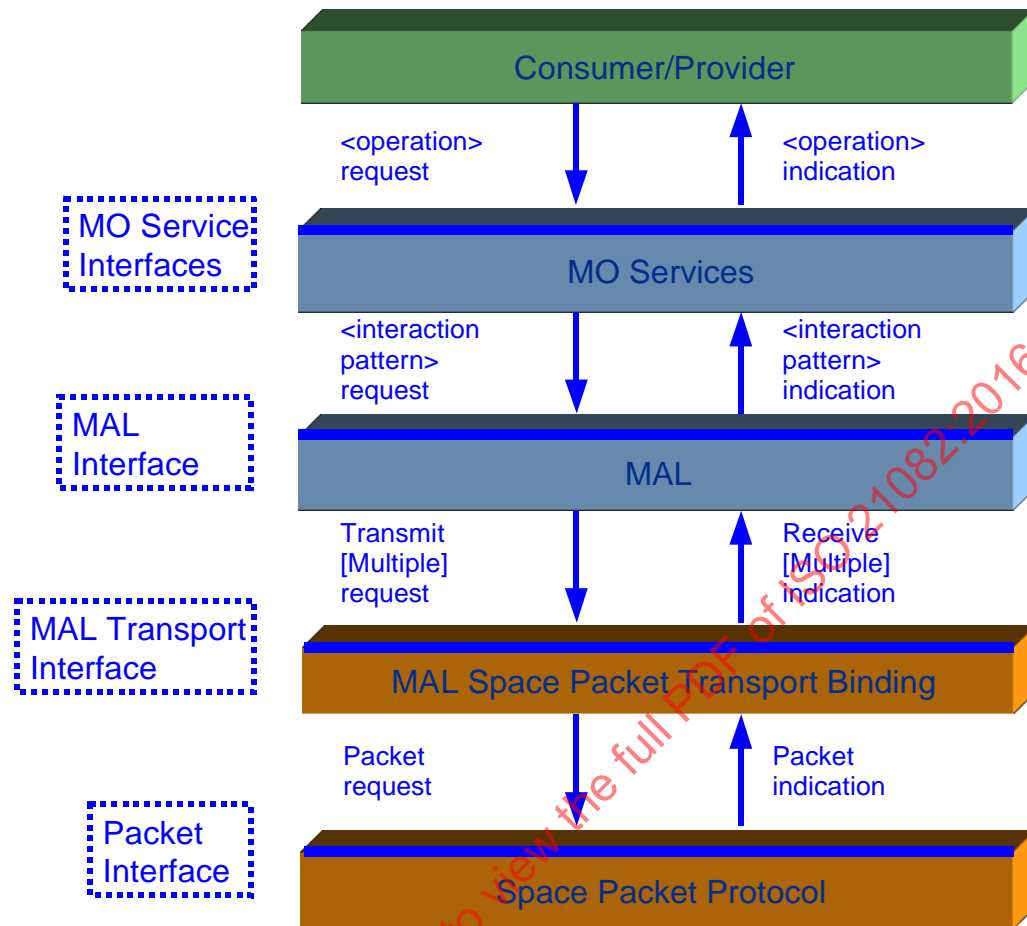
Figure 2-3: MO Service Framework above Space Packet Protocol

2.3 TYPICAL USE

Possible uses of the MAL to SPP binding may be between MO entities operating over a space link, for example:

a) ground and onboard applications interacting over a space link using Space Packets;

b) onboard components identified with distinct Application Process Identifiers and interacting with each other;

c) applications deployed in separate spacecraft and communicating using Space Packets carried over the Internet Protocols.

A typical deployment is illustrated in figure 2-4. In this example, the MO service framework is only used by the end nodes: the ground end node (e.g., in a control centre) and the onboard end node (e.g., in a spacecraft). The intermediary nodes are not represented in this figure so both end nodes are shown as though directly connected through the space-to-ground link. In real deployments the ground end node would use SLE services to connect to an Earth Space Link Terminal that would provide these space link services.

CCSDS RECOMMENDED STANDARD FOR MISSION OPERATIONS—MAL SPACE PACKET
TRANSPORT BINDING AND BINARY ENCODING

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.

A component may implement several layers of the abstract stack. For example, on the onboard end node, the MAL and Space Packet Transport Binding Layers are shown as being implemented by a single component called MAL/SPP.

The first concrete PDU is produced at the Binding level as the result of the mapping of the MAL message to the Space Packet Protocol. This mapping is explained in section 2.4 and explained in more detail in figure 2-5.

The lower protocol layers between the Space Packet Protocol and the space link are not represented.
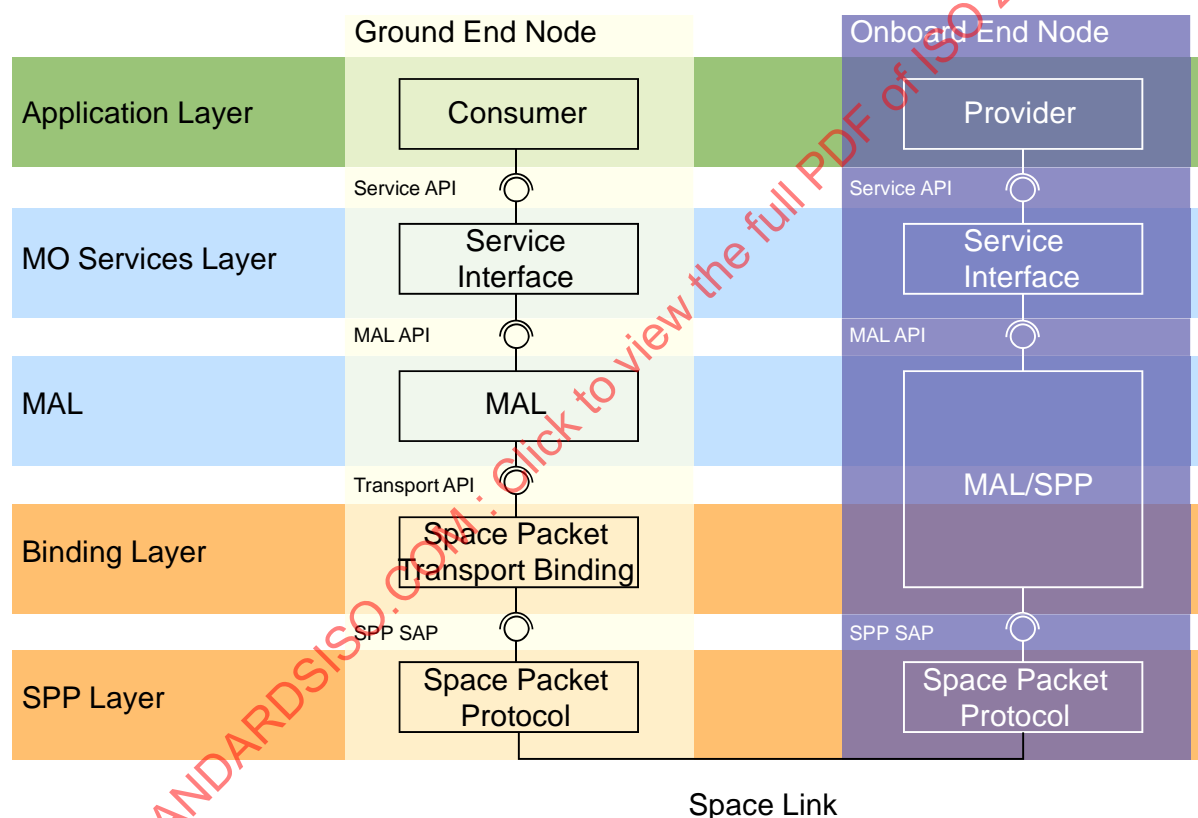
Figure 2-4: Typical Deployment of the MAL Space Packet Transport Binding

2.4 MAL MESSAGE MAPPING

2.4.1 MAPPING TO THE SPACE PACKET PROTOCOL

Each field of the MAL message needs to map to a field of the Space Packet, either in the primary header, the secondary header, or the user data field:

a) the Space Packet primary header is specified by the Space Packet Protocol (reference [1]);

b) the Space Packet secondary header and the user data field are defined and constrained by reference [1] and specified by this Recommended Standard.

This Recommended Standard uses the Space Packet primary header field 'APID' as an identifier of the sending or receiving application process that can be located either onboard or on the ground. In order to stay compliant with existing systems, the APID of a TC packet identifies the destination application, whereas the APID of a TM packet identifies the source application. Furthermore, forwarding of data along the logical data path is managed by the intermediate entities; there is no routing protocol.

The mapping to the primary header field 'Packet Type' is left as a configuration matter for the missions. The value assigned to the 'Packet Type' does not result from the mapping of any field of the MAL message. The Packet Type shall be used to distinguish Packets used for telemetry from Packets used for telecommand.

Figure 2-5 illustrates the mapping of the MAL message to the Space Packet. All MAL message fields are mapped to the Space Packet fields. 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) in a TC packet, the MAL header field 'URI From' is mapped to the secondary header fields 'Secondary APID', 'Secondary APID Qualifier', and 'Source Id'; the background colour is yellow;

b) in a TC packet, the MAL header field 'URI To' is mapped to the primary header field 'Application Process Identifier' and the secondary header field 'Destination Id'; the background colour is yellow;

c) in a TM packet, the MAL header field 'URI From' is mapped to the primary header field 'Application Process Identifier' and the secondary header field 'Source Id'; the background colour is yellow;

d) in a TM packet, the MAL header field 'URI To' is mapped to the secondary header fields 'Secondary APID', 'Secondary APID Qualifier', and 'Destination Id'; the background colour is yellow;

e) the MAL header fields 'Interaction Type' and 'Interaction Stage' are mapped to the secondary field 'SDU Type'; the background colour is orange.

The secondary header fields 'Source Id' and 'Destination Id' can be left out according to the secondary header fields 'Source Id Flag' and 'Destination Id Flag'.

Some MAL header fields are optional. Their presence in the secondary header is specified by the QoS properties defined in annex C. The optional fields are mapped to two fields in the secondary header: a presence flag that indicates whether the value is encoded in the secondary header or not, and a field that gives the value in case it is encoded. The background colour is green.

The MAL header fields cannot be NULL, especially the field 'Transaction Id', even in MAL messages whose interaction type is SEND.

Space Packet fields whose values do not result from the mapping of a MAL message field have a white background.

The secondary header field 'Segment Counter' is optional and depends upon the primary header 'Sequence Flags'. The field 'Segment Counter' is included if the Space Packet user data is segmented.

Finally, the MAL message body field and its equivalent Space Packet user data field have a grey background. If the MAL message body is larger than the maximum SPP user data field , then it must be segmented into several contiguous Space Packets.

The generated Space Packets resulting from the mapping of a MAL message need to be delivered to the right MAL recipient application, which is identified by the MAL header field 'URI To'. Space Packets whose packet type is TC are delivered according to the primary header field 'APID' and the parameter 'APID Qualifier' passed to the 'Packet request' primitive. TM packets are delivered according to the secondary header fields 'Secondary APID' and 'Secondary APID Qualifier'.

Both MAL and SPP protocol layers rely on an error control mechanism (i.e., a CRC field) that is outside the scope of this Recommended Standard.

The reasons why the MAL header fields are reordered when mapped to the Space Packet are given in 3.1.
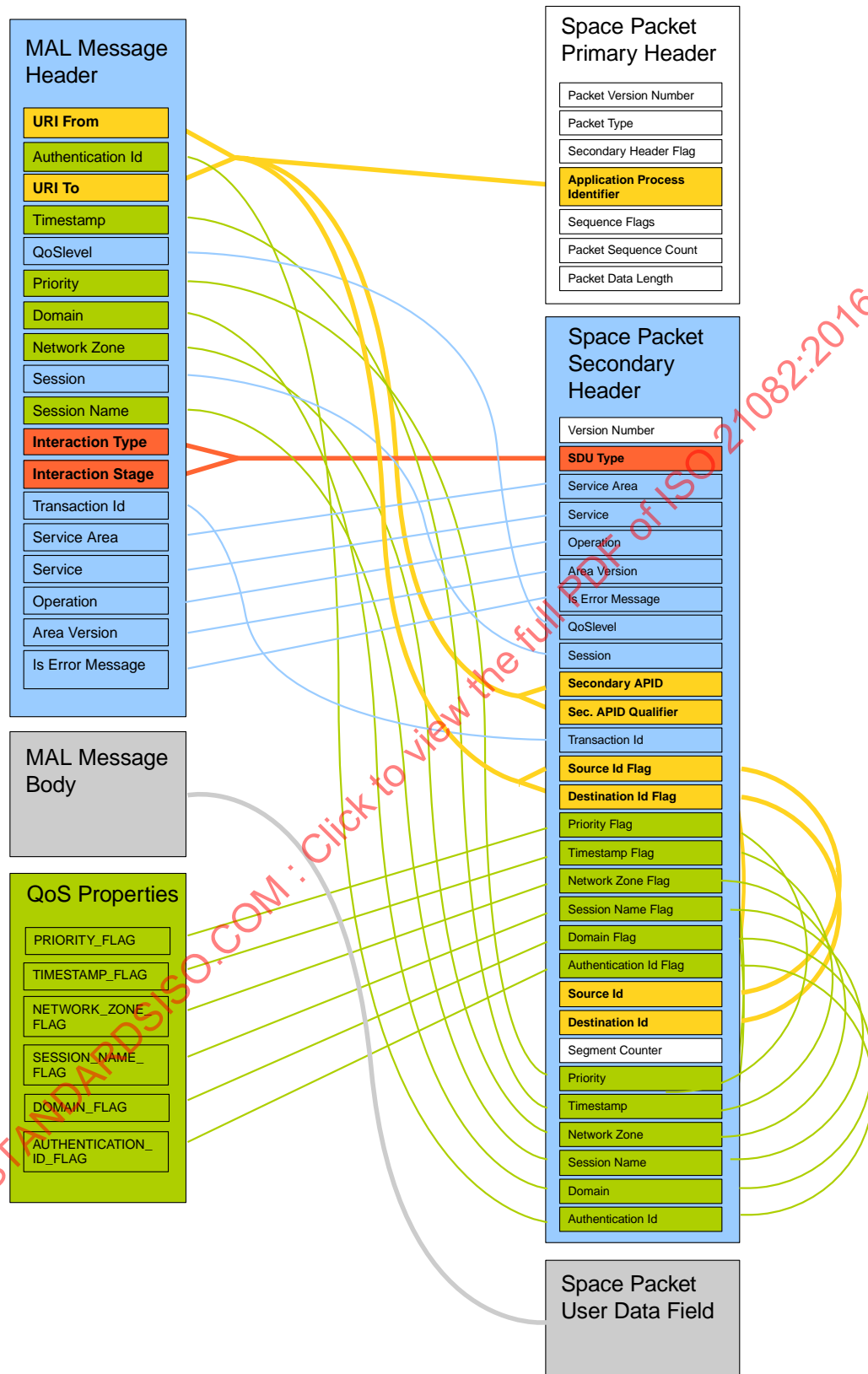
Figure 2-5: MAL Message Mapping to Space Packet

2.4.2 MAPPING CONFIGURATION PARAMETERS

This Recommended Standard defines parameters that allow to configure and optimize the MAL message mapping and the format of the Space Packet. Those parameters are called Mapping Configuration Parameters (MCPs). 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 space packet and interpreted at decoding time. The following encoding configuration options can be customized:

a) time code formats;

b) Packet Data Field size limit (in octets);

c) usage of the 'Varint' encoding format (supported or not);

d) default values of MAL header fields that are not encoded in the space packet, but that are assigned at decoding time to the resulting MAL header.

The MCPs are vital for interoperability. In case of misunderstanding related to these parameters, the space packet decoding would either fail with an error, or worse, would silently return a wrong result.

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.

2.4.3 MAPPING SPECIFICATION

The MAL Space Packet Transport Binding 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 Space Packet primary header, secondary header, and user data field. 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; the length of the format is put in brackets, either in bits or octets; the length can be variable; if the value to encode is directly given in binary format, then the encoding format is called 'Binary value';

c) 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 and APID, 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 format allows encoding of an integer using a number of octets that depends on the integer value. Each encoded octet begins with a continuation bit (Bit '0') indicating 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 4-octet integer can be encoded with 1, 2, 3, 4, or 5 octets. The signed and unsigned Varint formats are specified in 5.27 and 5.28.

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 Space Packet 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 Space Packet Protocol request by sending a Space Packet, and returning a reply;

b) a Space Packet Protocol indication initiating a MAL transport indication when receiving a Space Packet;

c) a MAL transport request returning a reply without calling the Space Packet Protocol 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 Space Packet.

Table 3-1 is taken from reference [2] 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 Space Packet primary and secondary headers.

The primary header is specified by the Space Packet Protocol (reference [1]). Table 3-2 expands the primary header specification with the encoding format of each field, as defined in section 5, and the static values to be assigned when using this Recommended Standard.

Table 3-2: Space Packet Primary Header Format

Packet Version Number	Packet Type	Secondary Header Flag	Application Process Identifier	Sequence Flags	Packet Sequence Count	Packet Data Length
Binary value (3 bits)	Binary value (1 bit)	Binary value (1 bit)	Unsigned 11-bit integer (11 bits)	Binary value (2 bits)	Unsigned 14-bit integer (14 bits)	Unsigned 16-bit integer (16 bits)
Always equal to '000'		Always equal to '1'				

The Space Packet secondary header is defined for MAL purposes by selecting option b) of subsection 4.1.3.2.1.5 of reference [1], the Ancillary Data Field being comprised of the fields 'Version Number', 'SDU Type', 'Service Area', 'Service', 'Operation', 'Area version', 'Is Error Message', 'QoSlevel', 'Session', 'Secondary APID', 'Secondary APID Qualifier', 'Transaction Id', 'Source Id Flag', 'Destination Id Flag', 'Priority Flag', 'Timestamp Flag', 'Network Zone Flag', 'Session Name Flag', 'Domain Flag', 'Authentication Id Flag', 'Source Id', 'Destination Id', 'Segment Counter', 'Priority', 'Timestamp', 'Network Zone', 'Session Name', 'Domain', and 'Authentication Id'.

The ordering of the secondary header fields is justified as follows:

a) the field 'Version Number' needs to come first in the Ancillary Data Field because it identifies the secondary header and the user data field 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 secondary header field 'Version Number';

c) the fields 'Secondary APID', 'Secondary APID Qualifier' identify the interacting peers; this information should come after the fields that identify the interaction (see above);

d) the fields 'QoSlevel' and 'Session' are inserted between the field 'Is Error Message' and the field 'Secondary APID' in order to reach an octet boundary $(1 + 2 + 2 + 11 = 16$ bits);

e) the field 'Transaction Id' refines the previous information about the interaction and the peers identification: it identifies the current interaction occurrence;

f) a flag is added to the secondary header for the optional source and destination identifiers, and for every MAL header field that is optional;

g) 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 APID and APID Qualifier;

h) the field 'Segment Counter' is used in case the MAL message is segmented in several Space Packets;

i) the field 'Priority' could be used as soon as the MAL message is delivered;

j) the fields 'Network Zone', 'Session Name', 'Domain', and 'Authentication Id' are relegated to the back of the secondary header in order to enable not to decode them in case the packet decoding is allowed to be interrupted; for example, in an onboard environment, only the fixed-length fields could be decoded (from 'Version' to 'Timestamp'); the variable-length fields (from 'Network Zone' to 'Authentication Id') and the MAL message body would be ignored.

The format of the packet secondary header is shown in table 3-3.

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 format 'Varint' is delineated by continuation bits as specified in 5.27;

b) the formats 'Blob', 'Identifier', and 'List' contain a length field as specified respectively in 5.7, 5.12, and 5.5.

The QoS properties defined by annex C allow to set the values of secondary 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 Space Packet;

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

d) the mapping of the MAL message body to the Space Packet.

Table 3-3: Space Packet Secondary Header Format

Version Number	SDU Type	Service Area	Service	Operation	Area Version	Is Error Message	QoSlevel	Session	Secondary APID	Secondary APID Qualifier	Transaction Id	Source Id Flag	Desti-nation Id Flag	Priority Flag	Time-stamp 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 2-bit Integer (2 bits)	Unsigned 2-bit Integer (2 bits)	Unsigned 11-bit Integer (11 bits)	Unsigned 16-bit Integer (16 bits)	Signed 64-bit Integer (64 bits)	Binary value (1 bit)	Binary value (1 bit)	Binary value (1 bit)	Binary value (1 bit)

Network Zone Flag	Session Name Flag	Domain Flag	Authen-tication Id Flag	Source Id	Destination Id	Segment Counter	Priority	Timestamp	Network Zone	Session Name	Domain	Authentication Id
Binary value (1 bit)	Binary value (1 bit)	Binary value (1 bit)	Binary value (1 bit)	Unsigned 8-bit Integer (8 bits)	Unsigned 8-bit Integer (8 bits)	Unsigned 32-bit Integer (32 bits)	UInteger (see 5.18) (var., mult. of octet)	Time (see 5.22) (var., mult. of octet)	Identifier (see 5.12) (var., mult. of octet)	Identifier (see 5.12) (var., mult. of octet)	List<Identifier> (see 5.5) (var., mult. of octet)	Blob (see 5.7) (var., mult. of octet)
				If 'Source Id Flag' is '1'	If 'Destination Id Flag' is '1'	If 'Sequence Flags' is either '00', '01', or '10'	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 SPP. The mapping is specified in 3.3.2 and 3.3.4.

   a)   the URI scheme name shall be 'malspp';

   b)   the scheme name shall be followed by a colon separator ':' and a rootless path, i.e., a
        path that does not begin with a slash '/';

   c)   the path shall begin with the APID qualifier, an integer represented in decimal;

   d)   the APID qualifier shall be a positive integer, including zero, strictly less than 2^{16};

   e)   the APID qualifier shall be followed by a slash separator '/' and the APID, an integer
        represented in decimal;

   f)   the APID shall be a positive integer, including zero, strictly less than 2047; the APID
        shall be compliant with the APID values listed in the SANA registry called 'Space
        Packet Protocol Application Process Identifier (APID)';

   NOTE –   The APID value of 2047, indicating the Idle Packet, is not allowed because it
            cannot be used to map a MAL message.

   g)   the APID may be followed by a slash separator '/' and an integer represented in
        decimal which is called the 'Source Id' for the field 'URI From' and the 'Destination
        Id' for the field 'URI To';

   h)   the source or destination identifier shall be a positive integer, including zero, strictly
        less than 256.

NOTES

1   The range of the APID qualifier (16 bits) should be long enough for most use cases.
        According to reference [1], if Space Packets are transferred over a space-to-ground or
        space-to-space communications link with one of the Space Data Link Protocols
        (references [G1]-[G2]), the APID qualifier is assigned with the Master Channel
        Identifier (MCID) which is the concatenation of the Transfer Frame Version Number
        (TFVN, 2 bits) and the Spacecraft Identifier (SCID, 10 bits).

2   The source or destination identifier is optional.

3   An example of URI is 'malspp:417/0/2'. This URI references the source or
        destination '2' provided by the APID '0' for the APID qualifier '417'.

3.2.2   The APID and APID qualifier shall uniquely identify a MAL application.

3.2.3   The source or destination identifier shall be unique for a given MAL application identified by its APID and APID qualifier.

NOTES

1   A single onboard application that is identified by a single APID and a single APID qualifier may represent several MAL entities. In order to uniquely address a single MAL entity, source and destination identifiers are used to refine the APID and APID qualifier.

2   The use of source and destination identifiers should be avoided because of the complexity of processing.

3.2.4   The scheme name 'malspp' shall be added to the SANA registry 'MAL Transport Binding URI Scheme Name' and shall refer to the Mission Operations—MAL Space Packet Transport Binding and Binary Encoding document 'CCSDS 524.1-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 Space Packet.

The mapping configuration parameters are defined in annex B.

3.3.2   URI FROM

3.3.2.1   If the MAL message is mapped to a TC packet, then the APID of the MAL header field 'URI From' shall be assigned to the secondary header field 'Secondary APID'.

3.3.2.2   If the MAL message is mapped to a TC packet, then the APID qualifier of the MAL header field 'URI From' shall be assigned to the secondary header field 'Secondary APID Qualifier'.

3.3.2.3   If the MAL message is mapped to a TM packet, then the APID of the MAL header field 'URI From' shall be assigned to the primary header field 'Application Process Identifier'.

3.3.2.4   If the MAL header field 'URI From' contains a Source Id, then this identifier shall be assigned to the secondary header field 'Source Id' and the secondary header field 'Source Id Flag' shall be set to the value '1'.

3.3.2.5   If the MAL header field 'URI From' does not contain a Source Id, then the secondary header field 'Source Id Flag' shall be set to the value '0' and the secondary 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 secondary 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 secondary header field 'Authentication Id Flag' shall be set to the value '0' and the secondary header field 'Authentication Id' shall be left out;

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

   c) if the MCP 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 TC packet, then the APID of the MAL header field 'URI To' shall be assigned to the primary header field 'Application Process Identifier'.

3.3.4.2   If the MAL message is mapped to a TM packet, then the APID of the MAL header field 'URI To' shall be assigned to the secondary header field 'Secondary APID'.

3.3.4.3   If the MAL message is mapped to a TM packet, then the APID qualifier of the MAL header field 'URI To' shall be assigned to the secondary header field 'Secondary APID Qualifier'.

3.3.4.4   If the MAL header field 'URI To' contains a Destination Id, then this identifier shall be assigned to the secondary header field 'Destination Id' and the secondary 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 secondary header field 'Destination Id Flag' shall be set to the value '0' and the secondary 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 secondary 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 secondary header field 'Timestamp Flag' shall be set to the value '0' and the secondary 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 secondary header field 'QoSlevel' as specified by table 3-4.

Table 3-4: 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 secondary 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 secondary header field 'Priority Flag' shall be set to the value '0' and the secondary header field 'Priority' shall be left out;

b)   if the MCP PRIORITY is defined, then its value shall be assigned to the MAL header field 'Priority';

c)   if the MCP 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 secondary 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 secondary header field 'Domain Flag' shall be set to the value '0' and the secondary header field 'Domain' shall be left out;

b) if the MCP DOMAIN is defined, then its value shall be assigned to the MAL header field 'Domain';

c) if the MCP 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 secondary 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 secondary header field 'Network Zone Flag' shall be set to the value '0' and the secondary header field 'Network Zone' shall be left out;

b) if the MCP NETWORK_ZONE is defined, then its value shall be assigned to the MAL header field 'Network Zone';

c) if the MCP 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 secondary header field 'Session' as specified by table 3-5.

Table 3-5: 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 secondary 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 secondary header field 'Session Name Flag' shall be set to the value '0' and the secondary header field 'Session Name' shall be left out;

b) if the MCP SESSION_NAME is defined, then its value shall be assigned to the MAL header field 'Session Name';

c) if the MCP 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 secondary header field 'SDU Type' as defined by table 3-6, 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.

CCSDS RECOMMENDED STANDARD FOR MISSION OPERATIONS—MAL SPACE PACKET
TRANSPORT BINDING AND BINARY ENCODING

Table 3-6: Interaction Type and Stage Mapping

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

3.3.13 TRANSACTION ID

The value of the MAL header field 'Transaction Id' shall be assigned to the secondary
header field 'Transaction Id'.

3.3.14 SERVICE AREA

The value of the MAL header field 'Service Area' shall be assigned to the secondary header field 'Service Area'.

3.3.15 SERVICE

The value of the MAL header field 'Service' shall be assigned to the secondary header field 'Service'.

3.3.16 OPERATION

The value of the MAL header field 'Operation' shall be assigned to the secondary header field 'Operation'.

3.3.17 AREA VERSION

The value of the MAL header field 'Area Version' shall be assigned to the secondary header field 'Area Version'.

3.3.18 IS ERROR MESSAGE

If the MAL header field 'Is Error Message' is TRUE, then the secondary header field 'Is Error Message' shall be assigned with the value '1', otherwise it shall be assigned with the value '0'.

3.4 SPACE PACKET SPECIFIC FIELDS

3.4.1 OVERVIEW

The following subsections specify the values to be assigned to the Space Packet fields that are not the result of the MAL header mapping.

NOTE – The Space Packet Protocol layer expects from the upper layer a fully formed Space Packet, as specified by subsection 3.3.3.2.2 of reference [1].

3.4.2 SPACE PACKET PRIMARY HEADER

3.4.2.1 Overview

The following subsections provide the values to be assigned to the primary header fields that are not the result of the MAL header mapping. The Space Packet header is shown in table 3-2.

3.4.2.2   Packet Version Number

The field 'Packet Version Number' shall be assigned with the binary value '000' as specified by reference [1] and declared in the SANA registry 'Packet Version Number' (reference [7]).

NOTE   –   Only Version 1 CCSDS Packet is supported by this Recommended Standard.

3.4.2.3   Packet Type

The field 'Packet Type' shall be '0' for telemetry and '1' for telecommand.

3.4.2.4   Secondary Header Flag

The field 'Secondary Header Flag' shall be assigned with the binary value '1'.

NOTE   –   The field 'Secondary Header Flag' indicates presence of packet secondary header.

3.4.2.5   Sequence Flags

The field 'Sequence Flags' shall be assigned according to the size of the octet stream that results from the MAL message body encoding (see 4.4).

3.4.2.6   Packet Sequence Count

The field 'Packet Sequence Count' shall be assigned with the sequential binary count of each Space Packet generated for every APID and APID qualifier.

3.4.2.7   Packet Data Length

The field 'Packet Data Length' shall be assigned with the total number of octets in the Packet Data Field minus 1.

3.4.3   SPACE PACKET SECONDARY HEADER

3.4.3.1   Overview

The following subsections provide the values to be assigned to the secondary header fields that are not the result of the MAL header mapping.

3.4.3.2   Version Number

3.4.3.2.1   The field 'Version Number' shall identify the secondary header and the user data field as defined by this Recommended Standard.

3.4.3.2.2   The field 'Version Number' shall be assigned with the binary value '000'.

3.4.3.2.3   The version number '000' shall be added to the SANA registry 'Mission Operations MAL Space Packet Transport Binding Version Number' and shall refer to the Mission Operations—MAL Space Packet Transport Binding and Binary Encoding document 'CCSDS 524.1-B-1'.

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

3.4.3.3   Segment Counter

The field 'Segment Counter' shall enable ordering of a sequence of Space Packets resulting from the segmentation of a MAL message, as specified in 4.4 and 4.6.

3.5   MAL MESSAGE BODY MAPPING

3.5.1   OVERVIEW

The following subsections specify how the MAL message body is mapped to the Space Packet.

3.5.2   BODY MAPPING

The MAL message body shall be encoded as a Body (see 3.5.3.2) and assigned to the Space Packet user data field.

3.5.3   BODY ENCODING

3.5.3.1   Overview

This subsection specifies how the MAL message body is encoded.

The encoding formats are defined in this subsection. The encoding formats for data are in section 5.

NOTE –   The body encoding format can be re-used by a MAL Transport Binding to a messaging technology that is not the Space Packet Protocol.

3.5.3.2   Body

3.5.3.2.1   Each element of the MAL message body shall be encoded as a Body Element in the same order as it is declared in the operation definition.

3.5.3.2.2   If the body declared by the operation is empty, then the encoding result shall be empty.

3.5.3.3 Body Element

3.5.3.3.1 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 [2]), then the body element shall be encoded as follows:

Area	Service	Area Version	Type	Body Element
Unsigned 16-bit Integer (2 octets)	Unsigned 16-bit Integer (2 octets)	Unsigned 8-bit Integer (1 octet)	Signed 24-bit Integer (3 octets)	Publish Update List (variable, multiple of octet)
If <<Update Value Type>> is abstract				

3.5.3.3.2 The field 'Area' shall be assigned with the area number of the update list actual type encoded as an Unsigned 16-bit Integer.

3.5.3.3.3 If the update list actual type is defined by a service, then the field 'Service' shall be assigned with the service number of the update list actual type encoded as an Unsigned 16-bit Integer.

3.5.3.3.4 If the update list actual type is not defined by a service, then the field 'Service' shall be assigned with the value '0' encoded as an Unsigned 16-bit Integer.

3.5.3.3.5 The field 'Area Version' shall be assigned with the area version of the update list actual type encoded as an Unsigned 8-bit Integer.

3.5.3.3.6 The field 'Type' shall be assigned with the short form part of the update list actual type encoded as a Signed 24-bit Integer.

3.5.3.3.7 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).

3.5.3.3.8 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 [2]), then the body element shall be encoded as an Element (see section 5).

3.5.3.3.9 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 either Identifier or 'List<UpdateHeader>', then the body element shall be encoded as an Element (see section 5).

3.5.3.3.10 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).

3.5.3.3.11 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).

3.5.3.3.12 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).

3.5.3.4   Publish Update List

3.5.3.4.1   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

3.5.3.4.2   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 [2]). Using a UInteger as length field limits the list length to $2^{32} - 1$ elements.

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

3.5.3.5 Publish Update

3.5.3.5.1 The following fields shall be encoded:

Presence Flag	Encoded Update Size	Update
Boolean (see 5.8) (1 octet)	UInteger (variable, multiple of octet)	Element (see 5.2) (variable, multiple of octet)
	If Presence Flag is TRUE	

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

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

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

NOTES

1   Using a UInteger as size field limits the size of the encoded update to $2^{32} - 1$ octets.

2   The Publish Update encoding format enables a MAL broker not to decode the updates when handling a publish message.

4   MAL TRANSPORT INTERFACE MAPPING

4.1   OVERVIEW

The MAL specification (reference [2]) 'Transport Interface' section defines the interface to be provided by the Space Packet 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 Space Packet 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 Space Packet Protocol (reference [1]) are used by the mapping:

a)   'Packet request';

b)   'Packet indication'.

The parameters are listed in table 4-2.

CCSDS RECOMMENDED STANDARD FOR MISSION OPERATIONS—MAL SPACE PACKET
TRANSPORT BINDING AND BINARY ENCODING

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: Packet Interface Primitives

Primitive	Parameters
Packet request	Space Packet Application Process Identifier (APID) APID Qualifier (optional) QoS Requirement (optional)
Packet indication	Space Packet APID APID Qualifier (optional)

CCSDS RECOMMENDED STANDARD FOR MISSION OPERATIONS—MAL SPACE PACKET
TRANSPORT BINDING AND BINARY ENCODING

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 Space Packets.

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 [2]) 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 Space Packet Protocol 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 or several Space Packets and send each packet by calling the primitive 'Packet request'.

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, especially in MAL messages whose interaction type is SEND.

4.4.4 The MAL message header fields and body elements shall be mapped to the Space Packet 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   If the MAL message is mapped to a TC packet, then the following rules shall be applied:

a)   SPP Packet Type shall be '1' (according to [1]);

b)   the APID of the MAL header field 'URI To' shall be passed to the 'Packet request' primitive;

c)   the APID qualifier of the MAL header field 'URI To' shall be passed to the 'Packet request' primitive;

d)   the Space Packets that result from the mapping of the MAL message shall be delivered to the MAL application identified by the APID and APID qualifier of the MAL header field 'URI To'; the APID and APID qualifier are passed as parameters of the 'Packet request' primitive.

4.4.7   If the MAL message is mapped to a TM packet, then the following rules shall be applied:

a)   SPP Packet Type shall be '0' (according to [1]);

b)   the APID of the MAL header field 'URI From' shall be passed to the 'Packet request' primitive;

c)   the APID qualifier of the MAL header field 'URI From' shall be passed to the 'Packet request' primitive;

d)   the Space Packets that result from the mapping of the MAL message shall be delivered to the MAL application identified by the APID and APID qualifier of the MAL header field 'URI To'; the APID and APID qualifier are passed in the secondary header fields 'Secondary APID' and 'Secondary APID Qualifier'.

NOTE   –   The APID qualifier is not an optional parameter in the call to the 'Packet request' primitive.

4.4.8   The QoS properties from the TRANSMIT request shall be passed to the 'Packet request' primitive.

4.4.9   If the size of the octet stream that results from the MAL message body encoding (see 3.5.3.2) added to the size of the secondary header is less than or equal to the value of the MCP PACKET_DATA_FIELD_SIZE_LIMIT, then the following rules shall be applied:

a)   the octet stream shall be sent in the user data field of one Space Packet;

b)   the field 'Sequence Flags' of the Space Packet shall be assigned with the binary value '11', unsegmented user data.

4.4.10 If the size of the octet stream that results from the MAL message body encoding (see 3.5.3.2) added to the size of the secondary header is strictly greater than the value of the MCP PACKET_DATA_FIELD_SIZE_LIMIT, then the following rules shall be applied:

a) the octet stream shall be split in several user data segments whose size added to the size of the secondary header is equal to PACKET_DATA_FIELD_SIZE_LIMIT or less for the last segment;

b) the user data segments shall be sent in a sequence of Space Packets;

c) the MAL message header shall be mapped to the primary and secondary headers of every Space Packet of the sequence;

d) a segment counter starting from zero shall be incremented for each user data segment and assigned to the secondary header field 'Segment Counter';

e) every message whose MAL header fields 'Interaction type', 'Transaction Id', 'URI From', 'URI To', 'Session', 'Session Name', 'Domain', 'Network Zone', 'Service Area', 'Service', and 'Operation' are the same shall use the same segment counter, without resetting the counter to zero after a MAL message has been segmented;

f) the field 'Sequence Flags' in the first Space Packet of the sequence shall be assigned with the binary value '01', first segment of user data;

g) the field 'Sequence Flags' in the last Space Packet of the sequence shall be assigned with the binary value '10', last segment of user data;

h) the field 'Sequence Flags' in the other Space Packets of the sequence shall be assigned with the binary value '00', continuation segment of user data.

NOTES

1   This Recommended Standard does not specify any packet retransmission protocol in case some packets are lost.

2   In order to achieve reliability SPP delegates packet retransmission to a lower transport layer.

3   Examples of appropriate CCSDS transports that ensure packet retransmission are DTN (references [G3] and [G4]), COP (reference [G5]), LTP (reference [G6]), and AMS (reference [G7]).

4.4.11 If the size of the secondary header is not strictly less than the value of the MCP PACKET_DATA_FIELD_SIZE_LIMIT, then the TRANSMIT ERROR primitive shall be called with the error number MAL::INTERNAL.

4.4.12 If an error is returned by the Space Packet sending request, then the TRANSMIT ERROR primitive shall be called with the error number MAL::INTERNAL.

4.4.13 If the packet is successfully sent, then the TRANSMIT ACK primitive shall be called.

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 or several Space Packets and translate the packets into a MAL message.

4.6.2 The RECEIVE indication primitive shall be called after the primitive 'Packet indication' has been triggered one or several times (in case of MAL message segmentation) by the Space Packet Protocol layer.

4.6.3 The primitive 'Packet indication' shall provide the parameter 'APID qualifier'.

4.6.4 If the field 'Sequence Flags' of the delivered Space Packet is equal to the binary value '11', unsegmented user data, then 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 Space Packet;

b) the parameter 'APID qualifier' of the primitive 'Packet indication';

c) the XML specification of the MO service (see section 6 of reference [2]) identified by the MAL header fields 'Service Area', 'Service', and 'Area Version'.

4.6.5 If the field 'Sequence Flags' of the delivered Space Packet is not equal to the binary value '11', unsegmented user data, then the following rules shall be applied:

a) the MAL message header fields shall be generated according to the specifications given in section 3 of this Recommended Standard, by using the Space Packet and the parameter 'APID qualifier' of the primitive 'Packet indication';

b) every Space Packet whose MAL header fields 'Interaction Type', 'Transaction Id', 'URI From', 'URI To', 'Session', 'Session Name', 'Domain', 'Network', 'Service Area', 'Service', and 'Operation' are the same shall be sequenced according to the secondary header field 'Segment Counter';

   c) a full Space Packet sequence shall be detected as follows:

      1) the first Space Packet of the sequence shall have the field 'Sequence Flags' equal to the binary value '01', first segment of user data;

      2) the last Space Packet of the sequence shall have the field 'Sequence Flags' equal to the binary value '10', last segment of user data;

      3) the other Space Packets of the sequence shall have the field 'Sequence Flags' equal to the binary value '00', continuation segment of user data;

      4) starting from the second Space Packet of the sequence, every Space Packet shall have the secondary header field 'Segment Counter' assigned with the value 1 added to the 'Segment Counter' of the previous Space Packet in the sequence;

   d) if a full Space Packet sequence is detected, then a MAL message shall be generated according to the specifications given in section 3 of this Recommended Standard, by using the following input data:

      1) the MAL message header generated from the first Space Packet of the sequence;

      2) the user data segments of the Space Packet sequence;

      3) the parameter 'APID qualifier' of the primitive 'Packet indication' triggered by the first Space packet of the sequence;

      4) the XML specification of the MO service (see section 6 of reference [2]) identified by the MAL header fields 'Service Area', 'Service', and 'Area Version';

   e) if a Space Packet sequence is not full after a mission-defined timeout, then the Space Packets of the sequence may be discarded.

4.6.6　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; if the field belongs to the MAL message header then the declared type is given by table 3-1; 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 [2]);

   b) actual type: the type of the element.

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 re-used by a MAL Transport Binding to a messaging technology that is not the Space Packet Protocol.

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<>; then the following fields shall be encoded:

Area	Service	Area Version	Type
Unsigned 16-bit Integer (2 octets)	Unsigned 16-bit Integer (2 octets)	Unsigned 8-bit Integer (1 octet)	Signed 24-bit Integer (3 octets)

NOTE   –   As specified by MAL (reference [2]), 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 field 'Area' shall be assigned with the area number of the element actual type
encoded as an Unsigned 16-bit Integer.

5.2.5 If the element actual type is defined by a service, then the field 'Service' shall be
assigned with the service number of the element actual type encoded as an Unsigned 16-bit
Integer.

5.2.6 If the element actual type is not defined by a service, then the field 'Service' shall be
assigned with the value '0' encoded as an Unsigned 16-bit Integer.

5.2.7 The field 'Area Version' shall be assigned with the area version of the element actual
type encoded as an Unsigned 8-bit Integer.

5.2.8 The field 'Type' shall be assigned with the short form part of the element actual type encoded as a Signed 24-bit Integer.

NOTE – The short form part is required to be a signed 32-bit integer by MAL (reference [2]). 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 $2^{23}-1$.

5.2.9 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 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 [2].

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 [2], 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 [2]). 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:

Presence Flag	Element
Boolean (1 octet)	Element (variable, multiple of octet)
	If Presence Flag is TRUE

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

CCSDS RECOMMENDED STANDARD FOR MISSION OPERATIONS—MAL SPACE PACKET
TRANSPORT BINDING AND BINARY ENCODING

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 [2]). 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

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

5.8.2 If the Boolean value is TRUE then the integer '1' shall be encoded otherwise the integer '0' shall be encoded.

5.9 DURATION

5.9.1 A MAL::Duration shall be encoded according to the CUC Time Code Format (reference [5]) given by the MCP DURATION_CODE_FORMAT.

5.9.2 Negative values shall be encoded using the two's complement representation of the MAL::Duration.

5.9.3 The Time Code Format P-Field shall not be encoded.

5.10 FLOAT

A MAL::Float value shall be encoded according to the binary interchange format of width 32 bits defined in reference [3].

5.11 DOUBLE

A MAL::Double shall be encoded according to the binary interchange format of width 64 bits defined in reference [3].

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

5.15.1 If the MCP VARINT_SUPPORTED is TRUE, then a MAL::Short shall be encoded as a 2-octet Signed Varint (see 5.28).

5.15.2 If the MCP VARINT_SUPPORTED is FALSE, then a MAL::Short shall be encoded as a Signed 16-bit Integer (see 5.26).

5.16 USHORT

5.16.1 If the MCP VARINT_SUPPORTED is TRUE, then a MAL::UShort shall be encoded as a 2-octet Unsigned Varint (see 5.27).

5.16.2 If the MCP VARINT_SUPPORTED is FALSE, then a MAL::UShort shall be encoded as an Unsigned 16-bit Integer (see 5.25).

5.17 INTEGER

5.17.1 If the MCP VARINT_SUPPORTED is TRUE, then a MAL::Integer shall be encoded as a 4-octet Signed Varint (see 5.28).

5.17.2 If the MCP VARINT_SUPPORTED is FALSE, then a MAL::Integer shall be encoded as a Signed 32-bit Integer (see 5.26).

5.18 UINTEGER

5.18.1 If the MCP VARINT_SUPPORTED is TRUE, then a MAL::UInteger shall be encoded as a 4-octet Unsigned Varint (see 5.27).

5.18.2 If the MCP VARINT_SUPPORTED is FALSE, then a MAL::UInteger shall be encoded as an Unsigned 32-bit Integer (see 5.25).

5.19 LONG

5.19.1 If the MCP VARINT_SUPPORTED is TRUE, then a MAL::Long shall be encoded as a 8-octet Signed Varint (see 5.28).

5.19.2 If the MCP VARINT_SUPPORTED is FALSE, then a MAL::Long shall be encoded as a Signed 64-bit Integer (see 5.26).

5.20 ULONG

5.20.1 If the MCP VARINT_SUPPORTED is TRUE, then a MAL::ULong shall be encoded as a 8-octet Unsigned Varint (see 5.27).

5.20.2 If the MCP VARINT_SUPPORTED is FALSE, then a MAL::ULong shall be encoded as an Unsigned 64-bit Integer (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 [2]). 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 [4]).

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 [5]) specified by the MCPs TIME_CODE_FORMAT and TIME_EPOCH.

5.22.2 The Time Code Format P-Field shall not be encoded.

5.23 FINETIME

5.23.1 A MAL::FineTime shall be encoded according to the CCSDS Time Code Format (reference [5]) specified by the MCPs FINE_TIME_CODE_FORMAT and FINE_TIME_EPOCH.

5.23.2 The Time Code Format P-Field shall not be encoded.

5.24 URI

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

5.25 UNSIGNED N-BIT INTEGER

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

5.25.2 The bit 'Bit N−1' shall designate the least significant bit of the unsigned integer.

5.25.3 Each bit of the unsigned integer shall be encoded from 'Bit 0' to 'Bit N−1'.

5.26 SIGNED N-BIT INTEGER

5.26.1 A signed N-bit integer shall be encoded according to its 2's complement representation on a given number of bits N.

5.26.2 The sign bit shall be assigned to 'Bit 0'.

5.26.3 Each bit of the signed integer shall be encoded from 'Bit 0' to 'Bit N−1'.

5.27 UNSIGNED VARINT

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

5.27.2 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.

5.27.3 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.

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

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

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

5.27.6 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'.

5.28 SIGNED VARINT

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

5.28.2 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 2'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; 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.

5.28.3 The resulting unsigned integer shall be encoded as an Unsigned Varint.