GSS

Global Specification for Short Range Communication

The platform
for
Interoperable
Electronic Toll Collection
and Access Control

kapsch >>> THALES

GSS

Global Specification for Short Range Communication

Kapsch TrafficCom AB

Kapsch Telecom GmbH

Thales e-Transactions CGA SA

LEGAL NOTICE

Use of or quotations to this documents is subject to the following restrictions and conditions

- THIS SPECIFICATION IS PROVIDED ON AN "AS IS" BASIS AND KAPSCH TRAFFICCOM AB, KAPSCH TELECOM GMBH AND THALES E-TRANSACTIONS CGA SA DO NOT MAKE ANY WARRANTY OF ANY KIND, WHETHER EXPRESS OR IMPLIED (INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE) THEREON FURTHER, KAPSCH TRAFFICCOM AB, KAPSCH TELECOM GMBH AND THALES E-TRANSACTIONS CGA SA DO NOT WARRANT, GUARANTEE OR MAKE ANY REPRESENTATION WHATSOEVER REGARDING THE USE, IMPLEMENTATION, OR THE RESULTS OF USE OR IMPLEMENTATION OF THIS SPECIFICATION OR WRITTEN MATERIALS IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, CORRECTNESS OR OTHERWISE. THE ENTIRE RISK AS TO THE USE AND/OR THE RESULTS OF USE OF THIS SPECIFICATION IS ASSUMED SOLELY BY THE USER.

- KAPSCH TRAFFICCOM AB, KAPSCH TELECOM GMBH OR THALES E-TRANSACTIONS CGA SA SHALL NOT BE LIABLE FOR DIRECT OR INDIRECT DAMAGE OR CONSEQUENCE TO THE USER OR ANY THIRD PARTY RESULTING FROM THE USE, IMPLEMENTATION OR THE RESULTS OF THE USE OR IMPLEMENTATION OF THIS SPECIFICATION AND SHALL BE INDEMNIFIED AND HELD HARMLESS BY THE USER THEREFOR.

- KAPSCH TRAFFICCOM AB, KAPSCH TELECOM GMBH OR THALES E-TRANSACTIONS CGA SA SHALL NOT BE LIABLE FOR INFRINGEMENT OF ANY INTELLECTUAL OR INDUSTRIAL PROPERTY RIGHTS OF THIRD PARTY/IES RESULTING FROM THE USE, IMPLEMENTATION OR THE RESULTS OF THE USE OR IMPLEMENTATION OF THIS SPECIFICATION AND SHALL BE INDEMNIFIED AND HELD HARMLESS BY THE USER THEREFOR.

- KAPSCH TRAFFICCOM AB, KAPSCH TELECOM GMBH AND THALES E-TRANSACTIONS CGA SA RESERVE THE RIGHTS TO MAKE MODIFICATIONS TO THIS DOCUMENT.

The content of this specification can be used or implemented free of charge (subject to the above restrictions).

In order to avoid ambiguity in the interpretation of this specification all conformance thereto is recommended to be validated by an independent certification body.

For further information contact:

Kapsch TrafficCom AB
Box 1063
SE-55110 Jönköping
Sweden
Tel.: +46 36 194300
Fax: +46 36 194301

Kapsch Telecom GmbH
Stuttgarter Straße 61
DE-71554 Weissach im Tal
Germany
Tel.: +49 7191 3559-0
Fax: +49 7191 3559 6080

Thales e-Transactions CGA SA
BP 57
FR-91229 Bretigny-sur-Orge
France
Tel : +33 1 6988 52 00
Fax : +33 1 6988 54 05

Foreword

This specification "GSS, Global Specification for Short range communication" was originally established by the three European companies Alcatel CGA Transport, Bosch Telecom and Combitech Traffic Systems in order to achieve interoperability of Electronic Toll Collection.

These three companies were already from the start involved in the standardization process within CEN concerning Dedicated Short Range Communication and Electronic Fee Collection in order to meet the market need for stability and interoperability. They identified however early that the result of the standardization effort would not lead to a standard specific enough to support interoperability as some other players required that the standard should support alternative options. To overcome the difficulties in the standardization work and to be able to reach consensus, alternative possibilities of solutions were included in the standard by parameter setting. The consequence is however that the system integrators can use the standard differently by different selection of parameters and parameter values. Therefore an agreement and a specification of how to use the standard is required to support interoperability. That was the reason for the three companies to form an expert team to write the specification GSS, Global Specification for Short range communication. It defines how to use the CEN standard for DSRC and specifies parameter values concerning:

- default values for physical (microwave) parameters
- timing parameter and data flow control
- uplink/downlink window management
- how to handle application data
- uniformed initialization procedure to establish interoperable communication
- how to allow flexibility for operator specifications
- how to achieve interoperability, enabling the vehicle to travel through different roadside systems

With the GSS specification the market requirement of interoperable communication between the vehicles and the roadside is met. The GSS specifies exactly how to achieve such a communication. There is still flexibility to perform very short and simple communication in order to support simple On Board Equipment with low functionality, or more advanced communication, if required by the application. This is achieved through a well-specified initialization phase for establishment of the communication when the vehicle enters the communication zone. In this initialization phase a number of parameters with default values are used. Without this well specified phase the interoperability is lost. After this initialization phase the communication may continue differently in a controlled way.

GSS has already been implemented and is used in many Electronic Toll collection systems. The market response has been very strong and positive, since GSS for the first time was demonstrated during the World Congress on ITS in Orlando, Florida October 1996.

This revised version GSS 3.2 is issued by the original issuers now under their new company names: Kapsch TrafficCom AB, Kapsch Telecom GmbH and Thales e-Transactions CGA SA. GSS 3.2 is a contribution from the companies in order to support interoperability and eliminate the uncertainty regarding selection, assessment and procurement of DSRC based systems. It is now available for the market to be used free of charge.

Change History

Version 1.00	May 1997	• Original version
Version 1.01	June 1998	• Minor editorial changes
		• Update of references
		• Correction of figure numbering
Version 2.0	Feb 1999	• New ‘Legal Notice’
		• Deleted Chapter 2.3.12 Postamble
		• Chapter 5.2.3.4: extended definition of obeStatus
		• Deleted state transition table in Chapter 5.2.4.2
		• Added Chapter 6 Inter Layer Management
		• Previous Chapter 6 is now Chapter 7
		• Update of references
Version 3.0	Dec 2001	• Foreword and Introduction updated
		• Chapter 2
		2.3.2 RSE class A removed
		2.3.9 Power limits referred to the OBU
		2.3.10 Frame length for wake-up adjusted
		2.4.5 Same data on both sidebands
		2.4.6 OBU max eirp adjusted
		2.4.7 Conversion gain is maximised
		2.4.11 Preamble length is adjusted
		2.4.12 Postamble removed
		• Chapter 3
		fig 3.1 Updated
		• Chapter 4
		4.2.5 TA time definition updated
		4.2.5 Private uplink window max time updated
		4.4 Max time to start in public uplink window updated
		• Chapter 5
		5.2.2.6 Profile 0 or 1 not in Profile List
		5.2.4.2 VST creation added
		fig 5.7 Updated
		5.3 Reporting of exceptions removed
		5.4 Use of ACTION for SET_MMI added
		5.5 Tables 5.12, 5.13, 5.14 updated
		• Chapter 6
		6.3.5 Ref. 17, 18, 33, 51 corrected
		6.3.6 44, 45 clarification added
		• Chapter 7
		7.1 References updated
		7.2 Abbreviations added
Version 3.1	Jan 2003	• Foreword and Introduction updated
		• Chapter 2
		2.3.13 New heading
		2.4.6 Values adjusted
		2.4.7 Max value and Item no:s adjusted
		• Chapter 4
		4.4 N1, N8 deleted
		• Chapter 5
		5.5.4 Uplink (3) corrected
		• Chapter 7
		7.1 References updated
Version 3.2	Aug 2003	• Foreword and Introduction updated

Table of Contents

1 Introduction

The interoperability between real DSRC communication systems used for RTTT application is essential and requires the definition of precise technical characteristics. This objective is achieved by this document written by the group of three companies Kapsch TrafficCom AB, Kapsch Telecom GmbH and Thales e-Transactions CGA SA.

The main criteria considered for the elaboration of this specification are the following:

- Prescriptive specification unless otherwise stated. The compliance to the « GSS specification » requires the implementation of all the functionality specified in this document.
- Compliance with ENs and prENs issued by CEN TC278 concerning DSRC communication stack, Profile definition and EFC Application Interface, Ref [1], [2], [3], [4], [5].
- Self explanatory document for a reader having a deep knowledge of DSRC communication.
- Reduction of the complexity and increasing of performances completed by the selection of a subset of consistent functionality which covers the actual and future needs of DSRC communication systems.
- Explicit position concerning the DSRC preENs issued by CEN.
  - all the functions that have to be implemented are mentioned or described with details in this document.
  - when a standardised definition is considered as well known and precise to allow interoperable implementation, only a reference to the normative document is mentioned. In the other case a new formulation is given.
- Accent on the communication protocol and the format of the information transmitted by the DSRC link between sub-system instead of the communication services internal to each subsystem.
- Highlight on how to link functionality of different layers.
- In order to avoid ambiguity in the interpretation of this specification all conformance thereto is recommended to be validated by an independent certification body.

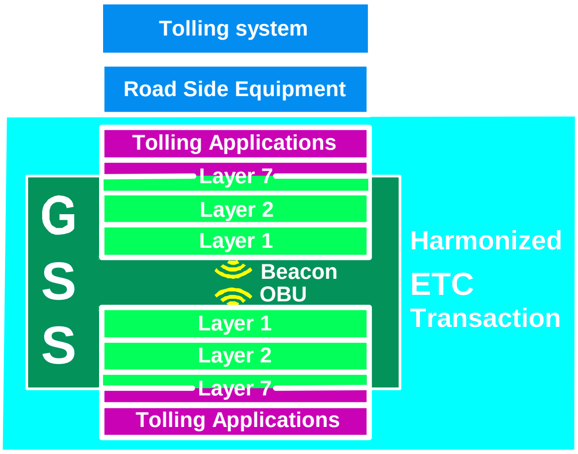

Figure 1.1: The scope of GSS

In accordance to the communication architecture chosen by CEN TC278 the GSS specification is based on a three Layers stack (1, 2, 7).

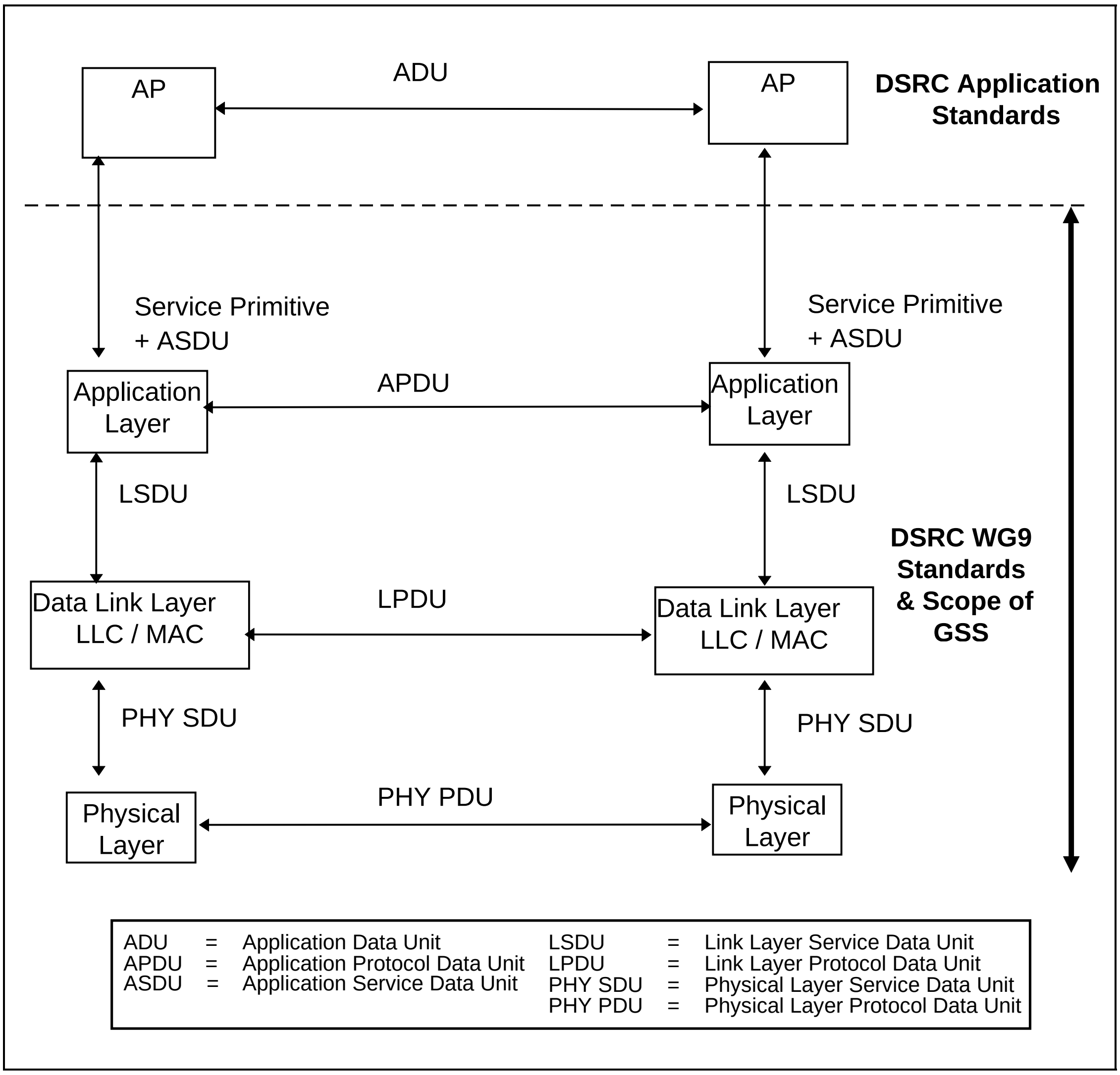
AP
ADU
AP
DSRC Application Standards
Service Primitive + ASDU
Service Primitive + ASDU
Application Layer
APDU
Application Layer
LSDU
LSDU
DSRC WG9 Standards & Scope of GSS
Data Link Layer LLC / MAC
LPDU
Data Link Layer LLC / MAC
PHY SDU
PHY SDU
Physical Layer
PHY PDU
Physical Layer
ADU = Application Data Unit
APDU = Application Protocol Data Unit
ASDU = Application Service Data Unit
LSDU = Link Layer Service Data Unit
LPDU = Link Layer Protocol Data Unit
PHY SDU = Physical Layer Service Data Unit
PHY PDU = Physical Layer Protocol Data Unit

Figure 1.2: Global data flow between communication Layers

On the air interface the Protocol Data Units (PDU) provided by each layer are encapsulated according to the Layer's order.

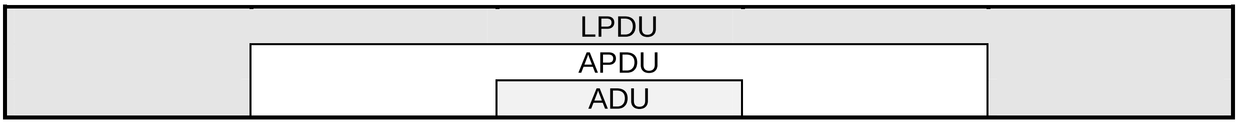

Figure 1.3: Encapsulation of Protocol Data Units

2 Physical Layer

2.1 Overview

The Physical Layer, at 5.8 GHz, communication requirements for the information from the RSE to the OBE are accounted for as downlink parameters, while the requirements associated with the information from the OBE to the RSE are accounted for as uplink parameters.

Only semi-passive transponders are supported by this specification. Due to the battery powered operation, power saving in a sleep mode is specified. The wake-up and sleep behaviour is not mandatory.

The parameter denominators are referred to [1].

2.2 Communication Zone Geometrical Requirements

To fulfil the purpose of this specification, i.e. to provide a foundation for interoperability of equipment from different manufacturers, it is crucial that not only the communication protocol is specified but also that the geometrical aspects of the communication are sufficiently well defined.

This section provides such requirements on the mounting position and orientation of the OBE that will enable the RSE to be designed and installed to achieve interoperability. **It is essential that all RSE have such design that they can handle all OBE including versions which meet the minimum requirements stated below.**

2.2.1 OBE Mounting Position

OBEs should be mounted in the lateral center of the vehicle as close as possible to a nominal height of 1,5 m.

In cars, this normally means a position behind the rear view mirror, in heavy commercial vehicles on the middle of the lower rim of the windscreen.

2.2.2 OBE Active Angle

The OBE downlink and uplink parameters stated below shall be met within a minimum solid angle, called the OBE Active Angle:

- horizontally: ±25º with respect to the vertical plane through the longitudinal axis of the vehicle.

- vertically: $35^{\circ} < \Theta < 80^{\circ}$, where Θ is the angle with respect to the horizontal plane as defined below.

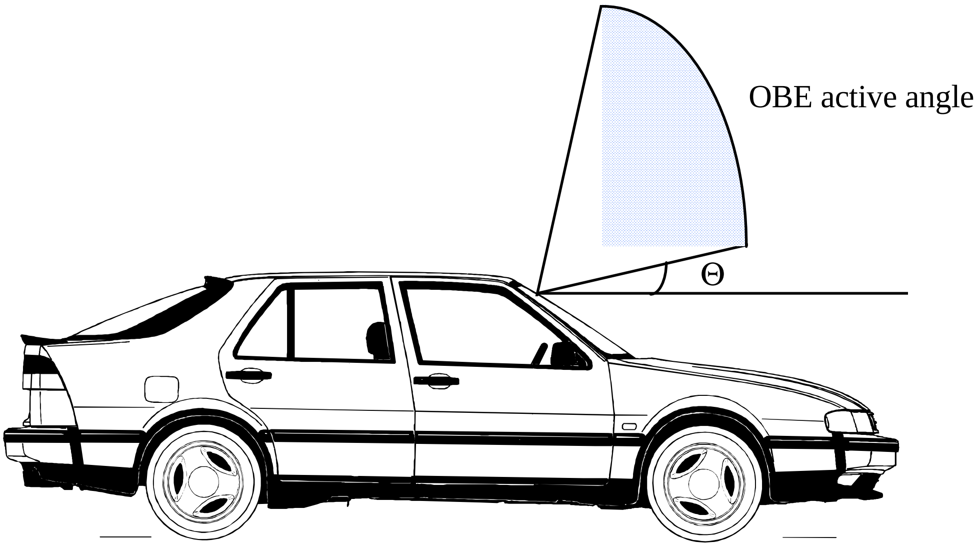

Figure 2.1: OBE Active Angle (vertically)

2.3 Downlink Parameters

2.3.1 Carrier Frequencies (D1, D3)

In Europe, there are four allowed carrier frequencies (D1) for the downlink channel:

- 5,7975 GHz
- 5,8025 GHz
- 5,8075 GHz
- 5,8125 GHz

The deviation of the carrier frequency from its nominal value shall be less than ±5 ppm for all operating conditions (D1a).

The OBE Minimum Receiver Bandwidth is 5.795 – 5.815 GHz (D3).

Other frequencies for applications outside Europe may be supported by the same equipment.

2.3.2 RSE Transmitter Spectrum Mask (D2)

As an overall requirement, the emitted EIRP shall be less than -30 dBm outside the used frequency band 5,795 - 5,815 GHz.

For in-band emissions the RSE transmitter spurious emissions shall comply with at least one of the classes below.

Co-channel uplinks are the uplink frequencies at f_c± 1,5 MHz and f_c± 2,0 MHz, where f_c denotes the utilised carrier frequency.

Adjacent channel uplinks are the uplink frequencies at f_a± 1,5 MHz and f_a± 2,0 MHz, where f_a denotes the remaining carrier frequencies of Chapter 2.3.1.

The requirements are defined for 500 kHz bandwidth.

	All classes, unmodulated carrier	Class B	Class C
Co-channel uplink at f_c ± 1,5 MHz	< - 27 dBm	< -17 dBm	< - 27 dBm
Co-channel uplink at f_c ± 2,0 MHz	< -27 dBm	< -27 dBm	< - 27 dBm
Adjacent channel uplinks	< -47 dBm	< - 37 dBm	< - 47 dBm

Table 2.1: Equipment Classes

2.3.3 Maximum EIRP (D4)

The maximum EIRP of the RSE transmitter shall be less than + 33 dBm (D4).

The RSE transmitter shall be oriented in such a way that the EIRP is kept below +18 dBm for angles larger than 70 degrees with respect to the vertical axis (D4a). (The main part of the emitted power should be directed towards the ground.)

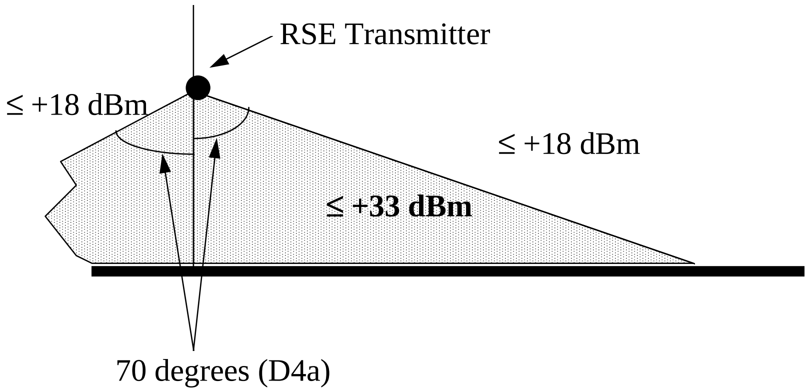

Figure 2.2: RSE antenna geometry

2.3.4 Polarisation (D5)

The polarisation shall be left hand circular (D5), i.e. when looking in the direction of wave propagation, the tip of the electrical vector shall rotate anti-clockwise.

The requirements below are applicable to the equipment itself not including other external effects.

The Cross Polarisation Discrimination (D5a), i.e. the residual right hand circular polarised wave shall be lower than the expected polarisation by a factor not less than:

• for RSE transmission in the boresight direction:	> 15 dB
• for OBE reception in the boresight direction:	> 10 dB
• for RSE transmission within the OBE Active Angle:	> 10 dB
• for OBE reception within the OBE Active Angle:	> 6 dB

2.3.5 Modulation and Eye Pattern (D6)

A two-level amplitude modulation shall be used, (D6) ensuring compliance with the RSE Transmitter Spectrum Mask requirements of Chapter 2.3.2.

The modulation index (D6a) is defined as $(V_{max} - V_{min})/ (V_{max} + V_{min})$,

It shall be within the range 0.5 - 0.9.

V_{max} = maximum amplitude during modulation
V_{min} = minimum amplitude during modulation

The eye pattern (D6b) defines the free decision distance of a digital signal pulse with respect to pulse width and amplitude. An ideal digital signal has a decision height of 100 % which is equal to the difference of high level and low level. Considering e.g. bi-phase coding, the ideal (=100 %) distance in width is equal to half the bit duration. With reference to figure 2.3 below the eye pattern requirements are:

Pulse width :	2B' / (A' + B')	> 0.90
Pulse amplitude :	2B / (A + B)	> 0.85

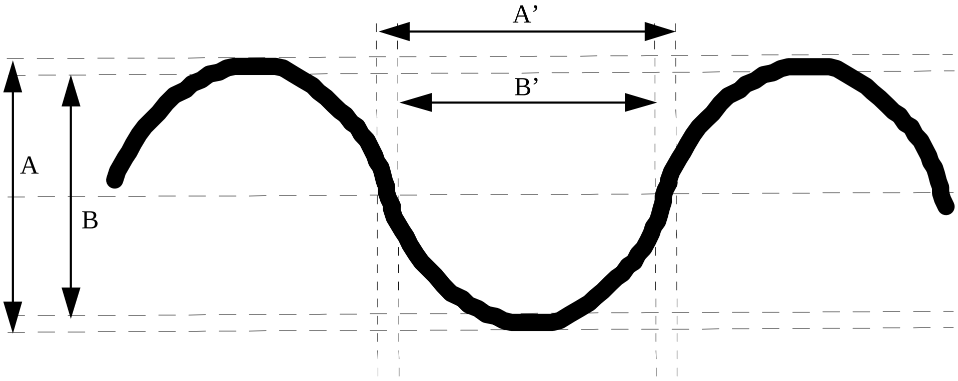

Figure 2.3: Definition of Eye Pattern

2.3.6 Data Coding (D7)

The bits are coded in FM0, i.e.

- there is always a transition at the beginning and end of the bit interval.
- a "0" bit has an additional transition in the middle of the bit interval.
- a "1" has no such additional transition in the middle of the bit interval.

2.3.7 Bit Rate (D8)

The downlink bit rate shall be 500 kBit/s ±100 ppm.

2.3.8 Communication Zone, Bit Error Rate for Communication (D11, D9a)

The communication zone is defined as the volume in space where the reference bit error rate (BER) for communication (D9a) equal to or lower than 10^{-6} can be achieved.

2.3.9 Power limit for communication zone (D11a, D11b)

The upper level of incident power referred to a lossless isotropic antenna (0 dB) in front of the OBE is -24 dBm (D11a-0). Below this level, but subject to D11b, communication is guaranteed with a specified BER (Communication may take place above this limit, but is not guaranteed.)

The lower level of incident power referred to a lossless isotropic antenna (0 dB) in front of the OBE is -43 dBm (D11b-0). Above this level, but subject to D11a, communication is guaranteed with a specified BER (Communication may take place below this limit, but is not guaranteed.)

D11a and D11b also specify the minimum dynamic range of the OBE receiver. Power values are measured without any additional losses due to rain or misalignment.

The requirement shall be met within the geometrical constraints of Chapter 2.2.2

2.3.10 Wake-up process (D10)

The OBE shall be capable of waking up on ordinary data frames, i.e. no special wake-up pattern shall be used.

The frame length required for wake-up shall not exceed 11 bytes, including flags (D10).

The start-up time from a sleep mode to full DSRC operation shall be less than 5 ms (D10a)

2.3.11 Preamble (D13a, D13b)

All data frames shall be preceded by a preamble of length 16±1 bits (32 µs ± 2 µs).

The preamble consists of a sequence of alternating high and low levels with pulse duration of 2 µs, which is equivalent to a sequence of FM0 coded ”1” bits.

2.3.12 Trailing Bits (D13c)

The RSE is permitted to transmit a maximum of 8 bits after the end flag.

An OBE is not required to take these additional bits into account in any way.

2.3.13 Cut-off power level of OBE (D12)

An incident power which is lower than –60 dBm shall not result in communication.

2.4 Uplink Parameters

2.4.1 Carrier Frequencies and Subcarrier Frequencies (U1)

The uplink carrier frequencies and frequency accuracy requirements are identical to those of Chapter 2.3.1.

The uplink data shall be modulated on to either one of two possible subcarrier frequencies:

- 1,5 MHz (U1-0) ± 0,1 % , or
- 2,0 MHz (U1-1) ± 0,1 %.

The OBE shall be able to respond by using the subcarrier frequency commanded from the RSE, in accordance with the chosen communication profile.

2.4.2 Uplink Data Coding (U7)

The uplink data shall be NRZI coded, i.e.:

- there is always a transition at the beginning of a ”0” bit.
- there is no transition at the beginning of a ”1” bit.
- the level is kept constant for the whole bit duration.

2.4.3 Uplink Data Rate (U8)

The uplink bit rate shall be 250 kbit/s ± 0,1 % .

2.4.4 Modulation of Subcarrier (U6a)

The modulation of the subcarrier shall be BPSK, i.e. bi-phase (0º , 180º) shift keying.

A phase shift of the subcarrier shall be generated at each transition of the uplink encoded data.

2.4.5 Modulation of Carrier (U6c, U1b)

The modulation of the carrier is a multiplication of the modulated sub-carrier with the carrier (U6c). The data shall be the same in both side bands (U1b).

2.4.6 OBE Transmitter Spectrum Mask (U2, U4)

The maximum EIRP at the used subcarrier shall be less than -14 dBm (U4a-0) in boresight and less than –17 dBm (U4b) at 35° away from boresight.

As an overall requirement, the emitted EIRP shall be less than -30 dBm (U2) in 1 MHz outside the used frequency band.

The in-band spurious emission at the co-channel uplink frequencies and all adjacent channel up-link frequencies shall be less than -35 dBm (U2) in 500 KHz.

2.4.7 OBE Conversion Gain (U12)

Difference between OBE EIRP within one side band and the carrier power incident on OBE. Conversion gain is measured at the minimum incident power within the communication zone given by U11. It shall be between 1 dB (U12a) and 10 dB (U12b)

Remark: Conversion Gain is equal to twice the OBE antenna gain minus the OBE losses.

2.4.8 Polarisation (U5)

The polarisation shall be left hand circular, i.e. when looking in the direction of wave propagation, the tip of the electrical vector shall rotate anti-clockwise (U5)

The requirements below are applicable to the equipment itself not including other external effects.

The Cross Polarisation Discrimination (U5a), i.e. the residual right hand circular polarised wave shall lower than the expected polarisation by a factor not less than:

• for RSE reception in the boresight direction:	> 15 dB
• for OBE transmission in the boresight direction:	> 10 dB
• for RSE reception within the OBE Active Angle:	> 10 dB
• for OBE transmission within the OBE Active Angle:	> 6 dB

2.4.9 Eye Pattern/ Duty Cycle (U6b)

For the uplink data and subcarrier either one of the following requirements shall be fulfilled:

Eye Pattern: > 90 % / > 90 %

Duty Cycle: 50 % ± 5 %.

2.4.10 Bit Error Rate (U9)

The bit error rate is the averaged number of erroneous bits relative to all transmitted bits.
The reference value for Layer 1 when measured with the incident power at the OBE at any level between the specified minimum and maximum values shall be equal or lower than 10^{-6}.
The effective BER within the communication zone may be different from the reference value because of time-variant and stochastic influences.

2.4.11 Preamble (U13,U13a)

All uplink data frames shall be preceded by a preamble which consists of two parts:

- first, 32 µs - 36 µs of unmodulated subcarrier,
- then, 8 bits of NRZI coded “0” bits.

2.4.12 Trailing Bits (U13b)

The OBE is permitted to transmit a maximum of 8 bits after the end flag.
A RSE is not required to take these additional bits into account in any way.

2.4.13 Communication Zone (U11)

The spatial region within which the OBE is situated such that its transmissions are received by the RSE with a BER of less than a specified value.

3 Frame Overview

3.1 Frame Format

The frame is a Data Link Layer entity which is the result of stepwise encapsulation of the application commands and data in line with the DSRC communication architecture. One or more Application Protocol Data Units, APDUs, can be encapsulated in one Link Protocol Data Unit, LPDU, encapsulated in the frame.

The relationships between frame, LPDU and APDU are depicted in the figure below.

The frame should be read from left to right, i.e. the leftmost field is transmitted first.

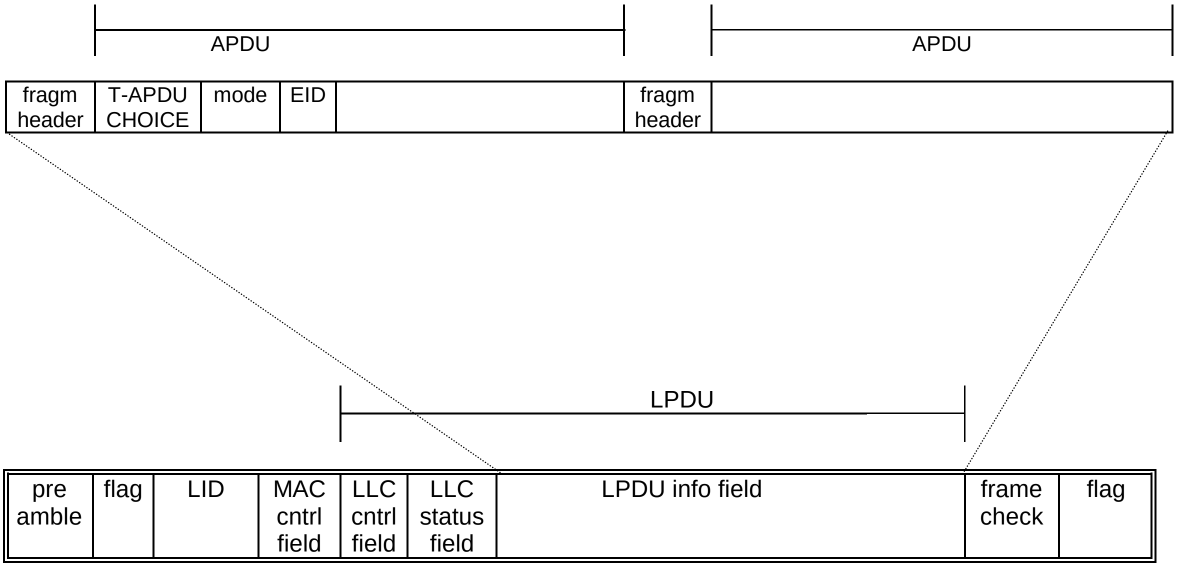

Figure 3.1: Frame format

The frame format is similar for the downlink and for the uplink. The downlink carries only Data Link Layer command frames, while the uplink can carry Data Link Layer command and response frames.

Each frame does not necessarily contain all of the fields depicted above. The most notable exceptions are

- Only response frames contain the LLC status field
- Response frames do not always contain an LPDU info field
- There are frames without LPDU (used for medium request / allocation)

3.2 Frame Size

The size of the frame, including the flags and anything between the flags but excluding the preamble is subject to the following restrictions:

in a downlink window	max 128 octets	(N2)
in a private uplink window	max 128 octets	(N3)
in a public uplink window	max 9 octets	(N4)

The number of bits transmitted is, however, dependant on bit stuffing.

3.3 Preambles

Information on the preambles is included in chapter 2, since they form a part of the definition of the physical layer. However, since they also form a part of the frame they are summarised here from a data link layer point of view.

	Length	**Content**
Downlink		
preamble	16 bit ± 1 bit	‘1’ bits coded according to FM0
Uplink		
preamble	8 - 9 bits + 8 bits	eight - nine '1' bits + eight '0' bits coded according to NRZI

Table 3.1: Preambles

3.4 Flags and bit stuffing

The flag is an eight bit sequence, namely 0111 1110, delimiting the content of the frame. Any sequence of 0111 1110 should be interpreted as a valid flag sequence.

All transmitters shall send only complete eight bit flag sequences. Two consecutive flags should thus be sixteen bits.

The occurrence of the flag sequence within a frame between the start and end flags shall be prevented via a bit stuffing procedure. The transmitter shall insert a 0 bit following five contiguous 1 bits anywhere between the start flag and the end flag of the frame.

The insertion of the 0 bit thus applies to the contents of the LID, the MAC control field, the LLC control field, the LLC status field, the LPDU info field and the FCS.

After receiving five contiguous 1 bits, the receiver shall inspect the following bit. If it is a 0, the five 1 bits are passed as data and the 0 is deleted. If the sixth bit is a 1, the receiver shall inspect the seventh bit. If this bit is a 0, the receiver shall interpret the bit sequence as a flag. If it is a 1, the frame is invalid.

In private uplink windows, the OBE may transmit several consecutive start flags.

3.5 Bit Order

3.5.1 Overview

Data transmission according to this specification occurs only in octets. Each frame is an ordered sequence of octets where octet#1 is transmitted first. In each octet, the least significant bit is transmitted first.

When data are depicted octetwise left to right the first octet is the leftmost octet. When they are depicted octetwise top to bottom the first octet is the top octet. When the octet is depicted bit by bit, the least significant bit is the rightmost bit.

3.5.2 One octet fields

The following fields are one octet fields: the Flag, the MAC control field, the LLC control field, the LLC status field, the LID field when the content is the broadcast LID and the fragmentation header field.

All those fields are transmitted least significant bit first.

3.5.3 Two octet fields

The FCS field is the only two octet field. The coefficient of the highest term is transmitted first.

3.5.4 Four octet fields

The LID field when the content is a private LID is the only four octet field.It is transmitted octet #1 first, and, for each octet, least significant bit first.

3.5.5 The APDU

Formally, the APDU is a bit sequence generated by the ASN.1 PER encoder of the transmitter. However, as a result of the agreed use of ASN.1, the PER encoder will always output a multiple of eight bits, i.e. octets. The octets are transmitted octet #1 first, and, for each octet, least significant bit first.

For further details see under Encoding in 5.1.2

3.6 Frame Check Sequence

All frames shall include a 16-bit Frame Check Sequence (FCS) just prior to the end flag.

The contents of the LID field, MAC control field, LLC control field, LLC status field and LPDU info fields shall be included in the calculation of the FCS.

The FCS shall be compliant with a 16-bit frame checking sequence as defined in ISO 3309 (clause 3.6.2). The generator polynomial shall be $X^{16} + X^{12} + X^{5} + 1$, and the initial value used shall be $FFFF_{16}$. The ones complement of the resulting remainder shall be transmitted as the 16-bit FCS.

The coefficient of the highest term shall be transmitted first.

4 Data Link Layer

4.1 Link ID

4.1.1 Overview

There are two types of Link IDs: private Link ID and broadcast Link ID. In [2], the Link ID is also called *link address*. Throughout the document we will use the abbreviation LID for Link ID.

Broadcast LID:
The broadcast LID is used to address all OBEs simultaneously. It can only be used in unacknowledged services in downlink frames.

Private LID:
The private LID is used to identify and address one distinct OBE. It is used in downlink frames as well as in uplink frames. When used in uplink frames, the private LID serves to identify the originator.

The private LID is temporary and random. The probability for two OBEs having received the same BST creating the same private LID shall be less than 2^{-25}. Every OBE shall create private LIDs having approximately a uniform distribution over the range of possible values.

All DSRC layers and the application inside RSE make use of this private LID. The applications inside RSE use the private LID to access data inside a specific OBE.

4.1.2 Format

Broadcast LID: one octet, all bits '1':

1	1	1	1	1	1	1	1
MSB							LSB

Figure 4.1: Broadcast LID format

Private LID: four octets, all bits but the LSB's are randomly chosen by the OBE:

x	x	x	x	x	x	x	0	octet #1
x	x	x	x	x	x	x	0	octet #2
x	x	x	x	x	x	x	0	octet #3
x	x	x	x	x	x	x	1	octet #4
MSB							LSB	

Figure 4.2: Private LID format

The bits marked with 'x' shall be selected by means of a random choice.

4.1.3 Private LID Establishment

A new private LID may be created by the OBE after having received a BST = Beacon Service Table which meets all necessary requirements. For details see Chapter 5.2.4.

4.1.4 Lifetime of private LID

As long as the private LID is valid, the OBE can be addressed by the RSE using this private LID. There are two events causing the private LID to become invalid:

- OBE creates a new private LID
- OBE receives a RELEASE command from the application layer of the RSE.

For details see Chapters 5.2.1 and 5.2.4.

4.2 Medium Access Control

4.2.1 Overview

The medium access control is characterised by half duplex mode of operation and asynchronous time division multiple access. It is unbalanced in that the RSE is always ultimately in control of the physical medium.

Downlink windows are used by the RSE when transmitting. Uplink windows are allocated by the RSE as indicated by the MAC control field of the downlink frame and follow immediately after the downlink window containing the frame. Uplink windows can be private or public.

The OBE can request access to the medium by a private window request transmitted in a public uplink window.

4.2.2 MAC Control field

The MAC control field is used to:

- indicate whether an LLC unit is available (to allow also for empty LPDU frames)
- indicate the transmission direction: downlink or uplink
- allocate public and private uplink windows
- request private uplink window
- specify type of LLC unit

The MAC control field is one octet. It contains the MAC control information and the command/response identifier of the LPDU. A distinction is made between downlink and uplink MAC control fields.

Bit	Name	Value	Remark
L	LPDU Existence	0 = no LPDU 1 = LPDU	Indicates if frame contains LPDU
D	Direction Identifier	0 = downlink 1 = uplink	
A	Medium Allocation	0 = no window allocated 1 = window allocated	Downlink only
R	Medium Request	0 = no window is requested 1 = window is requested	Uplink only
C/R	Command/ Response	0 = command LPDU 1 = response LPDU	(LLC functionality see 4.3)
S	MAC Sequence		Downlink only see 4.2.6

Table 4.1: MAC Control field bits

Downlink

L	D	A	C/R	S	-	-	-

MSB LSB

Figure 4.3 MAC control field on downlink

In the MAC Control field for the downlink the following bits have fixed values:

D	Direction Identifier	0 = downlink.
C/R	Command/Response	0 = command LPDU
-	unused bits	0

For the remaining bits the following combinations are valid

L	D	A	C/R	S	-	-	-	hex	Used for
0	0	1	0	S	0	0	0	20 / 28	Private Window Allocation
1	0	1	0	0	0	0	0	A0	UI command with allocation
1	0	0	0	0	0	0	0	80	UI command no allocation
1	0	1	0	S	0	0	0	A0 / A8	ACn command

MSB LSB

Table 4.2 MAC control field values on downlink

S denotes the MAC Sequence bit whose behaviour is specified in chapter 4.2.6

Uplink

L	D	R	C/R	-	-	-	-

MSB LSB

Figure 4.4 MAC control field on uplink

In the MAC Control field for the uplink the following bits have fixed values:

D	Direction Identifier	1 = uplink.
-	unused bits	0

For the remaining bits the following combinations are valid:

L	D	R	C/R	-	-	-	-	hex	Used for
0	1	1	0	0	0	0	0	60	Private Window Request
1	1	0	0	0	0	0	0	C0	Private UI command
1	1	0	1	0	0	0	0	D0	ACn response
MSB							LSB		

Table 4.3 MAC control field values on uplink

4.2.3 Window Definitions

Downlink window

A downlink window is when the RSE is transmitting a frame.

Private Uplink Window

A private uplink window may only be used by the OBE having a private LID equal to the LID of the frame allocating the window. For private uplink window timing see 4.2.5.

Public Uplink Windows

A public uplink window may be used by any OBE according to certain rules see 4.2.4.
For public uplink window timing see 4.2.5.

4.2.4 Window Management

Window allocation (downlink)

The RSE indicates that it allocates an uplink window by setting the A bit of the MAC control field of a frame to 1.

The distinction between allocation of public and private uplink windows is made by means of the LID of the frame allocating the window.

A private uplink window is allocated if the LID of the allocating frame is a private LID while public uplink windows are allocated if the LID of the allocating frame is broadcast.

When a private uplink window is allocated, only one window is allocated. The start of this window occurs a certain time after the end of the downlink window containing the frame allocating the window, see 4.2.5.

When public uplink windows are allocated, three consecutive windows are simultaneously allocated by one downlink frame. For the timing of these windows see 4.2.5.

This means that three possibilities are given: no allocation of uplink window, allocation of one private uplink window or allocation of three consecutive public uplink windows.

See figure 4.5 below.

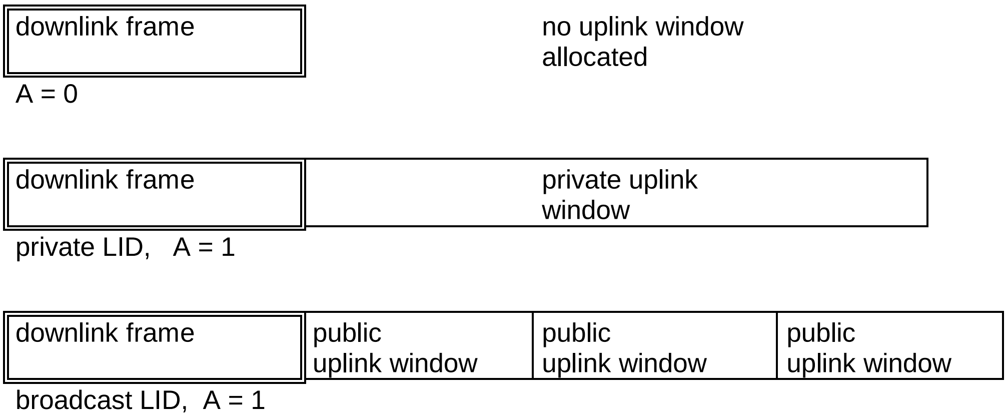

Figure 4.5: Window Allocation alternatives

Private Window Request (uplink)

The OBE indicates that it requests a private uplink window by setting the R bit of the MAC control field of a frame, containing no LPDU and transmitted in a public uplink window, to 1.

This request shall be granted by the RSE by allocating private uplink window, in a frame containing no LPDU, to that OBE before the next public uplink window allocation.

If the request is not granted by the RSE by transmitting a private window allocation before the next public uplink window allocation, the OBE shall retransmit its request.

In addition, if the request was granted by the RSE by transmitting a private window allocation and there is a subsequent public uplink window allocation before sending a frame with private LID to that OBE (other than private window allocation), the request shall be retransmitted either. This mechanism is an ‘implicit application layer acknowledge for uplink messages’, since the OBE will try to re-transmitt the message until it receives a new request from the RSE’s application layer.

The frame format of the Private Window Request is shown in figure 4.6.

The value of the MAC Control field is 60_{16} .

pre amble	flag	private LID	MAC cntrl field	frame check	flag

Figure 4.6: Private Window Request

Private Window Allocation (downlink)

The RSE can allocate a private uplink window without the transmission of an LPDU by setting the A bit of the MAC control field of a frame, containing no LPDU, to 1.

The frame format of the Private Window Allocation is shown in figure 4.7.

The value of the MAC Control field is 20_{16} or 28_{16} depending on the value of the S bit.

pre amble	flag	private LID	MAC cntrl field	frame check	flag

Figure 4.7: Private Window Allocation

4.2.5 Window Timing

Downlink windows

The start of a downlink window occurs at the start of the first bit of the preamble.

The end of a downlink window occurs at the end of the last bit of the end flag. The end of the downlink window is used as the reference point from which the timing of uplink windows is deduced.

The minimum time after an uplink window that a downlink window can start is 32 µs (T1).

After a downlink window that minimum time is 0µs (T2).

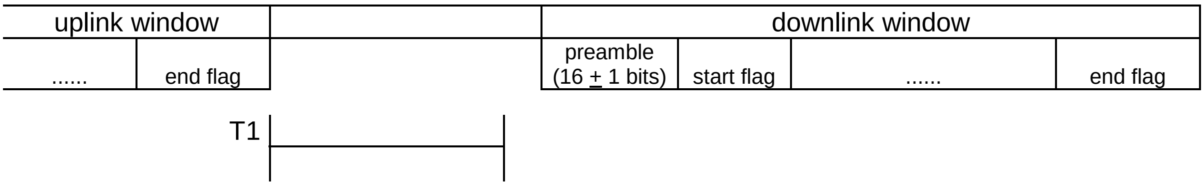

Figure 4.8: Timing of downlink window after uplink window.

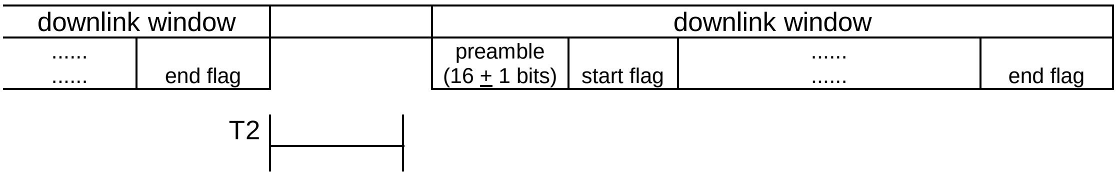

Figure 4.9: Timing of downlink window after downlink window

The capability to receive several consecutive downlink frames by one OBE is subject to implementation.

Private Uplink Windows

The start of a private uplink window occurs 160 µs (T3) after the end of the downlink window containing the frame allocating the private uplink window (see also 4.2.3)

The end of a private uplink window occurs 320µs (T4a) after the start of the window, if no OBE has started transmitting the first bit of its preamble before that time or otherwise at the end of the last bit of the end flag of the uplink frame transmitted.

The maximum time of a private uplink window is 5,5 ms.

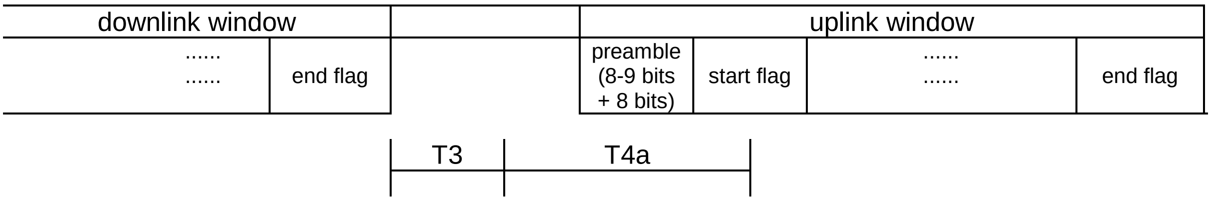

Figure 4.10: Private uplink window timing

Public Uplink Windows

Three consecutive windows are simultaneously allocated.

The start of the first public uplink window occurs 160 µs (T3) after the end of the downlink window containing the frame allocating the public uplink windows (see also 4.2.3).

The start of subsequent public uplink windows occurs immediately after the end of the previous window.

The end of a public window occurs 448 µs (T5) after the start of that window.

The transmission of the start of the first bit of the preamble in a public uplink window shall start before 32 µs (T4b) after the start of that window.

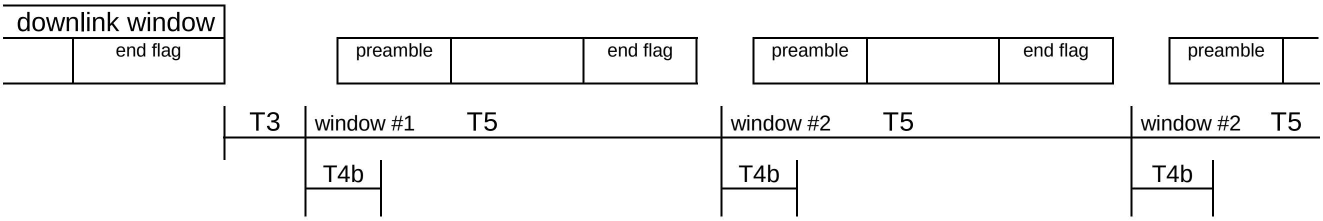

Figure 4.11: Public uplink window timing

4.2.6 The MAC Sequence bit, S

The MAC sequence bit is relevant in all downlink frames allocating private uplink window.

The first private window allocation for any new private LID uses the value S = 0. For any subsequent private uplink window allocation to that private LID the value shall toggle, except when the allocation is a reallocation.

The value of the MAC sequence bit for one private LID is independent of the value of the MAC sequence bit for another private LID.

Each time a private uplink window is allocated by the RSE, a transmission from the OBE, to which the window is allocated, is expected. If no valid frame was received, the RSE may reallocate the private uplink window, transmitting the Private Window Allocation (see figure 4.7) any time before the next public uplink window allocation. If a window is reallocated the MAC sequence bit shall have the same value as the first time the private uplink window was allocated.

4.2.7 Use of Public uplink windows

Public uplink windows are used by the OBE only to transmit a request for private window, i.e. in situations when a private uplink window is not automatically granted by the RSE.

When the OBE has decided to transmit in a public uplink window it shall randomly select one of the three windows allocated and transmit in that window.

4.2.8 Examples

Example #1

This example shows the OBE requesting private uplink window, having randomly choosen the first public window allocated. The private window allocation transmitted by the RSE is lost.

Since the OBE does not receive the private window allocation it does not transmit anything in the private uplink window allocated. After the timer has expired (T3 + T4a) the RSE reallocates the private uplink window, without toggling the S-bit.

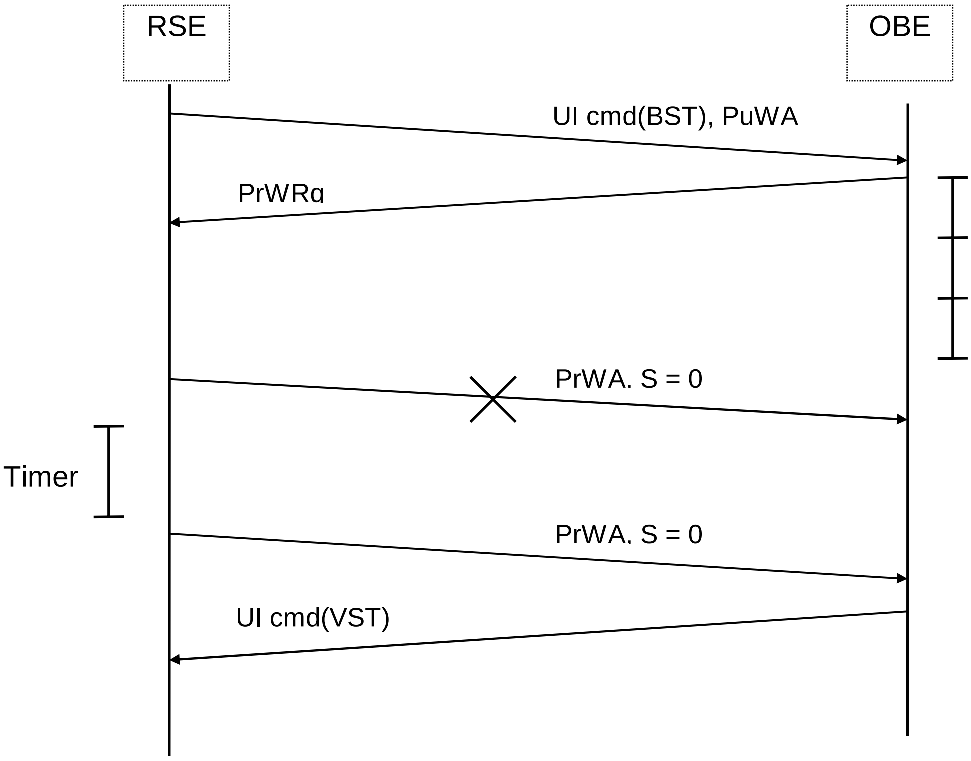

Figure 4.12: Private uplink window reallocation

Example #2

This example shows the OBE requesting private uplink window, having randomly choosen the first public window allocated. The resulting private window allocation is successful, but the transmitted VST is lost.

The RSE re-allocates that private window and the VST is transmitted successfully.

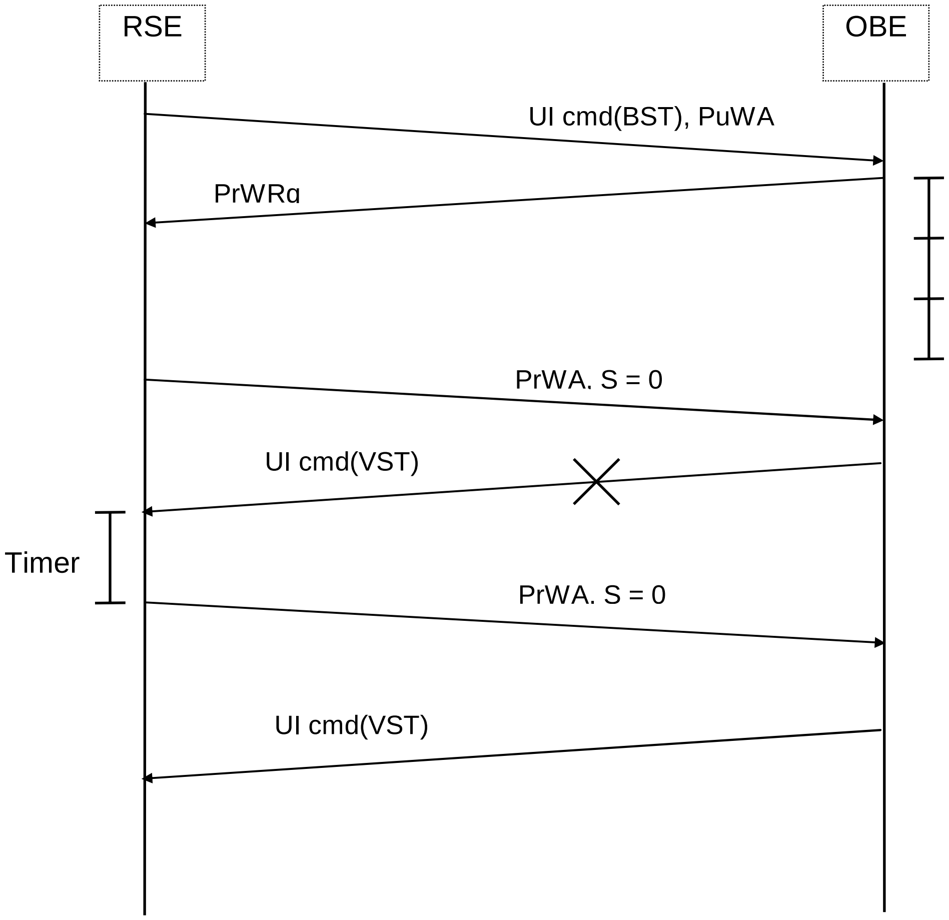

Figure 4.13: Re-allocation of private window

4.3 Logical Link Control

4.3.1 Overview

The LLC sub-layer provides for two types of services, unacknowledged services and acknowledged connection-less services. The functionality of the LLC sub-layer includes :

- organisation of data flow
- interpretation of received command PDUs
- generation of appropriate response PDUs
- Error control and recovery

Unacknowledged commands are used on both downlink and uplink and require no response by the data link layer. Acknowledged commands are used on downlink only and do always require an immediate response by the data link layer.

The LLC control field is one octet. It is used to indicate the type of command/response.

M	**M**	**M**	**P/F**	**M**	**M**	**-**	**-**

MSB LSB

Figure 4.14: General format of LLC control field

M denotes modifier bits, which have values specified for each command /response

P/F denotes the Poll bit (on the downlink) or the Final bit (on the uplink) respectively

- denotes bits not used that shall always be set to 1

4.3.2 Unacknowledged Commands

The unacknowledged services make use of the UI command. The command requires no response by the data link layer. UI commands are used on downlink and on uplink. On the downlink UI commands are used with broadcast LID or with private LID. On the uplink UI commands are used with private LID.

LLC control field content is UI.
The C/R bit of the MAC control field indicates Command.

Downlink or Uplink

M	M	M	P	M	M	-	-	hex	Command
0	0	0	0	0	0	1	1	03	UI command

MSB LSB

Table 4.4 LLC control field value for UI commands

4.3.3 Acknowledged Commands (downlink)

The acknowledged connection-less services make use of the AC command. The command always requires a response by the data link layer. AC commands are used on the downlink only, generating AC responses on the uplink.

LLC control field content for the command is ACn with p = 0 or ACn with p = 1.
The C/R bit of MAC control field indicates Command.

AC commands are always used with the private LID of the destination.

The P/F bit is used as the Poll bit, indicating whether an application layer response is requested to be returned on the uplink or not, see 4.3.6.

The n bit is the LLC sequence bit, indicating whether the command is the first transmission or a retransmission, see 4.3.7.

M	M	M	P	M	M	-	-	hex	Command
n	1	1	0	0	1	1	1	67 / E7	ACn with p = 0
n	1	1	1	0	1	1	1	77 / F7	ACn with p = 1

MSB LSB

Table 4.5 LLC control field for AC commands

4.3.4 Responses to Acknowledged Commands (uplink)

The acknowledged command always requires a response by the data link layer.
Each AC commands on the downlink generates an AC response on the uplink.

The response is transmitted in the private uplink window allocated by the command frame.

LLC control field content for the response is ACn with f = 0 or ACn with f = 1.
The C/R bit of MAC control field indicates Response.

AC responses are always used with the private LID of the originator.

The P/F bit is used as the Final bit, indicating whether an application layer response was requested or not, see 4.3.6.

The n bit is the LLC sequence bit, see 4.3.7.

M	M	M	F	M	M	-	-	hex	Response
n	1	1	0	0	1	1	1	67 / E7	ACn with f = 0
n	1	1	1	0	1	1	1	77 / F7	ACn with f = 1

MSB LSB

Table 4.6 LLC control field for AC responses

4.3.5 The LLC status field (uplink)

In every AC response the LLC status field shall be included.

The LLC status field is one octet.

It is used to indicate success or failure of the processing of the command.

The format of the LLC status field is found in table 4.7.

R	R	R	R	C	C	C	C	hex	Name	
0	1	0	0	0	0	0	0	40	NR_OK	Command accepted Response APDU not requested
0	0	1	1	0	0	0	0	30	NE_OK	Command accepted Response APDU not yet available
0	0	0	0	0	0	0	0	00	OK_OK	Command accepted Response APDU present

MSB LSB

Table 4.7: LLC status field for AC responses (uplink)

R denotes bits used to indicate the success in generating a response

C denotes bits used to indicate the success in interpretation of the command

4.3.6 The Poll / Final bit P/F

p = 0, f = 0
For an AC command ACn with p = 0 the corresponding response shall be ACn with f = 0, with LLC status field indicating NR_OK and with no LPDU info field included.

This command is used to pass application layer commands to the OBE.

p = 1, f = 1
For an AC command ACn with p = 1 the corresponding response shall be ACn with f = 1.

The response shall have an LLC status field indicating OK_OK and an LPDU info field included if the processing of the application command resulted in the availability of an application response.

The response shall have an LLC status field indicating NE_OK and no LPDU info field included if the processing of the application command did not result in the availability of an application response.

This command is used to pass application layer commands to the OBE and the associated responses to the RSE.

4.3.7 The LLC Sequence bit n

The first AC command for any new private LID uses the value n = 0. For any subsequent new AC command to that private LID the value shall toggle.

The value of the LLC sequence bit for one private LID is independent of the value of the LLC sequence bit for another private LID.

For each AC command transmitted by the RSE, an AC response is expected.

If no valid AC response was received, the RSE may retransmit the AC command.

The LLC sequence bit of the command shall then have the same value as the first time the AC command was transmitted.

The LLC sequence bit of an AC response shall have the complementary value to the LLC sequence bit of the AC command to which it is a response.

4.3.8 Example

This example shows the RSE transmitting an ACn command with p = 0 to the OBE, expecting an immediate response ACn with f = 0.

The OBE responds with an ACn response with f = 0 and the n-bit of the ACn command complemented. The response ACn is lost.

This causes the RSE to retransmit the ACn command without toggling the n-bit.

The OBE re-transmits the ACn response with f = 0.

Thereafter the RSE transmits another ACn command with p = 0 to the OBE.

The OBE responds with f = 0 and the n-bit of the ACn command complemented.

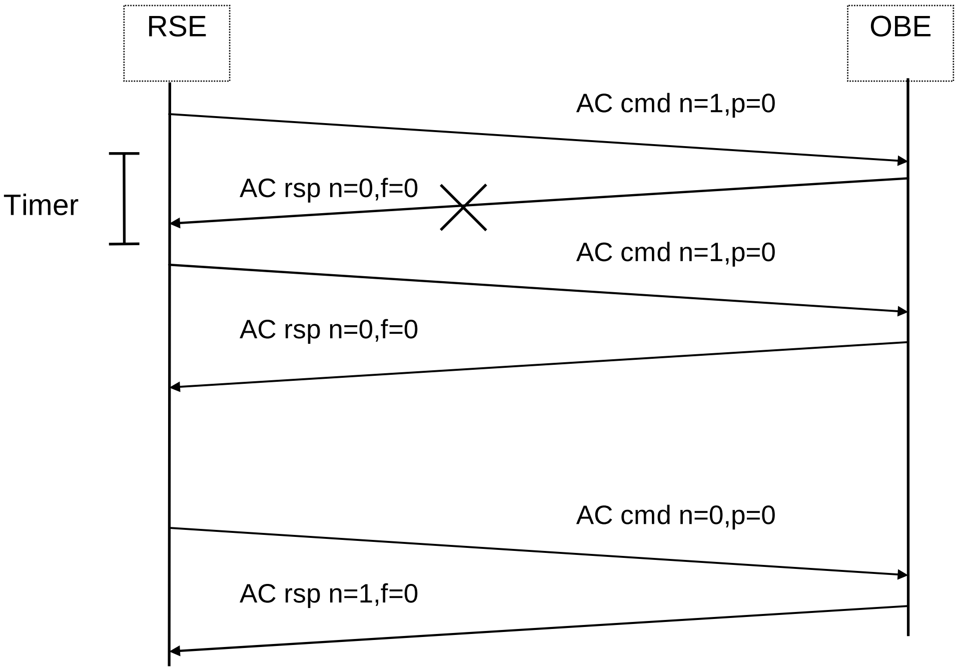

Figure 4.14 ACn command transmission and retransmission

4.4 DSRC Profiles 0 and 1

Each of the standards for the layers of DSRC contains some degree of variability, described by parameters defined in each standard.

Each set of parameter values is called a Profile and is assigned a unique identifier number.

This specification supports profiles 0 and 1 as defined in [4]. The choice of profile is primarily the responsibility of the RSE. All OBE shall support both profiles.

Below is a summary of important parameter values of the profiles 0 and 1.

Profile number		0	1	
D8	downlink bit rate	500	500	kBit/s
U 1	sub-carrier frequency	1.5	2.0	MHz
U8	uplink symbol (bit) rate	250	250	kBit/s
N2	max no of octets in frame in downlink window	128	128	
N3	max no of octets in frame in private uplink window	128	128	
N4	max no of octets in frame in public uplink window	9	9	
N5	no of simultaneously allocated public uplink windows	3	3	
N12	maximum private medium response time	1	1	
T1	minimum uplink to downlink turn around time	32	32	µs
T2	minimum downlink to downlink window time	0	0	µs
T3	downlink to uplink turn around time	160	160	µs
T4a	max time to start of transm. in private uplink window	320	320	µs
T4b	max time to start of transm. in public uplink window	32	32	µs
T5	time duration of public uplink window	448	448	µs
N13FE	acknowledgement time of fixed equipment	1	1	
N13ME	acknowledgement time of mobile equipment	1	1	

5 Application Layer

The application layer provides services for application data transfer and for Initialisation.

5.1 Application Data Transfer

5.1.1 Overview

The Application Layer provides the following services that enable application data transfer and remote application related operations:

- addressing of application data
- selection of services to access application data
- services: GET, SET, ACTION, EVENT-REPORT, INITIALISATION
- confirmed or unconfirmed mode of services
- encoding of application data and services
- identification of fragments
- chaining of fragments into chains
- concatenation of chains/fragments

5.1.2 Addressing of Application Data

Overview

Application data inside the OBE are organised by means of the concept of *elements.* Each element belongs to an application (e.g., EFC) and an application can have several elements. The application data itself are called *attributes*. The following figure illustrates the organisation of application data inside the OBE

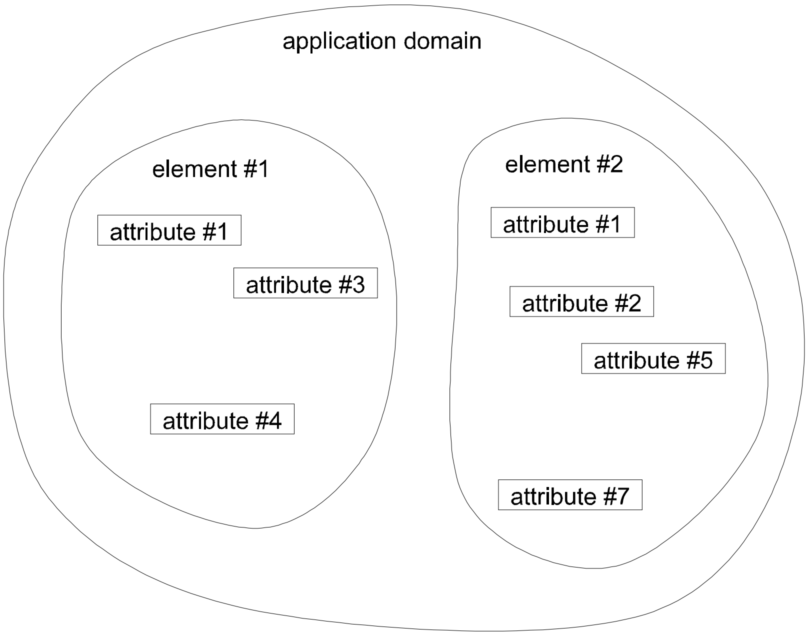

Figure 5.1: Organisation of application data inside OBE

Element

An element is identified by means of the Element ID (EID). Each element inside the OBE shall have an EID, unique within the context of the OBE. During Initialisation, the OBE sends the VST to the RSE (compare Chapter 5.2). The Application List contained in the VST provides the information, which applications are present in the OBE, and, which EIDs belong to which application.

According to [3] *Dsrc-EID*::=INTEGER(0..127,...). The extendibility is not used. Using PER, 8 bits are needed to encode the *eid.* The most significant bit is set to 0.

The value EID = 0 shall be used to address application-independent functions and components inside the OBE.

Attributes

Attributes contain the application-specific information. They are composed of one or more data elements of specified ASN.1 types. Each attribute is identified by means of an identifier (attributeID) which shall be unique within the context of an element. Hence, an attribute can be addressed uniquely using the EID + attributeID.

According to [3], attributeId::=INTEGER(0..127,...). The extendibility is not used. Using PER, 8 bits are needed to encode the attributeID*.* The most significant bit is set to 0.

Attributes are defined by the standardisation bodies responsible for RTTT applications, or, can be defined as 'private' attributes by the operators and manufacturers.

Invoker ID

In [3], the application layer services contain the optional element Invoker ID (IID). Since there is no specific use for the IID, it shall not be used in requests and responses (compare also remark on application multiplexing, 5.1.8).

5.1.3 Application Layer Services

In [3] there is defined a set of application layer services by means of a CHOICE type. The following table shows octet following the fragmentation header encoding the service primitives, see also example in Table 5.3.

MSB LSB	Service primitive
0000 xxxx	Action-Request
0001 xxxx	Action-Response
0010 xxxx	Event-Report-Request
0011 xxxx	Event-Report-Response
0100 xxxx	Set-Request
0101 xxxx	Set-Response
0110 xxxx	Get-Request
0111 xxxx	Get-Response
1000 xxxx	Initialisation-Request
1001 xxxx	Initialisation-Response

Table 5.1: Encoding of first octet of service primitives

In the following descriptions of the service primitives using ASN.1 notation, the word OPTIONAL indicates that the corresponding field is not always present. The presence is indicated by a bit in the first octet encoding the service primitive (one of the ‘x’ in above table).

GET

GET is used to retrieve (i.e. read) value(s) of the addressed attribute(s), a reply is always expected.

GET-Request

```
Get-Request ::= SEQUENCE {
 fill                 BIT STRING(SIZE(1)),
 eid                  Dsrc-EID,
 accessCredentials    OCTET STRING   OPTIONAL,
 iid                  Dsrc-EID,      OPTIONAL & not used,
 attrIdList           AttributeIdList OPTIONAL & always present
 }
```

GET-Request shall request the retrieval of the value(s) of the addressed attribute(s) in the Attribute Id List, a response shall always be expected.

`eid` identifies the element in the OBE containing the attributes identified with the `attrIdList`. `accessCredentials` are for optional use and may carry information necessary to fulfil the access conditions for the attributes. As mentioned above, `iid` shall not be used.

GET-Response

```
Get-Response ::= SEQUENCE {
  fill                BIT STRING(SIZE(1)),
  eid                 Dsrc-EID,
  iid                 Dsrc-EID,      OPTIONAL & not used,
  attributelist       AttributeList  OPTIONAL,
  ret                 ReturnStatus   OPTIONAL,
  }
```

GET-Response shall carry the retrieved value(s) of the addressed attribute(s) or/and the result of the corresponding GET-Request command.

`eid` identifies the element the attributes are retrieved from. The `attributelist`, if present, shall contain the attributes retrieved. If not all the addressed attributes were retrieved, a failure shall be indicated in the `ret` Code (see below).

SET

SET is used to set (i.e. write) value(s) of the addressed attribute(s).

SET-Request

```
Set-Request ::= SEQUENCE {
 fill                 BIT STRING(SIZE(1)),
 mode                 BOOLEAN,
 eid                  Dsrc-EID,
 accessCredentials    OCTET STRING   OPTIONAL,
 attrList             AttributeList,
 iid                  Dsrc-EID,      OPTIONAL & not used
 }
```

SET-Request can be used in confirmed (parameter `mode` = TRUE) or non-confirmed mode (parameter `mode` = FALSE). A response shall always be expected in the former case.

`eid` identifies the element in the OBE containing the attributes which are accessed. `attrList` contains the values for the attributes which shall be written. `accessCredentials` are for optional use and may carry information necessary to fulfil the access conditions for the attributes. As mentioned above, `iid` shall not be used.

SET-Response

```
    Set-Response ::= SEQUENCE {
      fill                  BIT STRING(SIZE(2)),
      eid                   Dsrc-EID,
      iid                   Dsrc-EID,      OPTIONAL & not used,
      ret                   ReturnStatus   OPTIONAL
      }
```

SET-Response shall explicitly convey the result of the corresponding SET-Request command. If not all the addressed attributes were set then a failure shall be indicated in the `ret` Code (see below). `eid` shall be the same as in the corresponding request.

ACTION

ACTION is used to invoke remote operations on the addressed resp. supplied attribute(s).

ACTION-Request

```
     ACTION-Request ::= SEQUENCE {
      mode                  BOOLEAN,
      eid                   Dsrc-EID,
      actionType            INTEGER (0..127,...),
      accessCredentials     OCTET STRING   OPTIONAL,
      actionParameter       Container      OPTIONAL,
      iid                   Dsrc-EID,      OPTIONAL & not used
      }
```

ACTION-Request can be used in confirmed (parameter `mode` = TRUE) or non-confirmed mode (parameter `mode` = FALSE). A response shall always be expected in the former case.

`eid` identifies the element in the OBE containing the attributes which are accessed. . `accessCredentials` are for optional use and may carry information necessary to fulfil the access conditions for the attributes. `actionParameter` is of type Container CHOICE. For the application EFC special CHOICE types have been defined in [5].

ACTION-Response

```
     ACTION-Response ::= SEQUENCE {
      fill                  BIT STRING(SIZE(1)),
      eid                   Dsrc-EID,
      iid                   Dsrc-EID,      OPTIONAL & not used,
      responseParameter     Container      OPTIONAL,
      ret                   ReturnStatus   OPTIONAL
      }
```

ACTION-Response shall explicitly convey the result of the corresponding ACTION-Request command. If some error occurred during execution of the command, the `ret` parameter shall contain an error indication (see below). `responseParameter` is of type Container CHOICE. For the application EFC special CHOICE types have been defined in [5].

INITIALISATION

The Initialisation-Request and -Response services are described in Chapter 5.2.

EVENT-REPORT

The Event-Report-Request and -Response services are described in Chapter 5.3.

Summary of Return Codes

The parameter `ret` inside the responses described above can have the following values.

returnStatus	associated command services
noError (0)	service successfully executed
accessDenied (1)	the requested operation was not performed for reasons pertinent to the security of the system
argumentError (2)	the requested operation was not performed because one or more arguments could not be handled
complexityLimitation (3)	the requested operation was not performed because a parameter was too complex
processingFailure (4)	a general failure in processing the operation was encountered
processing (5)	not used
chainingError (6)	command service not executed (see Chapter 5.1.6)

Table 5.2: Coding of returnStatus

5.1.4 Encoding procedure

The Application Layer shall encode the application service requests and responses according to ASN.1-BASIC-PER, UNALIGNED. The encodable ASN.1 types are specified in [3]. The encoding of a PDU results in an ordered sequence of octets as described in the two examples below.

Example #1 Encoding of ASN.1 type : INTEGER(0..65535) using PER UNALIGNED

This integer has a range of 64 K, so it has to be encoded as a positive-binary-integer in a bit-field with the minimum number of bits necessary to represent the range (16 bits).

The leading bit of the PER bit-field is defined as the most significant bit of the first octet(B15), and the trailing bit of the field is defined as the least significant bit of the second octet (B0)

octets	bits							
octet #1	B15	B14	B13	B12	B11	B10	B9	B8
octet #2	B7	B6	B5	B4	B3	B2	B1	B0
	MSB							LSB

Table 5.3: Encoding of an Integer

B15 : Most significant bit of the PER bit-field
B0 : Least significant bit of the PER bit-field

The Application Layer encoder transfers this PER bit-field as two octets with :

- bit B15 as the MSB of the first octet, bit B8 as the LSB of the first octet
- bit B7 as the MSB of the second octet, bit B0 as the LSB of the second octet

Example #2 :

The ASN.1 definition of the application layer service Get-Request was given above. using PER UNALIGNED, the resulting coding is given by the following table.

octets	MSB LSB	description
#1	0110	T-APDU: Get-Request (1st element in CHOICE codes with 0)
	0	no `accessCredentials`
	0	no `iid`
	1	`attrIdList` present
	0	`fill`
#2	0000 1010	`eid` = 0x0a = 10
#3	0000 0001	length of attrIdList = 1
#4	0000 0111	AttributeId = 0x07

Table 5.4: Coding of Get-Request

The encoded application layer service is called APDU and it is transmitted over the DSRC link with octet#1 first and the LSB of each octet transmitted first according to the data link layer rules.

5.1.5 Fragment Identification

Each encoded APDU is preceded by a fragment header. The fragment header is used to identify a fragment.

The general format of a fragment is as follows :

fragment header	encoded APDU
1 octet	N octets

Figure 5.2: Format of a fragment

Each fragment contains one complete encoded Application Layer service request or response.

As indicated in Chapter 3, each frame can contain one or more fragments.

The format of the fragment header octet is as follows :

MSB LSB	Description	Remark
1	Fragmentation Indicator	'1' indicates that the application layer service request or response fits into one fragment
xxx x	APDU Number : identifier of the fragment	(see below)
00	Fragment Counter	-
1	Extension Indicator	'1' indicates that no extension of the Fragment Counter is used

Table 5.5 : Format of the fragment header

The following rules apply to the usage of the APDU number:

- the values 0000_2 and 0001_2 are not used.
- the values of the APDU number of a fragment associated to a request service (e.g. Get-Request) and a fragment associated to the corresponding response service (e.g. Get-Response) shall be identical after the Initialisation phase
- the values of the APDU number of fragments belonging to the same chain shall be identical (see following subchapter).

5.1.6 Concatenation procedure

The concatenation procedure allows to map consecutive fragments on one frame as long as the size constraints of the frame are not violated and as long as a response is expected either for all service requests or for none of them.

fragment #1	fragment #2	fragment #3
PDUnumber = a	PDUnumber = b	PDUnumber = c

Figure 5.3: Concatenation of fragments

The PDU numbers of two consecutive fragments shall be different, except when the concatenated fragments are to be chained, see chapter 5.1.7.

The execution of service requests contained in concatenated fragments is sequential, in the order given by the position of the fragments inside the frame. The service request associated to the first fragment of the frame is executed first.

If a response is expected, there shall be a response frame containing a service response fragment for each processed service request, positioned in the same order as the request fragments in the command frame.

5.1.7 Chaining procedure

The chaining procedure allows to build up chains of fragments.

All PDU numbers of chained fragments shall be identical.

fragment 1	fragment 2	fragment 3
PDUnumber=a	PDUnumber=a	PDUnumber=a

Figure 5.4: Chain of fragments

A chain shall always fit into one frame. There shall be not more than one chain in one frame.

The execution of a chained service request is always dependent on the successful execution of all previous service requests in the chain. If a service request is not successfully executed, the subsequent service-requests are not executed.

If a response is expected, there shall be a response frame containing a service response fragment for each processed service request, positioned in the same order as the request fragments in the command frame.

If a chained service request can not be executed, and a response is expected, all subsequent service requests in the chain can be responded to with return status equal to chaining-error.

Example:

Assuming a request chain constituted with confirmed service requests as in the table below:

REQUEST CHAIN		
fragment #1	fragment #2	fragment #3
T-APDU Get-Request	T-APDU Set-Request	T-APDU Action-Request

Figure 5.5: Chain of Requests

The order of execution of the services is

1. Successful execution of the service GET
2. Processing failure of the service SET
3. No execution of the service ACTION

Resulting response chain:

RESPONSE CHAIN		
fragment #1 T-APDU Get-Response returnStatus = 0 (no error)	fragment #2 T-APDU Set-Response returnStatus = 4 (processingFailure)	fragment #3 T-APDU Action-Response returnStatus = 6 (chainingError)

Figure 5.6: Chain of Responses

5.1.8 Multiplexing procedure

Multiplexing of applications is supported by the fact that each application service response can be uniquely related to the corresponding application service request, which allows for directing the responses to the invoking application.

5.2 Initialisation

5.2.1 Overview

Before any two-way information exchange (down- and uplink) can be performed, the Initialisation procedure will have to take place. The Initialisation procedure is necessary to

- transmit the private LID to the RSE
- transmit the VST to the RSE

On reception of a new BST = Beacon Service Table matching the OBEs capabilities and applications, the OBE first creates a new private LID and then transmits the LID to the RSE together with the VST = Vehicle Service Table. In the following clauses the content of BST and VST are described as well as the precise procedure of Initialisation.

5.2.2 Content of BST

5.2.2.1 Overview

According to [3], the BST is a T-APDU called Initialisation-Request, which is periodically broadcasted using a UI command.

Content of the BST shall be according to [3]. The definition of the BST according to ASN.1 is

```
BST ::= SEQUENCE {
  beacon                beaconID,
  time                  Time,
  profile               Profile,
  mandApplications      ApplicationList,
  nonmandApplications   ApplicationList OPTIONAL & not used,
  profileList           SEQUENCE(0...127,...) OF Profile
  }
```

There shall be no `nonmandApplications` (i.e., non-mandatory applications) inside the BST.

In the following, the data elements contained in the BST are described in more detail.

5.2.2.2 BeaconID

According to [3] the BeaconID is defined as

```
BeaconID ::= SEQUENCE{
   manufacturerid        INTEGER(0..65535),
   individualid          INTEGER(0..2^27-1)
   }
```

Using PER, 16 bits are needed to code the manufacturerid and 27 bit to code the individualid. The manufacturerid is a number registered by CEN, see [3]. The individualid shall identify unambiguously a RSE issued by a manufacturer.

5.2.2.3 Time

According to [3], Time is defined as

```
Time ::= INTEGER(0..2^32-1)
```

Using PER, 32 bit are needed to code the Time.
The value of Time is the number of seconds passed since 1st January 1970, 0:00 am (UTC).

5.2.2.4 Profile

Profile is defined as

```
Profile ::= INTEGER(0..127,...)
```

The extendibility is not used. Therefore, the MSB is set to 0.

The Profile is according to [4]. The GSS Specification uses Profile =0 and =1, compare Chapter 4.4. The difference between Profile =0 and Profile =1 is the sub-carrier frequency to be used by the OBE in the uplink.

5.2.2.5 ApplicationList

ApplicationList is defined as

```
ApplicationList ::= SEQUENCE (0..127,...) OF
   SEQUENCE{
   aid                  DSRCApplicationEntityID,
   eid                  Dsrc-EID   OPTIONAL,
   parameter            Container  OPTIONAL
   }
```

It is required:

1. eid shall not be used within the BST
2. parameter shall not be used within the BST

This is in accordance with [5].

An ApplicationList having one application (=EFC) codes as follows:

octets	MSB LSB	description
#1	0000 0001	number of applications = 1
#2	0	no eid
	0	no parameter
	00 0001	aid = 1 = EFC

Table 5.6: Coding of ApplicationList

5.2.2.6 *ProfileList*

To provide a migration path from existing systems to GSS, the `profileList` may also contain other profiles than those used by the GSS specification. However, the extendibility of the maximum length of the list shall not be used and the `profileList` shall not contain profile 0 or 1.

5.2.2.7 *Example*

A complete frame (without bit-stuffing) containing an Initialisation-Request (BST) having an ApplicationList `mandApplications` with one entry equal to EFC, codes as follows:

octets	MSB LSB	description
#1	0111 1110	start flag
#2	1111 1111	Broadcast LID
#3	1010 0000	MAC control field
#4	0000 0011	LLC control field
#5	1xxx x001	fragmentation header (xxx x = PDU number, ≠ 0000 and ≠ 0001)
#6	1000	T-APDU: Initialisation-Request (1st element in CHOICE codes with 0!)
	0	no `nonmandApplications`
	000	`BeaconID.manufacturerid` = 1
#7	0000 0000	
#8	0000 1	
	001	`BeaconID.individualid` = 0x1234567 = 19088743
#9	0010 0011	
#10	0100 0101	
#11	0110 0111	
#12	0011 0010	`time` = 8514720001 = 0x32C06E81
#13	1100 0000	
#14	0110 1110	
#15	1000 0001	
#16	0000 0001	`profile` = 1
#17	0000 0001	number of mand. app. = 1
#18	0	no eid
	0	no parameter
	00 0001	`aid` = 1
#19	0000 0000	`profileList`, empty
#20	xxxx xxxx	FCS
#21	xxxx xxxx	
#22	0111 1110	end flag

Table 5.7: Frame containing a BST

5.2.3 Content of VST

5.2.3.1 *Overview*

According to [3], the VST is a T-APDU called Initialisation-Response, which is sent by OBEs using a private UI command.

The content of the VST shall be according to [3]. The definition of the VST using ASN.1 is

```
VST ::= SEQUENCE {
  fill                  BIT STRING(SIZE(4))
  profile               Profile,
  applications          ApplicationList,
  obeConfiguration      ObeConfiguration
  }
```

In the following, the data elements of the VST are described in more detail.

5.2.3.2 Profile

The `profile` returned in the VST shall match one of the `profiles` contained in the BST.

5.2.3.3 ApplicationList

According to Application Layer document [3] `ApplicationList` is defined as

```
ApplicationList ::= SEQUENCE (0..127,...) OF
   SEQUENCE{
   aid                  DSRCApplicationEntityID,
   eid                  Dsrc-EID   OPTIONAL & present,
   parameter            Container  OPTIONAL & present
   }
```

The following requirements apply for the ApplicationList of the VST

1. The `ApplicationList` of the VST shall contain a subset of those applications offered by the RSE inside the BST and which are supported by the OBE.
2. For each application in the VST's ApplicationList, there shall be at least one `eid` which is unique within the context of the OBE. Hence, by means of the `eid` data belonging to a certain application inside the OBE can be addressed.
3. The `parameter` shall be of type OCTET STRING lead by a Context Mark, and the decoding shall be unique under a given aid.

In [5], there is defined the coding of the parameter field for the application EFC.

5.2.3.4 ObeConfiguration

According to [3], `ObeConfiguration` is defined as

```
ObeConfiguration ::= SEQUENCE{
   equipmentClass       INTEGER(0..32767),
   manufacturerID       INTEGER(0..65535),
   obeStatus            INTEGER(0..65535) OPTIONAL & present
   }
```

The use and coding of `equipmentClass` is specified by the manufacturer of the equipment. Capabilities and limitations of a certain equipment class will be communicated to the implementors and operators of RSEs, in order to allow for adoption of the RSE's application according to the OBE's capabilities.

The field `obeStatus` shall always be present and the coding of `obeStatus` is specific to the GSS specification. Using PER, `obeStatus` codes with two octets.

- The first octet is used for transmitting the *OBE System status*
- The second octet in reserved for private use

Table 5.8 below contains an overview of the position of the status bits of `obeStatus`.
Not all status bits are relevant for all OBEs.

octets	MSB LSB	description (when applicable)
#1	x	Absence of ICC
	x	Unrecognised ICC
	x	Battery failure
	x	Peripheral interface Error
	x	Illegal tampering of the OBE
	abc	these three bits are interpreted as a number. It codes the last state of the OBE DSRC Kernel before SLEEP state, compare Chapter 6.3
		000 = BLOCKED
		001 = WAIT
		010 = INIT
		011 = READY
		100 = DATA
#2	xxxx xxxx	(reserved for private use)

Table 5.8: Coding of obeStatus

5.2.4 Procedure

In the underlying DSRC standards, the private LID Establishment procedure is described in the Data Link Layer and in the Application Layer documents ([2],[3]). In the following, the resulting behaviour is described.

Five major steps are needed:

1) Initialisation.Request: RSE broadcasts the BST periodically
2) Private LID Creation: On reception of a BST, the OBE creates a private LID (see below)
3) VST creation
4) Private LID Transmission: Transmits the chosen private LID to the RSE and requests the allocation of a private uplink window
5) Transmission of the VST in the allocated private uplink window

5.2.4.1 Private LID Creation

On reception of a BST the OBE performs the following actions:

1. Check whether the BeaconID contained in the BST is different from the last received BeaconID (which was stored in the OBE)
2. Check the Date and Time contained in the BST
3. Check the DSRC profile number, describing the setting of all parameters of Layer 1 and 2 which are important for interoperability.
4. Check the ApplicationList. If there is no match between the RSE's ApplicationList and the OBE's ApplicationList, no communication will take place.

5.2.4.2 VST creation

When all the necessary conditions for Initialisation are fulfilled, the OBE assembles the VST containing the ApplicationEntityIDs which exist in the ApplicationList of the OBE and which are included in the ApplicationList of the BST.

The following flow chart describes the behaviour in more detail.

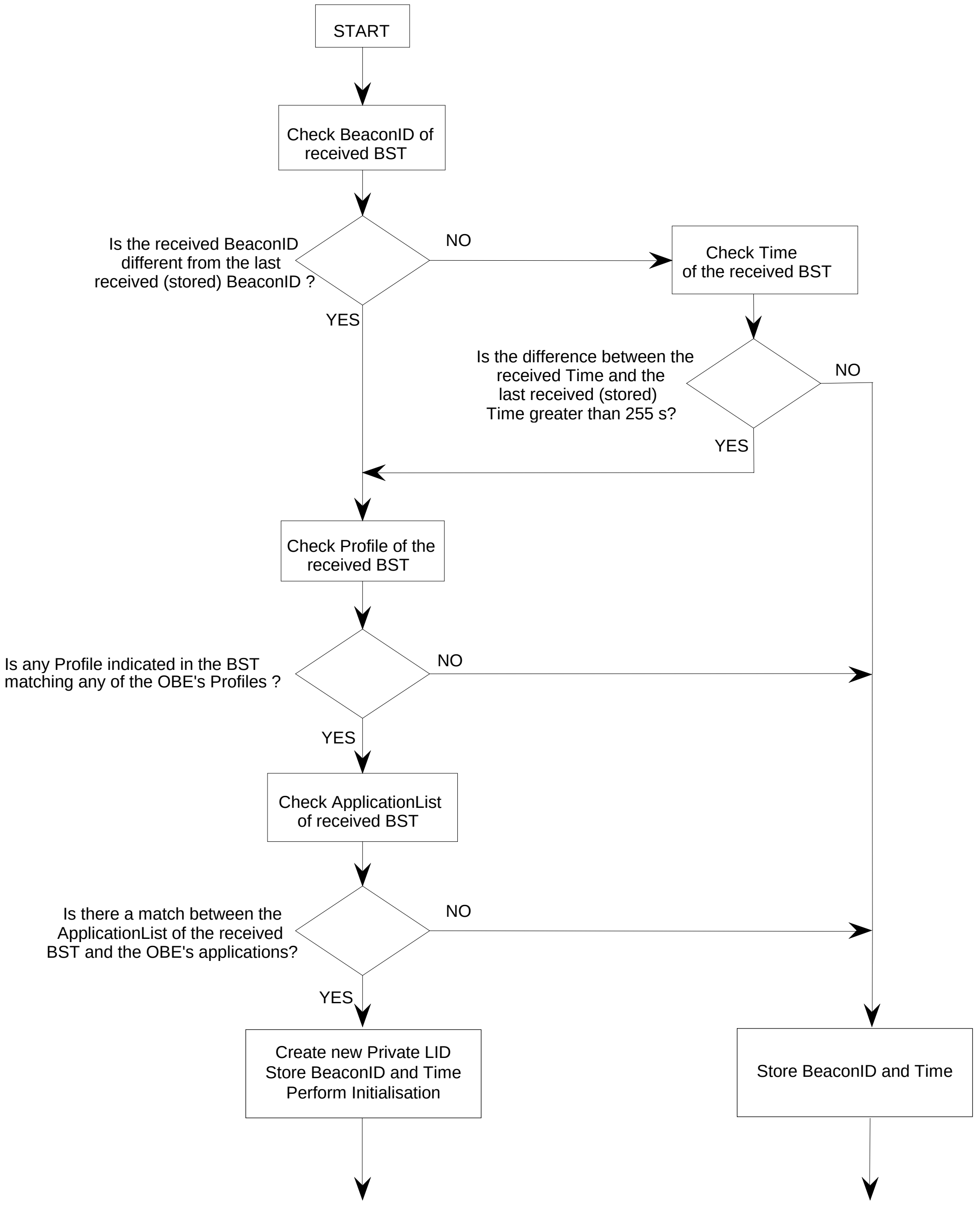
START
Check BeaconID of received BST
Is the received BeaconID different from the last received (stored) BeaconID ?
NO
YES
Check Time of the received BST
Is the difference between the received Time and the last received (stored) Time greater than 255 s?
NO
YES
Check Profile of the received BST
Is any Profile indicated in the BST matching any of the OBE's Profiles ?
NO
YES
Check ApplicationList of received BST
Is there a match between the ApplicationList of the received BST and the OBE's applications?
NO
YES
Create new Private LID
Store BeaconID and Time
Perform Initialisation
Store BeaconID and Time

Figure 5.7: Evaluation of BST and Creation of a new private LID

5.2.4.3 Transmission of VST

The Initialisation-Request is responded by those OBEs having created a new private LID.

1. The OBE randomly chooses one of the three allocated public up-link windows and sends a Private Window Request (PrWRq) containing the private LID.
2. On reception of a PrWRq, the RSE allocates a private window to that OBE having sent the PrWRq.
3. In the allocated private up-link window, the OBE sends the VST.

There are three mechanisms for recovery of lost messages:

1. On reception of BSTs, the OBE re-transmits the PrWRq
2. The RSE may re-transmit the PrWA until a VST is received.
3. The RSE may retransmit a BST, in which case the OBE transmits a PrWRq again. Only the reception of a frame with the OBE's private LID other than PrWA causes the OBE to stop requesting a private window for transmission of the VST. This mechanism is called 'implicit Application Layer Acknowledge for VST'.

The detailed behaviour is described in Chapter 6.3.

5.2.4.4 Examples

The described behaviour of creating new private LIDs and subsequent Initialisation has some consequences for the system operation. This can be illustrated with the following examples.

Example 1: If the OBE communicates with RSE 1 and after that it communicates with RSE 2, the private LID which had been used at RSE 1 will no more be available, even if OBE returns to RSE 1 in less than 255 s. The OBE will create a new private LID and use it for communication.

Example 2: If there is a shadowing situation (e.g., behind a heavy vehicle) lasting less than 255 s, there will be no new private LID and no VST will be sent after removal of the shadowing object. If the shadowing situation lasts more than 255 s, a new private LID will be created and Initialisation will be performed.

Example 3: Once the OBE has received a RELEASE command from a RSE (cmp. Chapter 5.3.3), the private LID is no more valid, and communication with the same RSE will be possible only after 255s have passed.

5.3 Use of the EVENT-REPORT

5.3.1 Overview

According to [3], the invocation of the Event-Report-Request service shall result in the notification of an event to a peer service user. This service is used by the RSE, to transmit a RELEASE command to the OBE

5.3.2 Format

According to [3], the ASN.1 definition of a Event-Report-Request is given by

```
Event-Report-Request ::= SEQUENCE {
  mode                  BOOLEAN,
  eid                   Dsrc-EID,
  eventType             INTEGER (0..127,...),
  accessCredentials     OCTET STRING   OPTIONAL,
  eventParameter        Container      OPTIONAL,
  iid                   Dsrc-EID,      OPTIONAL & not used
  }
```

The Event-Report-Request shall always be used in the non-confirmed mode, i.e. with parameter `mode` = 0.

5.3.3 Event-Report 'RELEASE'

The RELEASE command is transmitted by the RSE to close the communication with an OBE. As a result, the private LID will no longer be valid.

The RELEASE command is an Event-Report-Request having no `accessCredentials`, no `eventParameter`, and `eventType` = 0.

Using Packed Encoding Rules (PER), this command codes as follows:

MSB LSB	description
0010	T-APDU Event-Report-Request
0	no access credentials
0	no eventParameter
0	no iid
0	mode = FALSE
0000 0000	eid
0000 0000	eventType = 0

Table 5.10 : Encoding of the Event-Report-Request 'RELEASE'

The RELEASE command can only be transmitted with UI, compare table 5.11 on the supported combinations.

5.4 Use of the ACTION

5.4.1 Overview

According to [3], the invocation of the Action-Request service shall result in the performance of an action by a peer application. One use this service is to transmit a SET_MMI command to the OBE.

5.4.2 Format

According to [3], the ASN.1 definition of a Action-Request is given by

```
Action-Request ::= SEQUENCE {
  mode                   BOOLEAN,
  eid                    Dsrc-EID,
  actionType             INTEGER (0..127,...),
  accessCredentials      OCTET STRING   OPTIONAL,
  actionParameter        Container      OPTIONAL,
  iid                    Dsrc-EID,      OPTIONAL & not used
  }
```

5.4.3 Action 'SET_MMI'

The SET_MMI command is transmitted by the RSE to command the OBE to give an indication to the OBE user. The SET_MMI command is an Action-Request having `actionType` = 10, no `accessCredentials`, and `actionParameter` set to a value in the range 0...255.

Using Packed Encoding Rules (PER), this command codes as follows:

MSB LSB	**description**
0000	T-APDU Action-Request
0	no access credentials
1	actionParameter included
0	no iid
1	mode = TRUE
0000 0000	eid
0000 1010	actionType = 10
0000 0000	actionParameter = Container choice value = 0
0000 0000	MMI value

Table 5.11 : Encoding of the Action-Request ‘SET_MMI'

5.5 Supported Combinations

5.5.1 Overview

The standards for DSRC, [1], [2] and [3] contain a number of options and alternatives on each layer. Depending on the requirements and restrictions different options might be used by different applications and installations.

However in order to arrive at a harmonised protocol while maintaining both high performance and reasonable cost certain choices of combinations must be made. This subchapter gives a summary of the combinations supported by this specification. It is not required for all equipment to implement all supported combinations.

5.5.2 Downlink

Downlink Frames							
no	**LID**	**MAC**	**LLC**	**APDU**	**Remarks**	**RSE**	**OBE**
1	Private	20/28	none	none	Private Uplink Window Allocation	M	M
2	Broadcast	A0	03	INIT.request (BST)	Broadcast UI command with window allocation	M	M
3	Broadcast	80	03	SET. request, mode = 0 ACTION.request, mode = 0	Broadcast UI command no window allocation	opt	opt
4	Private	80	03	SET. request, mode = 0 ACTION. request, mode = 0	Private UI command no window allocation	opt	opt
5	Private	80	03	EVENT_REPORT. request (RELEASE), mode = 0	Private UI command no window allocation	M	M
6	Private	A0/A8	67/E7	SET. request, mode = 0 ACTION. request, mode = 0	Private AC command with p = 0 with window allocation	M	M
7	Private	A0/A8	77/F7	GET. request SET. request, mode = 1 ACTION. request, mode = 1	Private AC command with p = 1 with window allocation	M	M

Table 5.12: Supported combinations on downlink

M Indicates that it is mandatory to support the frame

opt RSE Indicates that it is optional for the RSE to support the frame

opt OBE Indicates that it is optional for the OBE to support the frame, and that information contained in the VST can be used to deduce whether the frame is supported or not

5.5.3 Uplink

Uplink Frames								
no	**LID**	**MAC**	**LLC**	**LLC status**	**APDU**	**Remark**	**RSE**	**OBE**
1	Private	60	none	none	none	Private Uplink Window Req. (in public uplink window only)	M	M
2	Private	C0	03	none	INIT.response (VST)	Private UI command no window request	M	M
3	Private	C0	03	none	GET. response, SET. response, ACTION. response	Private UI command no window request	M	opt
4	Private	D0	67/E7	40	none (APDU not requested)	Private ACn response f = 0 no window request	M	M
5	Private	D0	77/F7	30	none (APDU not available)	Private ACn response f = 1 no window request	M	opt
6	Private	D0	77/F7	00	GET. response, SET. response, ACTION. response	Private ACn response f = 1 no window request	M	M

Table 5.13: Supported combinations on uplink

5.5.4 Size of fields (octets)

	Downlink							Uplink					
Frame number	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(1)	(2)	(3)	(4)	(5)	(6)
Start flag	1	1	1	1	1	1		1	1	1	1	1	1
LID	4	1	1	4	4	4	4	4	4	4	4	4	4
MAC control field	1	1	1	1	1	1	1	1	1	1	1	1	1
LLC control field	0	1	1	1	1	1	1	0	1	1	1	1	1
LLC Status Field	0	0	0	0	0	0	0	0	0	0	1	1	1
Fragmentation header	0	1	1	1	1	1	1	0	1	1	0	0	1
Max APDU length	0	120	120	117	3	117	117	0	117	117*	0	0	116
FCS	2	2	2	2	2	2	2	2	2	2	2	2	2
End flag	1	1	1	1	1	1	1	1	1	1	1	1	1
Total maximum size	**9**	**128**	**128**	**128**	**14**	**128**	**128**	**9**	**128**	**128**	**11**	**11**	**128**

Table 5.14: Size of fields of supported combinations

* Please note that for interoperability between the two response modes defined in Layer 2 using uplink(3) or uplink(6), the APDU length of uplink (3) should be limited to 116.

6 Inter Layer Management

6.1 Introduction

The standards [1], [2], [3] and [4] specify an OSI stack of communication layers for DSRC. However, there is no complete description of the inter layer management. Especially for OBEs going to sleep while no communication is active, the transitions from SLEEP to ACTIVE states and vice versa require a detailed description of the MAC, LLC and Application Layer procedures.

Interoperability between RSEs and OBEs of different manufacturers not only is affected by the format of messages being exchanged but also by the procedures for handling of error situations. The philosophy behind the approach presented here is to require a specific behaviour from the OBEs in situations causing transitions between SLEEP and ACTIVE states. Then, the RSEs can be designed in such a way that interoperability with OBEs of different manufacturers even in critical situations like shadowing is given.

6.2 Mechanism for Slow Access to Data

Another highlight of the approach to Inter Layer Management presented here is a mechanism of data transfer in situations, where the processing inside the OBE takes long time. This may be the case when cryptographic operations are involved, or, when the OBE accesses a chipcard while executing the command issued by the RSE (for example a TRANSFER_CHANNEL command with channelId = ICC, compare Ref. [5]).

The core idea is to allow the OBE to actively request a private window when the response to a command has been generated. When the private window is granted, the data is transmitted to the RSE using a UI LCC service. This provides another example of an Inter Layer Management procedure. The following figure provides an overview of the procedure, while the detailed description is a part of the state transition table provided in Chapter 6.3.

XX.Request denotes an arbitrary Application Layer command causing a slow access to data inside the OBE.

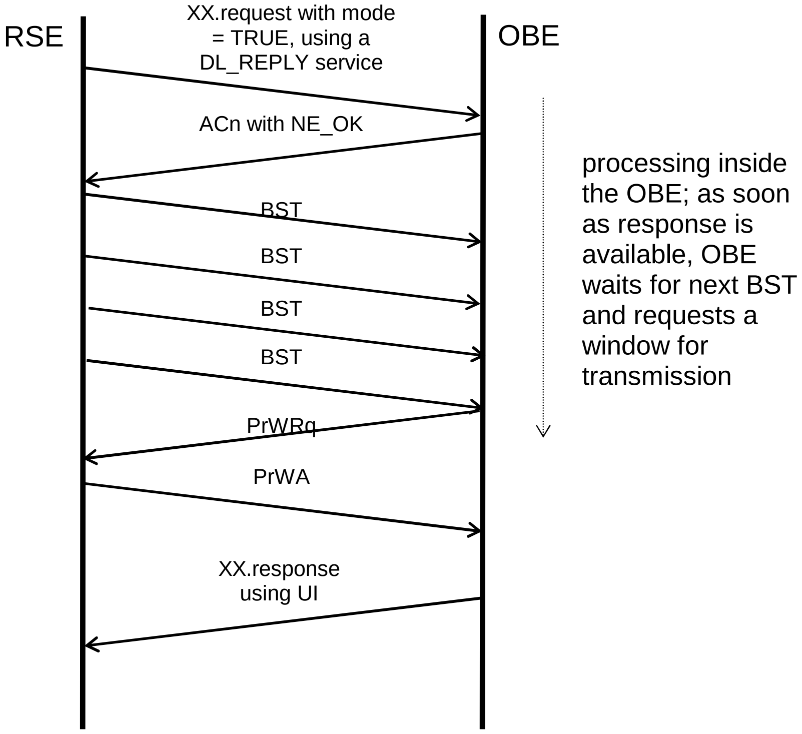
RSE
OBE
XX.request with mode = TRUE, using a DL_REPLY service
ACn with NE_OK
BST
BST
BST
BST
PrWRq
PrWA
XX.response using UI
processing inside the OBE; as soon as response is available, OBE waits for next BST and requests a window for transmission

Figure 6.1: Data exchange in case of slow access

6.3 State Transition Table for OBE DSRC Kernel

This chapter provides detailed description of the OBE’s behaviour with respect to

- wake-up and sleeping behaviour
- DSRC MAC functionality
- DSRC LLC functionality
- slow access to data
- Initialisation phase (I-Kernel functionality)

Since these functions interact with each other they are combined within one single state transition table. The following subchapters provide detailed descriptions for the states, the local variables used for the sub-qualification of events, the events, the actions and the complete state transition table. This state transition table is an extension of the LLC state transition tables defined in [2].

6.3.1 The States

State	Description
SLEEP	This is the normal sleeping state for saving battery power. In this state, the reception of frames is notified (see event Wake_Up_Signal), but the frames are not processed.
BLOCKED	The BLOCKED state is similar to the sleeping state, but in addition the frames are not notified, i.e. the wake-up signal is blocked. This state is used to reduce battery power consumption in certain situations.
WAIT	In the WAIT state power consumption may be reduced but it is mandatory to preserve the application context. Especially if an ICC is used, the ICC shall be powered in order not to loose information like session keys etc. Inside the OBE, all local variables (compare next chapter) shall be preserved.
COM_READY	In this state the OBE is able to receive and process broadcast frames. There is no valid LID and hence addressed frames can not be received and processed.
EVAL_BST	This is an OBE internal processing state and is left after the processing is finished. The last received BST is analysed and depending on the result actions are performed and new states are reached.
INIT	In the state INIT the OBE awaits a PrWA from the RSE in order to send the VST. The state INIT is left on any addressed frame other than PrWA (= ‘implicit application layer acknowledge for VST’)
READY	In state READY, the OBE can receive and process broadcast and private frames, since there is a valid LID.
BUSY	In the BUSY state the OBE is processing a command which could not be finalised within 160+320=480µs (=T3+T4a). This occurs mainly when an ICC was addressed. The BUSY state is an internal processing state and is left after the processing is finished.
DATA_1	In state DATA_1 the OBE has a valid response message in the SAVE buffer and it awaits a BST in order to send a PrWRq.
DATA_2	In state DATA_2, the OBE awaits a PrWA in order to send the message in the SAVE in the allocated private uplink window. The state DATA_2 is left on any addressed frame other than PrWA (= ‘implicit application layer acknowledge for SAVE’)

Table 6.1: States of the OBE DSRC Kernel

6.3.2 The Local Variables

Variable	Description
SavedState	This variable holds the last state before going to a sleeping state (WAIT, BLOCKED, INIT, READY). Furthermore, it can have the value DATA, see state transition table below. It shall be located in non-volatile memory. The contents of this variable is reported to the RSE within the VST (three least significant bits of obeStatus octet #1, see Chapter 5.2.3.4).
SavedLID	This variable holds the last valid LID before going to a sleeping state (SLEEP, BLOCKED or WAIT). It shall be located in non-volatile memory.
LID	This variable holds the actual valid LID to be used for receiving and transmitting private frames.
SavedBeaconId	This variable holds the last received BST-BeaconId.It shall be located in non-volatile memory.
SavedDateTime	This variable holds the last received BST-DateTime. It shall be located in non-volatile memory.
SavedSAVE	This variable holds content of the SAVE buffer before going to the WAIT state. It shall be located in non-volatile memory.
SAVE	This variable holds the generated response for delivery to the RSE, especially in cases of late responses.
VST	This is the prepared VST to be transmitted to the RSE.
V(RI)	This variable holds the value of the LLC sequence bit of the last transmitted response. The initial value is 0.

Table 6.2: Local Variables of the OBE DSRC Kernel

6.3.3 The Events

There are two kinds of events: a) real events and b) pseudo events which merely serve as a sub-qualification of real events, or, control the transitions from the internal processing states (EVAL_BST and BUSY). The first table summarises the real events, the second table shows the pseudo events.

Event	Description
Wake_Up_Signal	This event occurs whenever bit-modulated carrier (i.e., a frame) is detected.
TWait_expired	The timer controlling the duration of the WAIT state expired. TWait should be 255 s.
TBlocked_expired	The timer controlling the duration of the BLOCKED state. Tblocked should be in the order of 3 s.
TW_expired	The timer observing the wake-up signal expired. TW should be in the order of 100 ms.
Rec_Broadcast_UI (LSDU)	A UI command with public address and containing LSDU was received.
Rec_Private_UI (LSDU)	A UI command with private address and containing LSDU was received.
Rec_BST (BeaconId, DateTime)	A BST containing BeaconId and DateTime was received.
Rec_PrWA	A private window allocation message (PrWA) was received
Rec_ACn (SQC, P, INFO)	An ACn command with sequence bit SQC, poll bit P and information field INFO was received

Table 6.3: Events of the OBE DSRC Kernel

Pseudo Events	Description
Profile_Match=TRUE	There is one communication profile offered by the RSE which is supported by the OBE.
Profile_Match=FALSE	None of the communication profiles offered by the RSE inside the BST is supported by the OBE.
App_Match=TRUE	At least one application offered by the RSE inside the ApplicationList of the BST is supported by the OBE.
App_Match=FALSE	None of the applications offered by the RSE inside the ApplicationList of the BST is supported by the OBE.
ACCESS=FAST	The response frame could be generated before time T3+T4a = 480µs elapsed.
ACCESS=SLOW	The response frame could not be generated before time T3+T4a = 480µs elapsed.
Processing_Completed(LSDU)	The requested response LSDU is ready for transmission to the RSE.

Table 6.4: Pseudo Events of the OBE DSRC Kernel

6.3.4 The Actions

Action	**Description**
ResTW	The timer observing the time-out for the wake-up signal is reseted.
ResTBlocked	The timer guarding the duration for the BLOCKED state is reseted.
ResTWait	The timer guarding the duration for the WAIT state is reseted.
PowDown	Power down all circuits except the wake-up circuits.
CreateLID	Create a new LID and use it as the currently valid LID
Transmit_PrWRq	Transmit a private window request message
Transmit_UI(x)	Transmit a UI command to the RSE, containing ‘x’ as the LSDU
Transmit_ACn_RSP (SQR,F,STATUS,LSDU)	Transmit a ACn response frame with sequence bit SQR, final bit F, status subfield STATUS and the LSDU.
Retransmit_Previous_ Frame	Retransmit the previous frame; the frame shall not be deleted from the send queue after transmission since there might be again a retransmission.
UI_IND(x)	Inform the data link layer user on the reception of a UI command with given LSDU = ‘x’.
DATA_ACK_IND(x)	Inform the data link layer user on the reception of a ACn command with given LSDU = ‘x’.
REPLY_IND(x)	Inform the data link layer user on the reception of a DL_REPLY command with given LSDU = ‘x’.
GenerateLSDU	In cases of fast access to the data, the response LSDU is already generated during the resp. transition of the transition table. The generated LSDU is stored in the local variable SAVE.

Table 6.5: Actions of the OBE MAC/LLC/I-Kernel

6.3.5 State Transition Table

State	Ref	Event	Action(s)	Next State
WAIT	1	Wake_Up_Signal	ResTW, LID:=SavedLID, SAVE:=SavedSAVE	DATA_1
	2	TWait_expired	SavedState:=WAIT, PowDown	SLEEP
SLEEP	3	Wake_Up_Signal & SavedState=BLOCKED	ResTW	COM_READY
	4	Wake_Up_Signal & SavedState=WAIT	LID:=SavedLID, ResTW	COM_READY
	5	Wake_Up_Signal & SavedState=INIT	LID:=SavedLID, ResTW	COM_READY
	6	Wake_Up_Signal & SavedState=READY	LID:=SavedLID, ResTW	COM_READY
BLOCKED	7	TBlocked_expired	no action	SLEEP
COM_ READY	8	Rec_Broadcast_UI(LSDU)	UI_IND(LSDU)	COM_READY
	9	Rec_BST(BeaconId, DateTime)	no action	EVAL_BST
	10	TW_expired	PowDown	SLEEP
	11	(other events)	no action	COM_READY
EVAL_BST	12	BeaconId <> SavedBeaconId & Profile_Match=TRUE & App_Match=TRUE	SavedBeaconId:=BeaconId, SavedDateTime:=DateTime, CreateLID, Transmit_PrWRq	INIT
	13	BeaconId = SavedBeaconId & (DateTime – SavedDateTime) >= 255 s & Profile_Match=TRUE & App_Match=TRUE	SavedDateTime:=DateTime , CreateLID, Transmit_PrWRq	INIT
	14	BeaconId = SavedBeaconId & (DateTime – SavedDateTime) < 255 s & SavedState=READY	SavedDateTime:=DateTime	READY
	15	BeaconId = SavedBeaconId & (DateTime – SavedDateTime) < 255 s & SavedState=INIT	SavedDateTime:=DateTime, Transmit_PrWRq	INIT
	16	BeaconId = SavedBeaconId & (DateTime – SavedDateTime) < 255 s & SavedState=WAIT	SavedDateTime:=DateTime	READY
	17	BeaconId <> SavedBeaconId & (Profile_Match=FALSE or App_Match=FALSE)	SavedBeaconId:=BeaconId, SavedDateTime:=DateTime, SavedState=BLOCKED, ResTBlocked, PowDown	BLOCKED
	18	BeaconId = SavedBeaconId & (DateTime – SavedDateTime) >= 255 s & (Profile_Match=FALSE or App_Match=FALSE)	SavedDateTime:=DateTime, SavedState=BLOCKED, ResTBlocked, PowDown	BLOCKED
	19	BeaconId = SavedBeaconId & (DateTime – SavedDateTime) < 255 s & SavedState=BLOCKED	SavedDateTime:=DateTime, ResTBlocked, PowDown	BLOCKED

INIT	20	Rec_BST(BeaconId, DateTime) & BeaconId <> SavedBeaconId	SavedState:=INIT	EVAL_BST
	21	Rec_BST(BeaconId, DateTime) & BeaconId = SavedBeaconId	SavedDateTime:=DateTime, Transmit_PrWRq	INIT
	22	Rec_PrWA	Transmit_UI(VST)	INIT
	23	Rec_Broadcast_UI(LSDU)	UI_IND(LSDU)	INIT
	24	Rec_Private_UI(LSDU) & LSDU <> RELEASE	UI_IND(LSDU)	READY
	25	Rec_Private_UI(LSDU) & LSDU = RELEASE	SavedState=BLOCKED, ResTBlocked, PowDown	BLOCKED

	26	Rec_ACn (SQC=V(RI), P=*0*,
		INFO<>NULL)
			DATA_ACK_IND(INFO)
			Transmit_ACn_RSP(SQR=*1*-SQC, F=*0*,
			STATUS=NR_OK, LSDU=NULL)
			V(RI):=*1*-SQC
				READY

	27	Rec_ACn (SQC=V(RI), P=*1*, INFO<>NULL) & ACCESS=FAST	REPLY_IND(INFO) GenerateLSDU Transmit_ACn_RSP(SQR=*1*-SQC, F=*1*, STATUS=OK_OK, LSDU=SAVE) V(RI):=*1*-SQC	READY

	28	Rec_ACn (SQC=V(RI), P=*1*,
		INFO<>NULL)
		& ACCESS=SLOW
			REPLY_IND(INFO)
			Transmit_ACn_RSP(SQR=*1*-SQC, F=*1*,
			STATUS=NE_OK, LSDU=NULL)
			V(RI):=*1*-SQC
				BUSY

	29	TW_expired	SavedState=INIT, PowDown	SLEEP
	30	(other events)	no action	INIT
READY	31	Rec_PrWA	Retransmit_Previous_Frame	READY
	32	Rec_BST(BeaconId, DateTime) & BeaconId <> SavedBeaconId	SavedState=READY	EVAL_BST
	33	Rec_BST(BeaconId, DateTime) & BeaconId = SavedBeaconId	SavedDateTime:=DateTime	READY
	34	Rec_Broadcast_UI(LSDU)	UI_IND(LSDU)	READY
	35	Rec_Private_UI(LSDU) & LSDU <> RELEASE	UI_IND(LSDU)	READY
	36	Rec_Private_UI(LSDU) & LSDU = RELEASE	SavedState := BLOCKED ResTBlocked. PowDown	BLOCKED
	37	Rec_ACn (SQC=V(RI), P=*0*, INFO<>NULL)	DATA_ACK_IND Transmit_ACn_RSP(SQR=*1*-SQC, F=*0*, STATUS=NR_OK, LSDU=NULL) V(RI):=*1*-SQC	READY
	38	Rec_ACn (SQC=V(RI), P=*1*, INFO<>NULL) & ACCESS = FAST	REPLY_IND(INFO) GenerateLSDU Transmit_ACn_RSP(SQR=*1*-SQC, F=*1*, STATUS=OK_OK, LSDU=SAVE) V(RI):=*1*-SQC	READY

	39	Rec_ACn (SQC=V(RI), P=*1*,
		INFO<>NULL)
		& ACCESS = SLOW
			REPLY_IND(INFO)
			Transmit_ACn_RSP(SQR=*1*-SQC, F=*1*,
			STATUS=NE_OK, LSDU=NULL)
			V(RI):=*1*-SQC
				BUSY

	40	Rec_ACn (SQC<>V(RI), P=0, INFO<>NULL)	Transmit_ACn_RSP(SQR=*1*-SQC, F=0, STATUS=NR_OK, LSDU=NULL) (if P-bit of previous command was different, the behaviour of the OBE is undefined)	READY
	41	Rec_ACn (SQC<>V(RI), P=1, INFO<>NULL)	Transmit_ACn_RSP(SQR=*1*-SQC, F=1, STATUS=OK_OK, LSDU=SAVE) (if P-bit of previous command was different, the behaviour of the OBE is undefined)	READY
	42	TW_expired	SavedState := READY, PowDown	SLEEP
	43	(other events)	no action	READY

BUSY	44	Rec_Private_UI(LSDU) & LSDU <> RELEASE	UI_IND(LSDU)	BUSY
	45	Rec_Private_UI(LSDU) & LSDU = RELEASE	SavedState := BLOCKED, ResTBlocked. PowDown	BLOCKED
	46	Rec_ACn (SQC<>V(RI), P=*1*, INFO<>NULL)	Transmit(ACn_RSP(SQR=*1*-SQC, F=*1*, STATUS=NE_OK, LSDU=NULL))	BUSY
	47	Rec_PrWA	Retransmit_Previous_Frame	BUSY
	48	Processing_Completed(LSDU)	SAVE := LSDU	DATA_1
	49	(other events)	no action	BUSY
DATA_1	50	Rec_Private_UI(LSDU) & LSDU = RELEASE	SavedState := BLOCKED, ResTBlocked. PowDown	BLOCKED
	51	Rec_BST(BeaconId, DateTime) & BeaconId = SavedBeaconId	SavedDateTime:=DateTime, Transmit_PrWRq	DATA_2
	52	Rec_BST(BeaconId, DateTime) & BeaconId <> SavedBeaconId	SavedState := DATA	EVAL_BST
	53	Rec_Private_UI(LSDU) & LSDU <> RELEASE	UI_IND(LSDU)	DATA_1
	54	Rec_PrWA	Transmit_ACn_RSP(SQR=*1*-SQC, F=*1*, Status=OK_OK, LSDU=SAVE)	READY
	55	Rec_ACn (SQC<>V(RI), P=*1*, INFO<>NULL)	Transmit_ACn_RSP(SQR=*1*-SQC, F=*1*, Status=OK_OK, LSDU=SAVE)	READY
	56	TW_expired	ResTWait	WAIT
	57	(other events)	no action	DATA_1
DATA_2	58	Rec_Private_UI(LSDU) & LSDU <> RELEASE	UI_IND(LSDU)	READY
	59	Rec_Private_UI(LSDU) & LSDU = RELEASE	SavedState:=BLOCKED, ResTBlocked, PowDown	BLOCKED
	60	Rec_BST(BeaconId, DateTime) & BeaconId = SavedBeaconId	SavedDateTime:=DateTime, Transmit_PrWRq	DATA_2
	61	Rec_BST(BeaconId, DateTime) & BeaconId <> SavedBeaconId	SavedState := DATA	EVAL_BST
	62	Rec_PrWA	Transmit_UI(SAVE)	DATA_2
	63	Rec_ACn (SQC<>V(RI), P=*1*, INFO<>NULL)	Transmit_ACn_RSP(SQR=*1*-SQC, F=*1*, Status=OK_OK, LSDU=SAVE)	READY

	64	Rec_ACn (SQC=V(RI), P=*0*, INFO<>NULL)	DATA_ACK_IND(INFO)
			Transmit_ACn_RSP(SQR=*1*-SQC, F=*0*, STATUS=NR_OK, LSDU=NULL)
			V(RI):=*1*-SQC
				READY

	65	Rec_ACn (SQC=V(RI), P=*1*, INFO<>NULL)
		& ACCESS = FAST
			REPLY_IND(INFO)
			GenerateLSDU
			Transmit_ACn_RSP(SQR=*1*-SQC, F=*1*, STATUS=OK_OK, LSDU=SAVE)
			V(RI):=*1*-SQC
				READY

	66	Rec_ACn (SQC=V(RI), P=*1*, INFO<>NULL)
		& ACCESS = SLOW
			REPLY_IND(INFO)
			Transmit_ACn_RSP(SQR=*1*-SQC, F=*1*, STATUS=NE_OK, LSDU=NULL)
			V(RI):=*1*-SQC
				BUSY

	67	TW_expired	ResTWait	WAIT
	68	(other events)	no action	DATA_2

Table 6.6: State transition table of the OBE DSRC Kernel

6.3.6 Comments on the transitions

1. In the WAIT state, the context of the application is preserved. Therefore, on reception of a wake-up signal, it is possible to resume the transaction state, i.e. DATA_1. The previous LID and the saved LSDU will be used again. If the WAIT state was reached via a TW_expired in state DATA_2 (transition 67), the transaction can be resumed in state DATA_1 as well: on reception of a BST, state DATA_2 will be reached immediately (transition 51).
2. If TWait expired, the application context will not be preserved any longer, i.e. the ICC may be powered down. The variable SavedState remembers this situation and will be reported to a RSE during the initialisation (field OBEstatus inside VST).
3. This is the normal transition during wake-up after a previous transaction which was closed using the RELEASE command. The previous LID will not be used anymore.
4. Wake-up after an exceptional situation, see transition 2. The previous LID may be used again, if TWait was less than 255 s and the OBE wakes up under the same gantry again, compare subsequent transition 16.
5. Wake-up after reaching the SLEEP state via transition 29. The previous LID may be used again, if the duration of the SLEEP state was less than 255 s, compare subsequent transition 15.
6. Wake-up after reaching the SLEEP state via transition 42. The previous LID may be used again, if the duration of the SLEEP state was less than 255 s, compare subsequent transition 14.
7. After the time TBlocked expired (ca. 3 s), the BLOCKED state is left. In the SLEEP state, there will be a possibility to wake up on wake-up signals (transition 3).
8. This transition allows for OBEs which are able to receive broadcast UI commands without having previously performed an initialisation phase. This may be used for information services.
9. This is the normal case: after wake-up, the OBE receives the first BST. The date and time as well as the beacon Id are saved.
10. If the OBE woke up on a signal not stemming from a beacon (disturbing signals), it will reach the COM_READY state but it will not detect any further modulated carrier. Consequently, it will fall asleep again. The variable SavedState is not updated, since this case is not worth being reported to a beacon.
11. Other received frames will be discarded in state COM_READY, since there is no valid LID.
12. This is the normal transition for an OBE receiving the first BST from a new beacon. There is a supported profile and a known application offered by the beacon. The initialisation phase should be performed using a new LID.
13. The OBE woke up under the same gantry again and was asleep for more than 255 s. Therefore, a new LID is created and the Initialisation phase will be performed. (Note: If an OBE stays awake under a gantry, there will be no new initialisation after 255 s, because the stored date and time of the received BSTs is always updated, compare e.g. transition 33).
14. The OBE woke up under the same gantry again and was asleep for less than 255 s. The old LID will be used again. Since the SLEEP state was reached via the READY state (transition 42) it goes back to the READY state directly.
15. The OBE woke up under the same gantry again and was asleep for less than 255 s. The old LID will be used again. Since the SLEEP state was reached via the INIT state (transition 29), a PrWRq is transmitted and it goes back to the INIT state directly.
16. The OBE woke up under the same gantry again and was asleep for less than 255 s. The old LID will be used again. Since the SLEEP state was reached via the WAIT state (transition 2), the previous application context is lost and the transaction is resumed in state READY. (Note: This transition is possible only if TWait is set to less than 255 s).

17. There is no application offered by the beacon which is supported by the OBE. Therefore, the BLOCKED state is reached autonomously in order to save battery power.
18. The profile offered by the beacon is not supported by the OBE. Therefore, the BLOCKED state is reached autonomously in order to save battery power.
19. Since less than 255 have passed since reception of the last BST and the OBE was in the BLOCKED state, there is no need for communication at that beacon and the BLOCKED state is reached autonomously again. The date and time of the BST is saved. Therefore, the OBE will stay in a sequence of BLOCKED states, as long as it remains under that beacon.
20. This is an exceptional situation: the OBE receives a BST with a new beacon Id, but the previous initialisation phase was not completed. Mostly, this transition is preceded by the sequence of transitions 5-9-15. If there will be a new initialisation phase under the current beacon (depending on the content of the BST), this situation will be reported inside the VST, since the variable SavedState still has the value INIT.
21. This transition occurs if the OBE receives a BST before having received an implicit layer 7 acknowledge, i.e. an addressed frame other than PrWA. The OBE re-transmitts the PrWRq and stays in the INIT state.
22. A normal situation: after having issued a PrWRq (transition 12,13 or 21) the RSE will transmit a PrWA. The OBE then transmits the VST and remains in the state INIT until another addressed frame is received.
23. Broadcast UI commands are processed in every state which allows equally for processing of BSTs. The state is not changed.
24. Any command with private address is an implicit layer 7 acknowledge for the VST. The command is processed according to ist type and the state INIT is left. In the case of UI, the new state is READY.
25. A RELEASE command always causes the OBE to power down into the BLOCKED state, even if the initialisation was not completed.
26. Any command with private address is an implicit layer 7 acknowledge for the VST. The command is processed according to ist type and the state INIT is left. In the case of Acn with P=0, the new state is READY.
27. Any command with private address is an implicit layer 7 acknowledge for the VST. The command is processed according to ist type and the state INIT is left. In the case of Acn with P=1 and ACCESS()=FAST, the new state is READY.
28. Any command with private address is an implicit layer 7 acknowledge for the VST. The command is processed according to ist type and the state INIT is left. In the case of Acn with P=1 and ACCESS()=SLOW, the new state is BUSY.
29. If in state INIT a time-out of the wake-up signal occurred, the transaction is considered to be not successfully completed. However, no application specific postprocessing takes place. This exceptional situation is remembered in the variable SavedState=INIT.
30. Other events are not processed in state INIT.
31. On reception of a PrWA, the last transmitted command is repeated. For example, this may be an L2 acknowledge if the previous transition was no. 37.
32. In state READY, every BST is checked whether it contains a new beacon Id. If it is a new one, there is a transition into the state EVAL_BST and subsequently the initialisation may be performed. Note that the date and time of the BST is not checked in state READY, because the difference never will be more than 255 s. This is due to the fact that on every reception of a BST the stored date and time is updated. Note: for OBEs being always powered, this will be the normal transition.
33. On reception of a BST with the same beacon Id there is no transition into a new state. The stored beacon Id as well as the date and time are updated.
34. In state READY, a broadcast UI command is processed. The new state is again READY.

35. In state READY, an addressed UI command is processed. The new state is again READY.
36. A RELEASE command always causes the OBE to power down into the BLOCKED state. Since the OBE was in the READY state, the transaction is assumed to be closed without error.
37. Acn commands with P=0 are acknowledged with an empty L2 acknowledge (STATUS=NR_OK). The new state is READY.
38. Acn commands with P=1 and ACCESS=FAST are acknowledged with a L2 acknowledge containing already the response (STATUS=OK_OK). The new state is READY.
39. Acn commands with P=1 and ACCESS=SLOW are acknowledged with an empty L2 acknowledge (STATUS=NE_OK). The response will be delivered later. In the new state is BUSY, the response is generated.
40. This is a re-transmission of an earlier ACn command (transition 37). The acknowledge is re-transmitted.
41. This is a re-transmission of an earlier Acn command (transition 38). The acknowledge containing the response is re-transmitted.
42. A time-out in state READY causes the OBE to go to SLEEP state.
43. Other events are not processed in the READY state.
44. To allow for information services to be performed at AFC beacons, the BUSY state can be used to transmit information data using the UI command with private addresses. This service may not be available in all OBE
45. A RELEASE command always causes the OBE to power down into the BLOCKED state. Since the OBE was in the BUSY state, the transaction is assumed to be closed with an error. This state transition may not be available in all OBE
46. This is a re-transmission of an earlier ACn command (transition 39). The same answer is responded to the RSE.
47. The RSE issued a PrWA because the last message from the OBE to the RSE was lost (polling on MAC level). In BUSY state, the last message always is the empty L2 acknowledge with STATUS=NE_OK, compare transitions 39 and 46.
48. The BUSY state is left as soon as the internal OBE processing is finalised and the response was generated. The response LSDU is placed into the SAVE buffer and the new state is DATA_1 in order to await the BST.
49. Other events are not processed in the BUSY state. Note that there is no event TW_expired in state BUSY. This means that the BUSY state always is finalised by means of internal processing. The TW_expired can occur only in the next state DATA_1.
50. A RELEASE command always causes the OBE to power down into the BLOCKED state. Since the OBE was in the DATA_1 state, the transaction is assumed to be closed with an error.
51. This is the normal transition in state DATA_1. On reception of a BST the OBE transmits a PrWRq and reaches state DATA_2 in order to await the PrWA.
52. In state DATA_1, the reception of a BST with new beacon Id is considered to be an error. This error is reported inside the OBE_STATUS of the next transmitted VST (SavedState is part of OBE_STATUS). The new state is EVAL_BST which may subsequently lead to a new initialisation phase.
53. UI commands for information services are accepted as well in the DATA_1 state. Note that the RSE has no knowledge whether the OBE is in state BUSY or in state DATA_1.
54. A very special case: The RSE issues a PrWA if the last message from the OBE was lost. In state DATA_1, this last message must be an empty L2 acknowledge with STATUS=NE_OK, see transitions 39, 46 or 47. But now, the response is ready and therefore it is better to directly transmit the response in a way, as if the ACCESS was FAST! New state is READY.
55. Similar as transition 54. The difference is that the RSE did re-transmit on layer 2 instead on MAC level.

56. A time-out of the wake-up signal in state DATA_1 is a critical issue. If the OBE would go to SLEEP state, the transaction context may be lost without possibility for recovery. For example, session keys inside ICCs may be lost when powering down the ICC. Therefore, an intermediate state WAIT is taken, where the OBE is powered down only partially in order to allow for resuming transction in the same context (compare transition 1). The postprocessing closing the transaction is delayed until the WAIT state is left (transition 2). Note that this transitions are very important in double-gantry configurations with slow ICCs.
57. Other events are not processed in state DATA_1.
58. Any new command from the RSE is interpreted as in implicit layer 7 acknowledge, causing the OBE to transit into the READY state.
59. A RELEASE command always causes the OBE to power down into the BLOCKED state. Since the OBE was in the DATA_2 state, the transaction is assumed to be closed with an error.
60. On reception of a BST with same beaconId, in state DATA_2 the OBE re-transmits the PrWRq, compare transition 50.
61. In state DATA_2, the reception of a BST with new beacon Id is considered to be an error. This error will be reported in the field OBE_STATUS of the next VST, since SavedState is a part of the OBE_STATUS. The new state is EVAL_BST and subsequently there may be a new initialisation phase.
62. This is the normal transition is state DATA_2: on reception of a PrWA the OBE transmits the delayed response using a UI frame. The state DATA_2 is not left until an implicit layer 7 acknowledge was received.
63. A very special case: If the RSE re-transmits an ACn command this must be the one having caused the transition 39. Since the response is ready, it is better to answer directly as if was ACCESS=FAST.
64. Any new command from the RSE is interpreted as in implicit layer 7 acknowledge, causing the OBE to leave state DATA_2. In the case of ACn commands with P=0, the new state is READY.
65. Any new command from the RSE is interpreted as in implicit layer 7 acknowledge, causing the OBE to leave state DATA_2. In the case of ACn commands with P=1 and ACCESS=FAST, the new state is READY.
66. Any new command from the RSE is interpreted as in implicit layer 7 acknowledge, causing the OBE to leave state DATA_2. In the case of ACn commands with P=1 and ACCESS=SLOW, the new state is BUSY.
67. Same comment as for transition 56.
68. Other events are not processed in state DATA_2.

7 References, Terminology

7.1 References

[1] CEN/TC278, prEN 12253 “Road Transport and Traffic Telematics – Dedicated Short Range Communication (DSRC) - DSRC Physical Layer using Microwave at 5.8 GHz” – Jan 2003

[2] CEN/TC278, EN 12795 “Road Transport and Traffic Telematics – Dedicated Short Range Communication (DSRC) - DSRC Data Link Layer: Medium Access and Logical Link Control”

[3] CEN/TC278, EN 12834 “Road Transport and Traffic Telematics – Dedicated Short Range Communication (DSRC) - DSRC Application Layer”

[4] CEN/TC278, prEN 13372 “Road Transport and Traffic Telematics – Dedicated Short Range Communication (DSRC) - DSRC Profiles for RTTT Applications” – Jan 2003

[5] CEN/TC278, prEN ISO 14906
"Electronic Fee Collection, Application Interface Definition for Dedicated Short Range Communication" – 2002

[6] ISO IS 3309
Information processing systems - Data communication - High-level data link control procedures - Frame structure, version 5, 1993

7.2 Terminology

ACn	Acknowledged Connectionless ('n' denotes the sequence bit)
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation 1
BER	Bit Error Rate
BST	Beacon Service Table
CEN	Comité Européen de Normalisation / European Committee for Standardisation
DSRC	Dedicated Short Range Communication
EFC	Electronic Fee Collection
EID	Element ID
EIRP	Equivalent Isotropic Radiation Power
FCS	Frame Check Sequence
GSS	Global Specification for Short range communication
LID	Link ID
LLC	Logical Link Control
LSB	Least Significant Bit
LPDU	Link Protocol Data Unit
MAC	Medium Access Control
MSB	Most Significant Bit
OBE	On Board Equipment
OSI	Open Systems Interconnection
PER	Packed Encoding Rules
PDU	Protocol Data Unit
PrWA	Private Window Allocation
PrWRq	Private Window Request
RSE	Road Side Equipment
RTTT	Road Transport and Traffic Telematics
T-APDU	Transfer Application Protocol Data Unit
UI	Unnumbered Information
UTC	Universal Time Coordinated
VST	Vehicle Service Table