ETSI TS 102 357 V1.1.1 (2005-05)

Technical Specification

Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM) Common air interface specification; Satellite Independent Service Access Point (SI-SAP)

Reference

DTS/SES-00081

Keywords

air interface, broadband, multimedia, satellite

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Satellite Earth Stations and Systems (SES).

Introduction

The present document is a first release of the formal specification for the Satellite Independent Service Access Point (SI-SAP) interface. The SI-SAP defines the Satellite Independent to Satellite Dependent (SI-SD) interface as described in the BSM Services and Architectures report [1].

This release of the present document defines the main functional groupings for both the User-plane (U-plane) and the Control plane (C-plane) and defines the associated primitives. This provides an architectural framework for the development of the SI-SAP functions.

Future revisions of the present document may revise and extend the SI-SAP functions. Future releases should provide more detail to support realization (implementation) of the SI-SAP.

1 Scope

The present document specifies the Satellite Independent Service Access Point (SI-SAP) for the provision of standard Broadband Satellite Multimedia (BSM) services. The SI-SAP defines the Satellite Independent to Satellite Dependent (SI-SD) interface as described in the BSM Services and Architectures report [1]. This interface also corresponds to the endpoints of the BSM bearer services as illustrated in figure 1.

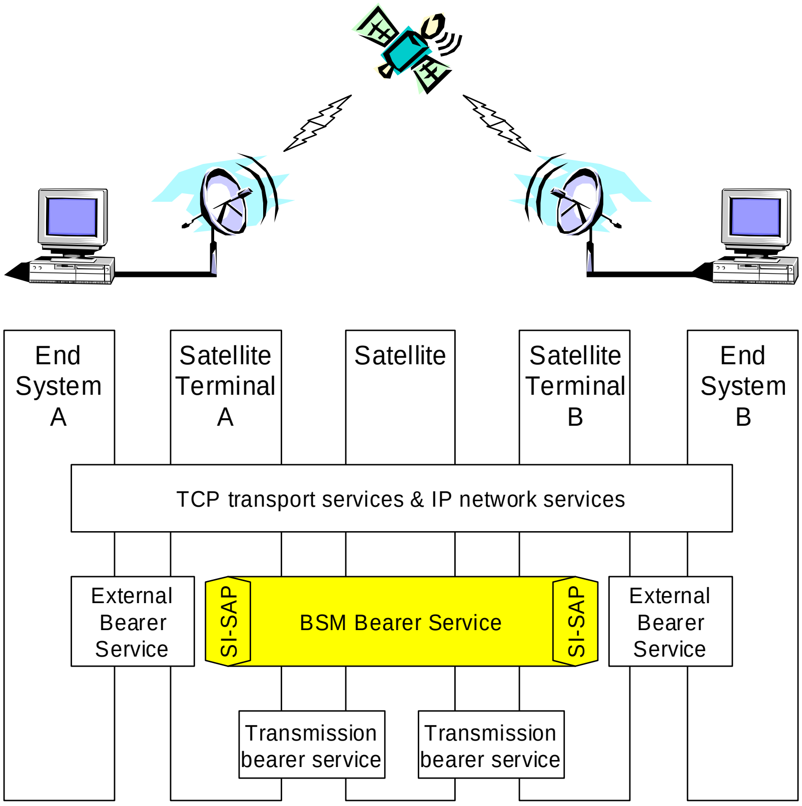

Figure 1: Satellite Independent Service Access Point (SI-SAP)

The SI-SAP services are accessed via the SI-SAP interface and they can be used to transport standard IP-based network services. These SI-SAP services are applicable to any IP-based satellite network, including both transparent and regenerative satellites, and including both star and mesh topologies. For example, in the case of a star network topology, one SI-SAP would be located in the remote Satellite Terminal (ST) and the other in the hub ST; whereas in the case of a mesh network topology, the SI-SAPs would be located in the source and destination STs.

The present document divides the services into user plane (U-plane), control plane (C-plane) and management plane (M-plane) services. This is a logical (functional) division but these different planes may also be physically separated. For example, in the case of a star topology being used to provide mesh connectivity using bridging at the hub, the SI-SAP will be located in the remote STs for the U-plane but will be located in the remote STs plus the hub ST for the C-plane and M-plane. Similar separations may be required for fully meshed network topologies.

The SI-SAP is a logical interface and the service primitives are defined in abstract functional terms. The present document is only intended to define the functionality of this interface and is not intended to constrain the implementation of the interface. The SI-SAP may be implemented as a physical interface (e.g. using message based exchanges); as a logical interface (e.g. using API function calls); as an internal interface (e.g. using embedded procedure calls) or in any other format that provides the specified functionality.

More details of the models and concepts used in the present document can be found in the associated SI-SAP guidelines [5].

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

[1] ETSI TR 101 984: "Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia; Services and Architectures".

[2] IETF RFC 2475: "An Architecture for Differentiated Service".

[3] IETF RFC 1633: "Integrated Services in the Internet Architecture: an Overview".

[4] IEEE 802-2001: "IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture".

[5] ETSI TR 102 353 "Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM); Guidelines for the Satellite Independent Service Access Point (SI-SAP)".

[6] IEEE EtherType Field Public Assignment Announcement http://standards.ieee.org/regauth/ethertype/eth.txt

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

BSM_IDentity (BSM_ID): BSM_ID is an SI-SAP address that identifies the BSM subnetwork point of attachment (SNPA)

Diffserv: IP Differentiated Services model for Quality of Service (QoS) [2]

NOTE: IP Differentiated Services (Diffserv) provides coarse-grained controls to aggregates of flows by defining Per-Hop Behaviour (PHB) based on defined Diffserv Codepoints (DSCP) in each IP packet header.

Intserv: IP Integrated Services model for Quality of Service (QoS) [3]

NOTE: IP Integrated Services (Intserv) provides fine-grained service guarantees to individual IP flows where bandwidth is reserved for a specific IP flow, and appropriate traffic conditioning and scheduling is installed in routers along the path.

Queue Identifier (QID): Queue_Identifier is an SI-SAP parameter that identifies an abstract queue at the SI-SAP

Service Access Point (SAP): service access point is a logical interface between two adjacent layers in a protocol stack

service primitive (primitive): primitives used to define layer to layer interactions and that represent, in an abstract way, the logical exchange of information and control between adjacent layers

NOTE: The service primitives do not specify or constrain implementation.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ARP	Address Resolution Protocol
BSM_ID	BSM IDentity
BSM_UID	BSM Unicast IDentity
BSM_GID	BSM Group IDentity
DTS	Data Transport Service
IANA	Internet Assigned Numbers Authority
IEEE	(The) Institute of Electrical and Electronic Engineers
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
LAN	Local Area Network
LL	Lower Layer
MAC	Medium Access Control
MTU	Maximum Transmission Unit
NCC	Network Control Centre
OUI	Organisationally Unique Identifier
PDU	Protocol Data Unit
QID	Queue IDentifier
QoS	Quality of Service
SAP	Service Access Point
SDU	Service Data Unit
SD	Satellite Dependent
SDAF	Satellite Dependent Adaptation Functions
SI	Satellite Independent
SIAF	Satellite Independent Adaptation Functions
SI-SAP	Satellite Independent SAP
SLC	Satellite Link Control
SNPA	SubNetwork Point of Attachment
ST	Satellite Terminal
UL	Upper Layer

4 SI-SAP overview

4.1 General

The SI-SAP is the formal service boundary between the satellite dependent part of the air interface and the satellite independent part. The introduction of the SI-SAP into the system design affords the following benefits:

- elements above the SI-SAP can be ported with greater ease to new satellite systems;
- extensibility to support new higher-layer functionality without major re-engineering of existing designs.

4.2 IP reference model

The SI-SAP corresponds to the interface between the Satellite Independent Layers and the Satellite Dependent Layers as illustrated in figure 2. Two adaptation functions are defined to adapt these layers to/from the SI-SAP:

- Satellite Independent Adaptation Functions (SIAF) to adapt between the IP network layers and the SI-SAP;
- Satellite Dependent Adaptation Functions (SDAF) to adapt between the SI-SAP and the native services of the satellite dependent SLC layer.

NOTE: The functionality of the SIAF and the SDAF may be NULL for some of the SI-SAP services.

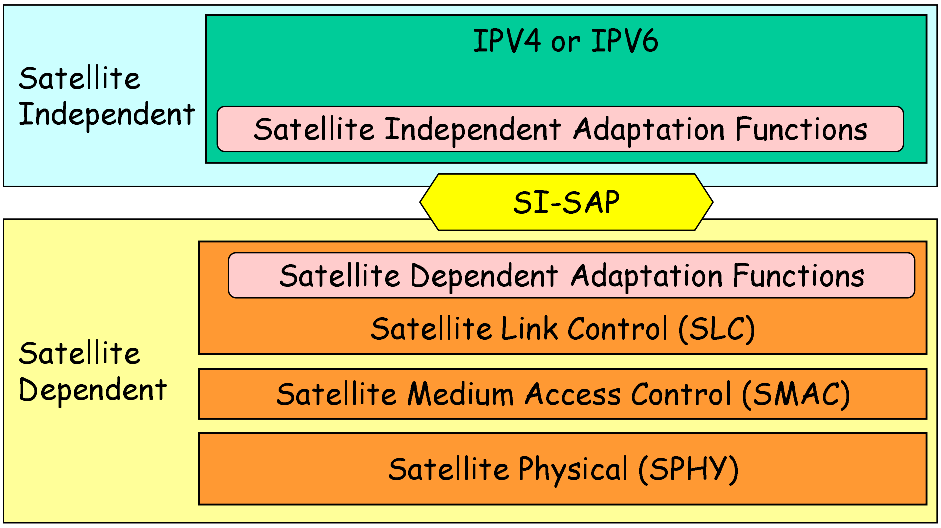

Figure 2: Satellite Independent Service Access Point (SI-SAP) IP reference model

4.3 Expanded reference model

For the purposes of the present document, the SI-SAP is logically divided into 3 separate SAPs as shown in figure 3:

a) the SI-U-SAP for U-plane services;

b) the SI-C-SAP for C-plane services;

c) the SI-M-SAP for M-plane services.

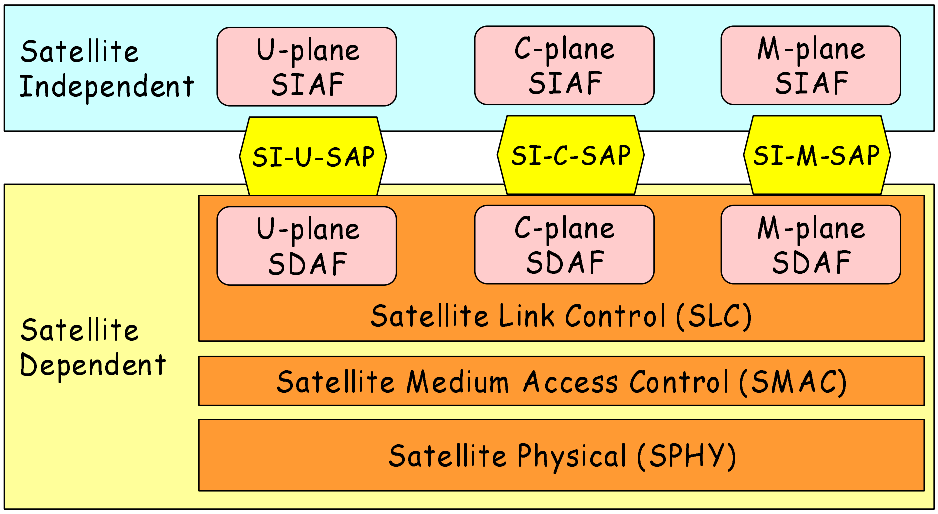

Figure 3: Satellite Independent Service Access Point (SI-SAP) Expanded reference model

Each SAP may support multiple service endpoints as defined in clauses 4.3.1 to 4.3.3.

4.3.1 User plane SAP (SI-U-SAP)

The SI-U-SAP is concerned with the services that are used to transport IP packets between STs. IP packets are interworked into the SI-U-SAP endpoint at the source ST, transported transparently to the corresponding SI-U-SAP endpoints at the destination STs where they are interworked into the destination ports.

NOTE: The U-plane services can be used to transport IP packets to/from internal ports as well as external ports. For example, ST management traffic based on SNMP can be transported via the U-plane.

4.3.2 Control plane SAP (SI-C-SAP)

The SI-C-SAP is concerned with control and signalling related to the user plane services:

- **Control** refers to local services, that are concerned with control of the behaviour of the lower layers of the local stack;
- **Signalling** refers to remote services that involve interaction with the Network Control Centre (NCC) or with the peer ST.

4.3.3 Management plane (SI-M-SAP)

The SI-M-SAP is concerned with management services.

The management plane services are outside the scope of the present document.

5 Common elements

5.1 Service primitives

Layer-to-layer interactions are specified in terms of SI-SAP primitives (service primitives). The primitives represent, in an abstract way, the logical exchange of information and control between the adjacent layers. They do not specify or constrain implementation.

The primitives that are exchanged between the upper layers (nominally the SIAF) and the lower layers (nominally the SDAF) are of the following four types: request, response, indication and confirm as illustrated in figure 4.

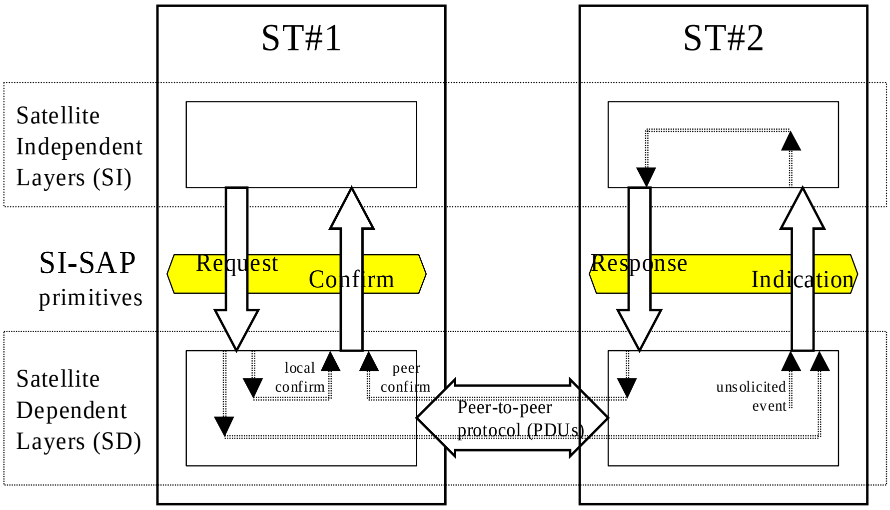

Figure 4: SI-SAP primitives

The REQUEST primitive type (-req) is used when the higher SI layer is requesting a service from the lower SD layer.

The INDICATION primitive type (-ind) is used by the lower SD layer to notify the higher SI layer of activities. The INDICATION primitive type may either be related to a primitive type REQUEST at the peer entity, or may be an indication of an unsolicited lower layer event.

The RESPONSE primitive type (-res) is used by the SI upper layer to acknowledge receipt, from the SD lower layer, of the primitive type INDICATION.

The CONFIRM primitive type (-cfm) is used by the SD lower layer to confirm that the activity requested by a previous REQUEST primitive has been completed. The CONFIRM primitive type may correspond to either a local confirmation or a peer confirmation.

5.2 BSM identities

5.2.1 BSM_Identity (BSM_ID)

The BSM_ID is an SI-SAP address that identifies the BSM subnetwork point of attachment (SNPA).

The format for the BSM_ID shall be a 48 bit address (6 octets) that conforms to the IEEE 802 [4] specification for LAN MAC addresses.

NOTE: The format for the BSM_ID permits a direct mapping to a L2 address based on IEEE 802 [4] MAC addresses. However, the IEEE 802 [4] address definition also permits locally administered addresses and this can be used to define a mapping to any other L2 address format with 46 bits or less.

The BSM_ID is unique within the entire BSM subnetwork and it is intended that each BSM_IDs is associated with a unique Layer 2 Address (e.g. by a direct mapping).

5.2.2 Unicast and group BSM_IDs

Both unicast and group BSM_IDs are defined:

- A BSM Unicast Id (UID) defines an individual SNPA. A given instance of an SI-SAP (SAPI) shall be associated one UID.
- A BSM Group Id (GID) defines a lower layer multicast group. A given instance of an SI-SAP may be associated with zero or more GIDs.

The UID and GIDs are assigned from the same numbering space. The UID and GID are distinguished by the I/G bit (Individual/Group bit) as defined in IEEE 802 [4].

5.3 Queue Identifiers(QIDs)

The Queue_Identifier (QID) is an SI-SAP parameter that identifies an abstract queue at the SI-SAP.

The format for the QID shall be a 24 bit integer.

NOTE 1: The format for the QID permits a large number of queues to be defined. This is required to permit different QIDs to be used for a range of QID related functions, including both resource reservation functions (see clause 6.3.2) and flow control functions (see clause 6.3.5).

NOTE 2: The QID format only applies at the SI-SAP. This QID provides an abstract label to identify a lower layer queue and the QID is not expected to be carried over the air interface as part of each data transfer.

A specific QID is used for every user data transfer via the SI-SAP. The satellite dependent layers are responsible for assigning satellite capacity to these abstract queues according to the specified queue properties (e.g. QoS). QIDs may be assigned statically (e.g. by management configuration) or dynamically using the SI-SAP resource reservation procedures defined in clause 6.3.2.

A QID is only required for submitting (sending) data via the SI-SAP and the QID is assigned when the associated queue is opened. An open queue is uniquely identified by the associated QID: in particular, the QID is used to label all subsequent data transfers via that queue.

5.4 Primitive parameters

5.4.1 General

Each SI-SAP function is defined as a list of primitives that are used to invoke the service.

5.4.2 Common parameters

The following parameters will be common to all of the SI-SAP primitives:

- Primitive ID; this is an abstract identity that is used to uniquely identify the primitive (name and type). All primitives shall be identified by the Primitive ID.

These common parameters are omitted from the following primitive definitions for simplicity.

5.4.3 Mandatory and optional parameters

The presence or absence of each parameter in a each type of primitive is defined using the following codes:

- Mandatory (M). A Mandatory parameter shall appear in all instances of that primitive.
- Optional (O). An Optional parameter may appear in a given instance.
- Equal (=). An equal symbol means that this parameter shall have the same value as the invoking parameter (i.e. it is a copy of the invoking parameter). A parameter in a Request primitive cannot have this status. This additional symbol can apply to either Mandatory (M=) or Optional (O=) parameters.
- Absent (-). An absent parameter shall not appear in that primitive.

5.4.4 Service Data Unit (SDU)

The Service Data Unit (SDU) corresponds to the "payload" of the primitive. The SDU is only used in the U-plane data transfer primitives.

For example, when a data transport SI-U-UNITDATA-req primitive is invoked at the sender, the SDU contains the IP packet(s) to be transported over the BSM network, and following their successful transport over the BSM network the peer ST delivers them to the upper layers using a data transport SI-U-UNITDATA-ind primitive with an SDU that contains the wanted IP packet(s).

6 SI-SAP service definitions

6.1 General

6.1.1 Service summary

The following tables summarize the services provided at the SI-SAP. The services are divided into functional groups for each plane (i.e. U-plane services, C-plane services and M-plane services).

Table 1: U-plane services

U-plane services		
Service	**Description**	**Primitives**
Data Transfer	Submission of data to the SD layers for transmission to peer destination; and receipt of data from the SD layers by that peer. This service can be used for both unicast and multicast data transfer. The type of transfer is defined by the addresses used in the primitive.	SI-U-UNITDATA-xxx

Table 2: C-plane services

C-plane services		
Service	**Description**	**Primitives**
Address Resolution	A mechanism to associate a BSM_ID address to a given IP address	SI-C-AR_QUERY-xxx SI-C-AR_INFO-xxx
Resource Reservation	A mechanism to open, modify and close SD layer queues for use by the SI layers.	SI-C-QUEUE_OPEN-xxx SI-C-QUEUE_MODIFY-xxx SI-C-QUEUE_CLOSE-xxx
Group Receive	A mechanism to activate and configure the SD layers to receive a wanted multicast service	SI-C-RGROUP_OPEN-xxx SI-C-RGROUP_CLOSE-xxx
Group Send	A mechanism to activate and configure the SD layers to send a wanted multicast service	For further study
Flow Control	A mechanism to activate and configure the SD layers to provide SI-SAP flow control on one or more of the SD layer queues.	SI-C-FLOW_ACTIVATE-xxx SI-C-FLOW_MODIFY-xxx SI-C-FLOW_DEACTIVATE-xxx

Table 3: M-plane services

M-plane services		
Service	**Description**	**Primitives**
None	No M-plane services are defined	

The suffix "xxx" on the primitive names in these tables refers to one or more of the primitive types defined in clause 5.1; namely REQUEST (req), CONFIRM (cfm), INDICATION (ind) or RESPONSE (res). Each service uses one or more different types of primitives: the primitive types that are used for each service are defined in clause 6.2 and clause 6.3.

6.1.2 Service assumptions

The SI-SAP services are defined using the following assumptions:

1) the SI-SAP services assume that primitives are exchanged across the interface without loss or corruption;.

2) each service invocation operates independently from any other service invocation at that same SI-SAP. Multiple parallel invocations of the same service may occur.

NOTE: Where appropriate, the primitives should be assigned a locally unique parameter that is used to associate pairs of requests and responses associated with a given invocation.

6.2 U-plane services

6.2.1 Data transfer

6.2.1.1 Description

SI-SAP data transfer services are used to send and receive user data via the SI-SAP. The following services are defined:

Table 4: Data transfer services

Service	Primitives	Comments
Send data	SI-U-UNITDATA-req	Used for both unicast and multicast
Receive data	SI-U-UNITDATA-ind	Used for both unicast and multicast

6.2.1.2 Service Data Unit (SDU)

The user data is transferred in the Service Data Unit (SDU) of the data transport primitives: each SDU shall contain a complete IP datagram.

In all cases, the SDU shall be capable of accepting a Maximum Transmission Unit (MTU) of at least 1 500 bytes.

NOTE 1: This minimum value of MTU is larger than the minimum defined by IETF for IPv6 (i.e. larger than 1 280 bytes). The value of 1 500 bytes is chosen to correspond to the widely used Ethernet packet size. A larger MTU may be needed for bridging applications.

NOTE 2: The SDU only defines the format of the data transferred at the SI-SAP. The lower layers (below the SI-SAP) may segment the IP packet as required for transmission over the air interface.

The SDU that is delivered at the destination shall be identical to the SDU that was submitted by the source. This requirement means that the SDU delivered at the destination ST can be submitted directly to the destination IP layer without requiring any reassembly above the SI-SAP.

6.2.1.3 IP transfer capability

An IP transfer capability is a set of network capabilities provided by IP based networks to transfer IP packets. At the SI-SAP the IP transfer capability is defined via an abstract representation of the underlying resources, including the associated set of underlying traffic control and congestion control functions.

For a given invocation of the data transfer services, the IP transfer capability shall be defined indirectly using the Queue_ID (QID) parameter. The QID defines the IP transfer capability provided by the underlying resources for that data transfer. The definition of the QID and the associated resource reservation is described in more detail in clause 6.3.2.

6.2.1.4 Lower layer data losses

Data transfers submitted to the lower layers for transmission may suffer errors or packet loss for various reasons such as packet drops and transmission errors. Different data transfer properties (e.g. different error rates and different levels of dropping) may apply to different QIDs. The expected properties should be indicated via the resource reservation procedures and associated quality of service as described in clause 6.3.2.

The lower layers may be able to detect some or all of these errors. However, any detected lower layer data errors are handled silently and are not reported to the upper layers via the U-plane.

NOTE: Data losses may be reported via the M-plane: this is the preferred route for these error reports.

6.3 C-plane services

6.3.1 Address resolution

6.3.1.1 Description

Address resolution services are used to associate an IPv4 unicast or IPv4 multicast address to the corresponding BSM_Identity (BSM_ID). A successful address resolution service returns the associated BSM_ID.

The BSM_ID can be either a Unicast_ID (BSM_UID) or a Group_ID (BSM_GID). The BSM_UID is the identity that is used for unicast services and the BSM_GID is the corresponding identity for multicast services.

The following address resolution service primitives are defined:

Table 5: Address resolution services

Service	Primitives	Comments
On-demand request to provide a new address association	SI-C-AR_QUERY-req SI-C-AR_QUERY-cfm	The same primitive is used for all variants of on-demand address resolution.
Unsolicited announcement of a new address association	SI-C-AR_INFO-ind	The same primitive is used for all variants of unsolicited address resolution.

Four variants of the address resolution services are defined as follows:

a) **Unicast sender address resolution** is used by a unicast sender to determine the "next-hop" BSM_UID that is associated with the given (next hop) IPv4 unicast address. This associated BSM_UID is used in the data transfer primitives when submitting unicast data for that destination to the lower layer. The BSM_UID is used by the lower layers to deliver the SDU over the BSM subnetwork to the intended next-hop IP destination (next hop router, or end host).

NOTE 1: This service is comparable to the service provided by ARP [RFC 826] to determine the destination MAC address in Ethernet given the associated IPv4 unicast address.

b) **Unicast receiver address resolution** is used by a unicast receiver to determine the BSM_UID that is associated with the given (local) IPv4 unicast address. This associated BSM_UID is used in the data transfer primitives when data for this unicast destination is delivered from the lower layer.

NOTE 2: The BSM_UID corresponds to the local SNPA that is associated with the local IP address (local router port, or end host).

c) **Multicast sender address resolution** is used by a multicast sender to determine the BSM_GID that is associated with the given IPv4 multicast destination address. The associated BSM_GID is then used in the data transfer primitives when submitting multicast data for that destination to the lower layer. The BSM_GID is used by the lower layer to deliver the SDU over the BSM subnetwork to the intended next-hop IP destination(s) (next hop routers, or end hosts).

d) **Multicast receiver address resolution** is used by a unicast receiver to determine the BSM_GID that is associated with the given IPv4 multicast destination address. The associated BSM_GID is then used in the data transfer primitives when data for this multicast destination is delivered from the lower layer.

NOTE 3: The BSM_GID refers to the local SNPA that is associated with a particular IPv4 Class D destination address. The same IPv4 Class D address may be associated with multiple SNPAs (i.e. this multicast group may be delivered to multiple local router ports, or multiple end hosts).

6.3.2 Resource reservation

6.3.2.1 Description

Resource reservation is the function that assigns the Queue_Identifier (QID) and defines or modifies the properties of the abstract queue that is associated with that QID.

Once assigned, the QID is used for user data transfer via the SI-SAP. The satellite dependent layers are responsible for assigning satellite capacity to these abstract queues according to the specified queue properties (e.g. QoS).

Resource reservation is used to open, modify and close queues for both unicast and multicast flows. Resource reservation is only required for sending data and is not required for receiving data. Every open queue is identified with an associated Queue_ID (QID). The QID uniquely identifies the queue and is used as the label for all subsequent use and control of those queues.

Resource reservation defines three methods for opening a queue and assigning a QID to that queue:

a) **Static queues**. These are queues that are created by management configuration without using any of the procedures defined in this clause.

NOTE: The QIDs that are associated with these static queues can be used to send data via the U-plane without any further activation; i.e. without requiring any use of the dynamic procedures defined in this clause.

b) **Dynamic queues**. These are queues that are created dynamically using the procedures defined in this clause. There are two variants:

1) queues that are dynamically invoked by the satellite independent adaptation functions (SIAF) at the source ST by issuing a SI-C-QUEUE_OPEN-REQ primitive;

2) queues that are dynamically invoked by the satellite dependent adaptation functions (SDAF) at the source ST using the SI-C-QUEUE_OPEN-IND primitive.

The following dynamic resource reservation service primitives are defined:

Table 6: Resource reservation services

Service	Primitives	Comments
Upper layer request to open a new resource reservation	SI-C-QUEUE_OPEN-req SI-C-QUEUE_OPEN-cfm	
Lower layer request to open a new resource reservation	SI-C-QUEUE_OPEN-ind SI-C-QUEUE_OPEN-res	Upper layer can reject using the response primitive
Upper layer request to modify an existing resource reservation	SI-C-QUEUE_MODIFY-req SI-C-QUEUE_MODIFY-cfm	
Lower layer request to modify an existing resource reservation	SI-C-QUEUE_MODIFY-ind SI-C-QUEUE_MODIFY-res	Upper layer can reject using the response primitive
Upper layer request to close an existing resource reservation	SI-C-QUEUE_CLOSE-req SI-C-QUEUE_CLOSE-cfm	
Lower layer request to close an existing resource reservation	SI-C-QUEUE_CLOSE-ind SI-C-QUEUE_CLOSE-res	Response primitive provides acknowledgement only - upper layer cannot reject a CLOSE

6.3.2.2 Queue properties

The resource reservation primitives define the properties of the queue via the following information:

1) **Quality of Service information:** this set of parameters define the QoS characteristics that are associated with the QID. These parameters shall support both the IETF DiffServ architecture [2] and the IETF IntServ architecture [3].

2) **Flow Control information:** this additional set of parameters define the flow control characteristics that are associated with the QID. These parameters shall support the flow control functions defined in clause 6.3.5.

The initial QoS and Flow Control parameters shall be defined when a queue is opened. These can subsequently be changed or extended using the resource modify procedures defined below.

6.3.2.3 Static queues

Static queues refers to the case where the queue and the associated QID are assigned via management configuration. The resource reservation primitives defined in this clause shall not be used to open, modify or close a statically defined queue and any resource reservation primitives that refer to a static queue should be ignored.

NOTE: A particular example of a static queue could be a default QID = N that corresponds to "*best effort unicast service for sending data to the default address (e.g. to the gateway)*".

6.3.2.4 Dynamic queues

Dynamic queues refers to the case where the queue and the associated QID are assigned using the resource reservation primtives defined in this clause. Dynamic queue reservations are divided into two variants:

- Upper Layer initiated requests (UL resource requests). In this case, the queue and the associated QID are assigned by the lower layers in response to a request from the upper layers.

- Lower Layer initiated requests (LL resource requests). In this case, the queue and the associated QID are assigned by the lower layers without any prior request from the upper layers.

An UL resource request may either be accepted or rejected by the lower layers. For example, if the higher layers attempt to request a type of service that is not supported by that port, the lower layers would reject the request.

Once open dynamic queues can only be modified or closed using the primitives defined in this clause. For example, the lower layers may close an existing queue at any time by issuing an unsolicited SI-C-QUEUE_CLOSE-IND primitive.

6.3.2.5 Queue modification

Any existing (open) dynamic queue (whether opened by a UL or a LL request) may be modified or closed at any time according to the following rules:

- An open dynamic queue may be closed at any time by either a request from the upper layers or an unsolicited indication from the lower layers.
- An open dynamic queue may be modified at any time by the lower layers. The upper layer may either accept the modification, or may reject the modification by closing the queue.
- The upper layers may request a modification to any open dynamic queue at any time. The lower layers may either accept the modification request or reject the modification request.

6.3.3 Group receive

6.3.3.1 Description

A multicast group address (e.g. an IPv4 Class D address) is associated with a series of lower layer (satellite dependent) parameters using the group receive services. Group receive may be either static or dynamic:

- Static group receive refers to static (pre-configured) reception of one or more groups. The primitives defined in this clause shall not be used to control the reception of these static groups.
- Dynamic group receive refers to dynamic reception of a specific group. The wanted group(s) are enabled on-demand using the primitives defined in this clause.

The primitives defined in this clause are used to dynamically configure the lower layers at the receiving ST (i.e. the destination of the data) to enable the wanted traffic to be received by the lower layers and delivered to the upper layer using the SI-U-UNITDATA-ind primitives.

The lower layer configuration should include all of the parameters that are needed to activate reception of the wanted data (e.g. set layer 2 and/or layer 1 filters).

Each invocation of the group receiving enable services refers to either a single value, or a range of values of the BSM_GID.

NOTE: This service operation may need to be performed many times triggered by protocol operations in the IP layer (e.g. IGMP).

The following group receive services are defined:

Table 7: Group receiver enable services

Service	Primitive	Comments
Upper layer request to activate a new Group for reception	SI-C-RGROUP_OPEN-req SI-C-RGROUP_OPEN-cfm	
Lower layer announcement of a new Group for reception	SI-C-RGROUP_OPEN-ind	No response required from upper layers
Upper layer request to deactivate an existing Group for reception	SI-C-RGROUP_CLOSE-req SI-C-RGROUP_CLOSE-cfm	
Lower layer announcement of closing an existing Group for reception	SI-C-RGROUP_CLOSE-ind	No response required from upper layers

6.3.3.2 Group reception

For the dynamic reception of group data, the destination ST lower layers are assumed to operate (by default) in a non-promiscuous mode, where received data is only delivered to the upper layers in response to these Group Enable services.

NOTE 1: This SI-SAP group reception service is only intended to enable the upper layers to control the lower layer filtering. Additional filtering of IP multicast groups may be provided by the upper layers. IP filtering may be needed if (for example) the lower layers are operated in a more promiscuous mode or if any of the statically enabled groups are not wanted.

The group open primitive activates the lower layer indirectly using a given BSM_GID to define the wanted group. The lower layers should respond to this request by activating and configuring the relevant lower layer functions. This includes, for example, adjusting the lower layer configuration to receive and demodulate the wanted physical channel(s); and setting lower layer filters to extract the wanted logical channels from those physical channel(s).

NOTE 2: For example, for a DVB-RCS network this service could be used by the lower layers in the destination ST to resolve and activate the PID value and tuning parameters associated with the relevant Transport Stream Logical Channel.

6.3.4 Group send

6.3.4.1 Description

Group send services are used to associate an IPv4 Class D address, or an IPv6 multicast group address with a series of lower layer (satellite dependent) parameters. This service is used to configure the lower layers at the sending ST (i.e. the source of the data) to enable the wanted traffic to be submitted to the lower layers by the upper layer using the SI-C-UNITDATA-req primitives.

This service is for further study.

6.3.5 Flow control

6.3.5.1 Description

The flow control primitives are used to activate and adjust lower layer flow control for a specific QID.

NOTE: This function should support satellite specific flow control such as managing satellite subnetwork congestion in either the uplink beam(s), the spacecraft or the downlink beam(s).

Flow control activates the flow control for the relevant flow by associating flow control with the relevant QID: all data transfers for the given QIDs are then subject to flow control.

Flow control can be activated statically or dynamically:

a) **Static flow control** refers to the case where the flow control is associated with a QID and is activated via management configuration. The primitives defined in this clause shall not be used to activate or deactivate a static instance of flow control and any primitives that refer to a static instance of flow control should be ignored.

b) **Dynamic flow control** refers to the case where the flow control is associated with a QID and is configured using the flow control primitives defined in this clause. Dynamic flow control services can be initiated by either the upper layers or lower layers as follows:

   1) Flow control can be invoked by the satellite upper layers by issuing a SI-C-FLOW_ACTIVATE-REQ primitive.

   2) Flow control can be invoked by the lower layers by issuing a SI-C-FLOW_ACTIVATE-IND primitive.

Flow control can operate on any existing lower layer resource; i.e. on static or dynamic QIDs as defined in clause 6.3.3. The SI-SAP should support multiple independent flow controls at the same time. The upper limit for the number of simultaneous flows is a lower layer implementation option: flow control requests that exceed this limit may be rejected.

Flow control operates using a credit based allowance, where the flow control functions are required to regularly update the credit for the controllable flow(s).

The following flow control services are defined:

Table 8: Flow control services

Service	Primitives	Comments
Upper layer request to activate a new flow control	SI-C-FLOW_ACTIVATE-req SI-C-FLOW_ACTIVATE-cfm	
Lower layer request to activate a new flow control	SI-C-FLOW_ACTIVATE-ind SI-C-FLOW_ACTIVATE-res	
Upper layer request to modify an existing flow control	SI-C-FLOW_MODIFY-req SI-C-FLOW_MODIFY-cfm	
Lower layer request to modify an existing flow control	SI-C-FLOW_MODIFY-ind SI-C-FLOW_MODIFY-res	
Upper layer request to deactivate an existing flow control	SI-C-FLOW_DEACTIVATE-req SI-C-FLOW_DEACTIVATE-cfm	
Lower layer request to deactivate an existing flow control	SI-C-FLOW_DEACTIVATE-ind SI-C-FLOW_DEACTIVATE-res	

6.3.5.2 Flow control credit

The flow control scheme is a credit-based scheme where the flow control procedures prescribe an amount of data, in bytes, that a given QID (or QIDs) may send to the lower layers.

The minimum credit the lower layers shall provide for a particular flow corresponds to one SDU of data, or 1 500 bytes, whichever is the greater.

7 U-plane primitives

7.1 General

The U-plane SAP (SI-U-SAP) is used to submit packets to the SDAF for transmission via the space segment, and receive packets from the SDAF that have been received via the space segment.

The upper layers are required to know (e.g. through signalling and management communications), what SI-SAP services are actually available so that each submitted packet can be classified and submitted to a valid service (i.e. to a valid queue).

7.2 Data transport primitives

7.2.1 Description

The data transfer primitives are used to send and receive user data via the SI-SAP.

The source ST sends data by submitting it to the SI-SAP using the SI-U-UNITDATA-REQ primitive. The primitive includes the BSM_ID of the destination ST. The data is delivered at the SI-SAP in the destination ST in the SI-U-UNITDATA-IND primitive.

When sending data, the associated resource is identified using the Queue Identifier (QID). Any data transfer that contains an unknown or undefined value of QID shall be dropped without notification to the upper layers. This event should be noted for performance management purposes.

A given value of QID defines the permitted range(s) of destination BSM_IDs that can be used for data transfers using that QID. Any data transfer that contains values of BSM_ID that are inconsistent with the permitted range(s) for that QID should be dropped without notification to the upper layers. This event should be noted for performance management purposes.

The user data is transferred in the Service Data Unit (SDU) of the data transport primitives: each SDU shall contain a complete IP datagram.

NOTE: The SDU only defines the format at the SI-SAP. The lower layers may segment the IP packet as required for transmission over the air interface.

7.2.2 Primitives

7.2.2.1 SI-U-UNITDATA

The SI-U-UNITDATA primitives are used to send and receive one packet of user data.

SI-U-UNITDATA primitives may be used for unicast, multicast and broadcast data transport services.

The SI layer at the source ST sends a data packet using the SI-U-UNITDATA-REQ primitive.

The SI layer at the destination ST receives a data packet using the SI-U-UNITDATA-IND primitive.

Table 9: SI-U-UNITDATA primitives

PRIMITIVE NAME	SI-U-UNITDATA-***
FUNCTION	Send and receive data SDUs

Primitive parameters:	**-req**		**-ind**		**Comments**
Destination BSM_ID	M		M		Specific values of BSM_ID are used to distinguish between unicast, broadcast and multicast data transfers
QID	M		-		
SDU Type	M		M=		IPv4; IPv6 etc.
SDU	M		M=		

7.2.3 Parameters

7.2.3.1 BSM_ID

The destination address shall be a valid BSM_IDs; this may be either a unicast ID or a multicast ID. BSM_ID are defined in clause 5.2.1.

7.2.3.2 SDU Type

The SDU Type shall be a 16 bit parameter that conforms to the IEEE defined EtherTypes [6].

NOTE: For more information on using these registry values see the IANA reference in bibliography.

7.2.3.3 Queue IDentifier (QID)

The Queue Identifier (QID) is defined in clause 5.3. The QID identifies the lower layer resources that are to be used for this data transfer.

A valid QID must be provided by the sending ST for every data transfer (i.e. in every SI-U-UNITDATA-request primitive). The QID is allocated using the resource reservation procedures defined in clause 8.3.

8 C-plane primitives

8.1 General

The C-plane primitives are used to exchange control information between the SI-layers and the SD-layers for controlling the transmission and reception of U-plane packets.

8.2 Address resolution

8.2.1 General

Address resolution query primitives are used to enable the upper SI-layers to determine the appropriate destination BSM_ID for a given destination network address (IP address).

Address resolution information primitives are used to enable the SD-layer to supply unsolicited address resolution information.

8.2.2 Primitives

8.2.2.1 SI-C-AR_QUERY

The SI layer requests an Address Resolution update using the SI-C-AR_QUERY-REQ primitive containing the Network Address of the destination.

The SD layer should respond to the request using the SI-C-AR_QUERY-CFM primitive which contains the BSM_ID associated with that Network Address.

Table 10: SI-C-AR_QUERY primitives

PRIMITIVE NAME	SI-C-AR_QUERY-***
FUNCTION	Request and receive address resolution information
PRIMITIVE TYPES	Request, Confirm
PRIMITIVE PARAMETERS	For further study.

8.2.2.2 SI-C-AR_INFO

The SD layer may supply unsolicited AR information (i.e. Network Address/ BSM_ID pairs) at any time using the SI-C-AR_INFO-IND primitive. This primitive can be used to announce the existence of multicast groups.

The SI layer should respond to unsolicited AR information using the SI-C-AR_INFO-RES primitive to acknowledge receipt of the AR information.

Table 11: SI-C-AR_INFO primitives

PRIMITIVE NAME	SI-C-AR_INFO-***
FUNCTION	Receive and acknowledge unsolicited address resolution information
PRIMITIVE TYPES	Indicate; Response
PRIMITIVE PARAMETERS	For further study.

8.3 Resource reservation

8.3.1 General

The resource reservation services are used by the SI upper layer to dynamically request the lower layer to open new resources; or to request the lower layers to modify or close an existing resource reservation.

The resource reservation services are also used by the lower layers to dynamically announce the opening of unsolicited resources for use by the upper layers; or modify or close an existing resource reservation.

Resource reservation is used to open both unicast and multicast resources. In both cases, the associated reservation procedures are only required to be invoked at the sending side (i.e. the source ST). Resource reservation is not used for receiving data.

8.3.2 Resource reservation primitives

8.3.2.1 SI-C-QUEUE_OPEN

The SI upper layer requests a new lower layer resource using the SI-C-QUEUE_OPEN-REQ primitive, and the SD layer responds to the request using the SI-C-QUEUE_OPEN-CFM primitive.

The SD layer indicates the opening of an unsolicited lower layer resource with the SI-C-QUEUE_OPEN-IND primitive, and the SI layer responds using the SI-C-QUEUE_OPEN-RES primitive.

Table 12: SI-C-QUEUE_OPEN primitives

PRIMITIVE NAME	SI-C-QUEUE_OPEN-***
FUNCTION	Request or indicate the opening of a lower layer resource
PRIMITIVE TYPES	Request; Confirm; Indicate; Response
PRIMITIVE PARAMETERS	For further study.

8.3.2.2 SI-C-QUEUE_MODIFY

The SI upper layer requests a modification to an existing lower layer resource using the SI-C-QUEUE_MODIFY-REQ primitive, and the SD layer responds to the request using the SI-C-QUEUE_MODIFY-CFM primitive.

The SD layer indicates the unsolicited modification of an existing lower layer resource with the SI-C-QUEUE_MODIFY-IND primitive, and the SI layer responds using the SI-C-QUEUE_MODIFY-RES primitive.

Table 13: SI-C-QUEUE_MODIFY primitives

PRIMITIVE NAME	SI-C-QUEUE_MODIFY-***
FUNCTION	Request or indicate a modification to a lower layer resource
PRIMITIVE TYPES	Request; Confirm; Indicate; Response
PRIMITIVE PARAMETERS	For further study.

8.3.2.3 SI-C-QUEUE_CLOSE

The SI layer requests the SD layer to close a resource using the SI-C-QUEUE_CLOSE-REQ primitive. The SD layer responds to the request using the SI-C-QUEUE_CLOSE-CFM primitive.

The SD layer indicates an unsolicited (or exceptional) close of an existing resource with the SI-C-QUEUE_CLOSE-IND primitive, and the SI layer responds to acknowledge the close using the SI-C-QUEUE_CLOSE-RES primitive.

Table 14: SI-C-QUEUE_CLOSE primitives

PRIMITIVE NAME	SI-C-QUEUE_CLOSE-***
FUNCTION	Request or indicate the closing of a lower layer resource
PRIMITIVE TYPES	Request; Confirm; Indicate; Response
PRIMITIVE PARAMETERS	For further study.

8.4 Group receive control

8.4.1 General

The group receive control services are used by the SI upper layer to dynamically request the lower layer to activate the reception of a given multicast group.

The group receive control services are also used by the lower layers to dynamically announce the activation of unsolicited groups for use by the upper layers.

Group receive control is only used for activating the reception of multicast groups: it is not used for receiving unicast data.

8.4.2 Group receive primitives

8.4.2.1 SI-C-RGROUP_OPEN

Table 15: SI-C-RGROUP_OPEN primitives

PRIMITIVE NAME	SI-C-RGROUP_OPEN-***
FUNCTION	Request the lower layers to start receiving the specified group.
PRIMITIVE TYPES	Request; Confirm; Indicate
PRIMITIVE PARAMETERS	For further study.

8.4.2.2 SI-C-RGROUP_CLOSE

Table 16: SI-C-RGROUP_CLOSE primitives

PRIMITIVE NAME	SI-C-RGROUP_CLOSE-***
FUNCTION	Request the lower layers to stop receiving the specified group.
PRIMITIVE TYPES	Request; Confirm; Indicate
PRIMITIVE PARAMETERS	For further study.

8.5 Group send control

8.5.1 General

The group send control services are used by the SI upper layer to dynamically request the lower layer to activate the sending of a given multicast group.

This service is for further study.

8.6 Flow control

8.6.1 General

The flow control services are used by the SI upper layer to dynamically request the lower layer to activate flow control for an existing resource; or to request the lower layers to modify or deactivate an existing flow control.

Flow control operates on a credit/demand scheme. Before sending data, the upper layers are responsible for identifying and quantifying IP flows that will be controlled by flow control. The upper layers start by opening a flow control for the relevant flow by associating flow control with the relevant QID. Once activated, the upper layer issues periodic demands for bandwidth and the lower layers respond with bandwidth credits. The lower layers are required to update the credit limit for each controlled flow to maintain the permitted flow. The upper layers end the flow control by deactivating the relevant flow control for that QID.

Flow control can be used for both unicast and multicast resources. In both cases, the associated flow control procedures are only required to be invoked at the sending side (i.e. the source ST). Flow control is not used for receiving data.

8.6.2 Flow control primitives

8.6.2.1 SI-C-FLOW_ACTIVATE

The SI upper layer requests the lower layer to activate flow control on an existing resource using the SI-C-FLOW_ACTIVATE-REQ primitive, and the SD layer responds to the request using the SI-C-FLOW_ACTIVATE-CFM primitive. This initial response should include an initial allocation of credit.

The SD layer subsequently updates each existing flow control of an existing resource with the SI-C-FLOW_ACTIVATE-IND primitive, and the SI layer responds using the SI-C-FLOW_ACTIVATE-RES primitive.

Table 17: SI-C-FLOW_ACTIVATE primitives

PRIMITIVE NAME	SI-C-FLOW_ACTIVATE-***
FUNCTION	Request the lower layers to activate flow control on the specified flow.
PRIMITIVE TYPES	Request; Confirm; Indicate; Response
PRIMITIVE PARAMETERS	For further study.

8.6.2.2 SI-C-FLOW_MODIFY

The SI upper layer requests a modification to an existing flow control using the SI-C-FLOW_MODIFY-REQ primitive, and the SD layer responds to the request using the SI-C-FLOW_MODIFY-CFM primitive.

The SD layer indicates an unsolicited modification to an existing flow control with the SI-C-FLOW_MODIFY-IND primitive, and the SI layer responds using the SI-C-FLOW_MODIFY-RES primitive.

In all cases, the flow control parameters in the SI-C-FLOW_MODIFY primitives replace (overwrite) any existing Flow control parameters.

Table 18: SI-C-FLOW_MODIFY primitives

PRIMITIVE NAME	SI-C-FLOW_MODIFY-***
FUNCTION	Request the lower layers to modify flow control on the specified flow.
PRIMITIVE TYPES	Request; Confirm; Indicate; Response
PRIMITIVE PARAMETERS	For further study.

8.6.2.3 SI-C-FLOW_DEACTIVATE

The SI upper layer requests the lower layer to deactivate flow control on an existing resource using the SI-C-FLOW_DEACTIVATE-REQ primitive, and the SD layer responds to the request using the SI-C-FLOW_DEACTIVATE-CFM primitive.

The SD layer indicates the unsolicited deactivation of flow control on an existing resource with the SI-C-FLOW_DEACTIVATE-IND primitive, and the SI layer responds using the SI-C-FLOW_DEACTIVATE-RES primitive.

Table 19: SI-C-FLOW_DEACTIVATE primitives

PRIMITIVE NAME	SI-C-FLOW_DEACTIVATE-***
FUNCTION	Request the lower layers to deactivate flow control on the specified flow.
PRIMITIVE TYPES	Request; Confirm; Indicate; Response
PRIMITIVE PARAMETERS	For further study.

Annex A (informative): Guidance on the primitive definitions

A.1 General

Primitives are defined in the present specification using a concise table format.

This annex provides guidance on how these tables should be interpreted.

A.2 The sections of a primitive table

Each primitive is defined using a generic table as shown in table A.1.

The table has two main sections:

- The header section; comprising two rows which list the PRIMITIVE NAME and the FUNCTION.
- The body section; comprising a variable number of rows which list the PARAMETERs that are used. The body section also contains 4 columns which are used to show which TYPEs of primitive are used.

The PRIMITIVE NAME is the formal label that defines the primitive. The name is related to the function and it contains several elements separated by hyphens: a common prefix ("SI"); a second part which defines the specific SAP ("U", "C" or "M"); a main part which gives the unique NAME of the primitive; and the last part always contains the wildcard "***" to indicate that the table may apply to several different TYPES of primitive (see below).

The FUNCTION is an informative description of the function. The function is usually reflected in the choice of the unique NAME.

The body section of the table contains a variable number of columns and rows.

The columns in the body section are used to define up to 4 different types of primitive with the same PRIMITIVE NAME. The possible types are: REQUEST (req); CONFIRM (cfm); INDICATION (ind) and RESPONSE (res) as defined in clause 5.1. The TYPES of primitive that are applicable for a given NAME are indicated by presence or absence of the relevant columns in the body section of the table: an empty column means that TYPE is not valid (not used).

The rows in the body section are used to define the list of parameters. Each parameter may apply to one or more different TYPES of primitive: the relevant body columns indicate if a given parameter applies to that TYPE of primitive using 3 possible symbols: "M" if the parameter is mandatory; "O" if the parameter is optional and "-" if the parameter is not used.

Table A.1: Generic primitive

PRIMITIVE NAME	The formal NAME of the primitive
FUNCTION	A brief description of the function

Primitive parameters:	**-req**	**-cfm**	**-ind**	**-res**	**Comments**
	These 4 column headings indicate which TYPES of primitive are used. If a column heading is missing, it means that TYPE of primitive is not used				
First parameter	M	M	M	M	EXAMPLE#1: Column entries here indicate that this parameter is Mandatory and that this parameter applies to all TYPES.
Second parameter	O			O	EXAMPLE#2: Column entries here indicate that this parameter is Optional and only applies to two different TYPES of primitive
etc					

A.3 Worked example

To illustrate the rules defined above, we can look at the SI-U-UNITDATA primitive as define in clause 6.

This is reproduced in table A.2.

Table A.2: SI-U-UNITDATA primitives

PRIMITIVE NAME	SI-U-UNITDATA-***
FUNCTION	Send and receive data SDUs

Primitive parameters:	**-req**		**-ind**		**Comments**
Destination BSM_ID	M		M		Specific values of BSM_ID are used to distinguish between unicast, broadcast and multicast data transfers
QID	M		-		
SDU Type	M		M=		IPv4; IPv6 etc.
SDU	M		M=		

In the header we see the formal name: SI-U-UNITDATA-***. This is interpreted as follows:

- SI-U-xxxxxx; indicates that this primitive is related to the U-plane of the SI-SAP;
- UNITDATA; this is the unique name of the primitive. It implies the function of the primitive (i.e. data transfer): the following row gives a longer description of the function.

In the body we see that only two columns contain TYPE headings: the "req" column and the "ind" column. This means that only two TYPES of primitive are used: SI-UNITDATA-REQ (defined by the "req" column) and SI-UNITDATA-IND (the "ind" column). The other two type columns are empty: the other types are therefore not used.

The body section lists a total 4 different parameters for this primitive; one parameter per row. This is interpreted as follows:

- The first parameter (Destination BSM_ID) is defined as "M"; i.e. this parameter is mandatory for both the -REQ and -IND primitive types.
- The second parameter (QID) is defined as "M" for the -REQ primitive type and "-" for -IND primitive type; i.e. this parameter is Mandatory in the -REQ primitive and Absent in the -IND primitive.

- The third parameter (SDU Type) is defined as "M" for the -REQ primitive and "M=" for the -IND primitive; i.e. as well as being Mandatory, the value of this parameter in the -IND primitive is equal to the value in the corresponding -REQ primitive.
- The last parameter is the SDU. This is a special type of parameter that contains the payload for data transfer. This parameter is defined as "M" in the -REQ primitive and "M=" in the corresponding -IND primitive.

A.4 Using primitives to define procedures

A primitive represents just one step of a procedure. The complete procedure for starting, modifying and completing a given function will generally involve more than one primitive.

The complete procedure may require a series of different primitives: this may include different TYPES of primitive (with the same NAME) as well as different NAMES of primitive.

Annex B (informative): IEEE 802 [4] address format

This annex provides an overview of the IEEE 802 [4] address format that is used for the BSM_ID.

NOTE: IEEE 802 [4] 48 bit address formats are defined in the IEEE 802 Overview and Architecture volume [4].

A 48-bit universal IEEE 802 [4] address consists of two parts. The first 24 bits (i.e. octets 0, 1 and 2) correspond to the Organisationally Unique Identifier (OUI) as assigned by the IEEE, except that the assignee may set the LSB of the first octet to 1 for group addresses or set it to 0 for individual addresses. The second part, comprising the remaining 24 bits, (i.e. octets 3, 4 and 5) is administered by the assignee. This is illustrated below:

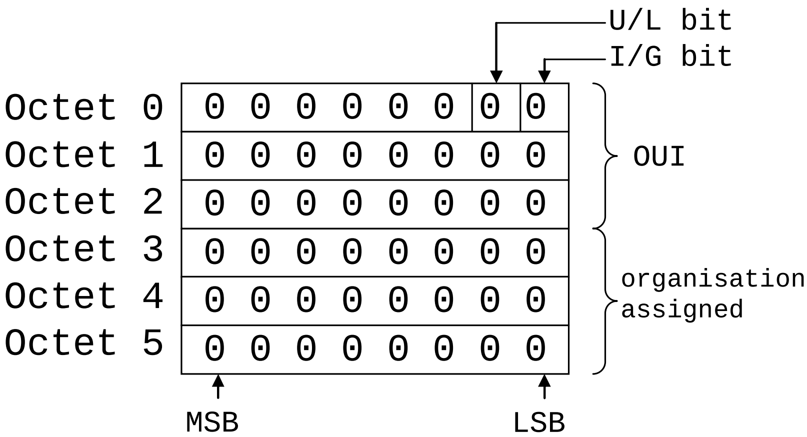

Figure B.1: IEEE 802 [4] Universal Address

The least significant bits of Octet 0 have special meanings as follows:

- The Individual/Group (I/G) address bit (LSB of octet 0) is used to identify the destination address as an individual address or a group address. If the I/G address bit is 0, it indicates that the address field contains an individual address. If this bit is 1, the address field contains a group address that identifies one or more (or all) stations connected to the LAN. The all-stations broadcast address is a special, predefined group address of all 1's.

- The Universally or Locally administered (U/L) address bit is the bit of octet 0 adjacent to the I/G address bit. This bit indicates whether the address has been assigned by a local or universal administrator. Universally administered addresses have this bit set to 0. If this bit is set to 1, the entire address (i.e. 48 bits) has been locally administered.

Annex C (informative): Bibliography

- ETSI TR 101 985 "Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia; IP over Satellite".
- IETF RFC 791: "Internet protocol" (STD 5).
- IETF RFC 2460: "Internet Protocol, Version 6 (IPv6) Specification".
- ETSI TS 102 295: "Satellite Earth Stations and Systems (SES); Broadband Satellite Multimedia (BSM) services and architectures; BSM Traffic Classes".
- IETF RFC 826: "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware".
- IANA defined ethernet numbers http://www.iana.org/assignments/ethernet-numbers

History

Document history		
V1.1.1	May 2005	Publication