INTERNATIONAL STANDARD

ISO 21218

Third edition
2018-06

Intelligent transport systems — Hybrid communications — Access technology support

Systèmes intelligents de transport — Communications hybrides — Support à la technologie d'accès

Reference number
ISO 21218:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

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 (see 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 (see 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 voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

This third edition cancels and replaces the second edition (ISO 21218:2013), which has been technically revised. It also incorporates the Amendment ISO 21218:2013/Amd 1:2014.

The main changes compared to the previous edition are as follows:

— ASN.1 has been aligned with the latest developments in ISO/TC 204;

— provisioning for path and flow management has been added;

— normative annex related to conformance testing, that contains the PICS proforma, has been added;

— management commands have been added;

— minor technical and editorial improvements.

Introduction

This document is part of a family of International Standards for communications in Intelligent Transport Systems (ITS) based on the ITS station and communication architecture specified in ISO 21217.

Hybrid communications, i.e. simultaneous support of different communication protocol stacks with different access technologies, is essential for ITS.

This document supports

— hybrid communications as requested, e.g. for Cooperative ITS in Europe, and

— path and flow management, enabling abstraction of applications from communications details.

It determines general technical details related to the access layer of an ITS station specified in ISO 21217 and illustrated in Figure 1 which are applicable to all or several access layer technologies. This includes especially:

— the IN-SAP offered to the ITS-S networking & transport layer for communication purposes;

— functions of the service primitives in the MI-SAP with reference to the generic services of the MI-SAP specified in ISO 24102-3.

The IN-SAP and the MI-SAP are presented in Figure 1. The specification of the SI-SAP is not within the scope of the present version of this document.

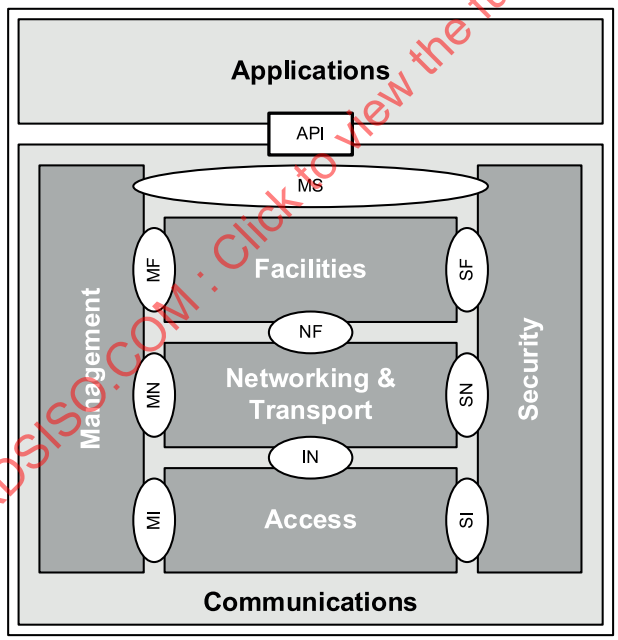

Figure 1 — ITS station reference architecture with named interfaces

This third edition cancels and replaces the second edition which has been revised and harmonized with newly developed C-ITS standards.

Intelligent transport systems — Hybrid communications — Access technology support

1 Scope

This document specifies general technical details related to the access layer of the ITS station reference architecture specified in ISO 21217 including:

— the service access point (SAP) of a communication interface (CI) as provided by the communication adaptation layer (CAL) for communication, named IN-SAP, and related service primitives and service primitive functions;

— the SAP provided by the CI management adaptation entity (MAE) for management of the communication interface, named MI-SAP, and related service primitives by reference to ISO 24102-3, and service primitive functions.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 8825-2, *Information technology — ASN.1 encoding rules: Specification of Packed Encoding Rules (PER) — Part 2*

ISO 21217, *Intelligent transport systems — Communications access for land mobiles (CALM) — Architecture*

ISO 24102-3, *Intelligent transport systems — ITS station management — Part 3: Service access points*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 21217, ISO 24102-3 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at http://www.electropedia.org/

— ISO Online browsing platform: available at https://www.iso.org/obp

3.1
(V)CI identifier
unique identifier of a (virtual) CI

3.2
communication interface
CI
instantiation of a specific ITS-S access layer technology and protocol

EXAMPLE An example of an ITS-S access layer technology and protocol is IR specified in[9].

3.3
medium
physical properties of a CI used to transmit a modulated signal, e.g. wireless or on a wire

Note 1 to entry: Medium is also referred to as access technology.

3.4
virtual communication interface
logical entity in a CI that is associated with a peer station

3.5
CI priority manager
logical entity in a CI that is managing priority queues

3.6
Link-ID
identifier of a link given by the address of a VCI

3.7
temps atomique international
time since 00:00:00 UTC, 1 January, 2004, identical with UTC except that no leap seconds need to be added

4 Abbreviated terms

APN	Access point name
BC-VCI	VCI for transmission to the broadcast MAC address
CAL	Communication adaptation layer
CEN	Comité Européen de Normalisation (European Committee for Standardization)
CI	Communication interface
CIC	Communication interface class
CIID	CI / VCI Identifier presented in a 64-bit EUI field
DLL	Data link layer
DNI	Distinct null identifier
DSRC	Dedicated short range communication (5,8 GHz back-scatter technology used in Europe for e.g. road tolling). This term is used in the USA to indicate IEEE 802.11 OCB communications in the 5,9 GHz band, see ISO 21215[10]
ETSI	European Telecommunications Standards Institute
EUI	ESxtended universal identifier
EUI-64	64-bit EUI
ICS	Implementation conformance statement
IN-SAP	Communication SAP as offered by the CAL to the ITS-S networking & transport layer
ITS-APDU	ITS station Access layer Protocol Data Unit
LocalCIID	CIID of a local CI

LSB	Least significant bit
MAC-48	48 bit MAC address
MAE	Management adaptation entity
MC-VCI	VCI for transmission to a multicast (group) MAC address
MI-SAP	Management SAP as offered by the ITS-S management towards the MAE
MSB	Most significant bit
OBU	On-board unit NOTE Term used for DSRC[16].
OSI	Open system interconnection
OUI	Organizational universal identifier
PIN	Personal identification number
RemoteCIID	CIID of a VCI enabling MAC groupcast transmissions and MAC unicast communication
RX/TX-CI	CI capable of operating in receive and transmit mode
RX-CI	CI capable of operating in receive mode only
RX-VCI	VCI for reception
SAE	Security adaptation entity
SIM	Subscriber identity module
SNAP	Sub-network access protocol
TAI	Temps Atomique International
TDMA	Time division multiple access
TX-CI	CI capable of operating in transmit mode only, either broadcast or multicast
TX-VCI	VCI for unicast transmission
UC-VCI	VCI for reception from and transmission to a unicast MAC address
UTC	Temps Universel Coordonné/Coordinated Universal Time
VCI	virtual communication interface
WAVE	Wireless access in vehicular environments NOTE IEEE work item related to [10].

5 Communication module adaptation

5.1 General

As ITS and the concept of an ITS station as a bounded secured managed domain (BSMD) specified in ISO 21217 does not only support access technologies (media) which are especially designed for implementations of ITS, there is a need to adapt the interfaces of these other access technologies to

those interfaces expected by the ITS networking & transport layer, the ITS-S management entity, and the ITS-S security entity.

For these other access technologies, the task is to adapt:

— the interface on top of the access technology to the IN-SAP by means of a communication adaptation layer (CAL);

— the management interface to the MI-SAP by means of a management adaptation entity (MAE);

— the security interface to the SI-SAP by means of a security adaptation entity (SAE).

The CI adaptation is outlined in Figure 2.

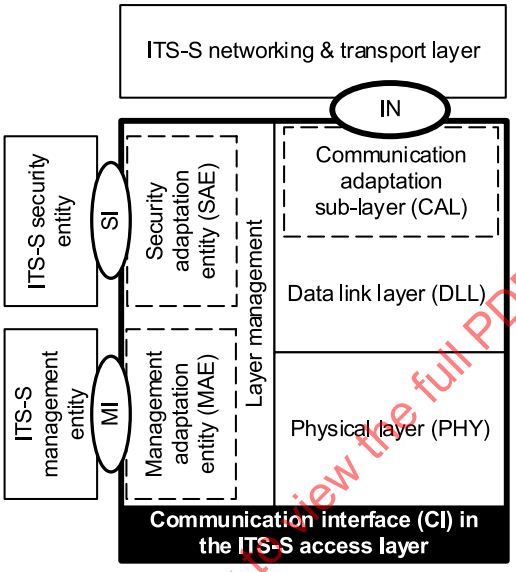

Figure 2 — Architecture

This document provides common basic functional specifications for the communication adaptation Layer, for the management adaptation entity (MAE), and for the security adaptation entity (SAE). It specifies the communication SAP (IN-SAP), the station management SAP (MI-SAP), and the security management SAP (SI-SAP).

5.2 Communication adaptation layer

The CIs built on different media are using the same ITS-S networking & transport layer. All CIs use the same type of IN-SAP between the ITS-S networking & transport layer and the CAL.

The medium-specific CAL provides an IN-SAP to the ITS-S networking & transport layer.

The CAL can be considered as an access technology (medium)-specific LLC or as an extension of an existing LLC providing the adaptation of the specific needs of an access technology (medium) to the common communication IN-SAP.

5.3 CI management adaptation entity

The CIs built on different media use the same ITS-S management, applying the functionality specified for the MI-SAP.

The MAE provides the MI-SAP to the ITS-S management following the same principles as outlined in ISO/IEC/IEEE 8802-11[2] with respect to the station management entity. The MI-SAP offers the services presented in Clause 9.

The MAE can be considered as medium-specific management providing the adaptation of the specific needs of an access technology (medium) to the common MI-SAP.

5.4 CI security adaptation entity

The current version of this document does not provide the specification of the SAE.

6 Communication interface

6.1 Architecture

This document uses the concept of

— Communication interface (CI), with

— virtual communication interfaces (VCIs).

A CI is a real communication equipment containing functionality of the ITS-S access layer. On top of a CI, one or several VCIs for transmission (TX-VCIs) to specific peer ITS-Ss, groups of ITS-Ss, or all ITS-Ss, and one or several receive VCIs (RX-VCIs), may be created.

NOTE The number of RX-VCIs is equal to the number of receive channels which can be operated simultaneously by the CI.

Further details on VCIs are specified in Clause 7.

6.2 Classification of CIs

6.2.1 CI Classes

CIs can be classified. Complementary classes shall be distinguished by means of CI classes presented in Table 1. A CI typically supports exctly one of the CI classes presented presented in Table 1. The set of applicable CI classes of a CI is given in I-Parameter CIclass of ASN.1 type `CIclass`. A single CI class is of ASN.1 type `CIclassSingle`.

NOTE New CI classes can be added as dynamic data, see Clause 11.

Table 1 — CI classes

Communication interface class	Definition and explanations
CIC-l1	CI that is capable of establishing simultaneous associations with different peer stations for unicast communication, and of receiving from and transmitting to broadcast and multicast (group) addresses. EXAMPLE Access technologies specified in [9],[10],[11]...
CIC-l2	CI that is capable of establishing a session via a single base station of a cellular network (typically to access Internet). Handover between different base stations may be possible, but not visible to the ITS upper layers and the ITS-S management. EXAMPLE Access technologies specified in [3],[7],[8]...
CIC-l3	CI that is capable to transmitting only on the basis of broadcast/multicast (group) addresses. EXAMPLE Access technologies specified in [9],[10],[11]...
CIC-l4	CI that is capable only of receiving frames from a broadcast station. EXAMPLE Satellite navigation receiver, satellite broadcast receiver, ...

Table 1 *(continued)*

Communication interface class	Definition and explanations
CIC-l5	CI that is capable only of performing communications between a car and a roadside station based on the master-slave principle with the roadside station being the master. Communication session establishment is done inside the CI. EXAMPLE Japanese DSRC, CEN DSRC, …
CIC-l6	CI for a cellular network technology that is capable of supporting one-to-many communications either with support of a base station of a cellular network (allocation of resources on a packet-per-packet basis) or without support of the base station (pre-configured resources). Example: LTE technologies specified in [4],[5].
CIC-il1	CI for station-internal network of an ITS station. Non-deterministic.
CIC-il2	CI for station-internal network of an ITS station. Deterministic.

6.2.2 CI access classes

Access to a remote station may require authentication, for example:

— PIN for a SIM card;

— operator data:

  — provider name;

  — APN;

  — user name;

  — password.

This is identified by means of complementary CI access classes, i.e. a CI supports exactly one of the CI access classes presented in Table 2. The applicable CI access classes of a CI is given in I-Parameter CIaccessClass of ASN.1 type `CIaClass`.

NOTE New CI access classes can be added as dynamic data, see Clause 11.

Table 2 — CI access classes

CI access class	Definition and explanations
CIAC-1	No user authentication required.
CIAC-2	CI requires access credentials, e.g. PIN and operator data.
CIAC-3	CI may require access credentials, e.g. PIN and operator data, dependent on the operational mode.

6.3 Link identifier

CIs and VCIs are referenced/addressed by a unique Link-ID. The Link-ID shall be constructed according to Figure 3.

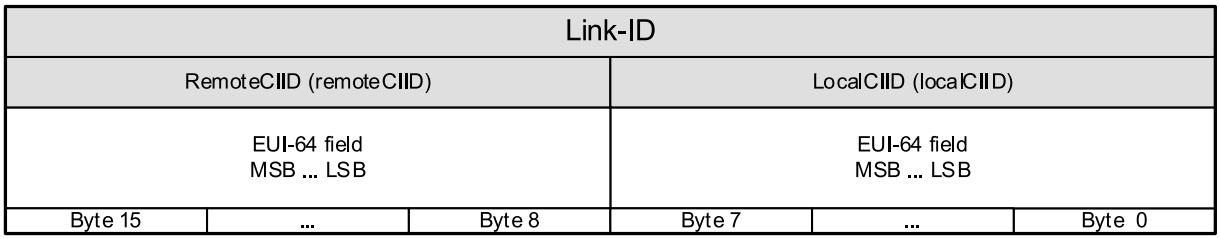

Figure 3 — Link-ID

The **LocalCIID** field identifies uniquely a specific CI in a specific ITS-S communication unit (ITS-SCU) in an instance of an ITS station.

NOTE A two octet ITS-SCU-ID specified in ISO 24102-4[13], identifying uniquely an ITS-SCU of an ITS-SU, can be derived from LocalCIID by means of a look-up table.

The **RemoteCIID** field identifies a VCI of the CI identified by LocalCIID which connects to a remote ITS-S unit (e.g. MAC unicast communication), or to a group of them (e.g. MAC broadcast or multicast communication). One reserved number of RemoteCIID identifies the CI which is addressed by the value of LocalCIID. This reserved number shall be

— the distinct null identifier (DNI) presented in D.2 for CIs supporting 48-bit MAC addresses;

— the "VCISerialNumber" field set to zero combined with the "UC/GC" field set to zero presented in D.3 for CIs which do not support 48-bit MAC addresses.

LocalCIID and RemoteCIID are presented in 64-bit global identifier (EUI-64) fields, see D.1, which may contain a 48-bit MAC address as illustrated in D.2.

For access technologies using 48-bit MAC addresses, LocalCIID may contain the globally unique MAC address of the CI, and RemoteCIID may contain either the individual MAC address reported in a received frame, or broadcast MAC address or a multicast MAC address.

Other access technologies use the numbering scheme specified in D.3.

The 48-bit MAC address is one example of a link layer address. Registered unique link layer addresses are presented in the medium-specific I-Parameter 8 "LLaddress". Locally assigned temporary values of a link layer address are stored in the I-Parameter 9 "LLaddressTemp".

6.4 CI procedures

6.4.1 General

The procedures as specified here use management services of the MI-SAP, as specified in Clause 9. The dynamic registration and deregistration procedures do not apply for CIs that are statically registered by implementation.

6.4.2 Registration

Registration of a CI at the ITS-S management is the process of making the CI known at the ITS-S management, and of making it addressable via a unique Link-ID. See the state machine in Figure 4.

The status of the CI before successful registration shall be CIstatus equal to "not existent".

A CI shall request registration of itself at the ITS-S management upon power-up, or upon physical insertion/activation of it. Two procedures are distinguished.

The following registration procedure shall be performed for CIs supporting 48-bit MAC addresses.

1) Create a Link-ID illustrated in Figure 3 with LocalCIID representing the globally valid unique MAC address of the CI as stored in I-Parameter **8** "LLaddress", with RemoteCIID equal to the "Distinct Null Indicator" (DNI) value presented in Annex C.

2) Send MI-REQUEST.request of ASN.1 type `RegistrationCI` indicating I-Parameter 14 "MedType" using the Link-ID constructed in step 1).

3) Set timer T_register to the value given in I-Parameter 7 "TimeoutRegister".

4) Await MI-REQUEST.confirm of ASN.1 type `RegistrationCIconf` providing the "ITS-SCU-ID" and "MedID" as long as T_register has not expired.

5) If the registration request was successfully performed, stop T_register and continue with the next step. If T_register had expired, start again with step 2).

6) Upon successful registration, set I-Parameter 4 "ITS-SCU-ID" as received in step 4. Set I-Parameter 12 "CIstatus" to the value "registered", and notify the new CIstatus value to the ITS-S management using MI-REQUEST.request of ASN.1 type `Event21218Notification` with event `E21218-5`.

The following registration procedure shall be performed for CIs not supporting 48-bit MAC addresses.

1) Create a preliminary Link-ID illustrated in Figure 3 with LocalCIID and RemoteCIID constructed as illustrated in Figure D.3 with:

   i) LocalCIID

      I) Set VCISerialNumber to the value zero, indicating the local CI.

      II) Set ITS-SCU-ID to the value zero, see ISO 24102-4[13].

      III) Set MedID to a value.

      IV) Set all bits in the UC/GC field to zero.

   The selected value of MedID might already be in use by another CI. Thus this value shall be confirmed by the ITS-S management entity in order to become valid.

   ii) RemoteCIID

      I) Set VCISerialNumber to the value zero, indicating the address of the CI.

      II) Set ITS-SCU-ID to the value zero.

      III) Set MedID to the same value as used in LocalCIID.

      IV) Set all bits in the UC/GC field to zero.

2) Send MI-REQUEST.request of ASN.1 type `RegistrationCI` indicating I-Parameter 14 "MedType".

3) Set timer T_register to the value given in I-Parameter 7 "TimeoutRegister".

4) Await MI-REQUEST.confirm of ASN.1 type `RegistrationCIconf` providing true values of "ITS-SCU-ID" and "MedID" as long as T_register has not expired.

5) If the registration request was successfully performed, stop T_register and continue with the next step. If T_register had expired, start again with step 1), using a different value for MedID.

6) Create the valid Link-ID of the CI using the values of ITS-SCU-ID, MedID as given by the ITS-S management in step 4).

7) Upon successful registration, set I-Parameter 4 "ITS-SCU-ID" and I-Parameter 5 "MedID"to the values received from the ITS station management. Set I-Parameter 12 "CIstatus" to the value "registered", and notify this value to the ITS management.

6.4.3 Deregistration

Deregistration of a CI at the ITS-S management is the reversal of the registration process of the CI. See the state machine in Figure 4.

Deregistration may be performed by the MAE or may be requested by the ITS-S management by sending the MI-COMMAND.request of ASN.1 type `CIstateChange` with the value "deregister".

Deregistration results in

— setting the ITS-SCU-ID to the value zero,

— deletion of all VCIs, and

— setting I-Parameter 12 "CIstatus" to the value "not existent".

The ITS- managament shall be notified of successful deregistration using the Link-ID as used for registration. Upon successful deregistration, the CI may be physically removed from the system.

6.4.4 Inactivation

Inactivation of a CI is the process to reset the CI and to block all subsequent communications. See the state machine in Figure 4.

Inactivation may be performed by the MAE or may be requested by the ITS-S management by sending the MI-COMMAND.request of ASN.1 type `CIstateChange` with the value "inactivate".

As a consequence of resetting a CI, no more VCIs exist, and thus there are no more pending data packets of the previously existing VCIs.

NOTE In a CI of class "CIC-l2" and access class "CIAC-2" such as specified in[7] or,[8] inactivation will result in disconnecting from the wireless service, i.e. ringing off.

The MAE shall set I-Parameter 12 "CIstatus" to the value "inactive" and shall notify the ITS-S management.

6.4.5 Activation

Activation of a CI is the process to enable communications in an inactive CI. See the state machine in Figure 4.

Activation may be performed by the MAE or may be requested by the ITS-S management by sending the MI-COMMAND.request of ASN.1 type `CIstateChange` with the value "activate".

This command shall trigger creation of VCIs as specified in 7.2.1.

The ITS-S management shall be notified of successful activation.

NOTE In a CI of class "CIC-l2" such as specified in [3],[7],[8] the state "active" indicates that the CI is within the communication zone of a base station and thus might connect to the service.

6.4.6 Suspension

Suspension of a CI is the process to put all communications of a CI on hold, without deleting any packets or state variables. See the state machine in Figure 4. A CI being in the state "suspended" shall still properly support the functionality of the primitives of the IN-SAP service IN-UNITDATA.

Suspension may be performed by the MAE or may be requested by the ITS-S management by sending the MI-COMMAND.request of ASN.1 type `CIstateChange` with the value "suspend".

All VCIs shall be maintained. No pending data packets shall be lost. An on-going frame transmission may be finalized. An on-going frame reception shall be finalized.

The MAE shall set I-Parameter 12 "CIstatus" to the value "suspended" and shall notify it to the ITS-S management.

6.4.7 Resuming

Resuming of a CI is the process to resume communications in a suspended CI. See the state machine in Figure 4.

Resuming may be performed by the MAE or may be requested by the ITS-S management by sending the MI-COMMAND.request of ASN.1 type `CIstateChange` with the value "resume".

The MAE shall set I-Parameter 12 "CIstatus" to the value "connected" and shall notify the ITS-S management. Pending packets shall be processed after resuming, if possible, otherwise pending packets may be deleted.

6.4.8 Connection

Connection of a CI is a process that depends on the CI access class. See the state machine in Figure 4.

For CI access class "CIAC-1" connection is established upon first usage of a TX-VCI or upon reception of a frame from a peer station.

For CI access classes "CIAC-2" connection is achieved upon confirmed establishment of a connection to the communication network. Connection may be requested by the ITS-S management by sending the MI-COMMAND.request of ASN.1 type `CIstateChange` with the value "connect".

The MAE shall set the I-Parameter 12 "CIstatus" to the value "connected" and shall notify the ITS-S management about the change of status.

6.4.9 Disconnection

Disconnection of a CI is a process that depends on the CI access class. See the state machine in Figure 4.

For CI access class "CIAC-1", disconnection is performed upon the event that no more TX-VCIs with a relation to a peer station are known.

For CI access classes "CIAC-2" and "CIAC-3" this is the termination of the connection to the communication network.

Disconnection may be requested by the ITS-S management by sending MI-COMMAND.request of ASN.1 type `CIstateChange` with the value "disconnect". There may be an implicit disconnection caused by deletion of a VCI.

The MAE shall set I-Parameter 12 "CIstatus" to the value "active" and shall notify the ITS-S management.

6.4.10 CI state machine

Figure 4 shows the state machine of a CI. It covers:

a) the start and end state:

   1) not_existent;

b) the interim states:

   1) existent;

   2) registered;

c) the operational states:

   1) active;

   2) connected, not applicable for receive-only CIs (CIC-l4);

d) the non-operational states, not applicable for receive-only CIs (CIC-l4):

   1) suspended;

   2) inactive;

   see I-Parameter 12 "CIstatus". The transitions between the states are:

   — power on/activate, see 6.4.2;

   — register, see 6.4.2;

   — deregister, see 6.4.3;

   — create VCI, see 7.2.1;

   — inactivate, see 6.4.4;

   — activate, see 6.4.5;

   — suspend, see 6.4.6;

   — resume, see 6.4.7;

   — connect, see 6.4.8;

   — disconnect, see 6.4.9;

   — delete VCI, see 7.2.3.

Requests to perform invalid transitions shall be acknowledged with error code `ErrStatus`="INVALID COMMAND/REQUEST VALUE" specified in ISO 24102-3.

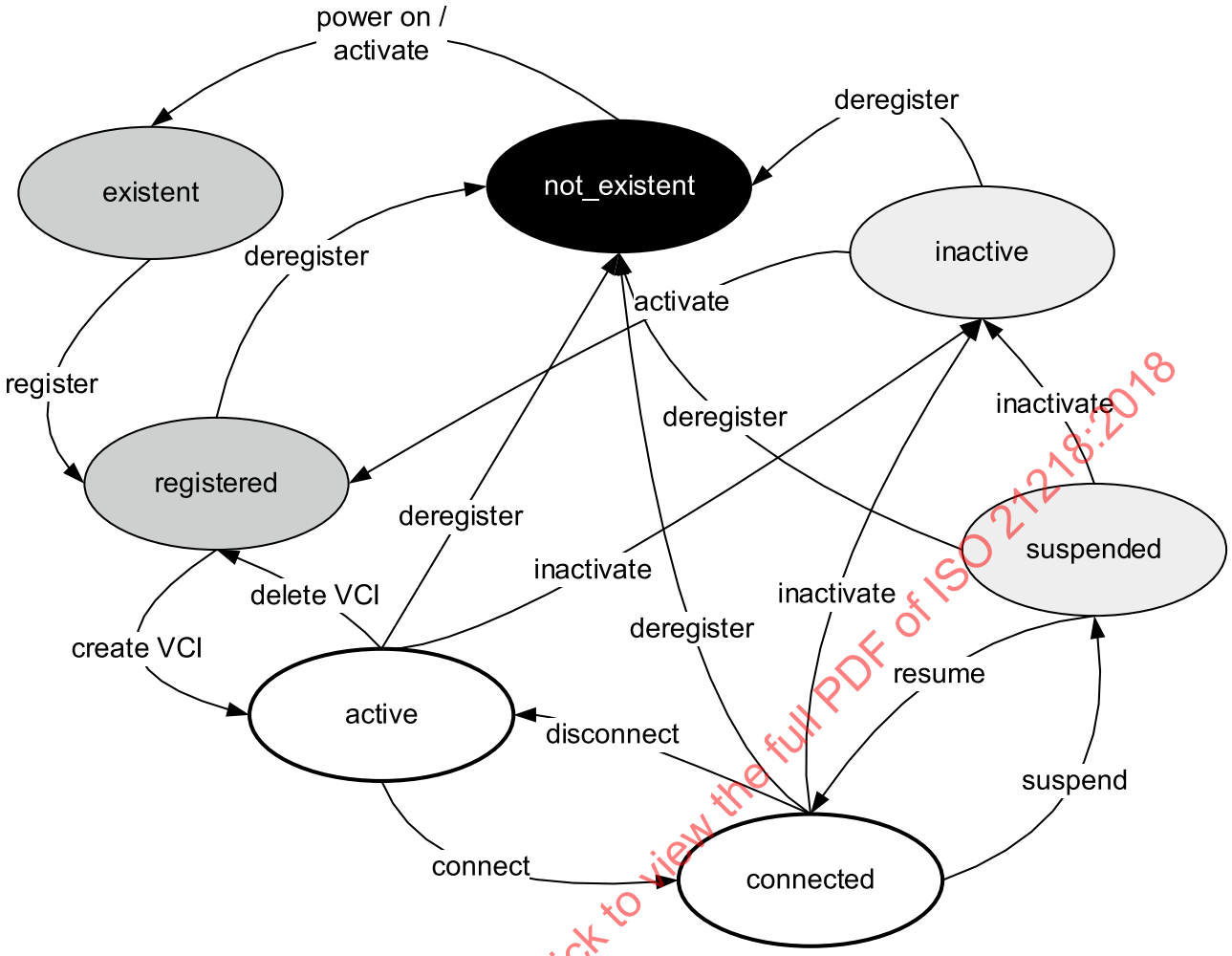

Figure 4 — CI state machine

6.4.11 Cross-CI prioritization

6.4.11.1 General

Wireless TX-VCIs in an instance of an ITS station might suffer from cross-interference. The procedure to synchronize transmission of multiple CIs based on user priority is called "Cross-CI prioritization". Cross-CI prioritization considers the case in which at least two local TX-VCIs need to be synchronized in order to avoid cross-interference.

The design and integration goal is to avoid such cross-interference to the greatest possible extent. A possible means of achieving this is proper assignment of allowed wireless communication channels to the CIs.

Priority management across CIs requires involvement of the ITS-S management for every packet to be prioritized.

The "Cross-CI prioritization" procedure is optional.

6.4.11.2 Registration of CI for prioritization REQUEST

A CI may register itself at the ITS-S management for the cross-CI prioritization procedure. This registration shall include

— the types of potentially interfering media, see I-Parameter 14 "MedType", and

— the prioritization timeout in milliseconds.

Registration for cross-CI prioritization is requested with the MI-REQUEST.request of ASN.1 type `PrioritizationRegistration`.

It is assumed that potentially interfering media are known *a priori* by the CI. Settings may be made by the manufacturer of the victim device. Settings may be overruled by the ITS-S management.

6.4.11.3 Prioritization request

Once registration for the cross-CI prioritization procedure is performed, intended prioritized transmission of a packet with a given high priority is notified via the ITS-S management to potential interferer CIs by means of a "dummy transmission request". Such dummy transmission requests are presented with MI-REQUEST.request of ASN.1 type `PrioritizationRequest` to the ITS-S management. The minimum required priority is specified in the I-Parameter `31`"MinPrioCrossCI".

— `PrioritizationRequest.seqNo` shall be set to a value unique for this CI;

— `PrioritizationRequest.priority` shall be set equal to the user priority of the pending packet.

NOTE The ITS-S management accepts a prioritization request only if PrioritizationRequest.priority is at least equal to MinPrioCrossCI.

Upon transmission of the dummy transmission request, the CI may start a timer T_DummyAckReq for this request.

In the case of protection only (see 6.4.12) the CI may try immediately to perform the intended transaction without awaiting receipt of an acknowledgement, if it will not cause interference to other CIs in this ITS station.

Otherwise, upon reception of the acknowledge MI-REQUEST.confirm of ASN.1 type `PrioritizationRequestConf` from the ITS-S management with:

— `PrioritizationRequestConf.seqNo` equal to the related request;

— `PrioritizationRequestConf.priority` equal to the granted priority;

— `PrioritizationRequestConf.status` equal to "granted";

the CI shall send the pending packet, and shall cancel the timer T_DummyAckReq.

If the acknowledgement shows `.status` equal to "ignored", the CI may either send the pending packet or delete it. The CI shall cancel the timer T_DummyAckReq. The MAE shall set I-Parameter 31 "MinPrioCrossCI" equal to the value provided in `.priority`.

Upon expiration of the timer T_DummyAckReq, if applicable, the CI may either send the pending packet or may delete it.

6.4.11.4 Prioritization release

Upon transmission or deletion of the pending packet, the CI shall release the prioritization request to the ITS-S management. An appropriate MI-REQUEST.request service primitive is of ASN.1 type `PrioritizationRelease`:

— `PrioritizationRequest.seqNo` shall be set equal to the value used in the prioritization request.

No acknowledgement is needed. An empty acknowledgement that may be used is of ASN.1 type `PrioritizationReleaseConf`.

The CI priority manager shall continue to serve the priority queues.

6.4.11.5 Interferer procedures

The information on a prioritization request, as contained in MI-REQUEST.request of ASN.1 type `PrioritizationRequest`, shall be used in the priority queue of the potential interferer CI. All potential interferers shall be notified by the ITS-S management. An appropriate MI-COMMAND.request is of ASN.1 type `PrioritizedRTS`:

— `PrioritizedRTS.seqNo` shall be set equal to the value of the related request from the victim VCI;

— `PrioritizedRTS.reqID` shall be set equal to LocalCIID of the related request from the victim VCI;

— `PrioritizedRTS.priority` shall be set equal to the granted priority of the related request from the victim VCI.

Once such a dummy entry in the priority queue is subject to transmission

— the dummy transmission request shall be acknowledged by means of MI-COMMAND.confirm of ASN.1 type `PrioritizedRTSConf` thus,

  — `PrioritizedRTSConf.reqID` shall be set equal to LocalCIID of the related request,

  — `PrioritizedRTSConf.seqNo` shall be set equal to the value of the related request, and

  — `PrioritizedRTSConf.status` shall be set to "granted", then

— the transmitter shall be disabled or tuned to an appropriate interference mitiagtion mode, and a timer T_dummyAckGrant for this request shall be started,

— the CI priority manager shall await either time out of T_dummyAckGrant or shall await release of this dummy transmission request by means of the MI-COMMAND.request of ASN.1 type `PrioritizedRTSrelease`, with the parameter set as follows:

  — `PrioritizedRTSrelease.seqNo` shall be set equal to the value of the related request from the victim VCI;

— the CI priority manager shall then delete the dummy transmission request from the queue and shall continue serving the priority queues in the previously used mode, and

— may acknowledge the release notification with `PrioritizedRTSreleaseConf`.

6.4.12 Protection of CI

Wireless transmitters and receivers integrated in an instance of an ITS station may suffer from cross-interference. Depending on user priorities, the interfering local CI transmitters are disabled for a defined period. This is called "Protection of CI".

NOTE An example of a CI in need of protection is a CEN DSRC OBU as widely used in payment and access control systems.

The design and integration goal is to avoid such cross-interference to the greatest possible extent.

"Protection of CI" shall make use of the "Cross-CI Prioritization" procedure.

Independent of the protection status, the CI may try to perform an intended communication at any time, unless it is requested to disable its transmitter due to a transmission request of a packet with higher priority, announced by means of the Cross-CI prioritization procedure.

The procedure for protection of a CI may be hard-wired in an implementation.

6.4.13 Regulatory information management

If regulation limits the capabilities of an access technology (medium), the CI controls proper settings of parameters in line with applicable regulation. The regulations could depend on, for example,

— the geographical location of the ITS station, or

— the legal type (owner) of the ITS station.

In general it is possible to get updates of the regulatory information with the following three options.

a) A CI may continuously monitor reception of frames with regulatory information, if applicable, in order to have an up-to-date regulatory information list.

b) Regulatory information may be derived from a local regulatory database located inside the CI using latitude and longitude of the geographical location of the ITS station. The geographical location of the ITS station is provided by a positioning service, e.g. ISO 21176[18], via the ITS-S management, otherwise directly from a positioning unit being integrated within the CI. The CI may request to receive updates of the kinematic vector, see I-Parameter 44 "KinematicVectorIn", containing latitude and longitude information by means of MI-REQUEST.request of ASN.1 type `PositionUpdateRequest`, which defines the update interval and activates/disables updates.

c) Regulatory information may become available to the ITS-S management by various means. Upon availability of new regulatory information, the ITS-S management presents it with MI-COMMAND. request of ASN.1 type `RIprovisioning`. Upon successful reception of an applicable regulatory requirement, the CI shall adopt the regulatory requirement.

NOTE Attempts of the ITS-S management to set parameters of the CI such that regulation would be violated can be accepted as the CI assumes being part of a BSMD, i.e. it trusts the ITS-S management.

Regulatory information is part of the"dynamic data" identified in Clause 11.

The details of regulatory information data content and format are outside the scope of this document; such information is part of the dynamic data, see Clause 11.

7 Virtual communication interface

7.1 Concept

The concept of a virtual communication interface (VCI) provides a quick and efficient method to set the properties of a CI on a packet-by-packet basis without the continuous involvement of the ITS-S management.

A TX-VCI is a software instance on top of a CI which is used for transmission towards either

— a specific receiver (UC-VCI), applying MAC unicast communications, or

— all possible receivers (BC-VCI), applying MAC broadcast communications, or

— a group of possible receivers (MC-VCI), applying MAC multicast communications.

The concept of a CI with TX-VCIs in relation to other instances of an ITS station is illustrated in Figures 5, 6 and 7. It is assumed, that in these figures only CIs of the same access technology (MedType *x*) are presented. Note that these figures do not show all functional blocks of an ITS station specified in ISO 21217.

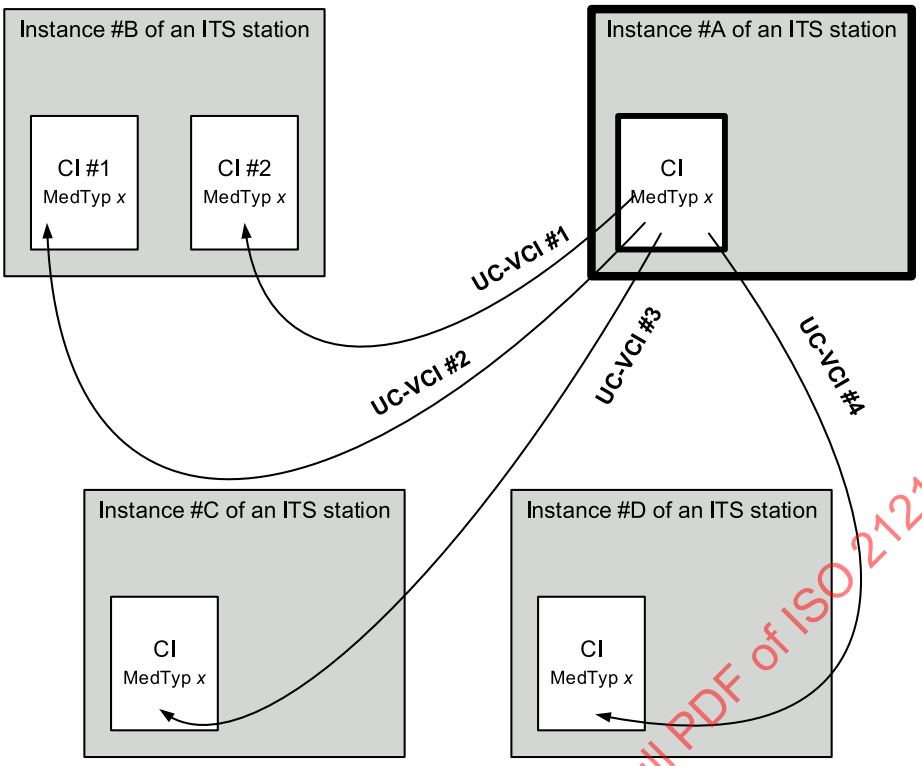

Figure 5 — UC-VCIs of instance #A of an ITS station

The CI in instance #A of an ITS station presented in Figure 5 maintains unicast links to four receivers installed in three instances of an ITS station, i.e. there are four UC-VCIs available, connecting to the RX-VCIs of the three other instances of an ITS station.

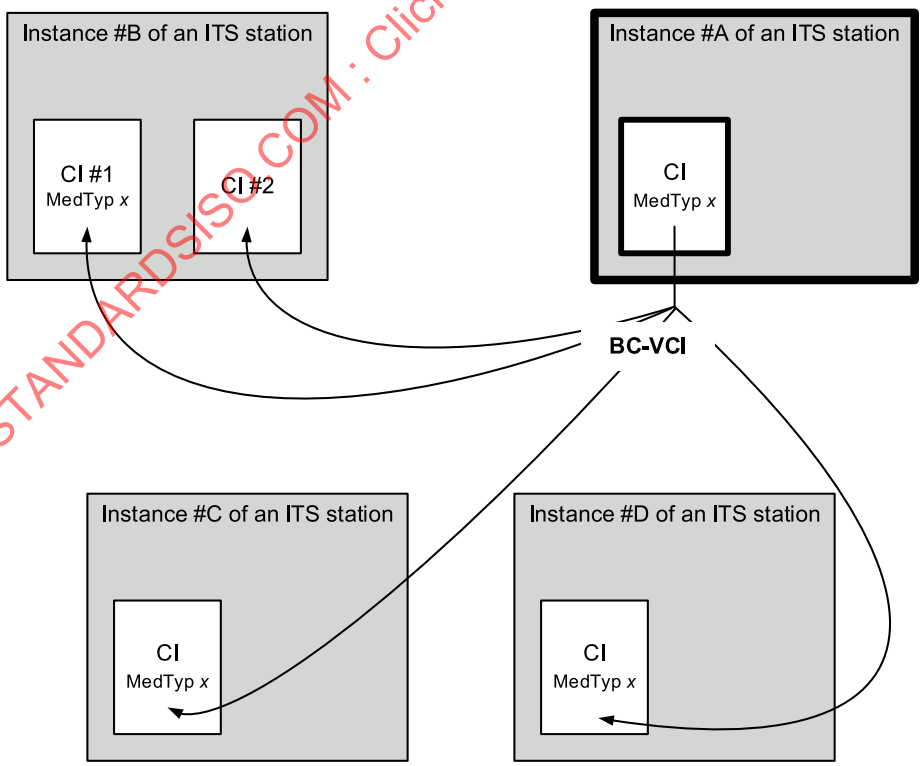

Figure 6 — BC-VCI of instance #A of an ITS station

The CI in instance #A of an ITS station presented in Figure 6 maintains one broadcast link, which in this given scenario leads to four receivers installed in three instances of an ITS station, i.e. there is one BC-VCI available, connecting to the RX-VCIs of the three other instances of an ITS station.

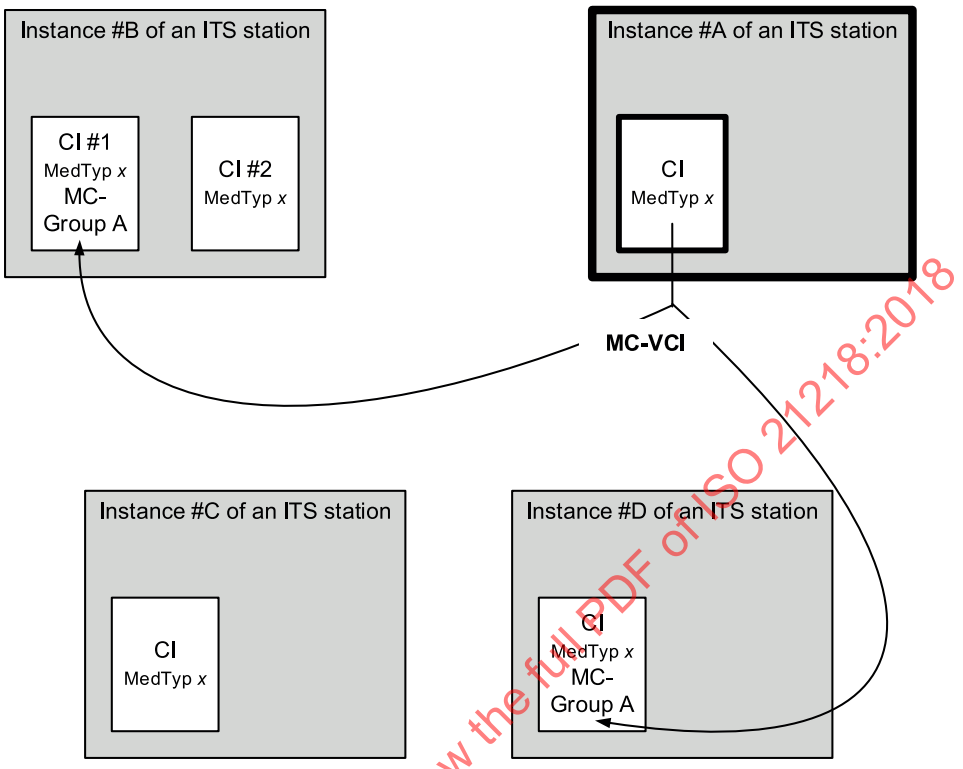

Figure 7 — MC-VCI of instance #A of an ITS station for MC-group A

The CI in instance #A of an ITS station maintains one multicast link for the MC-group A, which in this given scenario leads to two receivers installed in two instances of an ITS station, i.e. there is one MC-VCI available, connecting to RX-VCIs of two other instances of an ITS station. The other CIs also receive the related multicast frames, but these are ignored due to an invalid multicast address.

Each VCI may maintain its own set of I-Parameters in order to allow for an automatic packet-by-packet switching of transmit parameters without involving the ITS-S management, and without the need of additional parameters used in the IN-SAP service primitives. Different I-Parameter settings may apply for every association with a specific peer station, i.e. for every TX-VCI. Access to a TX-VCI may require a minimum user priority.

Each RX/TX-CI shall maintain at least one RX-VCI for reception of frames. The RX-VCI of a CI is shared by all of its TX-VCIs. There shall be one TX-VCI for transmission to the MAC broadcast address (BC-VCI), if applicable. There may be multiple TX-VCIs for transmission to MAC multicast (group) addresses (MC-VCI), if applicable. There is a single UC-VCI for every known peer CI.

NOTE Broadcast address and multicast addresses are referred to as groupcast addresses.

TX CIs, i.e. groupcast transmitters, contain BC-VCIs and/or MC-VCIs only. There is no RX-VCI.

RX CIs, e.g. groupcast or satellite positioning receivers, contain only one or several RX-VCIs. The number of RX-VCIs is equal to the number of receive channels which can be simultaneously operated.

Figure 8 explains how virtual entities of a CI and user priority are to be handled. After priority checking is performed by the CI priority manager, the CI I-Parameters are set to the values of the TX-VCI I-Parameters valid for the selected packet.

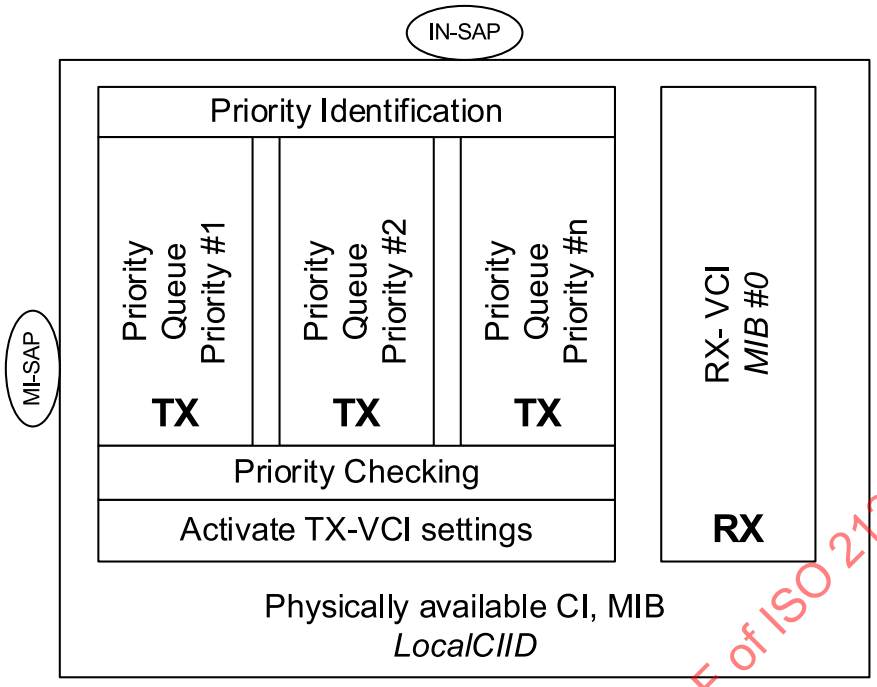

Figure 8 — CI priority queues and virtual communication interfaces

Each TX-VCI has its own set of I-Parameters. The differences between the I-Parameters of different VCIs of the same CI are only in

a) the TX-parameters,

b) the MAC addresses of the peer stations, and

c) the locally administered MAC address.

The CI shall use the transmit parameter settings as provided by the I-Parameters of the active VCI. These settings may be overruled by explicit change requests on a packet-by-packet basis, see "access parameters" in IN-UNITDATA.request and IN-UNITDATAACK.request in 8.2.1. Note that in simple implementations the transmit parameters of all VCIs of a CI may be identical.

User priority shall be handled according to the priority parameter provided by the ITS-S networking & transport layer in the service primitive for communication, e.g. IN-UNITDATA.request, see Clause 8. Priority queues for transmission shall be maintained either in the CAL, the LLC or the MAC of the CI. There shall be only a single set of queues for all TX-VCIs of a CI.

Multiple CIs of the same type, i.e. same medium, are allowed. If these CIs guarantee real simultaneous operation, these CIs may use the same or different priorities without a common priority checking entity. If these CIs operate simultaneously on interfering channels, a cross-CI prioritization mechanism may be applied.

NOTE Further on, it is allowed to provide a medium-specific bridge. This bridge can cover all CIs of the same type. Details are outside the scope of this document.

7.2 VCI procedures

7.2.1 Creation of VCI

7.2.1.1 General

Creation of VCIs either is performed by the MAE upon registration or activation of a CI, or upon request of the ITS-S management.

The procedure to create VCIs upon registration of a CI depends on the CI class.

Upon request from the ITS-S management by means of MI-COMMAND.request of ASN.1 type `VciCmd` with

— `VciCmd.linkID` presenting the Link-ID for the new VCI,

— `VciCmd.alive` presenting the value "true",

the MAE shall create a VCI, e.g. UC-VCI or MC-VCI, for a remote MAC address as requested in `.linkID`. The I-Parameters of this new VCI use default values in line with the settings of the shared RX-VCI, if applicable. Subsequently the actual set of I-Parameters of the new VCI may be modified by the ITS-S management.

NOTE A CI can have an RX-VCI operating on a different channel than the TX-VCIs. A CI can also have several RX-VCIs, operating on different channels.

The ITS-S management shall be notified about Creation of a VCI by means of MI-REQUEST.request of ASN.1 type `Event21218Notification`, reporting the Link-ID of the newly created VCI.

7.2.1.2 CIC-I1

Upon registration of an RX/TX-CI of CI class "CIC-l1", a BC-VCI and an RX-VCI shall be created by the MAE based on the default set of I-Parameters. I-Parameter 12 "CIstatus" shall be set to "active". Upon transmission request to a so far unknown peer station (i.e. unknown UC-VCI), a UC-VCI shall be created. The Link-ID.RemoteCIID of the UC-VCI shall identify the peer station as illustrated in D.2 or D.3, respectively. The ITS-S management shall be notified about the value of the MAC address or SerialNumber, respectively.

NOTE Different to the requirement in the previous edition of this document, due to the expected high rate of received messages from so far unknown peer stations, e.g. caused by CAM transmissions where a reply is never expected, UC-VCIs are not created upon reception of a messages from so-far unknown peer stations.

7.2.1.3 CIC-I2

Upon registration of an RX/TX-CI of CI class "CIC-l2", the CI shall create an RX-VCI and shall continuously monitor for a base station providing possible access to the communication network. Once a base station is identified that may provide access to the communication network, respective TX-VCIs (one or several dependend on the access technology and the supported communication modes) shall be created by the MAE based upon the default set of I-Parameters. I-Parameter 12 "CIstatus" shall be set to "active". Such a CI shall connect to the service automatically if it is of CI access class "CIAC-1" or "CIAC-3". If it is of CI access class "CIAC-2", initiation of connection depends on I-Parameter 16 "Connect". If "Connect" is set to "manual" it shall await a request from ITS-S management to connect by means of MI-COMMAND. request of ASN.1 type `CnConnect`. If "Connect" is set to "automatic" it shall connect automatically upon reception of the first data transmission request IN-UNITDATA.request.

7.2.1.4 CIC-I3

Upon registration of a TX-CI of CI class "CIC-l3", a BC-VCI shall be created by the MAE based on the default set of I-Parameters. I-Parameter 12 "CIstatus" shall be set to "active". Upon transmission request to a so far unknown multicast destination, an MC-VCI shall be created. The Link-ID.RemoteCIID of the

MC-VCI shall identify the peer destination as illustrated in D.2 or D.3, respectively. The value of the MAC address or SerialNumber, respectively, shall be notified to the ITS-S management.

7.2.1.5 CIC-I4

Upon registration of an RX-CI of CI class "CIC-l4", an RX-VCI shall be created by the MAE based on the default set of I-Parameters. I-Parameter 12 "CIstatus" shall be set to "active".

7.2.1.6 CIC-I5

Creation of VCIs does not apply.

7.2.1.7 CIC-I6

See CIC-l2 in 7.2.1.3.

7.2.1.8 CIC-iI1

Upon registration of an RX/TX-CI of CI class "CIC-il1", a BC-VCI and an RX-VCI shall be created by the MAE based on the default set of I-Parameters. I-Parameter 12 "CIstatus" shall be set to "active". Upon reception of a frame from a so far unknown peer unit (i.e. unknown UC-VCI), a UC-VCI shall be created. The Link-ID.RemoteCIID of the UC-VCI shall identify the peer unit as illustrated in D.2 or D.3, respectively. The value of the MAC address or SerialNumber, respectively, shall be notified to the ITS-S management.

7.2.1.9 CIC-iI2

Upon registration of an RX/TX-CI of CI class "CIC-il2" the procedure specified in 7.2.1.7 shall be performed.

7.2.2 Reset of VCI

Upon reset request by means of MI-COMMAND.request of ASN.1 type `VciCmd` with `.alive` set to "false" all state variables of the selected VCI and all related pending packets and frames shall be deleted. A relation to a peer station, if applicable, shall remain. All VCI I-Parameters shall be set to default values, if applicable.

Successful reset of a VCI shall be notified to the ITS-S management by means of MI-REQUEST.request of ASN.1 type `Event21218Notification`.

7.2.3 Deletion of VCI

The MAE may delete a UC-VCI in case no frames were received from the related peer station within a time span given by I-Parameter 13 "InactivityTimeLimit", see also Table 5. The time limit counter shall start on reception of every frame received.

Deletion of a VCI shall be notified to the ITS-S management by means of MI-REQUEST.request of ASN.1 type `Event21218Notification`.

Upon request from the ITS-S management, the MAE shall delete a VCI by means of MI-COMMAND. request of ASN.1 type `VciCmd` with `.alive` set to "false".

7.2.4 Association of peer with Link-ID

Packets received from a peer station are identified by means of the Link-ID of the related UC-VCI for this peer station.

Details on assignment of values and relation to MAC addresses are specified in Tables 3 and 4.

NOTE A CI can change its locally administered MAC address for the purpose of privacy. Procedures on how to manage a change of MAC address are medium-specific and outside the scope of this document. A change of MAC address without notification of the peer station will thus result in creation of a new VCI and termination of on-going communications at the higher OSI layers based on the old MAC address.

The MAE maintains information specified in the following Table 5 with the mappings of peer MAC address and related Link-ID together with time of last reception of a frame.

Table 5 — Peer-list

Peer MAC address	Link-ID	Time of last reception
As received from peer station.	Complete Link-ID according to Figure 3, and annexes D.2 and D.3, respectively.	ASN.1 `Time48IAT` with at least one byte for fractional seconds. See I-Parameter 25 "TimeOfLastReception" in Annex A.

The time of last reception may be used to estimate whether a peer station has left the communication zone. The time of last reception shall be initialized with time of creation of the VCI.

7.2.5 Change of I-Parameter settings

The CI may automatically change I-Parameter settings according to rules that are specific to the medium.

The ITS-S management may request change of I-Parameter settings according to rules specified in ISO 24102-1[12].

8 Communication SAP

8.1 Addressing

8.1.1 IN-SAP nt_protocol_id address

The IN-SAP *nt_protocol_id* address element of the IN-SAP service primitives specified in 8.2 contains an EtherType address identifying the protocol in the ITS-S networking & transport layer that uses the services of the IN-SAP. The address element *nt_protocol_id* is of ASN.1 type `Ethertype`.

The protocol in the ITS-S networking & transport layer shall be identified by a globally unique Ethertype value.

Allocated EtherType values are published at [21].

8.1.2 IN-SAP source and destination addresses

The IN-SAP *source_address* and *destination_address* of the IN-SAP service primitives specified in 8.2 contain a Link-ID identifying the ITS-SU that generated the packet and the ITS-SU(s) for which the packet is intended. *source_address* and *destination_address* are of ASN.1 types `SourceLinkID` and `DestinationLinkID`, respectively, specified in B.2.

Tables 8 and 9 specify the mapping of MAC addresses on Link-ID in source_address and destination_address, see also D.2. In case usage of MAC addresses in Link-IDs is not allowed or not possible, D.3 provides information on how to generate replacements for MAC addresses.

Table 3 — Address parameters in e.g. IN-UNITDATA.request for CIs using 48-bit MAC addresses

Source MAC address	Destination MAC address	RemoteCIID of source_address	RemoteCIID of destination_address	LocalCIID
Private MAC_{local} of transmitting local CI	BC-MAC	DNI, or ID containing UC-MAC address MAC_{local}	ID containing BC-MAC address	ID of CI to be used for transmission of the frame
	MC-MAC (MAC group) MAC_{group}		ID containing MC-MAC address MAC_{group}	
	Private MAC_{peer} of peer CI		ID containing UC-MAC address MAC_{peer}	

Table 4 — Address parameters in e.g. IN-UNITDATA.indication for CIs using 48-bit MAC addresses

Source MAC address	Destination MAC address	RemoteCIID of source_address	RemoteCIID of destination_address	LocalCIID
Private MAC_{peer} of transmitting peer CI	BC-MAC	ID related to MAC_{peer}	Containing BC-MAC address	ID of the CI that received the frame
	MC-MAC (MAC group) MAC_{group}		Containing MC-MAC address MAC_{group}	
	Private MAC_{local} of local CI		DNI, or ID containing UC-MAC address MAC_{local}	

8.2 Service primitives

8.2.1 IN-SAP functions

IN-SAP service primitives are used to exchange ITS-ASDUs between the ITS-S access layer and the ITS-S networking and transport layer specified in ISO 21217, distinguishing request and indication service primitives.

The available set of *request* service primitives:

— IN-UNITDATA.request;

— IN-UNITDATAACK.request;

is identified with the ASN.1 type `INsapRequestSps`. A single request service primitive is of ASN.1 type `INsapRequest`.

The available set of *indication* service primitives:

— IN-UNITDATA-STATUS.indication;

— IN-UNITDATA.indication;

is identified with the ASN.1 type `INsapIndicationSps`. A single request service primitive is of ASN.1 type `INsapIndication`.

The *request* service primitives and *indication* service primitives are part of the "dynamic data" identified in Clause 11. The functionality of the service primitives IN-UNITDATA.request, IN-UNITDATA-STATUS. indication, and IN-UNITDATA.indication shall be supported. The functionality of the service primitives IN-UNITDATAACK.request shall be supported for those access technologies that provide MAC acknowledgement.

8.2.2 IN-UNITDATA.request

8.2.2.1 Function

The service primitive IN-UNITDATA.request is used for transmission of data.

8.2.2.2 Semantics of the service primitive

```
IN-UNITDATA.request     (
                        addresses
                                (- flowID
                                 or
                                - nt_protocol_id,
                                - source_address,
                                - destination_address),
                        data,
                        priority OPTIONAL,
                        timeout) OPTIONAL,
                        access parameters OPTIONAL
                        )
```

The *flowID* parameter FlowID is to be selected when path and flow management specified in ISO 24102-6[14] applies.

The *nt_protocol_id* parameter is defined in 8.1.1; the *source_address* and *destination_address* parameters are defined in 8.1.2. These three parameters are to be selected when path and flow management specified in ISO 24102-6[14] does not apply.

The *data* parameter contains the data to be transferred as payload. This parameter is referred to as ITS-NTPDU. *data* is of ASN.1 type `INdata` specified in B.2.

The *priority* parameter is defined in 8.3. It is mandatory when path and flow management specified in ISO 24102-6[14] does not apply.

The optional *timeout* parameter is defined in 8.4. Without *timeout* information provided either explicitly or implicitly by FlowID, the access technology assumes "no time-out".

The optional *access_parameters* parameter is defined in 8.6. Applicability of *these access_parameters* depends on upper layer protocols.

IN-UNITDATA.request is of ASN.1 type `IN-UNITDATA-request` specified in B.2.

8.2.2.3 When generated

IN-UNITDATA.request is passed from the ITS-S networking & transport layer to the addressed CAL to request that an ITS-ASDU be sent to one or more remote IN-SAPs.

8.2.2.4 Effect on receipt

Receipt of IN-UNITDATA.request causes the CAL to attempt to send the ITS-ASDU.

NOTE A packet addressed to a non-existent VCI will be discarded.

Success or failure of the transmission attempt may be reported to the ITS-S networking & transport layer using the IN-UNITDATA-STAUS.indication service primitive.

8.2.2.5 Additional comments

IN-UNITDATA.request is independent of any connection with the remote IN-SAP.

8.2.3 IN-UNITDATA.indication

8.2.3.1 Function

The service primitive IN-UNITDATA.indication is used for reception of data.

8.2.3.2 Semantics of the service primitive

IN-UNITDATA.indication (
nt_protocol_id,
source_address,
destination_address,
data,
priority,
access_parameters
)

The *nt_protocol_id* parameter is defined in 8.1.1.

The *source_address* and *destination_address* parameters are defined in 8.1.2.

The *data* parameter contains the data that are to be transferred. *data* is of ASN.1 type `INdata` specified in B.2.

The *priority* parameter is defined in 8.3. It shall present a priority value derived in a medium-specific way from the MAC priority of the respective access technology.

The *access_parameters* parameter is defined in 8.6.

IN-UNITDATA.indication is of ASN.1 type `IN-UNITDATA-indication` specified in B.2.

8.2.3.3 When generated

IN-UNITDATA.indication is passed from the CAL to the ITS-S networking & transport layer in the case of error-free reception of the physical layer frame to indicate the arrival of an ITS-ASDU from the specific remote entity. In the case of fragmentation performed in the CI, this service primitive is passed on only upon complete error-free reception of all fragments of the same block.

8.2.3.4 Effect on receipt

The effect on receipt of IN-UNITDATA.indication by the ITS-S networking & transport layer depends on the network protocol addressed.

8.2.3.5 Additional comments

IN-UNITDATA.indication is independent of any connection to the remote IN-SAP. In the absence of errors, the content of the data parameter is logically complete and unchanged relative to the data parameter in the associated IN-UNITDATA.request primitive.

8.2.4 IN-UNITDATA-STATUS.indication

8.2.4.1 Function

The service primitive IN-UNITDATA-STATUS.indication is used to notify success or failure of a related IN-UNITDATA.request or IN-UNITDATAACK.request.

8.2.4.2 Semantics of the service primitive

IN-UNITDATA-STATUS.indication (
addresses
(- flowID
or
- nt_protocol_id,
- source_address,
- destination_address),
data,
priority OPTIONAL,
timeout) OPTIONAL,
access parameters OPTIONAL
transmission_status,
service_class
)

The *flowID* parameter FlowID is to be selected when path and flow management specified in ISO 24102-6[14] applies.

The *nt_protocol_id* parameter is defined in 8.1.1; the *source_address* and *destination_address* parameters are defined in 8.1.2. These three parameters are to be selected when path and flow management specified in ISO 24102-6[14] does not apply.

The *data* parameter contains the data that were to be transmitted. *data* is of ASN.1 type `INdata` specified in B.2.

The *priority* parameter is defined in 8.3. It shall

— present the value used for transmission of the related IN-UNITDATA.request in case of success indicated by the parameter *transmission_status*, and

— be identical to the one in the related IN-UNITDATA.request service primitive in case of failure.

The *timeout* parameter is defined in 8.4.

The *access_parameters* parameter is defined in 8.6.

The *service_class* parameter indicates whether MAC acknowledgement was used or not. *service_class* is of ASN.1 type `MACServiceClass` specified in B.2.

The parameter *transmission_status* is defined in 8.7

IN-UNITDATA-STATUS.indication is of ASN.1 type `IN-UNITDATA-STATUS-indication` specified in B.2.

8.2.4.3 When generated

IN-UNITDATA-STATUS.indication is passed from the CAL to the ITS-S networking & transport layer in order to notify success of failure of a previous transmission request. Details on how to detect a failure are outside the scope of this document.

8.2.4.4 Effect on receipt

The effect on IN-UNITDATA-STATUS.indication by the ITS-S networking & transport layer depends on the network protocol addressed.

8.2.5 IN-UNITDATAACK.request

8.2.5.1 Function

The service primitive IN-UNITDATAACK.request is used for transmission of data. It is identical with IN-UNITDATA.request except that it requires application of a MAC acknowledgement procedure, if supported by the access technology.

8.2.5.2 Semantics of the service primitive

```
IN-UNITDATAACK.request    (
                          addresses
                              (- flowID
                              or
                              - nt_protocol_id,
                              - source_address,
                              - destination_address),
                          data,
                          priority OPTIONAL,
                          timeout) OPTIONAL,
                          access parameters OPTIONAL
                          )
```

IN-UNITDATAACK.request is of ASN.1 type `IN-UNITDATA-ACK-request` specified in B.2.

8.2.5.3 When generated

IN-UNITDATAACK.request is passed from the ITS-S networking & transport layer to the addressed CAL to request that an ITS-ASDU be sent to one remote IN-SAP, requesting a MAC sub-layer acknowledgement procedure.

8.2.5.4 Effect on receipt

Receipt of IN-UNITDATAACK.request causes the CAL to attempt to send the ITS-ASDU applying MAC sub-layer acknowledgement procedure if supported by the access technology.

NOTE A packet addressed to a non-existent VCI will be discarded.

Success or failure of the transmission attempt may be reported to the ITS-S networking & transport layer using the IN-UNITDATA-STAUS.indication service primitive.

8.2.5.5 Additional comments

IN-UNITDATAACK.request is independent of any connection with the remote IN-SAP.

8.3 Priority

Parameter *priority* carries the user priority. *priority* is of ASN.1 type `UserPriority` specified in ISO 17419[17].

Values for the priority parameter are in the range from 0 to 255, where 0 indicates lowest priority. The choice of priority values depends on ITS-S application processes served by the communication system, and related ITS-S messages.

Mapping of MAC sub-layer priorities to user priorities and vice versa are subject to the CI as specified in the related standard.

A VCI may require a minimum user priority as an access prerequisite by setting the I-Parameter 19 "MinimumUserPriority", see Annex A, to a value larger than zero. Packets offered to a VCI for transmission with a user priority below the minimum user priority according to I-Parameter 19 "MinimumUserPriority" shall be deleted, and the ITS-S management shall be notified by means of MI-REQUEST.request of ASN.1 type `Event21218Notification` indicating "invalid user priority". The default priority value shall indicate lowest priority.

8.4 Timeout

The *timeout* parameter carries the absolute time until which a pending packet may be stored in the ITS-S access layer. The value zero indicates "no timeout". *timeout* is of ASN.1 type `Time48IAT` specified in ISO 17419[17].

8.5 Queue management

Pending packets of same priority (i.e. a priority range defined by properties of the CI access technology, or defined by implementation) are stored in a first-in first-out queue. The ITS-S management shall be notified by means of MI-REQUEST.request of ASN.1 type `Event21218Notification` about the following events:

— a queue is filled up above an alert threshold presented in I-Parameter 22 "QueueAlarmThreshold" defined in the standard of the access technology (medium), or defined by implementation;

— a queue is emptied below an alert-threshold presented in I-Parameter 21 "QueueLowThreshold" defined in the standard of the access technology (medium), or defined by implementation;

— a queue can no longer store anymore packets.

Packets pending in a VCI shall not be sent once the lifetime of these packets has expired, if applicable.

Priority management across CIs may be a somewhat slow process which requires involvement of the ITS-S management for every packet to be prioritized. A packet to be transmitted may be recognized in CIs that are not in charge of transmitting it by means of a dummy transmission request. Details are specified in 6.4.11.

For protection, see 6.4.12.

8.6 Access parameters

The *access_parameters* parameter:

— used in IN-UNITDATA.request and IN-UNITDATAACK.request, allows requesting settings of access layer parameters for the given packet prior to transmission of the packet;

— used in IN-UNITDATA.indication, allows reporting about access layer receive parameters from the receiver of the packet;

— used in IN-UNITDATA-STATUS.indication;

— present the value used for transmission of the related IN-UNITDATA.request in case of success indicated by the parameter transmission_status;

— is identical to the one in the related IN-UNITDATA.request service primitive in case of failure.

access_parameters is of ASN.1 type `AccessParameters` specified in B.2.

Access parameters are part of the"dynamic data" identified in Clause 11.

Further details of the *access_parameters* values depend on the access technology of the selected CI.

8.7 Transmission status

The transmission_status parameter identifies success or failure of a previous transmission attempt. Possible status values are presented in Table 5.

Table 5 — Transmission status

Mnemonic	Value	Comment
SUCCESS	0	The related IN-UNITDATA.request was successfully processed at the access layer
QUEUE_FULL	1	The related IN-UNITDATA.request could not be processed at the access layer as the waiting queue was full
TIMEOUT	2	Lifetime of pending packet expired prior to transmission of the packet
PRIORITY	4	Priority of the packet is smaller than the minimum required priority given in I-Parameter "MinimumUserPriority"
NO_VCI	8	Selected VCI is not available
NO_CI	16	Selected CI is not available
UNSPECIFIED_FAILURE	128	The related IN-UNITDATA.request could not be processed at the access layer due to an unspecified failure

9 Management SAP

9.1 MI-SAP services

The functionality of the basic services:

— MI-GET and MI-SET to read and write values of I-Parameters;

— MI-COMMAND and MI-REQUEST to command actions;

and the related service primitives of the MI-SAP specified in ISO 24102-3 shall be supported.

The .confirm service primitives of MI-COMMAND and MI-REQUEST contain an error code of ASN.1 type `ErrStatus` indicating success or failure of the related .request service primitives.

The MI-SAP specifications from this document are considered by the ISO 24102-3 registrar, see B.3.

9.2 MI-COMMAND service primitive functions

9.2.1 Overview

Table 6 presents a summary of defined MI-COMMANDs presented by the ITS-S management entity to the MAE. In Annex B appropriate ASN.1 types are specified for .request and .confirm service primitives.

If a requested MI-COMMAND feature is not supported by the CI, the request shall be confirmed with error code 11 "FEATURE NOT SUPPORTED" specified in ISO 24102-3.

Table 6 — ISO 24102-3 MI-COMMANDs overview

COMMAND name	Description
ChangePseudonymMACaddress	Command to request change of MAC address or other type of layer 2 address, e.g. LTE Layer-2 ID, which is visible in a link to a peer station. This may be due to privacy regulations.
CIstateChange	Change of CI status.
CnConnect	Request a CI of CI access class CIAC-2 to connect to the communication service as soon as possible.
MacManagementFrameTX	Command to request transmission of a data packet in a MAC management frame.
ManufacturerCommand	Allows for manufacturer-specific access to the CI. Used e.g. for test and maintenance purposes. NOTE Details are outside the scope of this document.
MonitorIparameters	Command to request monitoring of I-Parameters.
PrioritizedRequestToSend	Used for cross-CI prioritization. Information on an intended prioritized transmission of a victim CI presented to interferer CIs.
PrioritizedRTSrelease	Used for cross-CI prioritization. Releases a related PrioritizedRequestToSend.
RegulatoryInformation	Provisioning of regulatory information.
VCImanagement	Command to request creation, reset or deletion of a VCI. Setting of parameters different to the default values for a newly created VCI has to be done in subsequent MI-SET commands.
WakeUpSignal	Command to enable and disable transmission of wake-up signals in a CI. Details depend on the access technology.

9.2.2 ChangePseudonymMACaddress

MI-COMMAND "ChangeLocalCIID" is used by the ITS-S management to request a change of the LocalCIID due to privacy regulations.

Table 7 — MI-COMMAND ChangePseudonymMACaddrress

ASN.1 Type	Description
`.request: ChangePseudonymMACaddress`	This empty command requests change of MAC address or other type of layer 2 address, e.g. LTE Layer-2 ID, in a CI-specific way.
`.confirm: ChangePseudonymMACaddressConf`	Empty acknowledgement indication success or failure of the request. This empty acknowledgement is only required in case of failure.

9.2.3 CIstateChange

MI-COMMAND "CIstateChange" is used by the ITS-S management to request a change of CI state.

Table 8 — MI-COMMAND CIstateChange

ASN.1 Type	Description
.request: CIstateChange	Available commands are: — deregister; — activate; — resume; — connect; — disconnect; — suspend; — inactivate.
.confirm: CIstateChangeConf	Empty acknowledgement indication success or failure of the request. Note that change of CI state will be notified automatically. This empty acknowledgement is only required in case of failure.

9.2.4 CnConnect

MI-COMMAND "CnConnect" is used by the ITS-S management to request that a CI of CI access class CIAC-2 or CIAC-3 connects / disconnects to the communication service, or deletes access information.

Table 9 — MI-COMMAND CnConnect

ASN.1 Type	Description
.request: CnConnect	deleteAC: Delete access information I-parameters "SIMpin" and "ProviderInfo", if applicable. This automatically disconnects from the service, if applicable. connect: Connect asap with given access information. If this information is not available, confirm command with error code 7 "ACCESS VIOLATION" specified in ISO 24102-3. disconnect: Disconnect immediately without deleting access information.
.confirm: CnConnectConf	Empty acknowledgement indication success or failure of the request.

9.2.5 MacManagementFrameTX

MI-COMMAND "MacManagementFrameTX" is used by the ITS-S management to request transmission of a packet in a MAC management frame, if applicable.

Table 10 — MI-COMMAND MacManagementFrameTX

ASN.1 Type	Description
.request: MacManagementFrameTX	Transmission request of a management data packet.
MacManagementFrameTX .sourceAddr	Same as source_address in DL_UNITDATA.request.
MacManagementFrameTX .destAddr	Same as destination_address in DL_UNITDATA.request.
MacManagementFrameTX .data	Same as data in DL_UNITDATA.request.

Table 10 *(continued)*

ASN.1 Type	Description
`MacManagementFrameTX .priority`	Same as priority in DL_UNITDATA.request.
`MacManagementFrameTX .parameter`	Parameters to be set prior to transmission. Details depend on the access technology (medium). These parameters are part of the"dynamic data" identified in Clause 11.
`.conf: MacManagementFrameTXConf`	Empty acknowledgement indication success or failure of the request. This empty acknowledgement is only required in case of failure.

9.2.6 ManufacturerCommand

MI-COMMAND "ManufacturerCommand" is used by the ITS-S management to allow a manufacturer to have private access to a CI.

Table 11 — MI-COMMAND ManufacturerCommand

ASN.1 Type	Description
`.request: ManufacturerSpecificCiAccess`	Manufacturer-specific octet string.
`confirm: ManufacturerSpecificCiAccessConf`	Manufacturer-specific octet string.

9.2.7 MonitorIparameters

MI-COMMAND "MonitorIparameters" is used by the ITS-S management to request automatic notification of change of I-Parameter values.

Table 12 — MI-COMMAND MonitorIparameters

ASN.1 Type	Description
`.request: MonitorIparameter`	Sequence of `MonitorDetail` of indicating I-Parameters for which monitoring shall be started or stopped.
`MonitorDetail.paramNo`	Reference tag number of I-Parameter to be monitored.
`MonitorDetail.action`	0: stop monitoring 255: start monitoring
`.confirm: MonitorIparameterConf`	Sequence of information elements `MonitorError` presenting details on errors. Without errors there are zero such information elements.

9.2.8 PrioritizedRequestToSend

MI-COMMAND "PrioritizedRequestToSend" is used by the ITS-S management to request prioritization from potentially interfering CIs.

Table 13 — MI-COMMAND PrioritizedRequestToSend

ASN.1 Type	Description
`.request: PrioritizedRTS`	Information on an intended prioritized transmission of a victim CI presented to interferer CIs. Used for cross-CI prioritization.
`PrioritizedRTS.seqNo`	Sequential number to identify dummy request. Cyclic counter.
`PrioritizedRTS.reqID`	LocalCIID of CI announcing the dummy request.
`PrioritizedRTS.priority`	User priority of dummy request.

Table 13 *(continued)*

ASN.1 Type	Description
`.confirm: PrioritizedRTSConf`	Acknowledgement of dummy request
`PrioritizedRTSConf.seqNo`	Same value as in .request
`PrioritizedRTSConf.reqID`	Same value as in .request
`PrioritizedRTSConf.status`	ignored granted

9.2.9 PrioritizedRTSrelease

MI-COMMAND "PrioritizedRTSrelease" is used by the ITS-S management to release a previously requested prioritization.

Table 14 — MI-COMMAND PrioritizedRTSrelease

ASN.1 Type	Description
`.request: PrioritizedRTSrelease`	Releases a previously announced intended prioritized transmission. Used for cross-CI prioritization.
`PrioritizedRTS.seqNo`	Sequential number to identify dummy request. Cyclic counter.
`.confirm: PrioritizedRTSreleaseConf`	Optional empty acknowledgement of dummy request

9.2.10 RegulatoryInformation

MI-COMMAND "RegulatoryInformation" is used by the ITS-S management to provide updates of regulatory information.

ASN.1 Type	Description
`.request: RIprovisioning`	Regulatory information provided by the ITS-S management to a CI
`.confirm: RIprovisioningConf`	Empty acknowledgement indication success or failure of the request. Failure to apply the provided regulatory information shall be indicated with `ErrStatus` 4 "RI violation".

9.2.11 VCImanagement

MI-COMMAND "VCImanagement" is used by the ITS-S management to request creation, deletion or reset of a VCI.

Table 15 — MI-COMMAND VCIcmd

ASN.1 Type	Description
`.request: VciCmd`	Information on action to be performed on an existent or non-existent VCI.
`VciCmd.linkID`	Link-ID of VCI to be deleted, created, or reset.
`VciCmd.alive`	FALSE: Delete VCI, if alive. TRUE: Create VCI, if not existent, or reset it.
`.confirm: VciCmdConf`	Empty acknowledgement indication success or failure of the request. Note that change of VCI state will be notified automatically. This empty acknowledgement is only required in case of failure.

9.2.12 WakeUpSignal

MI-COMMAND "WakeUpSignal" is used by the ITS-S management to enable and disable transmission of wake-up signals in a CI. Details depend on the access technology (medium).

Table 16 — MI-COMMAND WakeUp

ASN.1 Type	Description
`.request: WakeUp`	0: Stops transmission of wake-up signal. 1 to 255: Starts repetitive transmission of wake-up signal with maximum interval in milliseconds.
`.confirm: WakeUpConf`	Empty acknowledgement indication success or failure of the request. This empty acknowledgement is only required in case of failure.

9.3 MI-REQUEST service primitive functions

9.3.1 Overview

Table 17 presents a summary of defined MI-REQUESTs presented by the MAE to the ITS-S management entity. In Annex B appropriate ASN.1 types are specified for .request service primitives.

If a requested MI-REQUEST feature is not supported by the ITS-S management entity, the request shall be confirmed with error code 11 "FEATURE NOT SUPPORTED" specified in ISO 24102-3.

Table 17 — ISO 24102-3 MI-REQUESTs overview

REQUEST name	Description
EventNotification	Notification of an event.
MacManagementFrameRX	Notification of a data packet received in a MAC management frame.
PositionUpdate	Requests to receive position updates with update interval as indicated in milliseconds, or to stop delivery of updates.
PrioritizationRegistration	Registration of a victim CI for Cross-CI Prioritization.
PrioritizationRequest	Used for cross-CI prioritization. Real-time request of a victim VCI to get prioritization.
PrioritizationRelease	Used for cross-CI prioritization. Releases a previously presented request of a victim VCI to get prioritization.
RegistrationCI	Request to register the CI.

9.3.2 EventNotification

MI-REQUEST "EventNotification" is used by the MAE to notify events.

Table 18 — MI-REQUEST EventNotification

ASN.1 Type	Description
`.request: Event21218Notification`	Notification of events to the ITS-S management. Events are specified in Table 25.
`.confirm Event21218NotificationConf`	Empty acknowledgement indication success or failure of the request. This empty acknowledgement should never be used.

9.3.3 MacManagementFrameRX

MI-REQUEST "9.3.3 MacManagementFrameRX" is used by the MAE to notify reception of a data packet in a MAC management frame.

Table 19 — MI-REQUEST MacManagementFrameRX

ASN.1 Type	Description
`.request: MacManagementFrameRX`	Reception of a data packet in a MAC management frame.
`MacManagementFrameRX .sourceAddr`	Same as source_address in DL_UNITDATA.indication.
`MacManagementFrameRX .destAddr`	Same as destination_address in DL_UNITDATA.indication.
`MacManagementFrameRX .data`	Same as data in DL_UNITDATA.indication.
`MacManagementFrameRX .priority`	Same as priority in DL_UNITDATA.indication.
`MacManagementFrameRX .parameter`	Parameters reporting details of frame reception. Details depend on the access technology (medium). These parameters are part of the"dynamic data" identified in Clause 11.
`.confirm: MacManagementFrameRXConf`	Empty acknowledgement indication success or failure of the request. This empty acknowledgement should never be used.

9.3.4 PositionUpdate

MI-REQUEST "PosUpdateReq" is used by the MAE to manage reception of updates of the actual position of the station.

Table 20 — MI-REQUEST PositionUpdate

ASN.1 Type	Description
`.request: PositionUpdateRequest`	0: Stop updates. >0: Updates with interval given in milliseconds. The updates shall be written in I-Parameter "KinematicVectorIn" by means of the MI-SET service.
`.confirm: PositionUpdateRequestConf`	Empty acknowledgement indication success or failure of the request. This empty acknowledgement should never be used.

9.3.5 PrioritizationRegistration

MI-REQUEST "PrioritizationRegistration" is used by the MAE of a victim CI to register at the ITS-S management for the cross-CI prioritization procedure.

Table 21 — MI-REQUEST PrioritizationRegistration

ASN.1 Type	Description
`.request: PrioritizationRegistration`	Registration of a victim VCI for "Cross-CI prioritization"
`PrioritizationRegistration .interferers`	MedType of potential interferers. Known *a priori* to CI.
`PrioritizationRegistration .timeout`	Timeout. Default value set by CI.
`.confirm: PrioritizationRegistrationConf`	Empty acknowledgement indication success or failure of the request.

9.3.6 PrioritizationRequest

MI-REQUEST "RrioritizationRequest" is used by the MAE of a victim CI to request and release cross-CI prioritization.

Table 22 — MI-REQUEST PrioritizationRequest

ASN.1 Type	Description
`.request: PrioritizationRequest`	Dummy Request To Send (RTS). Used for cross-CI prioritization.
`PrioritizationRequest .seqNo`	Sequential number to identify dummy request. Cyclic counter.
`PrioritizationRequest .priority`	User priority of dummy request.
`.confirm: PrioritizationRequestConf`	Acknowledgement of the dummy RTS.
`PrioritizationRequestConf .seqNo`	Same as in .request
`PrioritizationRequestConf .priority`	As granted.
`PrioritizationRequestConf .status`	ignored granted

9.3.7 PrioritizationRelease

MI-REQUEST "PrioritizationRelease" is used by the MAE of a victim CI to release cross-CI prioritization.

Table 23 — MI-REQUEST PrioritizationRelease

ASN.1 Type	Description
`.request: PrioritizationRelease`	Dummy Request To Send (RTS). Used for cross-CI prioritization.
`PrioritizationRelease .seqNo`	Sequential number to identify dummy request. Cyclic counter.
`.confirm: PrioritizationReleaseConf`	Optional empty acknowledgement of the dummy RTS.

9.3.8 RegistrationCI

MI-REQUEST "RegistrationCI" is used by the MAE to register a CI at the ITS-S management.

Table 24 — MI-REQUEST RegisterCI

ASN.1 Type	Description
`.request: RegistrationCI`	Request to register the CI of given access technology type.
`.confirm: RegistrationCIconf`	Acknowledgement of registration request.
`RegistrationCIconf .scuId`	Identifier of the ITS-SCU specified in ISO 17419[17].
`RegistrationCIconf .medID`	Sequential reference number of CI in the given ITS-SCU.

10 Events

Events are notified with the MI-Request service primtive Events of ASN.1 type `Event21218Notification`. The events presented in Table 25 are identified. The list of events may be extended in the future, even by other standards. Events are part of the"dynamic data" identified in Clause 11.

Table 25 — ISO 21218 events

Event ASN.1 type	Event description	Request.Value
`E21218-0`	A transmission request was rejected due to a user priority that was below the minimum required priority as defined by I-Parameter "MinimumUser-Priority".	I-Parameter "MinimumUserPriority" Link-ID of VCI
`E21218-1`	A transmission queue is filled above the threshold defined by I-Parameter "QueueAlarmThreshold".	Priority of the queue
`E21218-2`	A transmission queue is full.	Priority of the queue
`E21218-3`	A VCI was created.	Link-ID of VCI
`E21218-4`	A VCI was deleted.	Link-ID of VCI
`E21218-5`	An I-Parameter subject to notification has changed its value.	I-Param
`E21218-6`	A transmission queue is emptied below the threshold defined by I-Parameter "QueueLowThreshold".	Priority of the queue
`E21218-7`	A VCI was reset.	Link-ID of VCI
`E21218-8`	An address conflict was detected by a CI, i.e. a frame (ITS-APDU specified in ISO 21217) was received from a peer ITS-SU using the same address. This may apply e.g. for dynamically assigned MAC addresses.	Link-ID of CI that received the frame with the conflicting address.

11 Dynamic data

Dynamic data are data being registered at the ISO standards maintenance portal of ISO 21218[26]. New details of dynamic data may be specified in other standards based on the generic definitions in this document, and will become part of this document once the other standard applies for a registration of these dynamic data details.

Dynamic data are:

— CI classes, see 6.2.1;

— CI access classes, see 6.2.2;

— Regulatory information schemes 6.4.13;

— IN-SAP service primitives, see 8.2;

— Access parameters, see 8.6;

— MAC management frame parameters, see 9.2.

— Events, see Clause 10;

— I-Parameters, see Annex A;

— Medium-specific I-Parameter definitions, see Annex A.

12 Conformance

An implementation conformance statements (ICS) proforma is specified in Annex E.

NOTE Conformance with this document is always tested in combination with at least one access technology standard, e.g. IEEE 802.11 OCB profile ISO 21215[10], LTE-V2X[5].

13 Test methods

The test suite structure and test purposes (TSS&TP), and an abstract test suite (ATS) for conformance tests will be specified in another document based on the respective test suite standards for the predecessor of the present document, i.e. ETSI TS 102 760-2[23] and ETSI TS 102 760-3[24].

Conformance testing may use upper tester access in the System Under Test (SUT) applying ITS station-internal management communications specified in ISO TS 20026[1], ISO 24102-4[13], and ISO 24102-3[13].

Annex A
(normative)

I-Parameters

CI parameters also are referred to as I-Parameters. I-Parameters are of ASN.1 type `I-Param` specified in Annex B. I-Parameters are identfied by a unique reference number I-Param.No. Privately defined and used I-Parameters are allowed within the number range of I-Param.No from 65280 through 65535.

NOTE Usage of privately defined I-Parameters may result in unexpected behaviour due to multiple definitions for the same number.

Table A.1 presents the relation between parameter number "I-Param.No" and parameter name as used in the services MI-SET and MI-GET, and provides a description of the I-Parameters.

"Access" specifies the possible access to the I-Parameter:

— R: Read by ITS-S management only, includes notify by CI/VCI;

— W: Write by ITS-S management only;

— RW: Read and write;

— N: Notify by CI only;

— NW: Notify by CI and Write by ITS-S management;

— x: to be defined by standard of the access technology (medium).

"Owner" specifies the owner of the I-Parameter:

— VCI: a VCI;

— CI: a CI;

— x: to be defined by standard of the access technology (medium).

NOTE An I-Parameter can be medium-specific.

I-Parameters are part of the "dynamic data" identified in Clause 11, i.e.

— I-Parameters are partly defined in this document, and may be defined in other standards;

— I-Parameters from other standards will be added to the online registry for ISO 21218 I-Parameters on the ISO standards maintenance portal[26], and will be included in subsequent revisions of this document;

— related values for the reference number I-Param.No will be assigned by the ISO 21218 registrar.

Table A.1 — I parameters

I-Param No	I-Parameter name/ASN.1 Type	Comm-Profile	Access	Description	Owner
41	BlockLength/ `BlockLength`	Yes	RW	Maximum length of the ITS-APDU in octets	VCI
51	ChannelParam/ `ChannelParam`	Yes	RW	Usage specified in ISO 19079[6]	VCI
11	CIaccessClass/ `CIaClass`	Yes	R	CI access class.	CI
2	CI-Capabilities/ `CI-Capabilities`	No	R	This parameter was named "Properties" in earlier versions of this document. It is now renamed to "Capabilities" in order to align terminology with ISO 24102-6. Details of "Properties" will be specified in an Amendment of this document related to ISO 24102-6.	CI
10	CIclass/ `CIclass`	Yes	R	Communications interface class.	CI
12	CIstatus/ `CIstatus`	No	R	Status of CI.	CI
1	CommProfile/ `CommProfile`	No	R	Communication profile. Contains the set of parameter values that define the actual communication properties (capabilities) of a VCI. A default profile beneficially is to be defined for each access technology in the respective access technology standard. The column CommProfile of this table indicates which parameters may be part of the Communication profile.	VCI
24	CommRangeRef/ `CommuniationRangeReference`	Yes	R	Estimate of size of communication zone in 1/10 m. NOTE Value is derived from TXpower, RXsensitivity and the properties of the reference station. To be calculated by the MAE.	VCI
16	Connect/ `Connect`	Yes	R	Flag indicating whether CI will connect automatically or manually upon request.	CI
46	Cost/ `MediumCost`	No	R	Price information. Cost of communication in terms of money, e.g. Cost information is temporarily unavailable. Free of any charge. Fixed flat rate. Price per time unit. Price per amount of data. General variable cost according to contract. Specified in ISO 17419[17].	VCI

Table A.1 *(continued)*

I-Param No	I-Parameter name/ASN.1 Type	Comm-Profile	Access	Description	Owner
37	DataRateNW/ DataRateNetwork	Yes	R	Estimate of average data rate available at the IN-SAP in 100 bit/s. NOTE 1 The value of this parameter is based on the assumption of a reliable, error-free communications link. NOTE 2 The value of this parameter may depend on the actual operational load of the CI, e.g. in the case of a TDMA scheme and multiple simultaneous users. NOTE 3 In a TDMA scheme this value depends on the number of timeslots (=peer devices) served in a single TDMA frame.	VCI
39	DataRateNWreq/ DataRateNetworkRequired	No	RW	Minimum required value of DataRateNW. This value defines the possible number of time slots (=peer stations) served in a TDMA scheme.	CI
38	DataRatesNW/ DataRatesNetwork	Yes	R	Minimum and maximum possible value of DataRateNW.	VCI
40	Directivity/ Directivity	Yes	R, RW	Characteristics of beam. See e.g[9]. for more details.	CI, VCI
23	DistancePeer/ DistancePeer	No	R	Distance in 1/10 m to peer station estimated by the CI	VCI
43	FrameLengthMax/ FrameLengthMax	Yes	RW	Maximum length of a TDMA frame in a time unit. See e.g[9]. for more details.	VCI
42	FreeAirTime/ FreeAirTime	Yes	RW	Gap between subsequent TDMA given as a time. See e.g[9]. for more details.	VCI
50	FrequencyBand/ FrequencyBand	Yes	RW	Usage specified in ISO 19079[6]	VCI
26	InactivityTimeLimit/ InactTimeLimit	No	RW	Maximum allowed idle time of an RX-VCI with respect of a specific peer station, i.e. maximum allowed time without proper reception of a frame. 0: No limit. >0: Value ofimit .	VCI
55	ITSAID/ ITSaid	No	n.a.	ITS-AID specified in ISO 17419[17] and used on a packet-per-packet basis as access parameter for destination address management in the ITS-S access layer. Values are provided by the ITS-S facilities layer.	n.a.
4	ITS-SCU-ID/ ITS-scuId	No	RW	Unique identifier of the ITS-SCU in an ITS-S unit. Specified in ISO 17419[17].	CI

Table A.1 *(continued)*

I-Param No	I-Parameter name/ASN.1 Type	Comm-Profile	Access	Description	Owner
44	ITS-SCUkinematicState/ `ItsScuKinematicState`	No	W	Kinematic state of an ITS station unit. Date and universal time Latitude: specified in ISO 17419[17] Longitude: specified in ISO 17419[17] Altitude: specified in ISO 17419[17] Ground speed: Resolution 0,01 m/s True track angle: Resolution 0,1°	CI
45	KinematicVectorOut/ `KineVectOut`	No	R	Kinematic vector of ITS station as estimated by CI. Date and universal time Latitude: specified in ISO 17419[17] Longitude: specified in ISO 17419[17] Altitude: specified in ISO 17419[17] Ground speed: Resolution 0,01 m/s True track angle: Resolution 0,1°	CI
52	LimitChannelAccess/ `LimitChannelAccess`	Yes	RW	Allowed usage of a specific phyiscal communication channel of a CI expressed in percentage of time applicable for a defined observation time period	CI
36	LinkDataRate/ `DataRateLink`	Yes	RW	Data rate in a medium-specific format.	VCI
8	LLaddress/ `LLaddress`	Yes	R	Globally assigned link layer address of CI.	CI
9	LLaddressTemp/ `LLaddressTemp`	Yes	RW	Actually used link layer address of CI.	CI
6	LocalCIID/ `LocalCIID`	No	R	Identifier of local CI. E.g. 48-bit MAC address encapsulated in EUI-64 field.	CI
48	LogicalChannels/ `LogicalChannels`	Yes	RW	Mapping of logical channels on phyiscal channels	VCI
3	ManufacturerDeviceID/ `ManuDeviceID`	No	R	Text string to be defined by manufacturer, clearly identifying the CI.	CI
5	MedID/ `MedID`	No	RW	Sequential number indicating uniquely a CI within an ITS-SCU	CI
27	MediumUsage/ `MediumUsage`	No	R	Percentage of active usage of access technology (medium) for receive and transmit channel. 0: 0 %. 255: 100 %	CI
14	MedType/ `ITSatt`	Yes	R	Indicates type of access technology (medium). Specified in ISO 17419[17].	CI

Table A.1 *(continued)*

I-Param No	I-Parameter name/ASN.1 Type	Comm-Profile	Access	Description	Owner
28	MedUseObservationTime/ `MedUseObsTime`	No	RW	Observation time used to calculate MediumUsage. 0: No observation. >0: Size of gliding window for measurement of I-Parameter 27 "MediumUsage". NOTE An access technology (medium) can use the nearest available value and can change this parameter accordingly.	CI
19	MinimumUserPriority/ `MinimumUserPriority`	Yes	RW	Minimum value of user priority needed to use the VCI. NOTE Default value zero defined.	VCI
31	MinPrioCrossCI/ `MinimumCrossCiPriority`	No	RW	Minimum required user priority in order to be able to request cross-CI prioritization.	CI
13	Notify/ `Notify`	No	R	List of I-Parameter numbers indicating I-Parameters for automatic notification.	VCI
54	OperationalMode/ `OperationalMode`	Yes	RW	CIs than can be operated with different modes indicate the actually valid mode with this medium-specific I-Parameter	CI
29	PeerLLAddress/ `LLaddressPeer`	No	RW	Link layer address of peer station related to the VCI. Set to the "all bits zero" value if no relation to a peer station exists. Only applicable for UC-VCIs.	VCI
35	PeerTXpower/ `PeerTXpower`	No	R	TX power EIRP as reported by a peer station. Same presentation as TxPower.	VCI
53	PhysicalChannelIdentifier/ `PhysicalChannelIdentifier`	Yes	R	Identifier of a physical communication channel used by a VCI of a given CI	VCI
18	ProviderInfo/ `ProviderInfo`	No	W	Access information for a provider: Name of provider Name of access point Log-in name of user Password for log-in	CI
56	QoSrequirement/ `QoSrequirement`	No	n.a.	Medium specific information on QoS features requried for specific purposes and provided to a CI on a packet-by-packet basis.	n.a.
22	QueueAlarmThreshold/ `QueueAlarmThreshold`	No	RW	Threshold indicating the minimum level of usage of a transmit queue for a specific priority value in a CI above which the ITS-S management is notified. 0: Transmit queue is empty. 255: Transmit queue is full (100 %)	CI

Table A.1 *(continued)*

I-Param No	I-Parameter name/ASN.1 Type	Comm-Profile	Access	Description	Owner
20	QueueLevel/ `QueueLevelActual`	No	N	Actual level in a transmit queue for a specific priority value. 0: Transmit queue is empty. 255: Transmit queue is full (100 %)	CI
21	QueueLowThreshold/ `QueueLowThreshold`	No	RW	Threshold indicating the maximum level of usage of a transmit queue for a specific priority value in a CI below which the ITS-S management is notified. 0: Transmit queue is empty. 255: Transmit queue is full (100 %)	CI
15	RegulatoryInformation/ `RegulatoryScheme`	Yes	RW	RI data structure containing a regulatory information, or a statement that no regulation is known or applicable (identified by the reference value `c-RegSchemeNull`).	CI
47	Reliability/ `Reliability`	No	R	Real-time measure of reliability of CI. 0: unreliable 1: ½ percent reliabilty 2: 1 percent reliability ... 200: perfect (100 % reliability) 201 - 254: reserved 255: unknown reliability	VCI
65280 - 65535	reserved for private usage				
32	RXsensitivity/ `RxSens`	Yes	RW	Receiver sensitivity presented in a medium-specific format. Receiver sensitivity typically is without gain of receive antenna, i.e. corresponding to receive field strength in the air in front of the antenna.	VCI
17	SIMpin/ `SimPin`	No	W	Medium-specific access information needed to connect a CI to a network, e.g. PIN for 3G/4G.	CI
25	TimeOfLastReception/ `TimeOfLastReception`	No	R	Time when last frame was successfully received.	VCI
7	TimeoutRegister/ `TimeoutRegistration`	No	RW	Time out to be used during registration of a CI.	CI
49	TotalBatteryEnergy/ `TotalBatteryEnergy`	No	R	Usage specified in ISO 19079[6]	CI
33	TXpower/ `TxPowerActual`	Yes	RW	Actual transmit power EIRP presented in a medium-specific format.	VCI
34	TXpowMax/ `TxPowerMax`	Yes	R	Maximum allowed transmit power EIRP. Same presentation as TxPower.	CI
30	VirtualCIs/ `VirtualCIs`	No	R	RemoteCIID values of all VCIs of the selected CI being alive.	CI

Annex B
(normative)

ASN.1 definitions

B.1 Overview

The ASN.1 basic notation is specified in ISO/IEC 8824-1[34]. The following ASN.1 module is specified in this normative annex:

— ITSllsap {iso(1) standard(0) calm-ll-sap(21218) asnm-1 (1) version2 (2)}

In case the ASN.1 specifications given in this Annex are not compliant with illustrations or specifications provided elsewhere in this document, the specifications given in this Annex shall prevail.

Updates of this ASN.1 module will be published on the ISO standards maintenance portal at[26].

Applicable encodings of the types and values defined in the ASN.1 module specified in B.2 depend on the usage. ASN.1 BASIC-PER, UNALIGNED, as specified in ISO/IEC 8825-2, shall apply if no other explicit requirement on encoding is given.

B.2 Module ITSllsap

In order to achieve octet alignment enabling cheap implementations, "fill" bits were defined. All fill bits shall be set to the value '0'b.

```
ITSllsap {iso(1) standard(0) calm-ll-sap(21218) asnm-1 (1) version2 (2)} DEFINITIONS
AUTOMATIC TAGS::=BEGIN

-- EXPORTS;
IMPORTS

-- ISO 16460 ITSee
TXpower80211, ChannelNumber80211, DataRate80211 FROM ITSee1 { iso (1) standard (0)
localized(16460) ee(4) version1 (1)}

-- C-ITS Data Dictionary (still in ISO 17419)
Altitude, Latitude, Longitude, Logic, NullType, UserPriority, Time48IAT, Ethertype, Int1,
Int2, Int4, MACaddress, OneOCTETones, MACext, TwoOCTETones, MACoui, EUI64MAC48, EUI64,
MediumCost, TimeDurationValue FROM CITSdataDictionary1 {iso(1) standard(0) cits-applMgmt
(17419) dataDictionary (1) version1 (1)}

-- ISO 17419
LogicalChannelType, ITSatt, ITS-scuId, c-ITSatt-iso21215, c-ITSatt-iso17515, c-ITSatt-any,
ITSaid FROM CITSapplMgmtComm {iso(1) standard(0) cits-applMgmt (17419) applRegistry (2)
version2 (2)}

-- ISO 17515-3
LTE-V2X-OperationalMode, LTE-V2X-Layer2Address FROM ITSltev2x {iso (1) standard (0) lte
(17515) v2x (3) version0 (0)}

-- ISO 19079
TotalBatteryEnergy, FrequencyBand, ChannelParam FROM CALM6lowpan1 { iso (1) standard (0)
6lowpan (19079) asnm-1 (1) version1 (1)}

-- ISO 21215
ITS-M5-OperationalMode, RxSens80211, M5-MgmtSubTypes, ITS-M5-TIDcontrol FROM ITSm5
{iso (1) standard (0) calm-m5 (21215) asnm-1 (1) version1 (1)}

-- ISO 24102-6
```

```
FlowID FROM ITSpfm2 { iso (1) standard (0) calm-management (24102) pfm (6) asnm-1 (1)
version2(2) }
;

-- End of IMPORTS
-- General types

-- Class for medium specific parameters
MEDSPEC ::= CLASS {
    &medType ITSatt UNIQUE,
    &SpecFormat
    }
    WITH SYNTAX {&SpecFormat IDENTIFIED BY &medType}

-- General extension (profiling)
MediumSpecificFormats {MEDSPEC : SpecifcTypes} ::= SEQUENCE {
    mediumId      MEDSPEC.&medType({SpecifcTypes}),
    specificType  MEDSPEC.&SpecFormat({SpecifcTypes}{@.mediumId})
    }

/* For every medium-specific parameter
MediumSpecificFormat::= MediumSpecificFormats{{MediumSpecificTypeFormats}}

MediumSpecificTypeFormats MEDSPEC::={
    { MediumSpecifictype1 IDENTIFIED BY c-MediumReferenceNumber1 } |
    { MediumSpecifictype2 IDENTIFIED BY c-MediumReferenceNumber2 } ,
    ...
    }
*/
-- Data rate format
DataRate::= Int4                -- in 100 bit/s

-- Destination address LinkID
DestinationLinkID::=Link-ID

-- Distance in 10 cm
Distance::=Int2                 -- measured in 1/10 m

-- Ground speed
Gs::=Int2 -- Resolution 0,01 m/s

-- Kinematic vector options
KineVectOptions::=CHOICE{
    lat             [0] Latitude, -- latitude
    lon             [1] Longitude, -- longitude
    alt             [2] Altitude, -- altitude
    gs              [3] Gs, -- ground speed
    tta             [4] Tta -- true track angle
    }

KineVectOption::=SEQUENCE{
    fill            BIT STRING (SIZE(5)), -- '00000' PER octet alignment
    kineVectOptions  KineVectOptions
    }

-- Legacy CIID format
LegacyCIID::=SEQUENCE{
    selector1       OneOCTETones,
    iTS-scuId       ITS-scuId,
    selector2       TwoOCTETones,
    medID           MedID,
    vciSerialNumber  VCIserialNumber
    }

-- Link Identifier
Link-ID::=SEQUENCE{                 -- Link-ID
    remoteCIID      RemoteCIID,     -- CI in peer ITS-S(s)
    localCIID       LocalCIID       -- Unique ID of local CI
    }

-- Waiting queue level information
```

```
QueueLevel::=SEQUENCE{
    priority         UserPriority,
    level            QueueValue
    }

-- Waiting queue level value
QueueValue::=INTEGER(0..255)

-- Regulatory information scheme identifier
RefREGULSCHEME ::= INTEGER {
    c-RegSchemeNull    (0),
    c-RegScheme-iso21215    (c-ITSatt-iso21215),
    c-RegScheme-iSO17515    (c-ITSatt-iso17515)
    } (0..255)

-- Regulatory information scheme CLASS
REGULSCHEME::=CLASS {
    &regID    RefREGULSCHEME UNIQUE,
    &RegInfo
    }

-- A single regulatory information element
RegulatoryScheme::=SEQUENCE{
    reguIdent    REGULSCHEME.&regID({RegulSchemes}),
    reguInfo     REGULSCHEME.&RegInfo({RegulSchemes}{@reguIdent})
    }

-- All regulatory information information types
RegulSchemes REGULSCHEME::={nullRegScheme, ...} -- to be filled
nullRegScheme REGULSCHEME::={&regID c-RegSchemeNull, &RegInfo NullType}

-- Remote CIID
RemoteCIID::=EUI64

-- Source address LinkID
SourceLinkID::=Link-ID

-- Transmit power format
TxPower::=MediumSpecificFormats{{TxPowers}}
TxPowers MEDSPEC::={
    { TXpower80211    IDENTIFIED BY c-ITSatt-iso21215 } ,
    ...
    }

-- True track angle
Tta::=INTEGER(0..65535) -- Resolution 0,1°.

-- VCI serial number
VCIserialNumber::=INTEGER(0..65535)

-- I-Parameters
-- I-Param reference number
I-ParamNo::=INTEGER{
    c-commProfile              (1),
    c-ciCapabilities           (2),
    c-manuDeviceID             (3),
    c-iTS-scuId                (4),
    c-medID                    (5),
    c-localCIID                (6),
    c-timeoutReg               (7),
    c-llAddress                (8),
    c-llAddressTemp            (9),
    c-ciClass                  (10),
    c-ciaClass                 (11),
    c-ciStatus                 (12),
    c-notify                   (13),
    c-medType                  (14),
    c-regulatoryScheme         (15),
    c-connect                  (16),
    c-simPin                   (17),
    c-providerInfo             (18),
```

```
    c-minUserPriority         (19),
    c-queueLevel              (20),
    c-queueLowTh              (21),
    c-queueAlarmTh            (22),
    c-distancePeer            (23),
    c-commRangeRef            (24),
    c-timeOfLastRecep         (25),
    c-inactTimeLimit          (26),
    c-mediumUsage             (27),
    c-medUseObsTime           (28),
    c-llAddressPeer           (29),
    c-virtualCIs              (30),
    c-minPrioCrossCI          (31),
    c-rxSens                  (32),
    c-txPower                 (33),
    c-txPowMax                (34),
    c-peerTXpower             (35),
    c-dataRateLink            (36),
    c-dataRateNW              (37),
    c-dataRatesNW             (38),
    c-dataRateNWreq           (39),
    c-directivity             (40),
    c-blockLength             (41),
    c-freeAirTime             (42),
    c-frameLengthMax          (43),
    c-itsScuKinematicState    (44),
    c-kineVectOut             (45),
    c-cost                    (46),
    c-reliability             (47),
    c-logicalChannels         (48),
    c-totalBatteryEnergy      (49), -- medium-specific: ISO 19079
    c-frequencyBand           (50), -- ISO 19079
    c-channelParam            (51), -- ISO 19079    }
    c-limitChannelAccess      (52),
    c-physicalChannelId       (53),
    c-operationalMode         (54),
    c-itsAID                  (55),
    c-qosRequirements         (56)
-- reserved for private usage: 65280 through 65535
    }

-- I-Param CLASS
IPARAM ::= CLASS {
    &paramRef I-ParamNo UNIQUE,
    &Parameter
    }

-- A single I-Parameter
I-Param::=SEQUENCE{
    paramNo     IPARAM.&paramRef({I-Params}),
    parameter     IPARAM.&Parameter({I-Params}{@paramNo})
    }

-- All I-Params
I-Params IPARAM::={commProfile /*| ciCapabilities*/ | manuDeviceID | iTS-scuId | medID |
 localCIID | timeoutReg | llAddress | llAddressTemp | ciClass | ciaClass | ciStatus |
notify | medType | regulatoryScheme | connect | simPin | providerInfo | minUserPriority |
queueLevel | queueLowTh | queueAlarmTh | distancePeer | commRangeRef | timeOfLastRecep |
inactTimeLimit | mediumUsage | medUseObsTime | llAddressPeer | virtualCIs | minPrioCrossCI
| rxSens | txPower | txPowMax | peerTXpower | dataRateLink | dataRateNW | dataRatesNW |
dataRateNWreq | directivity | blockLength | freeAirTime | frameLengthMax |
itsScuKinematicState | kineVectOut | cost | reliability | logicalChannels |
totalBatteryEnergy | frequencyBand | channelParam | limitChannelAccess | operationalMode |
physicalChannelId | itsAID | qosRequirements, ...}

commProfile             IPARAM::={&paramRef c-commProfile, &Parameter CommProfile}
-- ciCapabilities       IPARAM::={&paramRef c-ciCapabilities, &Parameter CI-Capabilities}
manuDeviceID            IPARAM::={&paramRef c-manuDeviceID, &Parameter ManuDeviceID}
iTS-scuId               IPARAM::={&paramRef c-iTS-scuId, &Parameter ITS-scuId}
medID                   IPARAM::={&paramRef c-medID, &Parameter MedID}
localCIID               IPARAM::={&paramRef c-localCIID, &Parameter LocalCIID}
```

```
timeoutReg              IPARAM::={&paramRef c-timeoutReg, &Parameter TimeoutRegistration}
llAddress               IPARAM::={&paramRef c-llAddress, &Parameter LLaddress}
llAddressTemp           IPARAM::={&paramRef c-llAddressTemp, &Parameter LLaddressTemp}
ciClass                 IPARAM::={&paramRef c-ciClass, &Parameter CIclass}
ciaClass                IPARAM::={&paramRef c-ciaClass, &Parameter CIaClass}
ciStatus                IPARAM::={&paramRef c-ciStatus, &Parameter CIstatus}
notify                  IPARAM::={&paramRef c-notify, &Parameter Notify}
medType                 IPARAM::={&paramRef c-medType, &Parameter ITSatt}
regulatoryScheme        IPARAM::={&paramRef c-regulatoryScheme, &Parameter RegulatoryScheme}
connect                 IPARAM::={&paramRef c-connect, &Parameter Connect}
simPin                  IPARAM::={&paramRef c-simPin, &Parameter SimPin}
providerInfo            IPARAM::={&paramRef c-providerInfo, &Parameter ProviderInfo}
minUserPriority         IPARAM::={&paramRef c-minUserPriority, &Parameter
MinimumUserPriority}
                        -- DEFAULT 0
queueLevel              IPARAM::={&paramRef c-queueLevel, &Parameter QueueLevelActual}
queueLowTh              IPARAM::={&paramRef c-queueLowTh, &Parameter QueueLowThreshold}
queueAlarmTh            IPARAM::={&paramRef c-queueAlarmTh, &Parameter QueueAlarmThreshold}
distancePeer            IPARAM::={&paramRef c-distancePeer, &Parameter DistancePeer}
commRangeRef            IPARAM::={&paramRef c-commRangeRef, &Parameter
CommuniationRangeReference}
timeOfLastRecep         IPARAM::={&paramRef c-timeOfLastRecep, &Parameter
TimeOfLastReception}
inactTimeLimit          IPARAM::={&paramRef c-inactTimeLimit, &Parameter InactTimeLimit}
mediumUsage             IPARAM::={&paramRef c-mediumUsage, &Parameter MediumUsage}
medUseObsTime           IPARAM::={&paramRef c-medUseObsTime, &Parameter MedUseObsTime}
llAddressPeer           IPARAM::={&paramRef c-llAddressPeer, &Parameter LLaddressPeer}
virtualCIs              IPARAM::={&paramRef c-virtualCIs, &Parameter VirtualCIs}
minPrioCrossCI          IPARAM::={&paramRef c-minPrioCrossCI, &Parameter
MinimumCrossCiPriority}
rxSens                  IPARAM::={&paramRef c-rxSens, &Parameter RxSens}
txPower                 IPARAM::={&paramRef c-txPower, &Parameter TxPowerActual}
txPowMax                IPARAM::={&paramRef c-txPowMax, &Parameter TxPowerMax}
peerTXpower             IPARAM::={&paramRef c-peerTXpower, &Parameter PeerTXpower}
dataRateLink            IPARAM::={&paramRef c-dataRateLink, &Parameter DataRateLink}
dataRateNW              IPARAM::={&paramRef c-dataRateNW, &Parameter DataRateNetwork}
dataRatesNW             IPARAM::={&paramRef c-dataRatesNW, &Parameter DataRatesNetwork}
dataRateNWreq           IPARAM::={&paramRef c-dataRateNWreq, &Parameter
DataRateNetworkRequired}
directivity             IPARAM::={&paramRef c-directivity, &Parameter Directivity}
blockLength             IPARAM::={&paramRef c-blockLength, &Parameter BlockLength}
freeAirTime             IPARAM::={&paramRef c-freeAirTime, &Parameter FreeAirTime}
frameLengthMax          IPARAM::={&paramRef c-frameLengthMax, &Parameter FrameLengthMax}
itsScuKinematicState    IPARAM::={&paramRef c-itsScuKinematicState, &Parameter
                        ItsScuKinematicState}
kineVectOut             IPARAM::={&paramRef c-kineVectOut, &Parameter KineVectOut}
cost                    IPARAM::={&paramRef c-cost, &Parameter MediumCost}
reliability             IPARAM::={&paramRef c-reliability, &Parameter Reliability}
logicalChannels         IPARAM::={&paramRef c-logicalChannels, &Parameter LogicalChannels}
limitChannelAccess      IPARAM::={&paramRef c-limitChannelAccess, &Parameter
LimitChannelAccess}
physicalChannelId       IPARAM::={&paramRef c-physicalChannelId, &Parameter
                        PhysicalChannelIdentifier}
totalBatteryEnergy      IPARAM::={&paramRef c-totalBatteryEnergy, &Parameter
TotalBatteryEnergy}
frequencyBand           IPARAM::={&paramRef c-frequencyBand, &Parameter FrequencyBand}
channelParam            IPARAM::={&paramRef c-channelParam, &Parameter ChannelParam}
operationalMode         IPARAM::={&paramRef c-operationalMode, &Parameter OperationalMode}
itsAID                  IPARAM::={&paramRef c-itsAID, &Parameter ITSaid}
qosRequirements         IPARAM::={&paramRef c-qosRequirements, &Parameter QoSrequirement}

-- I-Parameter Types
CommProfile::=SEQUENCE OF I-Param

/* to be defined later in compliance with ISO 24102-6. New name of I-Parameter "Properties"
CI-Capabilities ::=
*/

ManuDeviceID::=UTF8String

MedID::=INTEGER(0..255)
```

```
LocalCIID::=EUI64

TimeoutRegistration ::= TimeDurationValue
LLaddress::=MediumSpecificFormats{{LLaddresses}}
LLaddresses MEDSPEC::={
    { MacAddressStatic IDENTIFIED BY c-ITSatt-iso21215 } |
    { LTE-V2X-Layer2Address IDENTIFIED BY c-ITSatt-iso17515 } ,
    ...
    }

MacAddressStatic::=MACaddress

LLaddressTemp::=MediumSpecificFormats{{LLaddressesTemp}}

LLaddressesTemp MEDSPEC::={
    { MacAddressTemp IDENTIFIED BY c-ITSatt-iso21215 } ,
    ...
    }
MacAddressTemp::=MACaddress

CIclass::=SEQUENCE OF CIclassSingle

CIclassSingle::=INTEGER{
    unknown         (0),
    cic-l1          (1),
    cic-l2          (2),
    cic-l3          (3),
    cic-l4          (4),
    cic-l5          (5),
    cic-l6          (6),
    cic-il1         (254),
    cic-il2         (255)
    } (0..255)

CIaClass::=INTEGER{
    unknown          (0),
    ciac-1           (1),
    ciac-2           (2),
    ciac-3           (3)
    } (0..255)

CIstatus::=INTEGER{
    not-existent     (0),
    existent         (1),
    unknown          (2),
    registered       (4),
    active           (8),
    connected        (16),
    suspended        (64),
    inactive         (128)
    } (0..255)

Notify::=SEQUENCE (SIZE(0..255)) OF INTEGER(0..255) -- valid parameter number

Connect::=INTEGER{
    automatic        (0),
    manual           (255)
    } (0..255)

SimPin::=MediumSpecificFormats{{SimPins}}

SimPins MEDSPEC::={
    { SimPinAny IDENTIFIED BY c-ITSatt-any } ,
    ...
    }

SimPinAny::=OCTET STRING
ProviderInfo::=SEQUENCE{
    provName     OCTET STRING (SIZE(0..255)), -- Name of provider
    apn          OCTET STRING (SIZE(0..255)), -- Name of access point
```

```
    username    OCTET STRING (SIZE(0..255)), -- Log-in name of user
    password    OCTET STRING (SIZE(0..255)) -- Password for log-in
    }

MinimumUserPriority::=UserPriority

QueueLevelActual::=QueueLevel

QueueLowThreshold::=QueueLevel

QueueAlarmThreshold::=QueueLevel

DistancePeer::=Distance

CommuniationRangeReference::=Distance

TimeOfLastReception::=Time48IAT

InactTimeLimit::=TimeDurationValue

MediumUsage::=SEQUENCE{
    receive     INTEGER(0..255),
    transmit    INTEGER(0..255)
    }

MedUseObsTime::=TimeDurationValue

LLaddressPeer::=MediumSpecificFormats{{LLaddressesPeer}}

LLaddressesPeer MEDSPEC::={
    { PeerMacAddress IDENTIFIED BY c-ITSatt-iso21215 } ,
    ...
    }

PeerMacAddress::=MACaddress

VirtualCIs::=SEQUENCE (SIZE(0..65535))OF RemoteCIID

MinimumCrossCiPriority::=UserPriority

RxSens::=MediumSpecificFormats{{RxSenss}}

RxSenss MEDSPEC::={
    { RxSens80211    IDENTIFIED BY c-ITSatt-iso21215 } ,
    ...
    }
TxPowerActual::=TxPower

TxPowerMax::=TxPower

PeerTXpower::=TxPower

DataRateLink::=MediumSpecificFormats{{DataRatesLink}}

DataRatesLink MEDSPEC::={
    { DataRateLinkAny IDENTIFIED BY c-ITSatt-any } |
    { DataRate80211 IDENTIFIED BY c-ITSatt-iso21215 } ,
    ...
    }

DataRateLinkAny::=DataRate

DataRateNetwork::=DataRate

DataRatesNetwork::=SEQUENCE{       -- available to the appl.
    minimum          DataRate,     -- minimum possible value
    maximum          DataRate      -- maximum possible value
    }

DataRateNetworkRequired::=DataRate
```

```
Directivity::=SEQUENCE{
    mode          DirMode,
    dirPredef     INTEGER(0..255), -- 0: see dirVar
                      -- >0: predefined direction
    dirVar        SEQUENCE (SIZE(0..255)) OF DirVar
    }

DirMode::=INTEGER{
    fixed             (0),
    tracking           (255)
    } (0..255)

DirVar::=SEQUENCE{
    bsAzimuth          INTEGER(-32768..32767) ,     -- -180 - +180
    bsElevation        INTEGER(-128..127) ,         -- -90 - +90
    openHorizontal     INTEGER(0..255) ,            -- 0 - 180
    openVertical       INTEGER(0..255)              -- 0 - 180
    }

BlockLength::=INTEGER(0..65535)

FreeAirTime::=TimeDurationValue

FrameLengthMax::=TimeDurationValue

ItsScuKinematicState::=SEQUENCE{
    dut               Time48IAT, -- date and universal time
    lat               Latitude, -- latitude
    lon               Longitude, -- longitude
    alt               Altitude, -- altitude / elevation
    gs                Gs, -- ground speed
    tta               Tta -- true track angle
    }

KineVectOut::=SEQUENCE{
    dut               Time48IAT, -- date and universal time
    options           SEQUENCE OF KineVectOption -- options
    }

Reliability::=INTEGER{
    unreliable      (0),
    halfPercent     (1),
    perfect         (200), -- 100%
-- (201) through    (254) reserved
    unknown         (255)
    }(0..255)

LogicalChannels::=SEQUENCE OF MappingLogicalPhysicalChannel

MappingLogicalPhysicalChannel::=SEQUENCE{
    logChannelID      LogicalChannelType,
    physChannelID     PhysicalChannelIdentifier
    }
-- Time limitation of access to physical channel (move to 21218)
LimitChannelAccess::=SEQUENCE{
    channel        PhysicalChannelIdentifier, -- unique in a CI
    percentage     LimitChannelAccessPercentage,
    period         LimitChannelAccessPeriod
    }

PhysicalChannelIdentifier::=MediumSpecificFormats{{PhysicalChannelIds}}

PhysicalChannelIds MEDSPEC::={
    { ChannelNumber80211 IDENTIFIED BY c-ITSatt-iso21215 } ,
    ...
    }

LimitChannelAccessPercentage::=INTEGER{
    noAccess       (0),
    onePercent     (2),
    twoPercent     (4),
```

```
    always          (200)
    }(0..255)       -- steps of 0,5 %, >200: no limit

LimitChannelAccessPeriod::=INTEGER{
noLimit     (0),
ms100       (1), -- 100ms observation interval
second      (10), -- 1s observation interval
max         (255) -- 25,5 s observation interval
} (0..255)    -- observation interval in 100 ms

OperationalMode::=MediumSpecificFormats{{OperationalModes}}

OperationalModes MEDSPEC::={
    { LTE-V2X-OperationalMode IDENTIFIED BY c-ITSatt-iso17515 } |
    { ITS-M5-OperationalMode IDENTIFIED BY c-ITSatt-iso21215 } ,
    ...
    }

QoSrequirement::=MediumSpecificFormats{{QoSrequirements}}

QoSrequirements MEDSPEC::={
    { ITS-M5-TIDcontrol IDENTIFIED BY c-ITSatt-iso21215 } ,
    ...
    }

-- IN Service Access Point
-- from ITS-S N&T Layer to ITS-S Access Layer
RefINsapRequest ::= INTEGER {
    c-inUnitdataRq        (0),
    c-inUnitdataAckRq     (1)
    } (0..255)

INSAPREQ ::= CLASS {
    &primitiveRef RefINsapRequest UNIQUE,
    &Primitive
    }

INsapRequestSps INSAPREQ::={inUnitdataRq | inUnitdataAckRq , ...}

INsapRequest::=SEQUENCE{
    spRef             INSAPREQ.&primitiveRef({INsapRequestSps}),
    servPrimitive     INSAPREQ.&Primitive({INsapRequestSps}{@spRef})
    }

-- For backward compatibility (used in ISO 24102-3)
INsapPrimitivesDown::=INsapRequest

inUnitdataRq       INSAPREQ::={&primitiveRef c-inUnitdataRq, &Primitive IN-UNITDATA-
request}
inUnitdataAckRq    INSAPREQ::={&primitiveRef c-inUnitdataAckRq, &Primitive IN-UNITDATA-
ACK-request}

IN-UNITDATA-request::=SEQUENCE{
    fill              BIT STRING (SIZE(4)), -- set to '0000'
    addresses          UnitdataAddresses,
    data               INdata,
    priority           UserPriority OPTIONAL,
    timeout            Time48IAT OPTIONAL,
    accessParams       AccessParameters OPTIONAL
    }

UnitdataAddresses::=CHOICE{
    flowID             [0] FlowID,
    explicitAddresses  [1] UnitdataExplicitAddresses
    }

UnitdataExplicitAddresses::=SEQUENCE{
    nt-protocol-id     Ethertype,
    source-addr        SourceLinkID,
    dest-addr          DestinationLinkID
    }
```

```
IN-UNITDATA-ACK-request::=SEQUENCE{
    fill                BIT STRING (SIZE(4)), -- set to '0000'
    addresses           UnitdataAddresses,
    data                INdata,
    priority            UserPriority OPTIONAL,
    timeout             Time48IAT OPTIONAL,
    accessParams        AccessParameters OPTIONAL
    }

-- from ITS-S Access Layer to ITS-S N&T Layer
RefINsapIndication ::= INTEGER {
    c-inUnitdataInd          (0),
    c-inUnitdataStatusInd    (1)
    } (0..255)

INSAPIND::= CLASS {
    &primitiveRef RefINsapIndication UNIQUE,
    &Primitive
    }

INsapIndicationSps INSAPIND::={inUnitdataInd | inUnitdataStatusInd , ...}

INsapIndication::=SEQUENCE{
    spRef            INSAPIND.&primitiveRef ({INsapIndicationSps}),
    servPrimitive    INSAPIND.&Primitive ({INsapIndicationSps}{@spRef})
    }

-- For backward compatibility (used in ISO 24102-3)
INsapPrimitivesUp::=INsapIndication

inUnitdataInd    INSAPIND::={&primitiveRef c-inUnitdataInd, &Primitive
IN-UNITDATA-indication}
inUnitdataStatusInd INSAPIND::={&primitiveRef c-inUnitdataStatusInd, &Primitive
IN-UNITDATA-STATUS-indication}

IN-UNITDATA-indication::=SEQUENCE{
    nt-protocol-id    Ethertype,
    source-addr       SourceLinkID,
    dest-addr         DestinationLinkID,
    data              INdata,
    priority          UserPriority,
    accessParams      AccessParameters
    }

IN-UNITDATA-STATUS-indication::=SEQUENCE{
    fill              BIT STRING (SIZE(4)), -- set to '0000'
    addresses         UnitdataAddresses,
    data              INdata,
    priority          UserPriority OPTIONAL,
    timeout           Time48IAT OPTIONAL,
    accessParams      AccessParameters OPTIONAL,
    txStatus          INtxStatus,
    macServiceClass   MACServiceClass
    }

-- IN-SAP function parameters
INdata::=OCTET STRING (SIZE(0..65535))

RefACCPARAM ::= INTEGER {
    c-nullAP       (0),
    c-txParamAP    (1),
    c-rxParamAP    (2)
    }

ACCPARAM ::= CLASS {
    &paramRef RefACCPARAM UNIQUE,
    &Parameter
    }
udNullAP ACCPARAM::={&paramRef c-nullAP, &Parameter NullType}
udTxAP ACCPARAM::={&paramRef c-txParamAP, &Parameter UdTxAP}
```

```
udRxAP ACCPARAM::={&paramRef c-rxParamAP, &Parameter UdRxAP}

UdTxAP::=SEQUENCE OF I-Param

UdRxAP::=SEQUENCE OF I-Param

AccParams ACCPARAM::={udNullAP | udTxAP | udRxAP, ...}

AccessParameters::=SEQUENCE OF AccessParameterSet

AccessParameterSet::=SEQUENCE{
    apRef         ACCPARAM.&paramRef ({AccParams}),
    aParameter    ACCPARAM.&Parameter ({AccParams}{@apRef})
    }

MACServiceClass::=INTEGER{
    macAckNotUsed    (0),
    macAckUsed       (255)
    } (0..255)

INtxStatus::=INTEGER{
    success          (0),
    queueFull        (1),
    timeout          (2),
    priority         (4),
    noVCI            (8),
    noCI             (16),
    unspecFailure    (128)
    } (0..255)

-- MI-SAP
-- MI-Commands (to be registered in ISO 24102-3)
ChangePseudonymMACaddress::=NullType

ChangePseudonymMACaddressConf::=NullType

CIstateChange::=INTEGER{
    deregister       (0),
    activate         (4),
    resume           (8),
    connect          (16),
    disconnect       (32),
    suspend          (64),
    inactivate       (128)
    }(0..255)

CIstateChangeConf::=NullType

CnConnect::=INTEGER{
    deleteAC         (0),
    connect          (1),
    disconnect       (255)
    }(0..255)

CnConnectConf::=NullType

MacManagementFrameTX::=UnitData

MacManagementFrameTXConf::=NullType

ManufacturerSpecificCiAccess::=OCTET STRING

ManufacturerSpecificCiAccessConf::=OCTET STRING

MonitorIparameter::=SEQUENCE OF MonitorDetail

MonitorDetail::=SEQUENCE{
    paramNo   I-ParamNo, -- valid parameter number
    action    MonitorDetailAction -- start / stop monitoring
    }
```

```
MonitorDetailAction::=INTEGER{
    stop  (0),
    start (255)
    }(0..255)

MonitorIparameterConf::=SEQUENCE OF MonitorError

MonitorError::=SEQUENCE{
    paramNo    I-ParamNo, -- I-Parameter number for which monitoring failed
    errorCode  MonitorErrorCode
    }

MonitorErrorCode::=INTEGER{
    success           (0),
    unspecified       (1),
    invalidParamNo    (2),
    accessViolation   (7)
    } (0..255)

PrioritizedRTS::=SEQUENCE{
    seqNo       RTSseqNo,
    reqID       RTScmdReqID,
    priority    UserPriority
    }

RTScmdReqID::=SEQUENCE{
    linkID    LocalCIID
    }

PrioritizedRTSConf::=SEQUENCE{
    seqNo    RTSseqNo,
    reqID    RTScmdReqID,
    status   RTSstatus -- only 64 and 128 applicable
    }

PrioritizedRTSrelease::=SEQUENCE{
    seqNo    RTSseqNo
    }

PrioritizedRTSreleaseConf::=NullType

RIprovisioning::=RegulatoryScheme

RIprovisioningConf::=NullType

VciCmd::=SEQUENCE{
    linkID   Link-ID,
    alive    Logic OPTIONAL
    }

VciCmdConf::=NullType

WakeUp::=INTEGER{
    stopTX           (0),
    repetition1ms    (1),
    repetition255ms  (255)
    }(0..255)

WakeUpConf::=NullType

-- MI-REQUESTs (to be registered in ISO 24102-3)
RefEVENT21218::=INTEGER{
    c-minUserPrio          (0),
    c-txQueueThreshold     (1),
    c-txQueueFull          (2),
    c-vciCreated           (3),
    c-vciDeleted           (4),
    c-paramMonitor         (5),
    c-txQueueLow           (6),
    c-vciReset             (7),
    c-macAddressConflict   (8)
```

```
    } (0..255)

EVENT21218::=CLASS {
    &eventRef RefEVENT21218 UNIQUE,
    &Event21218
    }

Event21218Notification::=SEQUENCE{
    eventNo  EVENT21218.&eventRef({Events-21218}),
    event    EVENT21218.&Event21218({Events-21218} {@eventNo})
    }

Events-21218 EVENT21218::={minUserPrio | txQueueThreshold | txQueueFull | vciCreated |
 vciDeleted | paramMonitor | txQueueLow | vciReset | macAddressConflict, ...}

minUserPrio
    EVENT21218::={&eventRef c-minUserPrio, &Event21218 E21218-0}
txQueueThreshold
    EVENT21218::={&eventRef c-txQueueThreshold, &Event21218 E21218-1}
txQueueFull
    EVENT21218::={&eventRef c-txQueueFull, &Event21218 E21218-2}
vciCreated
    EVENT21218::={&eventRef c-vciCreated, &Event21218 E21218-3}
vciDeleted
    EVENT21218::={&eventRef c-vciDeleted, &Event21218 E21218-4}
paramMonitor
    EVENT21218::={&eventRef c-paramMonitor, &Event21218 E21218-5}
txQueueLow
    EVENT21218::={&eventRef c-txQueueLow, &Event21218 E21218-6}
vciReset
    EVENT21218::={&eventRef c-vciReset, &Event21218 E21218-7}
macAddressConflict
    EVENT21218::={&eventRef c-macAddressConflict, &Event21218 E21218-8}
E21218-0::=SEQUENCE{
    priority    UserPriority,
    linkID    Link-ID
    }

E21218-1::=UserPriority

E21218-2::=UserPriority

E21218-3::=Link-ID

E21218-4::=Link-ID

E21218-5::=I-Param

E21218-6::=UserPriority

E21218-7::=Link-ID

E21218-8::=Link-ID

Event21218NotificationConf::=NullType

MacManagementFrameRX::=UnitData

MacManagementFrameRXConf::=NullType

PositionUpdateRequest::=INTEGER{
    stopUpdate       (0),
    update1ms        (1),
    update65535ms    (65535)
    }(0..65535)

PositionUpdateRequestConf::=NullType

PrioritizationRegistration::=SEQUENCE{
    interferers    PrioRegInterferers,
    timeout        PrioRegTimeout
```

```
    }
PrioRegInterferers::=SEQUENCE OF ITSatt

PrioRegTimeout::=TimeDurationValue

PrioritizationRegistrationConf::=NullType

PrioritizationRequest::=SEQUENCE{
    seqNo       RTSseqNo,
    priority    UserPriority
    }

RTSseqNo::=INTEGER(0..255)

PrioritizationRelease::=SEQUENCE{
    seqNo    RTSseqNo
    }

PrioritizationReleaseConf::=NullType

RTSstatus::=INTEGER {
    unknown      (0),
    ignored      (1), -- request ignored
    requested    (16),
    released     (32),
    granted      (128) -- request granted
    }(0..255)

PrioritizationRequestConf::=SEQUENCE{
    seqNo       RTSseqNo,
    priority    UserPriority,
    status      RTSstatus -- only 1 and 128 applicable
    }

RegistrationCI::=SEQUENCE{
    linkID     Link-ID,
    medType    ITSatt
    }

RegistrationCIconf::=SEQUENCE{
    scuId    ITS-scuId,
    medID    MedID
    }

-- MI-SAP general definitions
UnitData::=SEQUENCE{
    sourceAddr     Link-ID,
    destAddr       Link-ID,
    data           MframeData,
    priority       UserPriority,
    parameter      MframeParameter -- tbd dependent on medium
    }

MframeData::=OCTET STRING (SIZE(0..65535))

MFRAME::=CLASS {
    &medID MframeRef UNIQUE, -- identifies access technology
    &FrameParameter
    }

MframeRef::=ITSatt

MframeParameter::=SEQUENCE{
    medium       MFRAME.&medID({MframeParams}),
    parameter    MFRAME.&FrameParameter({MframeParams}{@medium})
    }

MframeParams MFRAME::={m5-MframeParameter, ...}

m5-MframeParameter     MFRAME::={&medID c-ITSatt-iso21215, &FrameParameter
                       M5-MgmtSubTypes}
```

```
-- Values

/*
   The ASN.1 specification has been checked for conformance to the ASN.1
   standards by OSS ASN.1 Syntax Checker, and by OSS ASN-1STEP
*/

END
```

B.3 Definitions to be added in ISO 24102-3

The following definitions are dynamic extensions of types defined in ISO 24102-3 with CLASS. An up-to-date version of the ASN.1 modules:

— ITSsapCmdMi { iso (1) standard (0) calm-management (24102) msap (3) commands (3) mi (4) version0 (0)};

— ITSsapReqMi { iso (1) standard (0) calm-management (24102) msap (3) requests (4) mi (4) version0 (0)}.

including all dynamic updates is published on the ISO standards maintenance portal at [25].

IMPORT statement to be added:

```
/* MI-COMMAND */
CIstateChange, CIstateChangeConf, CnConnect, CnConnectConf, MacManagementFrameTX,
MacManagementFrameTXConf, ManufacturerSpecificCiAccess, ManufacturerSpecificCiAccessConf,
MonitorIparameter, MonitorIparameterConf, PrioritizedRTS, PrioritizedRTSConf,
RIprovisioning, RIprovisioningConf, VciCmd, VciCmdConf, WakeUp, WakeUpConf,
INsapIndication, I-Param, I-ParamNo, Link-ID FROM ITSllsap {iso(1) standard(0)
calm-ll-sap(21218) asnm-1 (1) version2 (2)}
/* MI-REQUEST */
Event21218Notification, Event21218NotificationConf, MacManagementFrameTX,
MacManagementFrameRXConf, PositionUpdateRequest, PositionUpdateRequestConf,
PrioritizationRegistration, PrioritizationRegistrationConf, PrioritizationRequest,
PrioritizationRequestConf, RegistrationCI, RegistrationCIconf, INsapIndication, Link-ID
FROM ITSllsap {iso(1) standard(0) calm-ll-sap(21218) asnm-1 (1) version2 (2)}
```

To be added to MI-Command:

```
cIstateChange           MISAP-CR::={&mxref c-cIstateChange, &MXParam CIstateChange}
cNconnect               MISAP-CR::={&mxref c-cNconnect, &MXParam CnConnect}
macManagementFrameTX    MISAP-CR::={&mxref c-macManagementFrameTX, &MXParam
                        MacManagementFrameTX}
manufacturerSpecificCiAccess    MISAP-CR::={&mxref c-manufacturerSpecificCiAccess,
                        &MXParam ManufacturerSpecificCiAccess}
monitorIparameter       MISAP-CR::={&mxref c-monitorIparameter, &MXParam
                        MonitorIparameter}
prioritizedRTS          MISAP-CR::={&mxref c-prioritizedRTS, &MXParam PrioritizedRTS}
rIprovisioning          MISAP-CR::={&mxref c-rIprovisioning, &MXParam RIprovisioning}
vciCmd                  MISAP-CR::={&mxref c-vciCmd, &MXParam VciCmd}
wakeUp                  MISAP-CR::={&mxref c-wakeUp, &MXParam WakeUp}

MI-Command MISAP-CR::={cIstateChange | cNconnect | macManagementFrameTX |
manufacturerSpecificCiAccess | monitorIparameter | prioritizedRTS | rIprovisioning |
vciCmd | wakeUp, ...}

cIstateChangeConf           MISAP-CC::={&mxref c-cIstateChange, &MXParam
                            CIstateChangeConf}
cNconnectConf               MISAP-CC::={&mxref c-cNconnect, &MXParam CnConnectConf}
macManagementFrameTXConf    MISAP-CC::={&mxref c-macManagementFrameTX, &MXParam
MacManagementFrameTXConf}
manufacturerSpecificCiAccessConf    MISAP-CC::={&mxref c-manufacturerSpecificCiAccess,
                            &MXParam ManufacturerSpecificCiAccessConf}
monitorIparameterConf       MISAP-CC::={&mxref c-monitorIparameter, &MXParam
                            MonitorIparameterConf}
prioritizedRTSConf          MISAP-CC::={&mxref c-prioritizedRTS, &MXParam
                            PrioritizedRTSConf}
rIprovisioningConf          MISAP-CC::={&mxref c-rIprovisioning, &MXParam
                            RIprovisioningConf}
```

```
vciCmdConf                  MISAP-CC::={&mxref c-vciCmd, &MXParam VciCmdConf}
wakeUpConf                  MISAP-CC::={&mxref c-wakeUp, &MXParam WakeUpConf}

MI-CmdConfirm MISAP-CC::={cIstateChangeConf | cNconnectConf | macManagementFrameTXConf |
manufacturerSpecificCiAccessConf | monitorIparameterConf | prioritizedRTSConf |
rIprovisioningConf | vciCmdConf | wakeUpConf, ...}
```

To be added to MI-Request:

```
MI-Request MISAP-RR::={event21218Notification | macManagementFrameRX |
positionUpdateRequest
| prioritizationRegistration | prioritizationRequest | registrationCI , ...}
event21218Notification        MISAP-RR::={&mxref c-event21218Notification, &MXParam
                              Event21218Notification}
macManagementFrameRX          MISAP-RR::={&mxref c-macManagementFrameRX, &MXParam
                              MacManagementFrameRX}
positionUpdateRequest         MISAP-RR::={&mxref c-positionUpdateRequest, &MXParam
                              PositionUpdateRequest}
prioritizationRegistration    MISAP-RR::={&mxref c-prioritizationRegistration, &MXParam
                              PrioritizationRegistration}
prioritizationRequest         MISAP-RR::={&mxref c-prioritizationRequest, &MXParam
                              PrioritizationRequest}
registrationCI                MISAP-RR::={&mxref c-registrationCI, &MXParam
                              RegistrationCI}

MI-ReqConfirm MISAP-RR::={event21218Notification | macManagementFrameRX |
positionUpdateRequest | prioritizationRegistration | prioritizationRequest |
registrationCI , ...}
event21218NotificationConf    MISAP-RC::={&mxref c-event21218Notification, &MXParam
                              Event21218NotificationConf}
macManagementFrameRXConf      MISAP-RC::={&mxref c-macManagementFrameRX, &MXParam
                              MacManagementFrameRXConf}
positionUpdateRequestConf     MISAP-RC::={&mxref c-positionUpdateRequest, &MXParam
                              PositionUpdateRequestConf}
prioritizationRegistrationConf    MISAP-RC::={&mxref c-prioritizationRegistration,
                              &MXParam PrioritizationRegistrationConf}
prioritizationRequestConf     MISAP-RC::={&mxref c-prioritizationRequest, &MXParam
                              PrioritizationRequestConf}
registrationCIConf            MISAP-RC::={&mxref c-registrationCI, &MXParam
                              RegistrationCIconf}
```

The following ASN.1 value definitions (values 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o') will be completed by ISO TC204, i.e. numerical values will be assigned. The result will be published in the respective ISO registry on the ISO standards maintenance portal at [25].

```
c-cIstateChange                 RefMISAP-C ::= <'a' tbd>
c-cNconnect                     RefMISAP-C ::= <'b' tbd>
c-macManagementFrameTX          RefMISAP-C ::= <'c' tbd>
c-manufacturerSpecificCiAccess  RefMISAP-C ::= <'d' tbd>
c-monitorIparameter             RefMISAP-C ::= <'e' tbd>
c-prioritizedRTS                RefMISAP-C ::= <'f' tbd>
c-rIprovisioning                RefMISAP-C ::= <'g' tbd>
c-vciCmd                        RefMISAP-C ::= <'h' tbd>
c-wakeUp                        RefMISAP-C ::= <'i' tbd>

c-event21218Notification        RefMISAP-R ::= <'j' tbd>
c-macManagementFrameRX          RefMISAP-R ::= <'k' tbd>
c-positionUpdateRequest         RefMISAP-R ::= <'l' tbd>
c-prioritizationRegistration    RefMISAP-R ::= <'m' tbd>
c-prioritizationRequest         RefMISAP-R ::= <'n' tbd>
c-registrationCI                RefMISAP-R ::= <'o' tbd>
```

Annex C
(normative)

Extended universal 64 bit identifier

C.1 EUI-64 format

The IEEE defined 64-bit extended unique identifier (EUI-64) is illustrated in Figure C.1 .

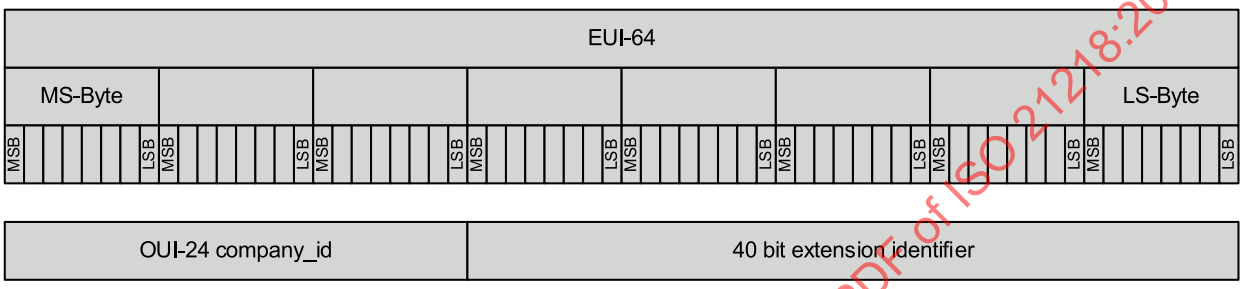

Figure C.1 — EUI-64 format

EUI-64 is given by the concatenation of two parts:

— 24 bit "Universal Object Identifier" (OUI-24) company_id;

— 40 bit extension identifier.

The "Distinct Null Identifier" (DNI) is given by all bits in EUI-64 being set to '1', i.e. the value FF-FF-FF-FF-FF-FF-FF-FF$_{16}$.

IEEE_RA administers the assignment of OUI-24 values.

The following values are prohibited as EUI-64 values in order to enable encapsulation of EUI-48 numbers and 48-bit MAC addresses in the EUI-64 field:

a) cc-cc-cc-FF-FE-ee-ee-ee;

b) cc-cc-cc-FF-FF-ee-ee-ee.

where the letters 'c' and 'e' represent hexadecimal digits. The reserved pattern FF-FE$_{16}$ identifies encapsulation of an EUI-48 value in an EUI-64 field. The reserved pattern FF-FF$_{16}$ identifies encapsulation of a 48-bit MAC address in an EUI-64 field. According to [20] common practice in IPv6 is to use the reserved pattern FF-FE$_{16}$ to identify encapsulation of a 48-bit MAC address in an EUI-64 field.

C.2 Encapsulation of 48-bit MAC addresses

The encapsulation of 48-bit MAC addresses in an EUI-64 field as specified in [20] is illustrated in Figure C.2.

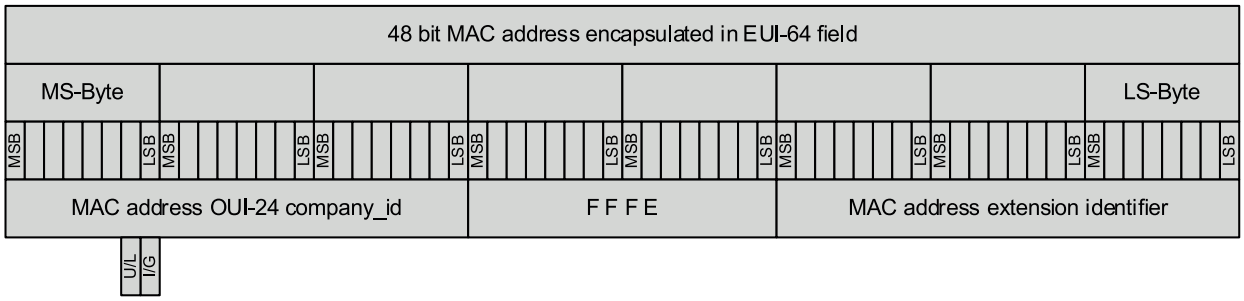

Figure C.2 — Encapsulation of 48-bit MAC address in EUI-64 field[20]

The "U/L"-bit indicates, whether the MAC address is:

— a universal address (globally unique), indicated by the value '0';

— or a locally administered address, indicated by the value '1'.

The "I/G"-bit indicates, whether the MAC address is:

— an individual address (for unicast communication), indicated by the value '0';

— or a group address (for broadcast or multicast communication), indicated by the value '1'.

With "U/L" set to '1', all remaining bits of the MAC address may be used to distinguish locally administered addresses.

C.3 Encapsulation of identifiers specific to ITS

Identifiers for usage in Link-ID (RemoteCIID, LocalCIID) being specific to ITS-S are needed for CIs which do not support 48-bit MAC addresses. Such identifiers are referred to as LegacyCIID. The approach is illustrated in Figure C.3.

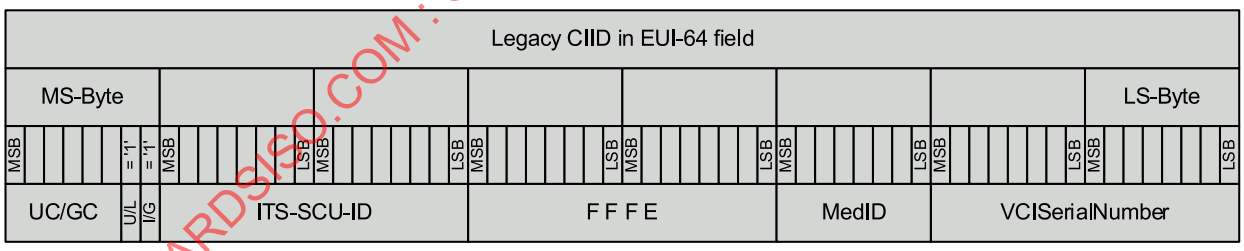

Figure C.3 — Encapsulation of identifiers specific to ITS

The encapsulated LegacyCIID shall be constructed in a unique way as follows.

a) Set the "UC/GC" field illustrated in Figure C.2 to either '111111'$_2$ indicating broadcast communication, or to '000000'$_2$ indicating unicast communication, or to one of the other possible values to indicate multicast communication. This allows for the definition of 62 multicast groups.

b) Set the "U/L" bit to '1'$_2$ and the "I/G" bit to '1'$_2$ indicating the encapsulation mode specified here.

c) The "ITS-SCU-ID" field contains the ITS-SCU-ID of ASN.1 type ITS-scuId specified in ISO 17419. ITS-SCU-ID is introduced in ISO 24102-1[12] and used in e.g. ISO 24102-4[13] and ISO 29281-1[19].

d) The "FF EE" field contains the reserved pattern FF-FE$_{16}$, identifying an encapsulated MAC address field.

e) The "MedID" field contains a CI identifier (MedID) being unique in an ITS-SCU.

f) The "VCISerialNumber" field contains a serial number (VCISerialNumber).

- The "VCISerialNumber" $FFFF_{16}$, is used to identify a remote broadcast or multicast receiver. Any other value is used if either multicast communication or unicast communication is selected by the "UC/GC" field.
- The "VCISerialNumber" 0000_{16} in combination with "UC/GC" set to $'000000'_2$ is used to identify the local CI.
- The "VCISerialNumbers" 0001_{16} through $FFFF_{16}$ in combination with "UC/GC" set to $'000000'_2$ are used to identify a remote unicast receiver.

Other rules may apply for specific access technologies, e.g. LTE-V2X specified in ISO 17515-3[5].