INTERNATIONAL STANDARD

ISO/IEC 4005-2

First edition
2023-03

Telecommunications and information exchange between systems — Unmanned aircraft area network (UAAN) —

Part 2: Physical and data link protocols for shared communication

Télécommunications et échange d'information entre systèmes — Réseau de zone de drones (Unmanned aircraft area network - UAAN) —

Partie 2: Protocoles de liaison de données et physiques pour la communication partagée

Reference number
ISO/IEC 4005-2:2023(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

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

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

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

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*.

A list of all parts in the ISO/IEC 4005 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

Unmanned aircrafts (UAs) operating at low altitude will provide a variety of commercial services in the near future. UAs that provide these services are distributed in the airspace. In level II, many people operate their own UAs without the assignment of communication channels from a central control centre.

This document describes shared communication, which is a wireless distributed communication. Shared communication allows all units related with UAs to communicate with UAs when necessary. Shared communication can support communication between UAs, UAs and controllers, UAs and ground equipment, UAs and landing devices, and UAs and obstacle devices. A wireless distributed communication described by this document is intended to be used in licensed frequency bands.

The ISO/IEC 4005 series consists of the following four parts:

— ISO/IEC 4005-1: To support various services for UAs, it describes a wireless distributed communication model and the requirements that this model shall satisfy.

— ISO/IEC 4005-2 (this document): It describes communication in which all units involved in UA operations can broadcast or exchange information by sharing communication resources with each other.

— ISO/IEC 4005-3: It describes the control communication for the controller to control the UA.

— ISO/IEC 4005-4: It describes video communication for UAs to send video to a controller.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of patents.

ISO and IEC take no position concerning the evidence, validity and scope of these patent rights.

The holders of these patent rights have assured ISO and IEC that they are willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Telecommunications and information exchange between systems — Unmanned aircraft area network (UAAN) —

Part 2: Physical and data link protocols for shared communication

1 Scope

This document describes communication protocols for the physical and data link layer of shared communication, which is a wireless distributed communication network for units related with UAs in level II.

Physical layer includes frame structure, encoding procedure, physical layer procedure and coexistence operations. Data link layer includes channel and slot, resource management, broadcast and exchange of data, synchronization, security, and interface with upper layer and other communication layers.

2 Normative references

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

ISO/IEC 4005-1, *Telecommunications and information exchange between systems — Unmanned aircraft area network (UAAN) — Part 1: Communication model and requirements*

ISO/IEC 4005-3:2023, *Telecommunications and information exchange between systems — Unmanned aircraft area network (UAAN) — Part 3: Physical and data link protocols for control communication*

ISO/IEC 4005-4:2023, *Telecommunications and information exchange between systems — Unmanned aircraft area network (UAAN) — Part 4: Physical and data link protocols for video communication*

ISO 21384-4, *Unmanned aircraft systems — Part 4: Vocabulary*

3 Terms and definitions

For the purposes of this document, the terms and definitions defined in ISO/IEC 4005-1, ISO 21384-4 and the following apply.

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

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

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

3.1
broadcast slot
slot used to broadcast the information of a unit

3.2
data slot block
block in which multiple slots are grouped together for efficient resource allocation

3.3
linearization slot
slot used to maintain the linearity of the power amplifier

3.4
response clearing
transmission of a tone signal in subslot 0 of the subslot set corresponding to the same slot of this frame, in order to respond to a talk packet received in a slot of the previous frame

3.5
slot block
block in which four slots are grouped together to perform efficient competition

3.6
super frame slot
slot with a frame period greater than one second

3.7
talk slot
slot used to exchange information with a specific counterpart

3.8
tone slot block
block in which multiple slots are grouped together to perform efficient competition

4 Abbreviated terms

CC	Control Communication
CRC	Cyclic Redundancy Check
CSCH	Control Subchannel
DL	Data Link
DLL	Data Link Layer
DQPSK	Differential Quadrature Phase Shift Keying
GF	Galois Field
LFSR	Linear Feedback Shift Register
PCCC	Parallel Concatenated Convolutional Code
PB	Parsing Block
PF	Parsing Field
PN	Pseudo Noise
SA	Source Address
SC	Shared Communication
SRRC	Square Root Raised Cosine
TX	Transmission
UTC	Coordinated Universal Time

VC Video Communication

VSCH Video Subchannel

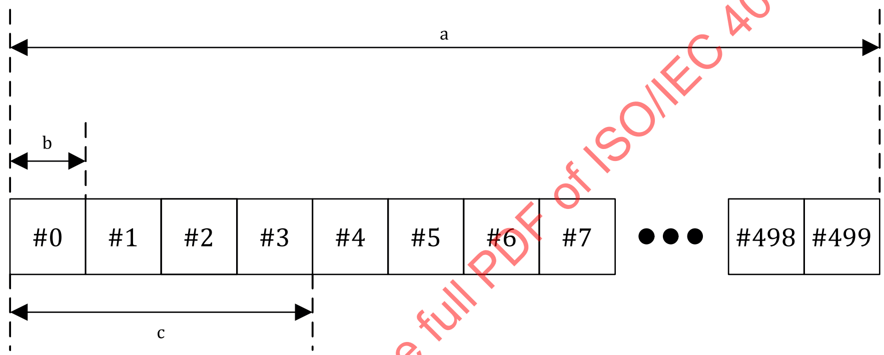
5 Physical layer

5.1 Physical layer frame structure

5.1.1 Data channel

5.1.1.1 Frame structure

The frame length of the data channel is 1 second and consists of 500 slots. The one slot time T_s is 2 ms. A data slot block has four slots. Therefore, there are 125 slot blocks in one frame, and the slot block is 8 ms in length as shown in Figure 1. The frame number changes from 0 to 59 in 1 min interval.



a 1 frame, T_f=1 second=500 T_s.

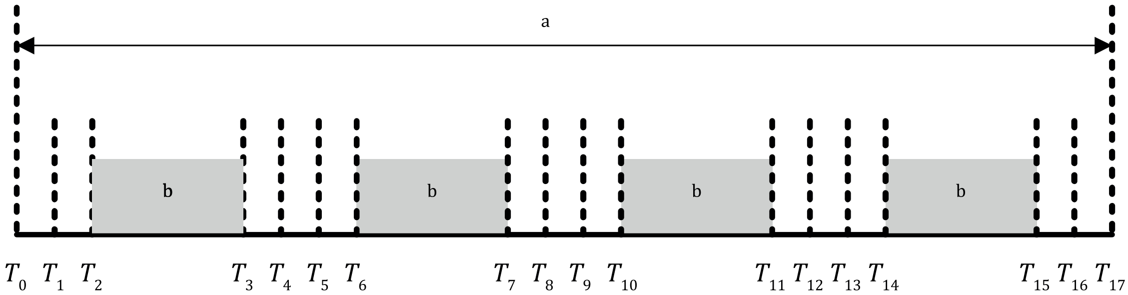
b 1 slot, T_s=2 ms.

c 1 slot block, T_{sb}=8 ms=4 T_s.

Figure 1 — Data channel frame structure

5.1.1.2 Slot block transmit time mask

The transmission time mask of a slot block is shown Figure 2.



Key

T_0	0 µs
T_1, T_2, ..., T_{17}	symbol time offsets from T_0
a	8 ms.
b	Modulated signal.

Figure 2 — Transmission time mask of a slot block

T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}, T_{11}, T_{12}, T_{13}, T_{14}, T_{15}, T_{16}, T_{17} are symbol offsets from T_0, and symbol time is 1/672000 second. Each value is as follows.

T_1 is 152, T_2 is 154, T_3 is 1449, T_4 is 1451, T_5 is 1456.5, T_6 is 1459.5, T_7 is 2754.5, T_8 is 2756.5, T_9 is 2763, T_{10} is 2765, T_{11} is 4060, T_{12} is 4062, T_{13} is 4068.0, T_{14} is 4070.5, T_{15} is 5365.5, T_{16} is 5366.5, T_{17} is 5376.

T_0 is 0 µs as the start time of the slot block. T_1, T_5, T_9 and T_{13} are offsets when the power amplifier is gated on, and unmodulated fine signals begin to be transmitted. T_2, T_6, T_{10}, and T_{14} are offsets at which modulation signal transmission starts. T_3, T_7, T_{11}, and T_{15} are offsets at which the transmission of the modulated signal ends. T_4, T_8, T_{12}, and T_{16} are offsets at which the power amplifier is gated off, and transmission of unmodulated fine signals is stopped. The transmit powers of T_1 to T_2, T_3 to T_4, T_5 to T_6, T_7 to T_8, T_9 to T_{10}, T_{11} to T_{12}, T_{13} toT_{14} and T_{15} to T_{16} shall be at least 50 dB less than the modulation signal transmit power.

5.1.2 Tone channel

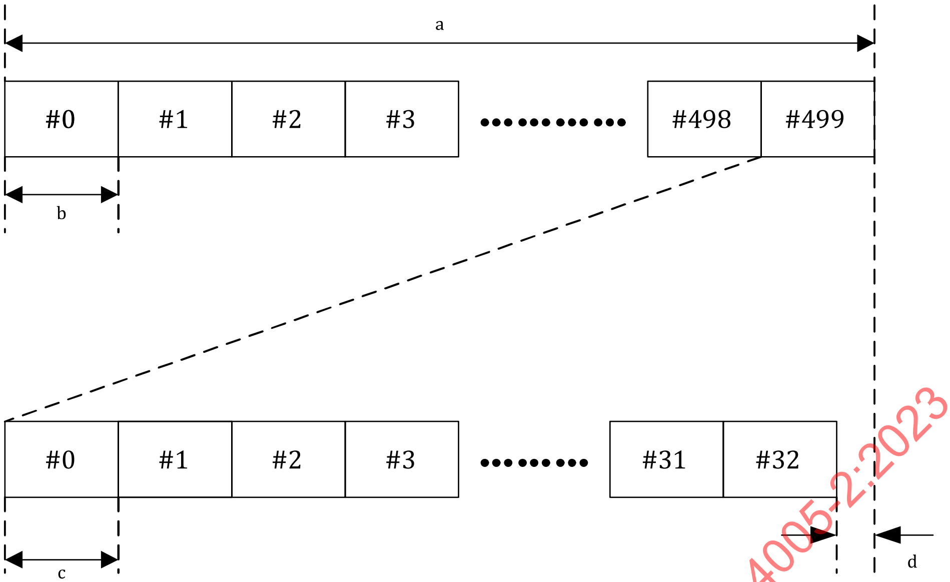
5.1.2.1 Frame structure

5.1.2.1.1 General

There are two types of tone channels. One is a competition tone channel and the other is an information tone channel. The competition tone channel is used for resource allocation, occupation, and management operations. Information tone channels are used to convey simple information. In this document, the frame structure for the information tone channel is described, but whether and how to use the information tone channel is determined by the upper layer.

5.1.2.1.2 Information tone channel frame structure

The frame length of information tone channel is 1 sec and the number of slots per frame is 500. One slot is divided into 33 subslots, and the length T_{ss} of a subslot is 60 µs as shown in Figure 3. The information given to each subslot is determined by the upper layer. SS_x means the x-th subslot (see Annex G).



a 1 frame, T_f = 1 second = 500 T_s.

b 1 slot, T_s = 2 ms.

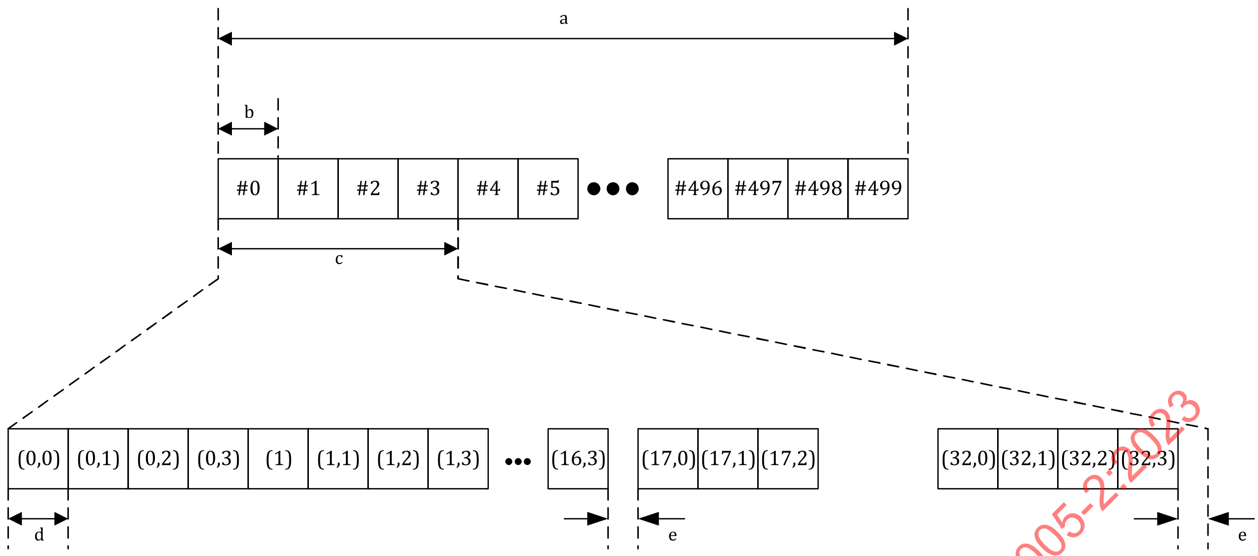
c T_{ss} = 60 µs.

d 20 µs.

Figure 3 — Frame structure of information tone channel

5.1.2.1.3 Competition tone channel frame structure

The frame length of competition tone channel is 1 sec and the number of slots per frame is 500. Four tone slots constitute one slock block. Thus, there are 125 slot blocks in one second frame. There are a total of 132 subslots in one slot block. The length T_{ss} of the subslot is 60 µs. The 132 subslots are divided into four parts, as shown in Figure 4, according to each slot numbers.



a 1 frame, T_f = 1 second = 500 T_s.

b 1 slot, T_s = 2 ms.

c 1 slot block, T_{sb} = 8 ms.

d T_{ss} = 60 µs.

e 40 µs.

Figure 4 — Frame structure of competition tone channel

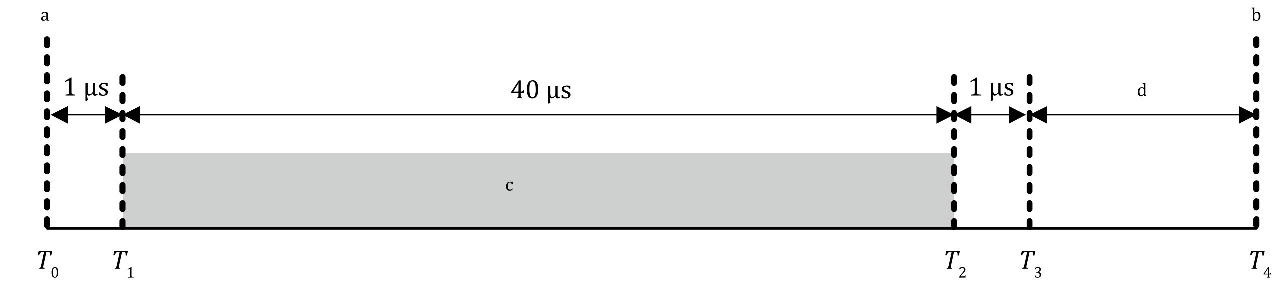
The n-th slot block is composed of the following subslots.

— (0, 4n), (0, (4n + 1)), (0, (4n + 2)), (0, (4n + 3)), (1, 4n), (1, (4n + 1)), (1, (4n + 2)), (1, (4n + 3)),... , (32, 4n), (32, (4n + 1)), (32, (4n + 2)), (32, (4n + 3))

where (x, y) is the x-th subslot of the y-th subslot set. The 132 subslots constitute 4 subslot sets.

5.1.2.2 Subslot transmit time mask

Subslot transmission time mask is shown in Figure 5.



Key

T_1, T_2, T_3, and T_4 are time offsets from T_0

T_1 is 1 µs

T_2 is 41 µs

T_3 is 42 µs

T_4 is 60 µs

T_0 is the start time of the subslot, the power amplifier is gated on and the unmodulated fine signal starts to be transmitted

T_1 is the time at which transmission of the modulated signal begins

T_2 is the time at which transmission of the modulated signal is terminated

T_3 is the time when the power amplifier is gated off and the transmission of unmodulated fine signals is stopped

a Subslot start.

b Subslot end.

c Tone signal.

d Guard time.

Figure 5 — Subslot transmission time mask

The transmission power between T_0 and T_1 and between T_2 and T_3 shall be 50 dB less or less than the maximum transmission power of the modulated signal.

5.1.2.3 Subslot signal waveform

The modulation scheme of subslot signal is on-off keying. The subslot transmission signal is transmitted in the 40 µs interval from T_1 to T_2 in Figure 5. The waveform of the subslot transmission signal uses a raised cosine function. The subslot signal is generated by the following formula.

$$g(t;\alpha)=\frac{\cos(\pi\alpha(t-2T))}{1-(2\alpha(t-2T)/T)^2}\operatorname{sinc}\left(\frac{(t-2T)}{T}\right),\qquad 0\le t\le 4T \tag{1}$$

where

α is 0,75 as a roll-off factor;

T is 10 µs as a raised cosine period.

See Annex D.

5.2 Encoding procedure

The encoding follows the following procedure as shown in Figure 6. CRC encoding, turbo coding, rate matching, interleaving, modulation mapping, burst mapping, and pulse mapping are performed in this order.

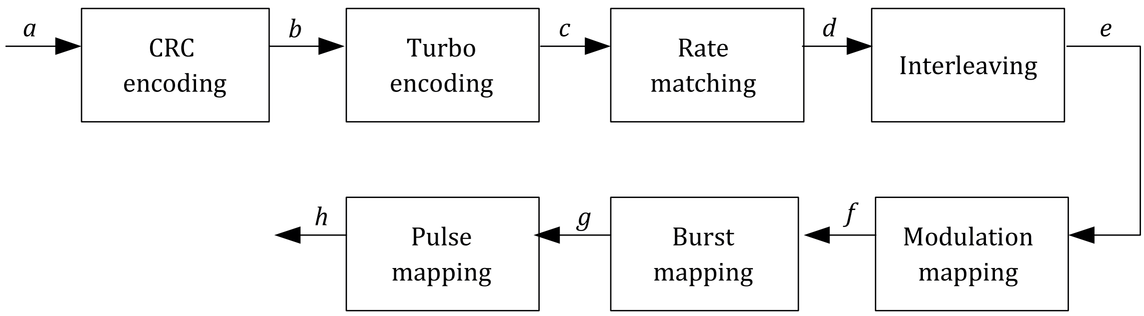


Figure 6 — Encoding procedure

The number of symbols according to each encoding stage is shown in Table 1.

Table 1 — Number of symbols at each encoding stage

Stage	Number of symbols
a	792 (binary)
b	816 (binary)
c	2460 (binary)
d	2432 (binary)
e	2432 (binary)
f	1216 (Complex)
g	1288 (Complex)
h	1295×*OS* (Complex)

5.2.1 CRC encoding

The input bits are defined as $a_0, a_1, a_2, a_3, \ldots, a_{A\text{-}1}$ and parity bits as $p_0, p_1, p_2, p_3, \ldots, p_{23}$ where, A represents the number of input sequences. Parity bits are generated through CRC generation polynomial as follows.

$$g_{\text{CRC}}(D) = D^{24} + D^{22} + D^6 + D^5 + D + 1 \tag{2}$$

The encoding performed through the cyclic generator polynomials has a systematic form as follows. The resulting polynomial has zero remainder when it is divided by $g_{\text{CRC}}(D)$ on GF(2).

$$a_0 D^{A+23} + a_1 D^{A+22} + \ldots + a_{A\text{-}1} D^{24} + p_0 D^{23} + p_1 D^{22} + \ldots + p_{22} D^1 + p_{23} \tag{3}$$

After CRC insertion, bits are represented by $b_0, b_1, b_2, b_3, \ldots, b_{B\text{-}1}$ (where $B = A + 24$), and the relationship between a_k and b_k is as follows.

$$b_k = \begin{array}{ll} a_k, & \mathit{for}\, k = 0,1,2,\cdots,A-1 \\ P_{k-A}, & \mathit{for}\, k = A, A+1, A+2, \cdots, A+23 \end{array} \tag{4}$$

5.2.2 Turbo encoding

The turbo encoder consists of Parallel Concatenated Convolutional Code (PCCC) with two 8-state constituent encoders and one turbo coded internal interleaver. The coding rate of the turbo encoder is 1/3. The structure of the turbo encoder is shown in Figure 7. The PCCC transfer function is as follows:

$$G(D) = [1, g_1(D)/g_0(D)] \tag{5}$$

where $g_0(D) = 1{+}D^2{+}D^3$, $g_1(D) = 1{+}D{+}D^3$.

When the input bits of the turbo encoder are encoded, the initial values of the shift registers of the 8-state constituent encoder shall all be zero.

For $k = 0, 1, 2, \ldots, K\text{-}1$, the output value of the turbo encoder is expressed as follows:

$$
\begin{aligned}
c_{3k} &= x_k \\
c_{3k+1} &= z_k \\
c_{3k+2} &= z'_k
\end{aligned}
\tag{6}
$$

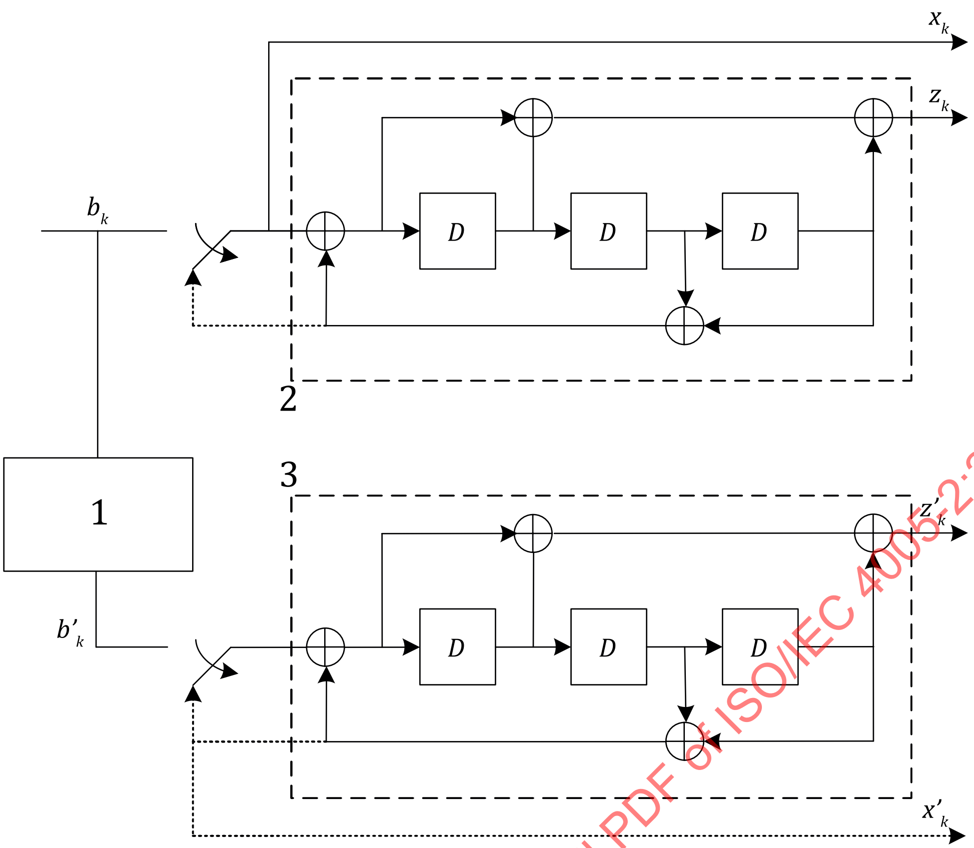
Output bits of the first and second 8-state constituent encoders for turbo encoder input bits b_0, b_1, b_2, b_3, ..., $b_{B\text{-}1}$ are z_0, z_1, z_2, z_3, ..., $z_{B\text{-}1}$ and z'_0, z'_1, z'_2, z'_3, ..., $z'_{B\text{-}1}$, and the output bits through the turbo code internal interleaver that is described in Annex A are represented by b'_0, b'_1, b'_2, b'_3, ..., $b'_{B\text{-}1}$ and these output bits are used as inputs for the second 8-state constituent encoder. The turbo code internal interleaver shall use Table A.1 in Annex A.

Trellis termination is performed by taking tail bits from shift register feedback after all information bits have been encoded. The generated tail bits are added after encoding of the information bits.

The first three tail bits are used for the first constituent encoder termination and not the second constituent encoder. The remaining three tail bits are used for the termination of the second constituent encoder and not the first constituent encoder.

The bits transmitted for trellis termination are determined as follows:

$$
\begin{aligned}
&c_{3B} = x_B,\ c_{3B+3} = z_{B+1},\ c_{3B+6} = x'_B,\ c_{3B+9} = z'_{B+1} \\
&c_{3B+1} = z_B,\ c_{3B+4} = x_{B+2},\ c_{3B+7} = z'_B,\ c_{3B+10} = x'_{B+2} \\
&c_{3B+2} = x_{B+1},\ c_{3B+5} = z_{B+2},\ c_{3B+8} = x'_{B+1},\ c_{3B+11} = z'_{B+2}
\end{aligned}
\tag{7}
$$



Key

1	turbo code internal interleaver
2	first constituent encoder
3	second constituent encoder
D	register
b_k	a k-th bit of turbo encoder input
b'_k	a k-th bit of turbo code internal interleaver output
x_k	a k-th systematic bit of turbo encoder output
z_k	a k-th bit of first constituent encoder output
x'_k	a k-th bit of second constituent encoder output for trellis termination
z'_k	a k-th bit of second constituent encoder output

Figure 7 — Turbo encoder structure

Input bit sequence of turbo code internal interleaver, b_0, b_1, b_2, b_3, ..., $b_{B\text{-}1}$ and output bit sequence generated from turbo code internal interleaver, b'_0, b'_1, b'_2, b'_3, ..., $b'_{B\text{-}1}$ have the following relationship.

$$b'_i = b_j \qquad (8)$$

where the mapping between the output bit index i and the input bit index j shall follow Table A.1 in Annex A. Where j and i are as follows, and row and column numbers start at zero.

$$j = (\text{number shown in table}) - 1$$

$$i = (\text{row number in table})\times 16+(\text{column number in table}) \qquad (9)$$

5.2.3 Rate matching

Rate matching outputs d_0, d_1, d_2, d_3, ..., $d_{D\text{-}1}$ by puncturing the input bits c_0, c_1, c_2, c_3, ..., $c_{C\text{-}1}$. The puncturing bit numbers are as follows.

— 43, 131, 217, 305, 391, 479, 565, 653, 739, 827, 913, 1001, 1087, 1175, 1261, 1349, 1435, 1523, 1609, 1697, 1783, 1871, 1957, 2045, 2131, 2219, 2305, 2393

5.2.4 Interleaving

The interleaver uses block interleaving with 38 rows and 64 columns.

$$e_m = d_n$$

$$m = (n\times 64)\%2432 + \lfloor n/38 \rfloor \quad (10)$$

where $\lfloor x \rfloor$ means the largest integer among integers less than or equal to x and $0 \le n \le 2431$.

5.2.5 Modulation mapping

Modulation mapping generates a complex symbol f_n from the input bit e_m, $0 \le n \le 2431$, $0 \le m \le 1215$. Two input bits are mapped to one complex number as shown in Table 2.

Table 2 — modulation mapping

$e_{2n}e_{2n+1}$	00	01	10	11
f_n	$exp(j/4\pi)$	$exp(j\cdot 7/4\pi)$	$exp(j\cdot 3/4\pi)$	$exp(j\cdot 5/4\pi)$

5.2.6 Burst mapping

Output complex symbols $g_0, g_1, ..., g_{1287}$ are generated from the input complex symbols $f_0, f_1, ..., f_{1215}$.

$$g_n = \prod_{k=0}^{n} c(k) \quad (11)$$

where, $c(n)$ is shown in Table 3.

Table 3 — $c(n)$

n	$c(n)$	Number of symbols
0, 1	$TSS(n)$	2
2, ..., 37	$PTS1(n\text{-}2)$	36
38, ..., 443	$f_{n\text{-}38}$	406
444, ..., 459	$PTS2(n\text{-}444)$	16
460, ..., 865	$f_{n\text{-}54}$	406
866, ...,881	$PTS2(n\text{-}866)$	16
882, ...,1285	$f_{n\text{-}70}$	404
1286, 1287	$TSS(n\text{-}1286)$	2

where $TSS(n)$, $PTS1(n)$, and $PTS2(n)$ are shown in Table 4, Table 5 and Table 6 respectively.

Table 4 — *TSS*(*n*)

TSS(0)	*TSS*(1)
$exp(j{\cdot}3/4\pi)$	$exp(j{\cdot}7/4\pi)$

Table 5 — *PTS1*(*n*)

n	*PTS*1(*n*)	*n*	*PTS*1(*n*)	*n*	*PTS*1(*n*)
0	$exp(j{\cdot}5/4\pi)$	12	$exp(j{\cdot}5/4\pi)$	24	$exp(j{\cdot}7/4\pi)$
1	$exp(j{\cdot}7/4\pi)$	13	$exp(j/4\pi)$	25	$exp(j{\cdot}5/4\pi)$
2	$exp(j{\cdot}7/4\pi)$	14	$exp(j/4\pi)$	26	$exp(j{\cdot}7/4\pi)$
3	$exp(j{\cdot}5/4\pi)$	15	$exp(j{\cdot}5/4\pi)$	27	$exp(j/4\pi)$
4	$exp(j/4\pi)$	16	$exp(j{\cdot}7/4\pi)$	28	$exp(j{\cdot}5/4\pi)$
5	$exp(j/4\pi)$	17	$exp(j/4\pi)$	29	$exp(j{\cdot}3/4\pi)$
6	$exp(j{\cdot}3/4\pi)$	18	$exp(j{\cdot}5/4\pi)$	30	$exp(j{\cdot}3/4\pi)$
7	$exp(j{\cdot}5/4\pi)$	19	$exp(j{\cdot}3/4\pi)$	31	$exp(j/4\pi)$
8	$exp(j{\cdot}3/4\pi)$	20	$exp(j{\cdot}7/4\pi)$	32	$exp(j/4\pi)$
9	$exp(j/4\pi)$	21	$exp(j/4\pi)$	33	$exp(j{\cdot}5/4\pi)$
10	$exp(j{\cdot}5/4\pi)$	22	$exp(j/4\pi)$	34	$exp(j{\cdot}3/4\pi)$
11	$exp(j{\cdot}5/4\pi)$	23	$exp(j{\cdot}3/4\pi)$	35	$exp(j{\cdot}7/4\pi)$

Table 6 — *PTS*2(*n*)

n	*PTS*2(*n*)	*n*	*PTS*2(*n*)	*n*	*PTS*2(*n*)
0	$exp(j/4\pi)$	6	$exp(j{\cdot}5/4\pi)$	12	$exp(j{\cdot}3/4\pi)$
1	$exp(j{\cdot}3/4\pi)$	7	$exp(j{\cdot}3/4\pi)$	13	$exp(j{\cdot}3/4\pi)$
2	$exp(j/4\pi)$	8	$exp(j{\cdot}5/4\pi)$	14	$exp(j/4\pi)$
3	$exp(j{\cdot}7/4\pi)$	9	$exp(j{\cdot}7/4\pi)$	15	$exp(j{\cdot}7/4\pi)$
4	$exp(j{\cdot}7/4\pi)$	10	$exp(j{\cdot}5/4\pi)$		
5	$exp(j{\cdot}3/4\pi)$	11	$exp(j{\cdot}7/4\pi)$		

5.2.7 Pulse mapping

The complex symbol g_m is converted into a complex signal h_n where the oversampling ratio of the filter is OS times and depends on implementation. For $0 \le n < 1295 \times OS$, the complex signal is defined as follows.

$$h_n = w\left(\frac{nT_s}{OS}\right)\sum_{m=0}^{1287} p\left(\left(\frac{n}{OS} - m - 4\right)T_s\right)g_m \tag{12}$$

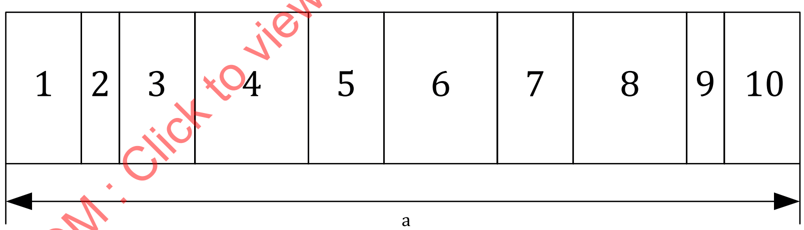
where symbol duration T_s is the 1/672000 second and pulse shape $p(t)$ is defined as SRRC function of roll-off factor 0,35 as follows.

$$p(t) = \frac{1}{1+\frac{(1-\alpha)\pi}{4\alpha}} \cdot \frac{\cos\left(\frac{(1+\alpha)\pi t}{T_s}\right) + \frac{\sin((1-\alpha)\pi t / T_s)}{4\alpha t / T_s}}{1-(4\alpha t / T_s)^2} \tag{13}$$

Here window function $w(t)$ to process tapering at signal start and end is introduced for good frequency characteristics reducing spurious emission. The $w(t)$ is defined as follows.

$$w(t) = \begin{cases} (1/2)(1-\cos(\pi t/2T_s)), & 0 \le t < 2T_s \\ 1, & 2T_s \le t < 1293T_s \\ (1/2)\left(1-\cos\left(\frac{\pi}{2T_s}(t-1295T_s)\right)\right), & 1293T_s \le t < 1295T_s \\ 0, & otherwise \end{cases} \tag{14}$$

The modulated signal is shown in Figure 8. D represents the modulation mapping output. Timing of modulated signal transmission is as described in 5.1.1.2 i.e. the modulated signals are transmitted in the time intervals of T_2 to T_3, T_6 to T_7, T_{10} to T_{11}, and T_{14} to T_{15} in Figure 2.



Key

1 filter delay, 4 symbols
2 TLS , 2 symbols
3 PTS_1, 36 symbols
4 data symbol, 406 symbols
5 PTS_2, 16 symbols
6 data symbol, 406 symbols
7 PTS_2, 16 symbols
8 data symbol, 404 symbols
9 TLS , 2 symbols
10 filter delay, 3 symbols
a Modulated signal, 1295 symbol.

Figure 8 — Modulated signal structure

5.3 Physical layer procedure

5.3.1 Slot synchronization

The synchronization mode of the unit includes 'A sync', 'B sync' and 'C sync'.

— A sync is synchronization obtained from UTC.

— B sync is secondary synchronization acquired from the synchronization signal of the A sync unit.

— C sync is sync status within 20 seconds after sudden loss of sync in A or B sync mode.

A sync unit shall know the date, hour, minute, second, slot number.

The time error of A sync shall be within ±0,4 μs. The time error of B sync shall be within ±4 μs. The time error of C sync shall be within ±5 μs.

The frequency error of A sync shall be within ±0,1ppm. The frequency error of the B sync shall be within ±0,2ppm. The frequency error of the C sync shall be within ±0,3ppm.

5.3.2 Transmit power control

The upper layer can designate the transmit powers for specific slot blocks. All units have the same transmission power for the same slot block. Normal slot transmit power is PTXnormalSCH. The transmission power of the tone subslot is a value obtained by adding PTX_SCHTCH_differ to the transmission power of the related data slot block.

5.3.3 Measurements

5.3.3.1 General

The physical layer shall have the ability to measure the following parameters. It shall be possible to measure the tone subslot received signal power, the data slot received signal power, and the propagation delay time of the received data signal. The received power decision point shall be the receiving antenna connector.

5.3.3.2 Slot map

The slot map consists of 500 bit strings. The *n*-th bit indicates whether *n*-th data slot is usable or not. '1' means the slot is available, and '0' means the slot is not available. The physical layer shall dynamically update the slot map for every slot block interval.

The availability of the corresponding data slot is determined by receiving a signal of the slot clearing subslot in the tone subslot set mapped with the corresponding data slot. If the power of the slot clearing subslot signal is greater than PRXtoneCompeteThre, it is determined that the data slot mapped thereto is occupied.

5.4 Coexistence operation

If the hardware of shared communication described in this document and the hardware of control communication described in ISO/IEC 4005-3, and the hardware of video communication described in ISO/IEC 4005-4 are completely physically isolated and do not affect each other at all, it shall be allowed that they do not perform coexistence operations, which is implementation dependent. In general, the three communications affect each other, and in this case, the following coexistence operations shall be performed.

The TX operation of a shared slot includes the TX operation of the corresponding shared slot and the mapped tone subslot set. The TX operation of a control communication includes TX of the mapped tone subslot set and CSCH TX. The TX operation of video communication includes TX of the mapped tone subslot set and VSCH TX.

When a UA periodically broadcasts its information to a shared slot of a shared channel, a shared slot and a tone subslot set mapped to the shared slot generally require 1 slot and 4 slots, respectively, for TX operation. If the TX operation of the shared slot used for mandatory periodic broadcasting and the TX operation of the control channel overlap, the TX operation of the shared slot shall be performed.

The talk slot TX operation of a shared channel shall not overlap with the periodic broadcast slot TX operation of the conversation partner, and also the conversation partner's CSCH TX and the VSCH TX.

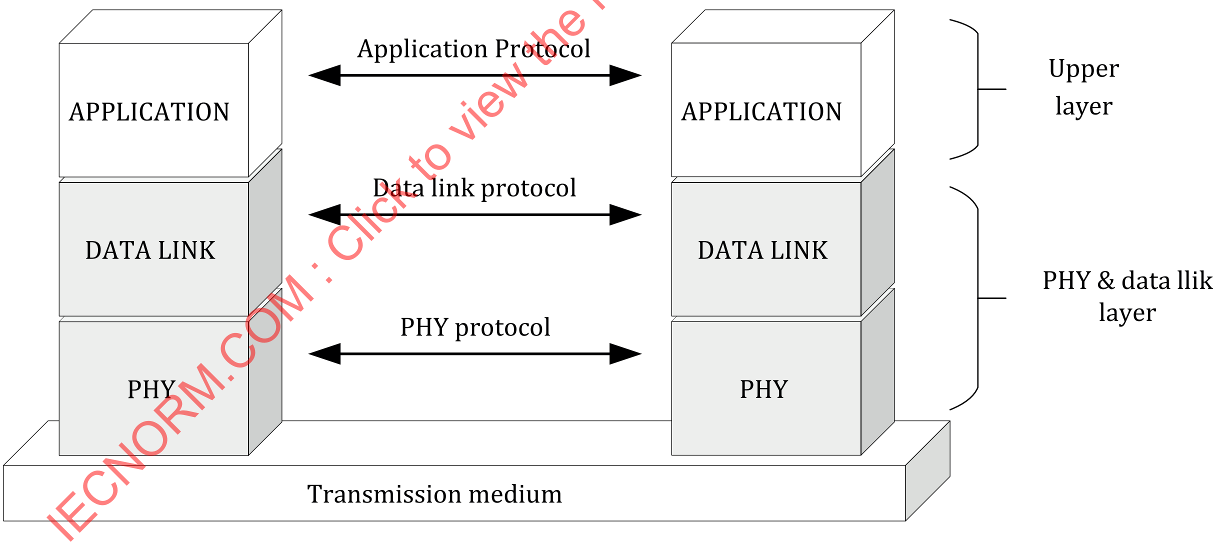
6 Data link layer

6.1 General

The data link layer describes the behaviour of units for broadcast links and unicast links. This document describes the allocation, occupation and return of data slots which are wireless communication resources.

Since this document uses a synchronous distributed communication, units allocate slot resources by themselves, occupy the allocated slot resources by themselves, and return slot resources by themselves. When occupying the allocated slot resource, collision check is continuously performed for the slot resource.

This data link describes only the basic functions of generating, maintaining, and terminating links. If necessary, additional functions for the link are implemented in the upper layer. The upper layer is mainly an application layer that provides services, as shown in Figure 9.



Key
PHY physical layer

Figure 9 — Protocol stack structure

Slots and channels are used in this document. There are data channel and tone channels, and one data channel and one tone channel work by mapping to each other.

The upper layer can flexibly design the frames and slots of the data channel, i.e. the length of the frame can be longer than 1 sec for certain slots or slots for a specific unit can be designated or transmission power of specific slots can be designated.

6.2 Channel and slot

6.2.1 General

The data channel means a channel through which information is transmitted. The tone channel refers to a channel in which competition for using a data channel is performed or information tones are transmitted.

The data channel consists of data slots. There are two types of data slot: broadcast slot and talk slot. The broadcast slot is a slot for broadcasting information. The talk slot is a slot for exchanging information with a specific counterpart.

There are two types of tone channels: information tone channels and competition tone channels. The information tone channel is composed of information tone slots, and the competition tone channel is composed of competition tone slots alone or a combination of competition tone slots and information tone slots.

The competition tone slot is a slot in which competition for allocating data slots is performed. An information tone slot is a slot for the purpose of transmitting information.

6.2.2 Information tone slot

An information tone slot is a slot to which meaning is given to transmit information. One meaning can be given to one tone slot, or different subslots in the tone slots can have different meanings. Meanings to be allocated are defined at the upper layer. When meaning is given to a subslot, one meaning can be given to one subslot, or many subslots can be grouped into one subslot group to give a meaning to the subslot group. Meaning related with the subslot group is defined in the upper layer.

In Figure 24, the tone slot block mapped to the linearization slot is configured as information tone slots. As such, the competition tone slot mapped to the dedicated data slot can be utilized as an information tone slot.

6.3 Allocation and occupation and return of data slot

6.3.1 General

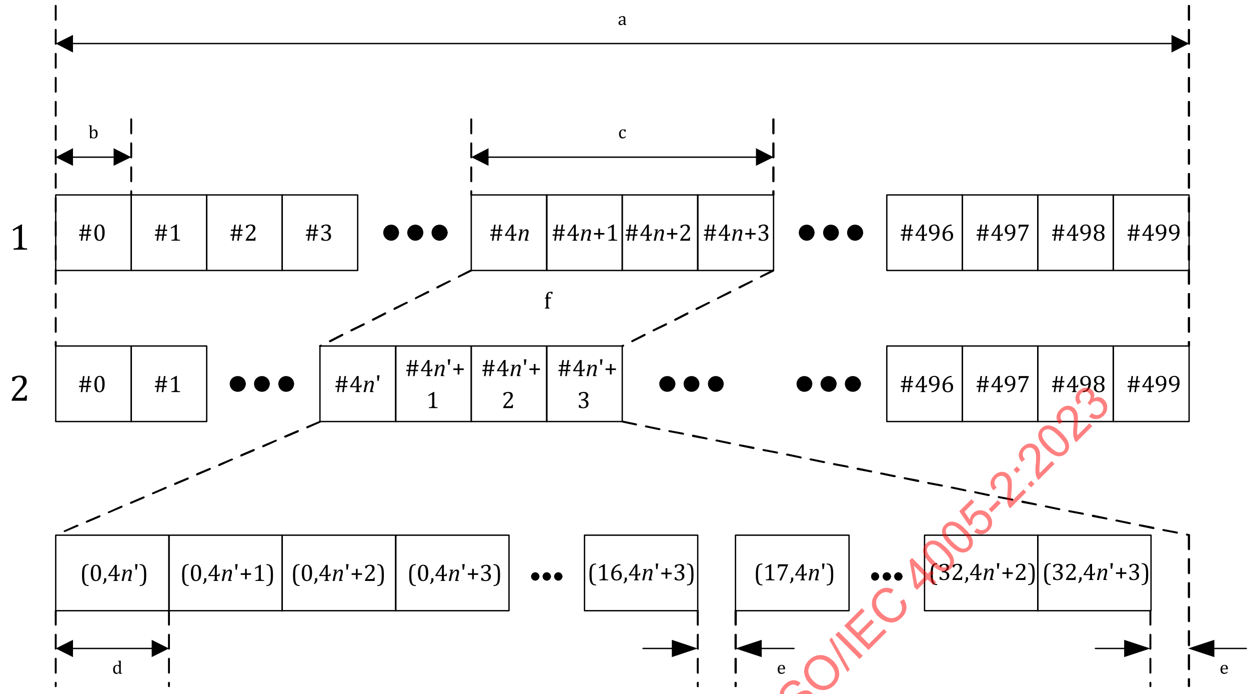
A unit can allocate, occupy, or return a broadcast slot. On the other hand, a unit can allocate a talk slot, but cannot occupy and return.

A unit can allocate only one data slot in one slot block. In addition, a unit cannot allocate a data slot in a contiguous slot block, i.e. a unit that allocates a data slot in an n-th slot block cannot allocate a data slot in the (n-1)-th or (n+1)-th slot block.

6.3.2 Mapping of data slots and competition tone slots

The competition for allocating a data slot is performed in the competition tone slot blocks mapped thereto. Starting in this subclause, for convenience, 'competition' is omitted from “competition tone slot” and is referred to as the “tone slot”.

Tone slot block n' is mapped to data slot block n, where n' is (n+124) *mod* 125, where *mod* stands for modulo operation. One exception is that the data slot block 0 is mapped to the tone slot block 124 of the previous frame.



Key

1 data channel

2 tone channel

a 1 frame, T_f = 1 second = 500 T_s.

b 1 slot, T_s = 2 ms.

c slot block n, T_{sb} = 8 ms.

d T_{ss} = 60 µs.

e 40 µs.

f Mapping relation.

Figure 10 — Mapping of data slots and tone subslot sets

The data slot numbers for constituting the n-th data slot block is $4n$, $4n$+1, $4n$+2, and $4n$+3. The tone slot block configuration is shown in Figure 10.

A tone subslot sets mapped to data slots $4n$, $4n$+1, $4n$+2, and $4n$+3 are as follows.

$$\{S_{4n}\} = (SS_0, 4n'), (SS_1, 4n'), ..., (SS_{32}, 4n')$$

$$\{S_{4n+1}\} = (SS_0, (4n' + 1)), (SS_1, (4n' + 1)), ..., (SS_{32}, (4n' + 1))$$

$$\{S_{4n+2}\} = (SS_0, (4n' + 2)), (SS_1, (4n' + 2)), ..., (SS_{32}, (4n' + 2))$$

$$\{S_{4n+3}\} = (SS_0, (4n' + 3)), (SS_1, (4n' + 3)), ..., (SS_{32}, (4n' + 3)) \quad (15)$$

where

$\{S_n\}$ is a tone subslot set mapped to the data slot S_n;

SS_x is the x-th tone subslot constituting $\{S_n\}$.

In this document, $\{F_x, S_n\}$ is defined as a tone subslot set mapped to $<F_x, S_n>$, and $<F_x, S_n>$ means data slot *y* in frame *x*, where F_x means frame *x* and S_n means slot *n*. In addition, $\{F_x, S_n, SS_z\}$ means the *z*-th subslot of $\{F_x, S_n\}$.

$\{S_n\}$ can be shown as Figure 11.

$\{S_n\}$	0	1	2	3	● ● ●	31	32

Key

$\{S_n\}$ tone subslot set mapped to data slot S_n

Figure 11 — Pictorial representations of $\{S_n\}$

6.3.3 Allocation and occupation and return of broadcast slot

6.3.3.1 General

Broadcast slots are generally assigned through competition, and collision checking is performed on the occupied slots. The upper layer can designate a broadcast slot that can be allocated without competition.

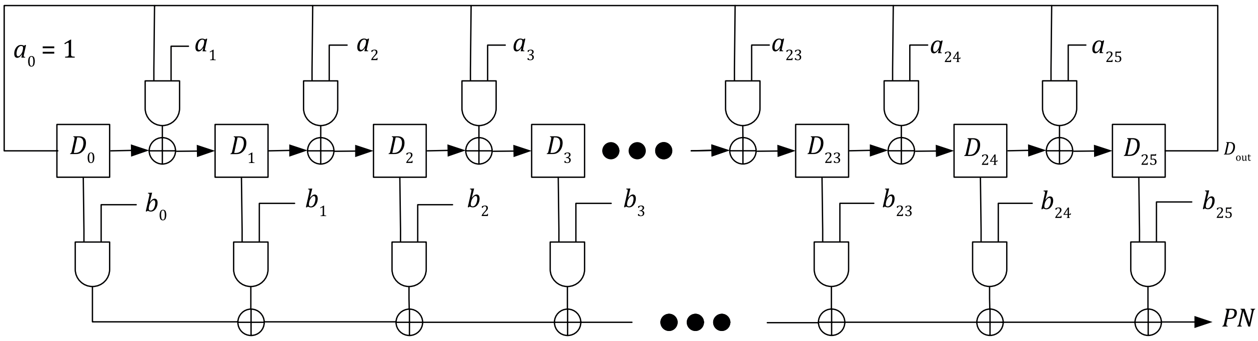
6.3.3.2 Broadcast slot allocation

6.3.3.2.1 Selection of a broadcast slot to attempt allocation

After receiving the slot allocation command UPtoDL.ReqGetSCH from the upper layer, the data link shall check the broadcast slots that can be allocated within the time period *TselectRegion*. At this time, if the reception power of the slot clearing subslot is greater than or equal to PRXtoneCompeteThre, it is determined as an already occupied slot. PRXtoneCompeteThre is received as UPtoDL.InfoPowerParamSCH from the upper layer.

If the number of allocable slots is $NUM_{\text{selectable\_slot}}$, the unit selects nMax allocable slots using a PN code generator. The unit selects nMax of *Nordinal*(*n*)-th slots, where *n* is 0, 1, 2, ..., nMax-1. nMax is received from the upper layer through the UPtoDL.InfoSharedChannel interface. As for the order from the 0-th slot to the ($NUM_{\text{selectable\_slot}}$-1)-th slot, the order of the small slot numbers is determined faster. When calculating *Nordinal*(*n*), the previously selected slots corresponding *Nordinal*(0),... , *Nordinal*(*n*-1) shall be excluded from the list of allocable slots.

The *Nordinal*(*n*) is calculated with PN code generator shown in Figure 12 as follows, where $[x_{i-1}, x_{i-2}, ..., x_0]$ represents *x* as a binary number with *i* bits. x_{i-1} is the MSB.



Key

a_k a k-th bit for AND operation with D_{out}

b_k a k-th bit for AND operation with D_k

D_k a k-th register

PN PN code generator output

Figure 12 — PN code generator

$$[D_{25}(0), D_{24}(0), ..., D_0(0)] = [SA_{25}, SA_{24}, ..., SA_0]$$

$$[b_{25}, b_{24}, ..., b_{17}] = [SA_8, SA_7, ..., SA_0]$$

$$[b_{16}, b_{15}, ..., b_{11}] = [f_5, f_4, ..., f_0]$$

$$[b_{10}, b_9, ..., b_0] = [M_5, M_4, M_3, M_2, M_1, M_0, H_4, H_3, H_2, H_1, H_0]$$

$$[a_{25}, a_{24}, ..., a_0] = [SA_0, SA_1, ..., SA_{24}, 1]$$

$$PN(clk) = (D_0(clk) \;\&\; b_0) \wedge (D_1(clk) \;\&\; b_1) \wedge ... \wedge (D_{25}(clk) \;\&\; b_{25})$$

$$N_{PN}(n) = [PN(16n+15), PN(16n+14), ..., PN(16n+1), PN(16n)]$$

$$Nordinal(n) = floor((N_{PN}(n) \times NUM_{selectable\_slot})/2^{16}) \quad (16)$$

where

$D_x(clk)$	is a x-th bit of shift register D at clock clk;
SA_x	is a x-th bit of source address, the number of SA bits is greater than or equal 26;
f_x	is a x-th bit of frame number, the number of f bits is 6;
M_x	is a x-th bit of minute, the number of M bits is 6;
H_x	is a x-th bit of hour for 24-hour clock, the number of H bits is 5 and H has the value from 0 to 23;
$PN(clk)$	is a pseudo random bit of PN code generator at clock clk;
&	is AND bit operation that means a logical multiplication;
^	is exclusive OR bit operation that means exclusive logical sum operation;
$floor(x)$	means the largest integer among integers smaller than or equal to x.

The B sync unit that does not know the current time sets each bit of $[b_{16}, b_{15}, ..., b_0]$ to [1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1].

When there is no broadcast slot that can be allocated, the unit shall report this to the upper layer with DLtoUP.NotiGetSCH.

Figure 13 shows the selection procedure for a broadcast slot to attempt allocation.

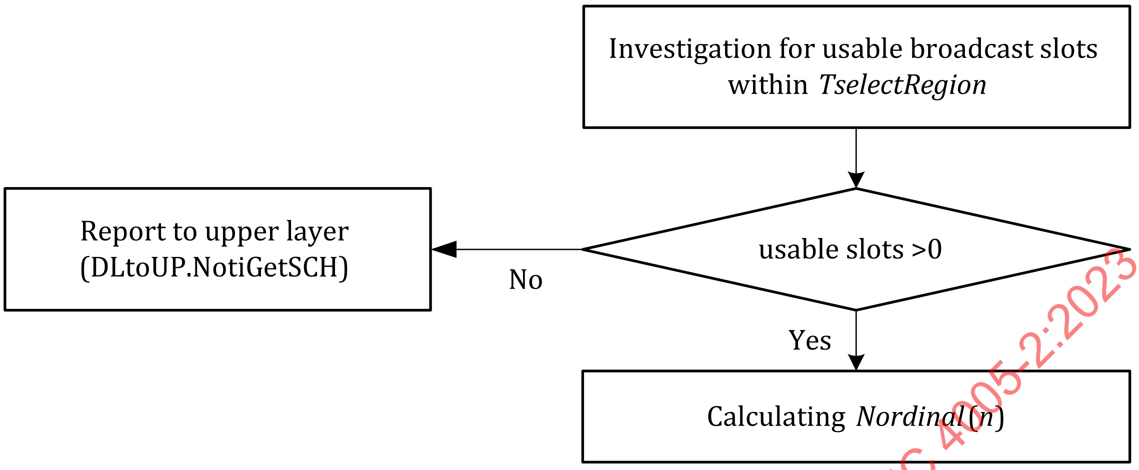


Figure 13 — Selection procedure for a broadcast slot to attempt allocation

6.3.3.2.2 Competition method for broadcast slot allocation

The tone subslot set consists of 33 tone subslots. The competition for allocating data slot n is performed twice in the same tone subslot set $\{S_n\}$. These are the first competition and the second competition.

The first competition is performed as follows:

a) The first subslot, SS_{first} is selected among the subslots from SS_{start} to SS_{32}.

b) Carrier sensing is performed until before SS_{first}.

c) If the carrier sensing result is 'signal detection', it is determined that the competition is lost, where the reception power threshold value for determining tone detection is PRXtoneCompeteThre.

d) If the carrier sensing result is 'no signal detection', the tone signal transmission starts from the first subslot.

The second competition is performed only by the unit winning the first competition. If the first subslot number is 32, the second competition is not performed.

The second competition is performed as follows:

a) The second subslot, SS_{second} is selected among the subslots from the next subslot of SS_{first} to SS_{32}.

b) The unit performs carrier sensing instead transmitting a competition tone signal in the second subslot.

c) If the sensing result is 'signal detection', it is determined that the competition is lost and unit stops to transmit a competition tone signal.

d) If the sensing result is 'no signal detection', a competition tone signal is transmitted from the next subslot of SS_{second} to SS_{32}.

Figure 14 shows the competition procedure for slot allocation.

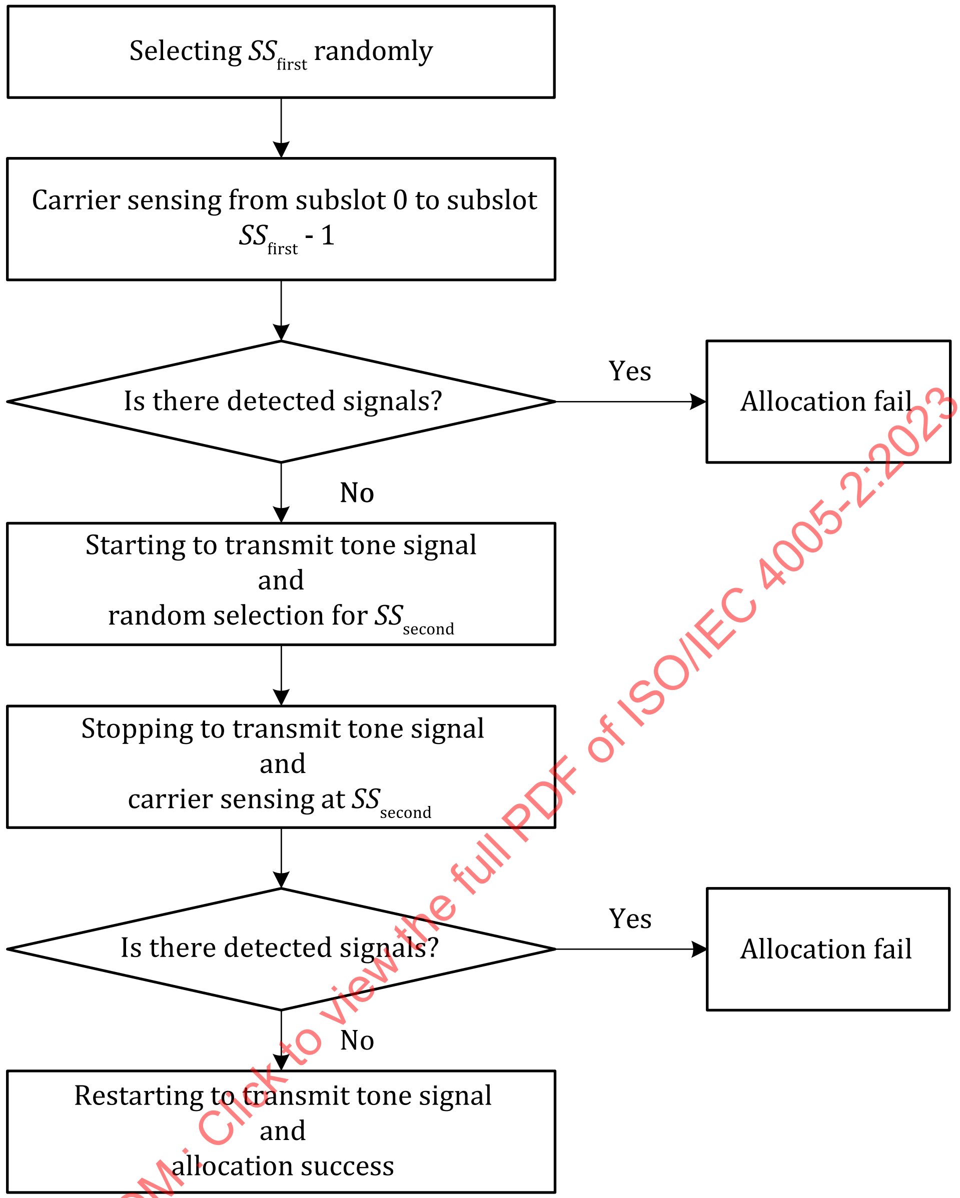


Figure 14 — Competition procedure for slot allocation

The first subslot and the second subslot are calculated in the PN code generator in Figure 12, where the used parameter values are as follows:

$[D_{25}(0), D_{24}(0), \ldots, D_0(0)] = [SA_{25}, SA_{24}, \ldots, SA_0]$

$[b_{25}, b_{24}, \ldots, b_{17}] = [s_8, s_7, \ldots, s_0]$

$[b_{16}, b_{15}, \ldots, b_{11}] = [f_5, f_4, \ldots, f_0]$

$[b_{10}, b_9, \ldots, b_0] = [M_5, M_4, M_3, M_2, M_1, M_0, H_4, H_3, H_2, H_1, H_0]$

$[a_{25}, a_{24}, \ldots, a_0] = [SA_0, SA_1, \ldots, SA_{24}, 1]$

$PN(clk) = (D_0(clk) \,\&\, b_0) \wedge (D_1(clk) \,\&\, b_1) \wedge \ldots \wedge (D_{25}(clk) \,\&\, b_{25})$

$N_{PN1} = [PN(12), PN(11), \ldots, PN(1), PN(0)]$

$N_{PN2} = [PN(25), PN(24), \ldots, PN(14), PN(13)]$

$$SS_{first} = SS_{start} + floor((N_{PN1} \times (33 - SS_{start}))/2^{13})$$
$$SS_{second} = SS_{first} + 1 + floor((N_{PN2} \times (32 - SS_{first}))/2^{13}) \quad (17)$$

where

s_x is a x-th bit of slot number, the number of bits of s is 9;

others are the same with Formula (16).

The B sync unit that does not know the current time sets each bit of $[b_{16}, b_{15}, ..., b_0]$ to '1'.

See Annex E.

6.3.3.2.3 Broadcast slot allocation procedure

The data link receives a broadcast slot allocation request UPtoDL.ReqGetSCH from the upper layer. Thereafter, a procedure for allocating a broadcast slot by a unit is as follows:

a) Data link selects slots for allocation competition using 6.3.3.2.1, where *TselectRegion* is 1 sec and k is 0. If there is no available slot, it is reported as DLtoUP.NotiGetSCH at the upper layer.

b) The data link reselects the fastest slot among the selected usable slots. In addition, the data link performs the allocation competition in 6.3.3.2.2 for the selected slot, where SS_{start} is 1.

c) If data link wins the slot allocation competition, this is reported to the upper layer as DLtoUP. NotiGetSCH and then upper layer requests a packet transmission on the allocated data slot with UPtoDL.ReqTxBslot.

d) If the data link loses the slot allocation competition, it increases k by 1. If k is kMax, the DLL reports the failure to the upper layer. When k is less than kMax, if allocable usable slots remain, b) is performed again. If there are no allocable usable slots, perform a) again, where kMax s is received from the upper layer through the UPtoDL.InfoSharedChannel interface.

Figure 15 shows the broadcast slot allocation procedure.

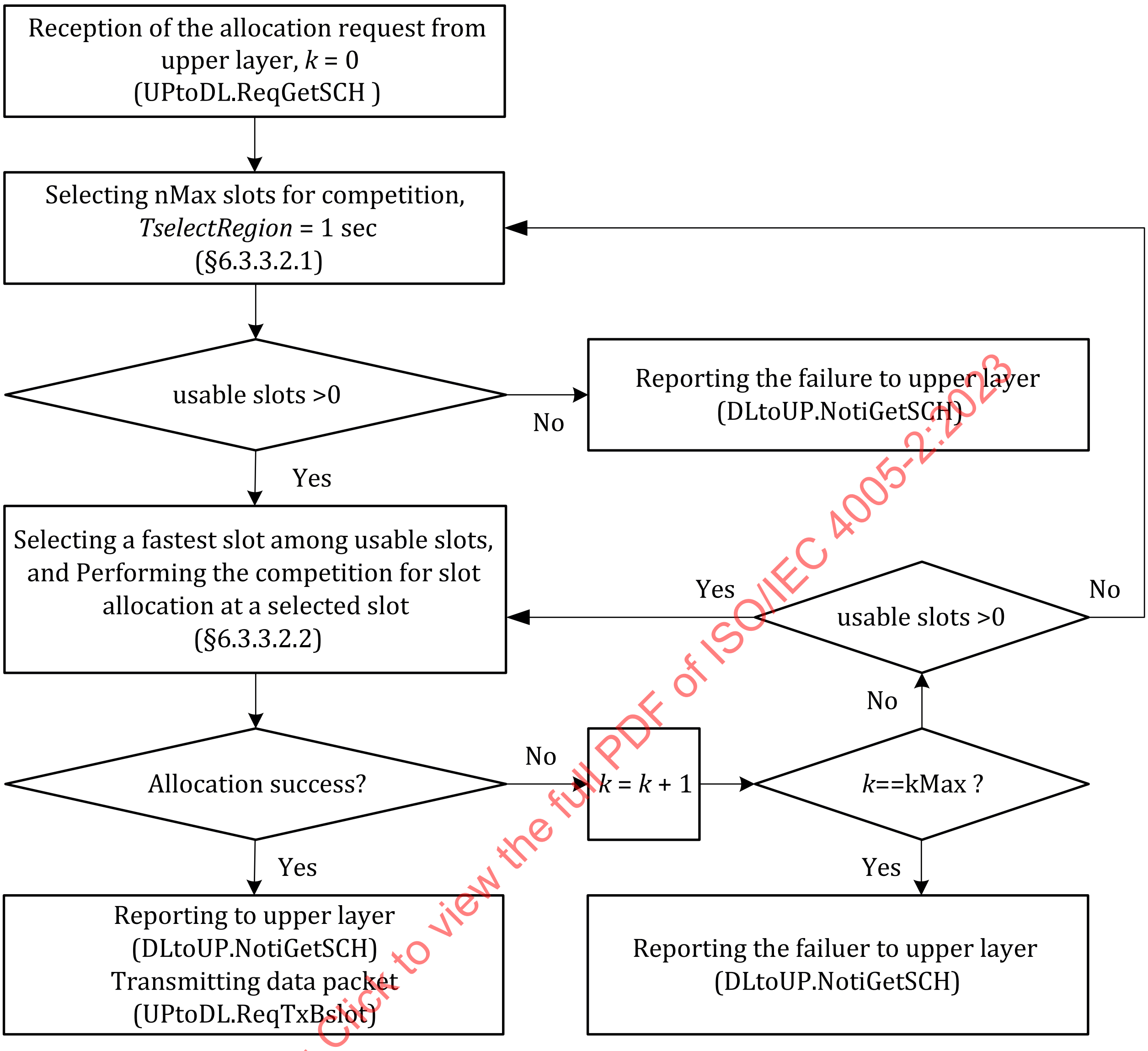


Figure 15 — Broadcast slot allocation procedure

6.3.3.3 Occupation and collision management and return of a broadcast slot

6.3.3.3.1 General

The unit can continuously occupy the allocated broadcast slot from the next frame of the allocation frame by performing slot clearing. The occupation decision on the allocated broadcast slot is determined by the upper layer. Returning of the occupied slot is performed by not performing slot clearing. If the slot allocated by the unit is used only in the frame, slot clearing is not performed in the next frame.

The unit shall continuously monitor the resource collision of the occupied broadcast slots. For this monitoring, the unit transmits a collision tone in the subslot set mapped with the occupied broadcast slots. If the detected power of received collision tone exceeds the predefined threshold, the unit shall stop slot clearing in the occupied broadcast slot.

When the occupied slot is returned because of the collision, the unit automatically can attempt to allocate a new broadcast slot. Upper layers can cancel or limit such an attempt.

6.3.3.3.2 Slot clearing

'Slot clearing' means that a unit transmits a tone signal in subslot 0 of the tone subslot set mapped with the allocated data slot in order to continuously use the data slot allocated in the previous frame.

After allocating a slot through competition, as shown in Figure 16, the unit occupies the slot by performing slot clearing from the next frame in the tone subslot set mapped to the allocated data slots. If slot clearing is stopped, the slot is considered returned. In Figure 16, <F_x, S_n> means data slot y of frame x and {F_x, S_n} means a tone subslot set mapped to <F_x, S_n>. See Annex F for a slot clearing example.

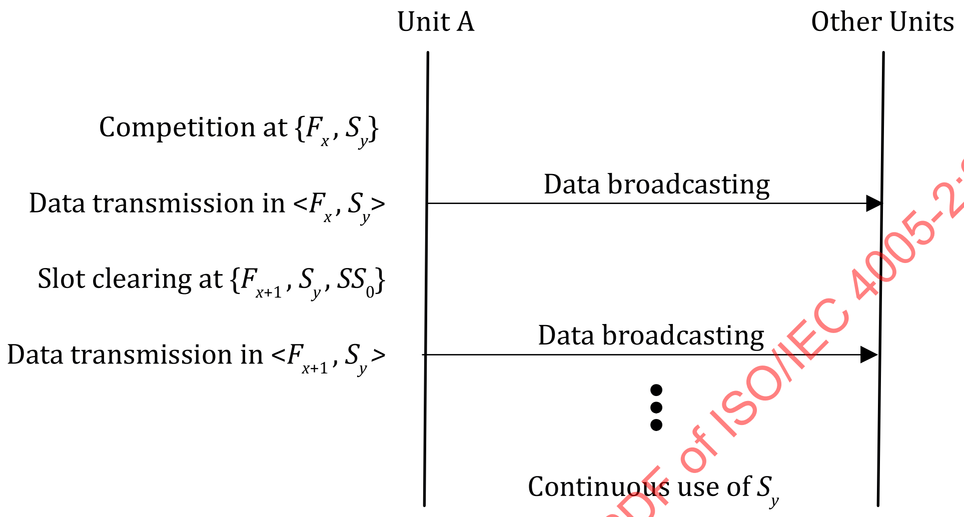
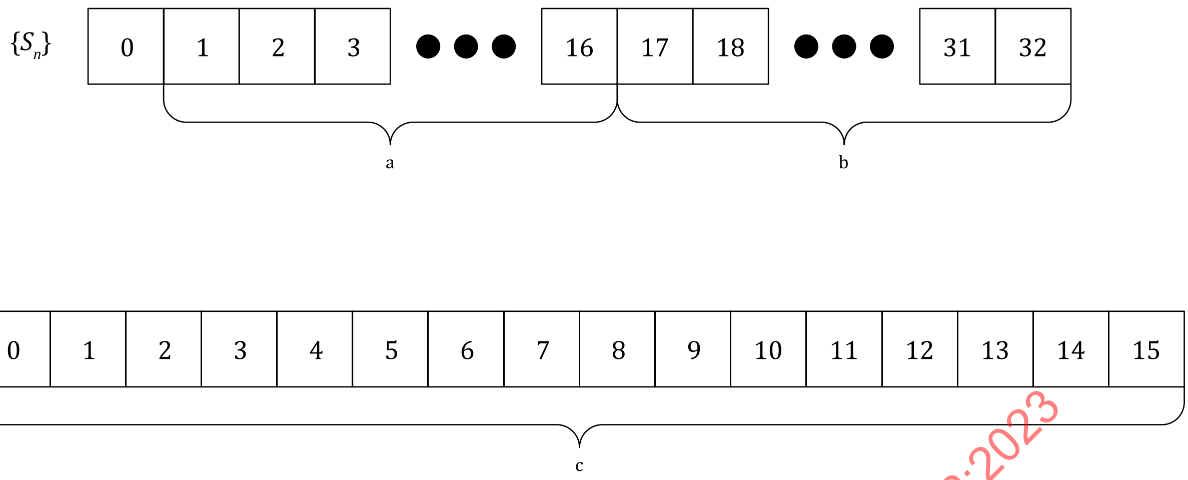


Figure 16 — Procedure for allocation and occupation of broadcast slot

6.3.3.3.3 Collision management of a broadcast slot

A unit performs collision management in the subslot set mapped to the occupied broadcast slot, i.e. collision management is performed only when slot clearing is performed. The unit transmits three collision tones of the corresponding subslot set and detects the collision tones of other units in the subslots in which the collision tones are not transmitted.

The transmission region of the collision tone is divided into two parts as shown in Figure 17. The first transmission region is from subslot 1 to subslot 16, and the second transmission region is from subslot 17 to subslot 32. The lengths of the first transmission region and the second transmission region are both 16.



Key

$\{S_n\}$ tone subslot set mapped to data slot S_n

[a] First transmission region.

[b] Second transmission region.

[c] Renumbering of each transmission region.

Figure 17 — Transmission region for collision tones

When the unit transmits the collision tone, the unit whose last bit of *SA* is 0 uses the first transmission region in the even-numbered frame and uses the second transmission region in the odd-numbered frame. The unit whose last bit of *SA* is 1 uses the second transmission region in the even-numbered frame and uses the first transmission region in the odd-numbered frame.

The unit transmits its tone signal three times in one of the two transmission regions. The unit shall detect the tones of the other unit by performing carrier sensing on subslots that do not transmit their collision tones among the subslots of the two regions.

Subslot numbers that transmit their tones in a transmission region are determined using the PN code generator shown in Figure 12. First, $N_{CT}(0)$, $N_{CT}(1)$, $N_{CT}(2)$ are calculated as follows:

$$[D_{25}(0), D_{24}(0), ..., D_0(0)] = [SA_{25}, SA_{24}, ..., SA_0]$$

$$[b_{25}, b_{24}, ..., b_{17}] = [s_8, s_7, ..., s_0]$$

$$[b_{16}, b_{15}, ..., b_{11}] = [f_5, f_4, ..., f_0]$$

$$[b_{10}, b_9, ..., b_0] = [M_5, M_4, M_3, M_2, M_1, M_0, H_4, H_3, H_2, H_1, H_0]$$

$$[a_{25}, a_{24}, ..., a_0] = [SA_0, SA_1, ..., SA_{24}, 1]$$

$$PN(clk) = (D_0(clk) \,\&\, b_0) \wedge (D_1(clk) \,\&\, b_1) \wedge ... \wedge (D_{25}(clk) \,\&\, b_{25})$$

$$N_{PN1}(n) = [PN(12n{+}11), PN(12n{+}10), ..., PN(12n{+}1), PN(12n)]$$

$$N_{CT}(0) = floor((16\times N_{PN1}(0))/2^{12})$$

$$N_{CT}(1) = floor((15\times N_{PN1}(1))/2^{12})$$

$$N_{CT}(2) = floor((14\times N_{PN1}(2))/2^{12}) \qquad (18)$$

where

s_x is a x-th bit of slot number, the number of bits of s is 9;

others are the same with Formula (16).

The positions N_{C0}, N_{C1}, and N_{C2} of the three collision tones are as follows:

$$
\begin{aligned}
N_{C0} &= N_{CT}(0) + N_{CToffset} \\
N_{C1} &= N_{CT}(1) + N_{CToffset,} && \text{if } N_{CT}(1) < N_{CT}(0) \\
&= N_{CT}(1) + 1 + N_{CToffset,} && \text{if } N_{CT}(0) \le N_{CT}(1) \\
N_{C2} &= N_{CT}(2) + N_{CToffset,} && \text{if } N_{CT}(2) < N_{CT}(0) \text{ and } N_{CT}(2) < N_{CT}(1) \\
&= N_{CT}(2) + 1 + N_{CToffset,} && \text{if } N_{CT}(0) \le N_{CT}(2) < N_{CT}(1) \text{ or } N_{CT}(1) \le N_{CT}(2) < N_{CT}(0) \\
&= N_{CT}(2) + 2 + N_{CToffset,} && \text{if } N_{CT}(0) \le N_{CT}(2) \text{ and } N_{CT}(1) \le N_{CT}(2)
\end{aligned}
\tag{19}
$$

where $N_{CToffset}$ is 1 for the first transmission region and 17 for the second transmission region.

The B sync unit that does not know the current time sets each bit of [b_{16}, b_{15}, ..., b_0] to '1'.

There are three collision detection conditions. First, the slot return is determined when two or more collision tones with received power greater than PRXcollsiontoneThre0 are detected in the subslot set.

Second, the slot return is determined when one tone with a received power greater than PRXcollsiontoneThre0 is detected in one frame and another one tone was detected in the previous frame.

Third, when *FrmAvgDT* is equal to or greater than PRXcollsiontoneThre2, the slot return is determined. Here, *FrmAvgDT* is an average power of one frame average power during PRXnumAvgFrm. One frame average power is calculated in a subslot set by averaging received power values of detected tones those received powers are equal to or greater than PRXcollsiontoneThre1. If the number of detected tones is greater than 3 in the subslot set, the one frame average power is calculated with the three values of the largest received power.

When the return of the slot is determined, the unit informs this return to the upper layer as DLtoUP. NotiGetSCH and returns the collision slot after the TschReturn frame from the current frame. TschReturn is greater than 0 because the unit cannot return a slot in the frame for which the slot return was determined.

Figure 18 shows slot return procedure based on collision tone detection.

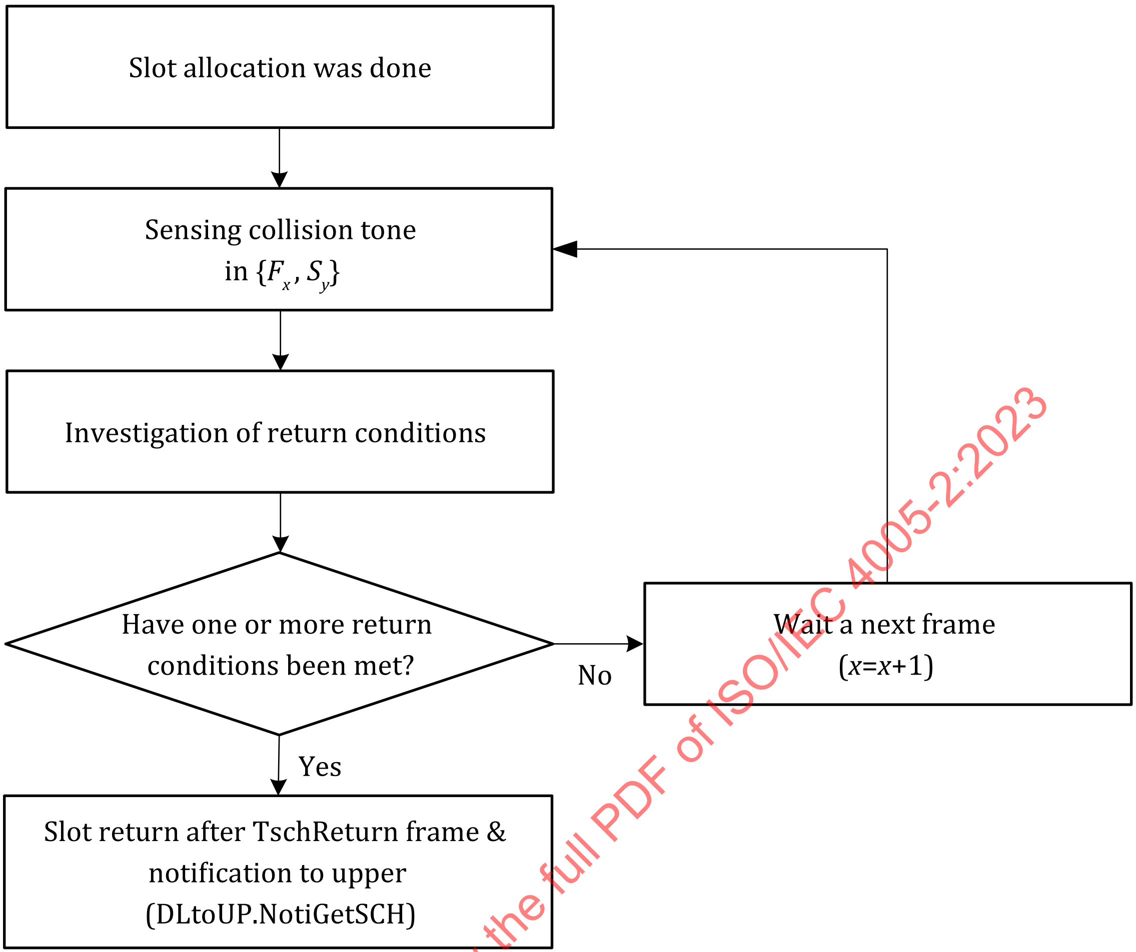


Figure 18 — Slot return procedure based on collision tone detection

If a collision occurs and it is determined that the broadcast slot is returned, the upper layer determines whether to reassign the broadcast slot. If the reallocation is determined at the upper layer, the unit performs the allocation procedure in 6.3.3.2.3 again.

Even when no collision occurs, the upper layer can return the occupied broadcast slots as needed.

6.3.3.4 Transmission of a broadcast slot

When the DLL receives UPtoDL.ReqTxBslot from the upper layer, it transmits the designated data in the designated slot. When a broadcast slot is allocated and occupied, the upper layer shall send UPtoDL.ReqTxBslot related to the corresponding broadcast slot to DLL.

6.3.4 Talk slot allocation transmission and response transmission

6.3.4.1 General

Talk slots are allocated through competition. The unit to which the talk slot has been allocated transmits information through this slot, and the unit which has received the talk slot transmits response information using response clearing. The talk slot is not occupied. Thus, no resource collision check is performed for the talk slot. One unit transmits the information through the one-time allocation, and the other unit receiving the talk slot transmits the response through the one-time allocation.

Talk slot transmission occurs when TslotTxEvent occurs. TslotTxEvent refers to an event that transmits a talk slot. TslotTxEvent occurs when UPtoDL.ReqTxTslot is received from the upper layer, CSCH number negotiation is performed, or VSCH number negotiation is performed. When UPtoDL.ReqTxTslot is received from the upper layer, the SC DLL reports the success or failure of the talk slot transmission to the upper layer using DLtoUP.NotiTxTslot.

6.3.4.2 Talk slot allocation

When a TslotTxEvent is generated, the procedure for the data link to allocate the talk slot is as follows:

a) DLL uses the method of 6.3.3.2.1 to select a talk slot. In this case, the same process is performed for the talk slot instead of the broadcast slot, where *TselectRegion* is 1 sec, and when the data link selects a talk slot, nMaxTalk slots are selected from among $NUM_{\text{selectable\_slot}}$ allocable talk slots, i.e. *Nordinal*(0), *Nordinal*(1),..., *Nordinal*(nMaxTalk-1) are calculated. When calculating *Nordinal*(n), the previously selected slots corresponding *Nordinal*(0),..., *Nordinal*(n-1) shall be excluded from the list of allocable slots. If there is no allocable slot, the unit fails to transmit talk slot.

b) DLL selects one among the nMaxTalk slots that is fastest to the current time.

c) In the selected slot, allocation competition is performed using the method of 6.3.3.2.2, where, SS_{start} is 1.

d) If DLL wins the slot allocation competition, the talk slot allocation request packet is transmitted in the allocated slot.

e) If DLL loses the slot allocation competition, the slot closest to the current time is selected from among the selected slots, and c) is performed again.

f) If allocation fails in all of the selected slots, the unit fails to transmit talk slot.

If necessary, the SC DLL notifies the upper layer or CC DLL or VC DLL of the success or failure of the talk slot transmission.

Figure 19 shows the allocation procedure of talk slot for talk request.

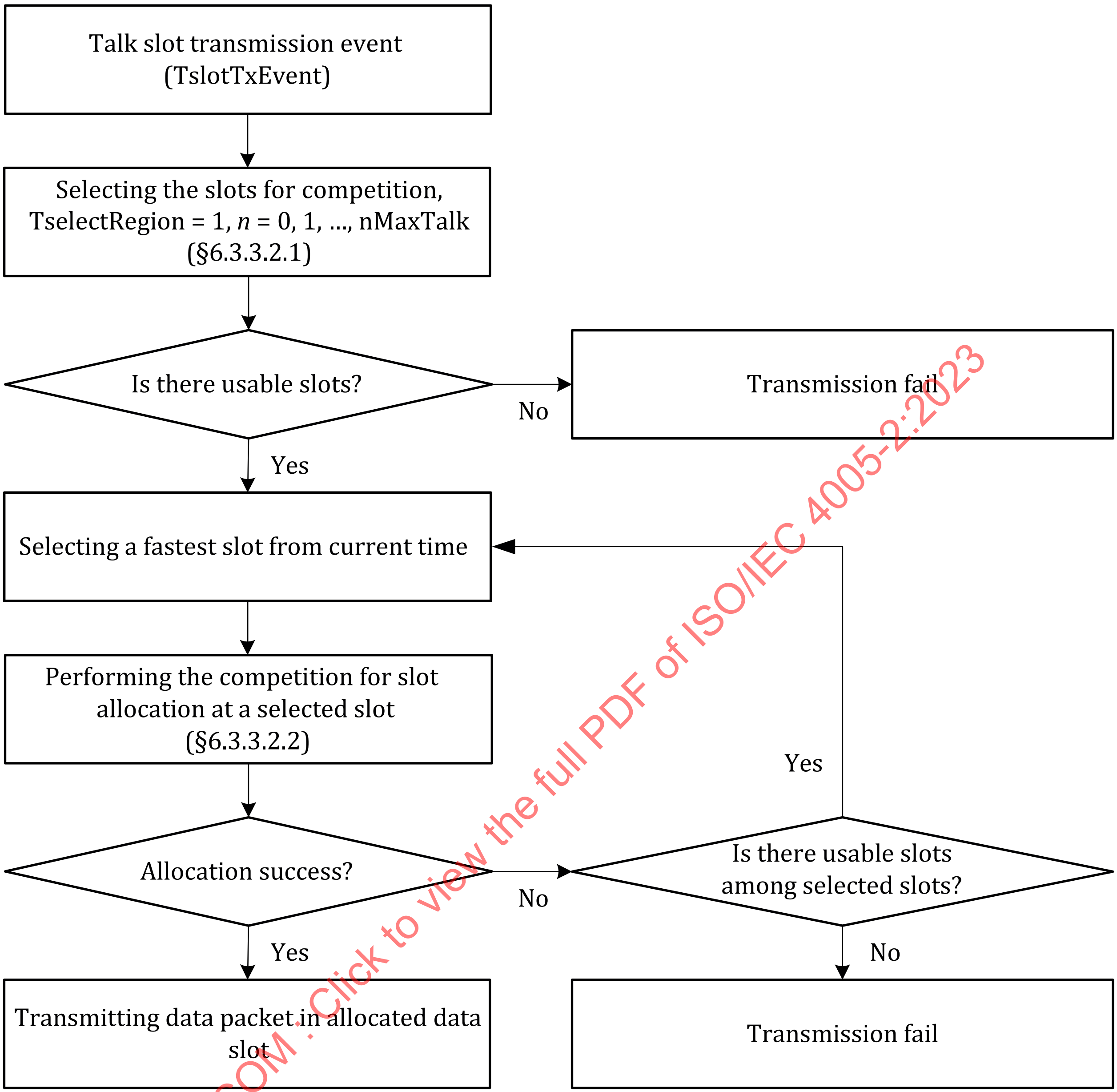


Figure 19 — Allocation procedure of talk slot for talk request

If the unit succeeds in allocating and transmitting the talk slot, the response clearing is performed once in the same slot of the next frame. The subslot related to response clearing is the first subslot.

6.3.4.3 Talk response slot allocation

The unit receiving a talk request packet shall transmit a response packet when the corresponding destination address matches its own address and the corresponding reception packet requests a response. The slot allocation procedure for response is as follows:

First, the talk request packet is received at $<F_x, S_y>$.

Second, response clearing is performed in $\{F_{x+1}, S_y, SS_0\}$.

Third, by using the method of 6.3.3.2.2, slot allocation competition is performed in $\{F_{x+1}, S_y\}$, and if the unit wins the competition, a response packet is transmitted, and if the unit loses, a response packet is not transmitted, where, SS_{start} is 1.

When the talk response packet is generated from the upper layer, the upper layer transmits UPtoDL. ReqTxTslot to the DLL after the first step. At this time, ResponseClearing is set to '1'.

Figure 20 shows the allocation procedure of talk slot for talk response.

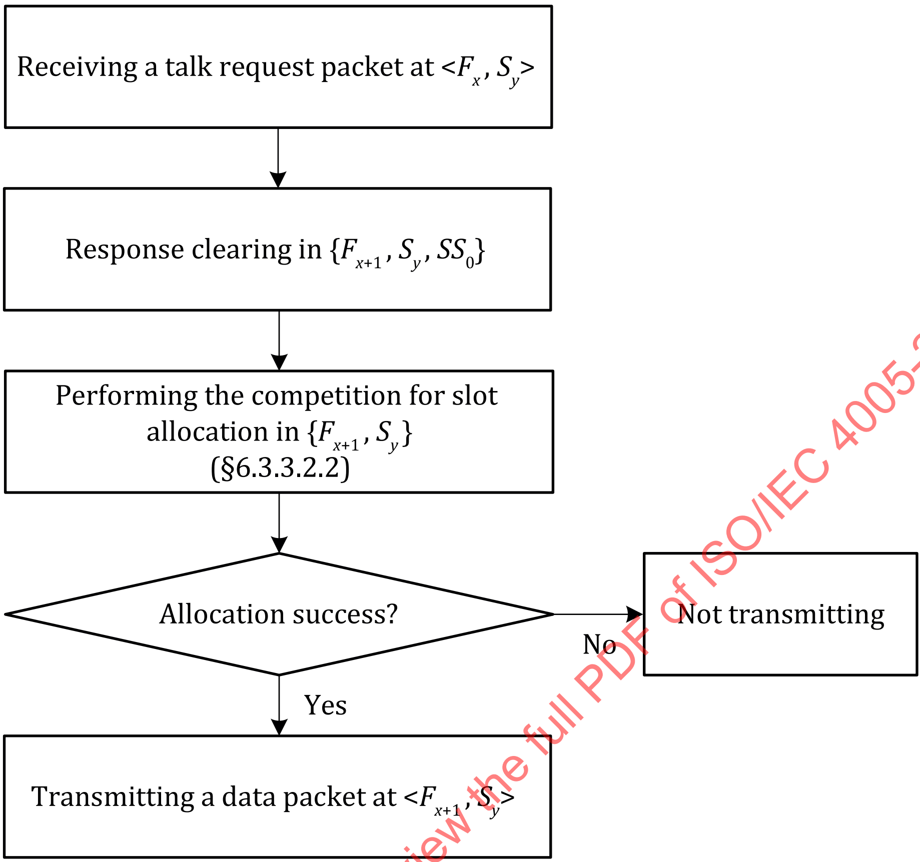


Figure 20 — Allocation procedure of talk slot for talk response

6.4 Data broadcast and exchange

6.4.1 Data packet format

The data packet format has two types, one is a broadcast packet used in a broadcast slot as shown in Figure 21 and the other is a talk packet used in a talk slot as shown in Figure 22.

The packet header determines the packet type. In this document, packet headers, 0x00, 0x02, and 0x04 are broadcast packet headers, and 0x01, 0x03, 0x05 and 0x06 are talk packet headers.

Packet header (8 bits)	Sync field (2 bits)	Source address (26 bits)	Trust field (8 bits)	Data (748 bits)	CRC (24 bits)

816 bits

Figure 21 — Broadcast packet format

Packet header (8 bits)	Sync field (2 bits)	Source address (26 bits)	Destination address (26 bits)	Trust field (8 bits)	Data (722 bits)	CRC (24 bits)
816 bits						

Figure 22 — Talk packet format

A broadcast packet consists of packet header, sync field, source address, trust field, data field, and CRC. The talk packet consists of packet header, sync field, source address, destination address, trust field, data field, and CRC.

6.4.1.1 Data packet header

The packet header consists of 8 bits. The packet header is determined at the upper layer except for some headers. Some packet headers shown in Table 7 are determined by the data link layer.

Table 7 — Packet header list predefined by data link layer

Packet header	Description
0x00	General broadcast slot header
0x01	General talk slot header
0x02	Broadcast slot header including parsing block for talk
0x03	Talk slot header including parsing block for broadcast
0x04	Broadcast slot header for control channel operation
0x05	Talk slot header for control channel operation
0x06	Talk slot header for video channel operation
0x07 ~0xFF	TBD for the upper layer

6.4.1.2 Sync field

The sync field is 2 bits. The meaning of the sync field is shown in Table 8.

Table 8 — Sync field meaning

Sync field	Description
0x0	A sync
0x1	B sync
0x2	C sync
0x3	reserved

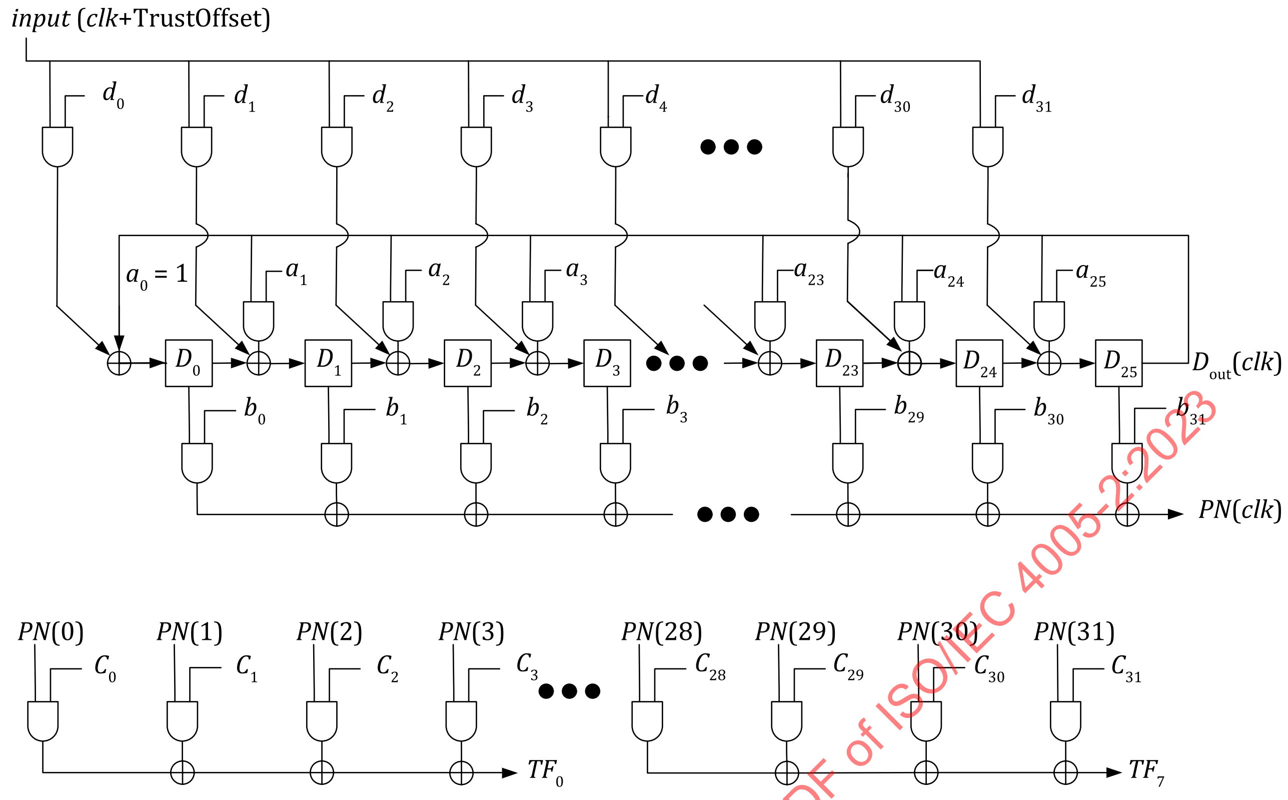
6.4.1.3 Source address and destination address

It is recommended to set the source address and the destination address to 26 bits. However, the upper layer can set the unit address bit number larger. In this case, the upper layer shall shorten the data field length by the increased number of bits.

6.4.1.4 Trust field

The trust field shall be received by all units, but the trust check shall be performed only by an authorized receiver.

The trust field generation method uses the LFSR of the following structure as shown in Figure 23.



Key

Input(*clk*+TrustOffset)	a (*clk*+TrustOffset)-th bit of a data field
a_k	a *k*-th bit for AND operation with D_{out}
b_k	a *k*-th bit for AND operation with D_k
c_k	a *k*-th bit for AND operation with *PN*(*k*)
d_k	a *k*-th bit for AND operation with *input*(*clk*+TrustOffset)
D_k	a *k*-th register
$D_{out}(clk)$	output of shift register D_{31} at clock *clk*
PN(*clk*)	PN coded generator output for trust field generation at clock *clk*
TF_x	a *x*-th bit of trust field

Figure 23 — Trust field generation LFSR structure

$[D_{31}(0), D_{30}(0), \ldots, D_{26}(0)] = [\mathrm{K}_{101}, \mathrm{K}_{100}, \ldots, \mathrm{K}_{96}]$

$[D_{25}(0), D_{24}(0), \ldots, D_0(0)] = [SA_{25}, SA_{24}, \ldots, SA_0]$

$[d_{31}, d_{30}, \ldots, d_{23}] = [s_8, s_7, \ldots, s_0]$

$[d_{22}, d_{21}, \ldots, d_{17}] = [f_5, f_4, \ldots, f_0]$

$[d_{16}, d_{15}, \ldots, d_{11}] = [M_5, M_4, M_3, M_2, M_1, M_0]$

$[d_{10}, d_9, \ldots, d_6] = [H_4, H_3, H_2, H_1, H_0]$

$[d_5, d_4, \ldots, d_0] = [J_4, J_3, \ldots, J_0, \mathrm{K}_0]$

$[b_{31}, b_{30}, \ldots, b_0] = [\mathrm{K}_{95}, \mathrm{K}_{94}, \ldots, \mathrm{K}_{64}]$

$[c_{31}, c_{30}, \ldots, c_0] = [\mathrm{K}_{63}, \mathrm{K}_{62}, \ldots, \mathrm{K}_{32}]$

$[a_{31}, a_{30}, \ldots, a_0] = [\mathrm{K}_{31}, \mathrm{K}_{30}, \ldots, \mathrm{K}_1, 1]$

$$PN(clk) = (D_0(clk) \& b_0) \wedge (D_1(clk) \& b_1) \wedge \dots \wedge (D_{31}(clk) \& b_{31})$$

$$TF_n = (PN(4n) \& c_{4n}) \wedge (PN(4n+1) \& c_{4n+1}) \wedge (PN(4n+2) \& c_{4n+2}) \wedge (PN(4n+3) \& c_{4n+3}),\ n=0, 1, \dots, 7,\ [c_{4n}, c_{4n+1}, c_{4n+2}, c_{4n+3}] \neq [0, 0, 0, 0] \quad (20)$$

where

s_x	is a x-th bit of slot number, the number of bits of s is 9;
J_x	is a x-th bit of a date and J has a value from 1 to 31;
K_x	is a x-th bit of a secret key between the trust check system and the unit and K has a length of 102 bits.;
TF_n	is a n-th trust bit of trust field. n=0, 1, ..., 7;
others	are the same as Formula (16).

The LFSR is updated as follows.

$$D_0(clk+1) = (D_{31}(clk) \& a_0) \wedge (input(\mathrm{TrustOffset} + clk) \& d_0)$$

$$D_i(clk+1) = D_{i-1}(clk) \wedge (D_{31}(clk) \& a_i) \wedge (input(\mathrm{TrustOffset} + clk) \& d_i),\ i = 1, \dots, 31 \quad (21)$$

where '*input*' means bits of a data field and TrustOffset is determined by an upper layer.

B sync units that does not know the current time set each bit of $[d_{22}, d_{21}, \dots, d_1]$ to '1'.

6.4.1.5 Data field

6.4.1.5.1 General

The data field consists of parsing blocks. The parsing block consists of a parsing header and a parsing field. The parsing header is 8 bits, and the length of the parsing field depends on the parsing header. Each parsing block can be inserted anywhere in the data field. However, the parsing block and the parsing block shall be attached without empty space, and the first parsing block shall be located immediately after the trust field.

The upper layer shall transmit data of a length less than or equal to the length of the data field to the data link layer. If the length of information provided from the upper layer is smaller than the length of the data field, the remaining bits are processed as follows.

If the length of unused extra data bits is less than 8, the extra data bits are filled with '0'. If the length of unused extra data bits is greater than or equal to 8, the extra data bits are filled with a parsing header '0x80' and a padding bit string of length 97 indicating padding. The padding bits are constructed by using a string of bits of length 97 repeatedly. A 97-bit sequence consists of the next 96-bit sequence followed by a bit of '1'.

— ‘0x71E5477D_A5B32BF7_E5469C8E’

where '_' is used to distinguish 32 bits without meaning and ‘0x’ means hex notation.

6.4.1.5.2 Parsing header and parsing field

6.4.1.5.2.1 General

The parsing header consists of 8 bits. If the most significant bit of the parsing header is '1', it has a unique meaning regardless of the packet header. If the most significant bit of the parsing header is '0', it has a different meaning depending on the packet header, and each meaning is determined by the upper layer. However, some parsing headers are exceptionally determined in this document.

Among the parsing headers with the most significant bit '1', parsing headers not described in this document are determined in the upper layer as shown in Table 9.

Table 9 — Parsing header list predefined by data link layer

Parsing header	Description
0x00~0x7F	Reserved for the upper layer
0x80	Data padding
0x81	This is used in talk packet. This requests to transmit a response on a talk slot by using broadcast slot only or both broadcast slot and talk slot.
0x82	Current time including date, hour, minute, second, slot number
0x83	Current time including slot number
0x84	Request for current time information
0x85	This is used in talk packet when sending a response on the data received in a broadcast slot by using a talk slot.
0x86	Position and power information related with control channel (described in ISO/IEC 4005-3)
0x87	CSCH map (described in ISO/IEC 4005-3)
0x88	Position and power information related with video channel (described in ISO/IEC 4005-4)
0x89	VSCH map (described in ISO/IEC 4005-4)
0x8A	Position and power information related with video channel and control channel (described in ISO/IEC 4005-4)
0x8B	Indicating that its parsing field is a parsing block for the control channel (described in ISO/IEC 4005-3)
0x8C	Indicating that its parsing field is a parsing block for the video channel (described in ISO/IEC 4005-4)
0x8D	Position and velocity information of a UA and a controller
0x8E	Position and way point information of a UA and a controller
0x8F~0xFF	Reserved for the upper layer

6.4.1.5.2.2 Parsing field of 0x80

The parsing field of parsing header 0x80 means an unfilled portion of the data field. Parsing fields are handled in the manner described in 6.4.1.5.

6.4.1.5.2.3 Parsing field of 0x81

This parsing field is used in talk packet. This requests to transmit a response on a talk slot by using broadcast slot. There is one parameter in this PF as shown in Table 10.

Table 10 — Parsing field of PB 0x81

Bits`	Description
[0]	RequestMethod '0' – Broadcast slot only '1' – Both broadcast slot and talk slot

6.4.1.5.2.4 Parsing field of 0x82

Table 11 — Parsing field of 0x82

Bits`	Description
[0:8]	Current slot number 0~499
[9:14]	Second 0~59
[15:20]	Minute 0~59
[21:25]	Hour 0~23
[26:30]	Date 1~31

Parsing header 0x82 provides the current time including day, hour, minute, second, and slot number as shown in Table 11.

6.4.1.5.2.5 Parsing field of 0x83

Table 12 — Parsing field of 0x83

Bits`	Description
[0:8]	Current slot number 0~499

Table 12 shows the parsing field of 0x83.

6.4.1.5.2.6 Parsing field of 0x84

There is no parsing field for parsing header 0x84. The unit that does not know current time requests current time information from another unit by parsing header 0x84.

6.4.1.5.2.7 Parsing field of 0x85

There is no parsing field for parsing header 0x85. This parsing header is used in talk packet when sending a response on the data received in a broadcast slot by using a talk slot.

6.4.1.5.2.8 Parsing field of PB 0x8D

This parsing block provides information about the location and speed of the UA and controller. The first two bits of the parsing field indicate the information type. Table 13, Table 14, Table 15, and Table 16 show the parsing field of PB 0x8D according to information types. The details of each field shall follow Annex B.

Table 13 — Parsing field of PB 0x8D with information type '00'

Bits`	Description
[0:1]	'00'. This Information type means UA location information.
[2:25]	Own latitude
[26:49]	Own longitude
[50:61]	Own altitude

Table 14 — Parsing field of PB 0x8D with information type '01'

Bits`	Description
[0:1]	'01'. This Information type means UA location and velocity information
[2:25]	Own latitude
[26:49]	Own longitude

Table 14 *(continued)*

Bits`	Description
[50:61]	Own altitude
[62:69]	Own horizontal ground speed
[70:77]	Own heading, degree
[78:84]	Own vertical speed

Table 15 — Parsing field of PB 0x8D with information type '10'

Bits`	Description
[0:1]	'01'. This Information type means UA and controller location information
[2:25]	Own latitude
[26:49]	Own longitude
[50:61]	Own altitude
[62:77]	Latitude difference between the controller and the UA
[78:97]	Longitude difference between the controller and the UA
[98:105]	altitude difference between the controller and the UA

Table 16 — Parsing field of PB 0x8D with information type '11'

Bits`	Description
[0:1]	'10'. This Information type means UA and controller location and UA velocity information
[2:25]	Own latitude
[26:49]	Own longitude
[50:61]	Own altitude
[62:69]	Own horizontal ground speed
[70:77]	Own heading, degree
[78:84]	Own vertical speed
[85:100]	Latitude difference between the controller and the UA
[101:120]	Longitude difference between the controller and the UA
[121:128]	Altitude difference between the controller and the UA

6.4.1.5.2.9 Parsing field of 0x8E

This parsing block provides information about the location of the UA and the controller, and information about the UA's way point. The first bit of the parsing field indicates the information type. Table 17 and Table 18 show the parsing field of PB 0x8E according to information types. The details of each field shall follow Annex B and Annex C.

Table 17 — Parsing field of PB 0x8E with information type '0'

Bits`	Description
[0]	'0'. This Information type means UA information only
[1:2]	The number of way points is '[1:2]+1'.
[3:26]	Own latitude (see B.1).
[27:50]	Own longitude (see B.2).
[51:62]	Own altitude (see B.3).
[63:62+53×([1:2]+1)]	Way points information, the length is 53 × ([1:2]+1).

Table 18 — Parsing field of PB 0x8E with information type '1'

Bits`	Description
[0]	'1'. This Information type means UA and controller location and UA way points information
[1:2]	The number of way points is '[1:2]+1'.
[3:26]	Own latitude (see B.1).
[27:50]	Own longitude (see B.2).
[51:62]	Own altitude (see B.3).
[63:78]	Latitude difference between the controller and the UA (see B.7).
[79:98]	Longitude difference between the controller and the UA (see B.8).
[99:106]	Altitude difference between the controller and the UA (see B.9).
[107:106+53×([1:2]+1)]	Way points information, the length is 53 × ([1:2]+1).

6.4.2 Slot planning

6.4.2.1 General

One data channel and one tone channel mapped to it each have 500 slots in one second frame. Each of the 500 slots is predefined to talk slots or broadcast slots. Detailed slot planning is the responsibility of the upper layer.

6.4.2.2 Slot planning rules

6.4.2.2.1 General

Slot planning in the upper layer shall satisfy the following rules:

a) The data channel and the tone channel mapped to it are composed of slot block units. See Figure 10.

b) The data slot block 1 is all dedicated slots, the data slot 3 is a linearization data slot, the tone slot 3 is a linearization tone slot. Tone slots from tone slot 0 to tone slot 2 are information tone slots. They consist of 99 subslots. See Figure 24.

c) Both talk slot and broadcast slot can exist in one slot block.

d) A slot block can select only one slot type among dedicated slot, fixed slot and normal slot.

e) The transmit power of all slots in one slot block shall be the same.

f) Tone slots mapped to dedicated slots can be used as information tone slots, in which case the transmit power of the tone slots is determined by the upper layer.

g) Even if there are a plurality of upper layers, slot planning for one physical channel is the same.

h) If there are multiple channels, slot planning of each channel shall be designed to minimize mutual interference.

Interfaces used for slot planning are UPtoDL.InfoApprovedSlotMap, UPtoDL.InfoBslotMap, UPtoDL.InfoTslotMap, UPtoDL.InfoSuperFrameSlot and UPtoDL.InfoDedicatedSlot.

6.4.2.2.2 Super frame slot

A super frame slot is a slot whose frame period is longer than 1 second. The super frame slot shall be predefined in the upper layer. Possible periods of super frame slot, T_{SF} are 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60 seconds. The super frame slot S_x with T_{SF} period is extended as T_{SF} slot resources as follows:

$$S_{xi}\,,\ 0 \leq i < T_{SF} \tag{22}$$

The configuration of the super frame slot is determined by the upper layer. If video communication is used with shared communication, it is strongly recommended that super frame is not used.

6.4.2.2.3 Dedicated slot and fixed slot and normal slot

The upper layer can reclassify the broadcast slot and the talk slot into a dedicated slot, a fixed slot, and a normal slot as shown in Table 19.

The dedicated slot is a preallocated slot by the upper layer, and unit shall use this slot without competition.

The fixed slot is a predefined slot by the upper layer, and only a unit satisfying a specific condition or a unit previously authorized by the upper layer can allocate this slot through competition.

The normal slot means a slot allocated by the general unit through competition.

Table 19 — Slot classification by transmission type and allocation type

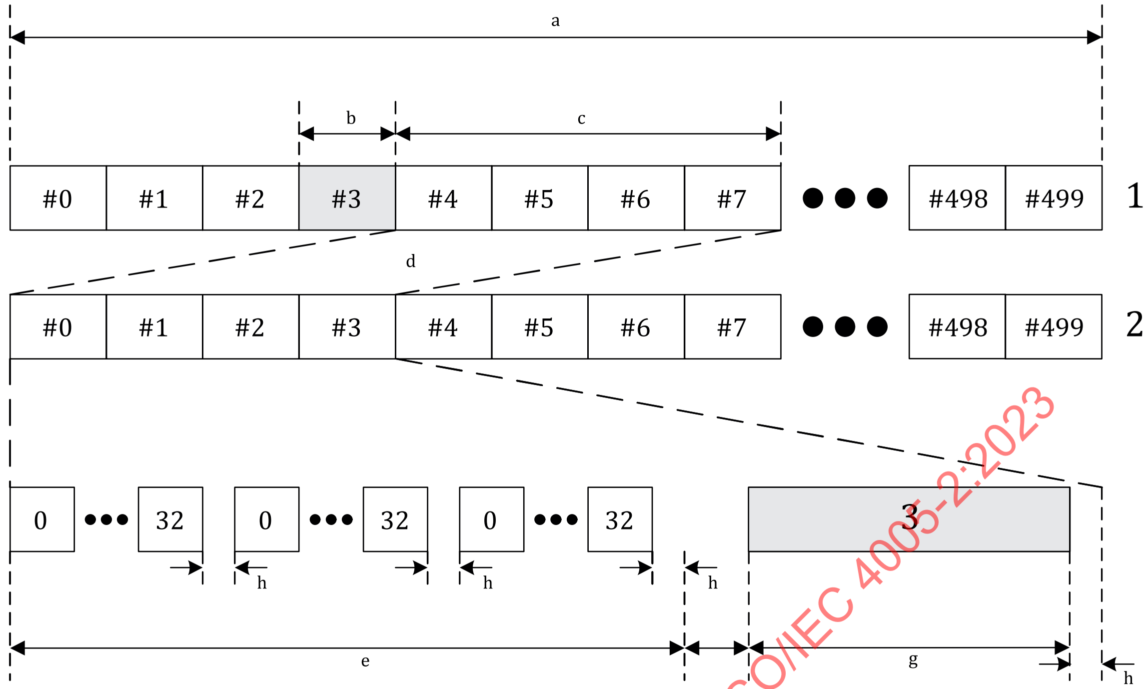
Transmission type	Allocation type		
	dedicated	**fixed**	**normal**
broadcast	dedicated broadcast slot	fixed broadcast slot	normal broadcast slot
talk	dedicated talk slot	fixed talk slot	normal talk slot

6.4.2.2.4 Information tone slot

The competition tone slot mapped with the dedicated data slot do not perform competition. Accordingly, the upper layer can define this slot as an information tone slot. In this document, tone slots from slot 0 to slot 2 are set as information tone slots.

6.4.2.2.5 Linearization slot

The linearization slot is predefined for signal transmission used to linearize the transmitter's power amplifier. The data linearization slot is data slot 3, which is the interval of T_{14} to T_{15} in Figure 2. The tone linearization slot is tone slot 3. The tone linearization interval is 6060 µs to 7980 µs based on the frame start time.



Key

1 data channel

2 tone channel

3 linearization interval

a 1 frame, T_f = 1 second = 500 T_s.

b Linearization slot.

c Dedicated slot block.

d Mapping relation.

e 6 ms (Information tone slots).

f 60 µs (guard time).

g 1920 µs.

h 20 µs.

Figure 24 — Linearization slot and timing

6.4.3 Data broadcasting

6.4.3.1 General

Methods for data broadcasting include one-time data broadcasting and continuous data broadcasting.

6.4.3.2 One-time data broadcasting

One-time data broadcasting can be performed through a broadcast slot or a talk slot.

One-time data broadcasting through a broadcast slot is achieved by not performing slot clearing after broadcast slot allocation.

There are two ways for one-time data broadcasting through a talk slot. The first method is not to perform response clearing after talk slot allocation. In this case, 0x03 is used for the packet header, and all destination address fields shall be filled with '0'.

The second method is to use a parsing header that means a data broadcasting in the talk packet. In this case the talk packet has a valid destination address field. Parsing headers for broadcasting shall be predefined by the upper layer.

6.4.3.3 Continuous data broadcasting

Continuous data broadcasting is performed through a broadcast slot, i.e. the unit performs slot clearing after broadcast slot allocation to continuously broadcast data.

6.4.4 Data exchange

6.4.4.1 General

There are two methods of data exchange: exchange using a talk slot and exchange using a broadcast slot.

6.4.4.2 Data exchange using talk slots

6.4.4.2.1 One-to-one data exchange

6.4.4.2.1.1 General

The unit can establish one or a plurality of one-to-one data links.

6.4.4.2.1.2 Single link

A unit can send and receive data using the destination address of another unit in the manner described in 6.3.4. In the case of a single link, there is no parsing block that requests to talk with the other unit except the unit of destination address in the talk packet.

In addition, the unit can transmit a talk request using the talk slot, and the destination unit can transmit a talk response using the broadcast slot. In this case, the talk request unit shall include a parsing block '0x81' in the talk packet. At this time, the parsing field RequestMethod is set to '0', and response clearing is not performed. If the parsing field RequestMethod is set to '1', the unit performs response clearing.

The destination unit that has received the talk packet containing the parsing block '0x81' shall transmit a response through the broadcast slot. At this time, if the parsing field RequestMethod is set to '1', the unit shall respond with the same response through the talk slot.

The parsing block that performs the response is associated with the application service and is defined at the upper layer.

6.4.4.2.1.3 Multiple link

A unit can send and receive data using the destination address of another unit in the manner described in 6.3.4, and at the same time, by using PBs, transmit information to other units that have the different destination addresses with the packet destination address. These PBs are mainly designed by upper layer.

In this case, the unit pointed by a PB cannot transmit the response information using the talk slot. The unit shall unilaterally receive the information or give a response through the broadcast slot occupied by itself. When responding with a broadcast slot, the related parsing block shall be used. The related parsing block is determined at the upper layer.

6.4.4.3 Data exchange using broadcast slots

The unit can send information to a specific unit by using a parsing block in a broadcast slot occupied by the unit. This parsing block contains the destination address and has the purpose for sending information to another unit.

When the unit of the destination address included in the parsing block receives the broadcast packet including the parsing block, the receiving unit generally responds using the broadcast slot occupied by the receiving unit. In this case, a parsing block including a destination address is similarly used.

If the upper layer requests a response by allocating a talk slot, the unit allocates a talk slot and then transmits a response including a 0x85 parsing header immediately after the trust field in the allocated talk slot. The detailed method for this is decided by the upper layer.

6.4.5 Interworking with CC and VC

6.4.5.1 General

If the SC DLL, CC DLL and VC DLL exist separately, the SC DLL, CC DLL and VC DLL shall interwork with each other. If the SC DLL, CC DLL and VC DLL are integrated into one DLL, the interworking can be done in one DLL, which is implementation dependent.

6.4.5.2 Interworking with CC

When the SC DLL receives CCtoSC.ReqNegoCSCH from the CC DLL, it shall perform subchannel negotiation according to ISO/IEC 4005-3: 2023, 6.3.2. When the SC DLL receives CCtoSC.PBTxReq from the CC DLL, it generates TslotTxEvent and shall notify SCtoCC.PBTxNoti of the success or failure of transmission. When the SC DLL receives the control channel related packet, it shall transmit it to the CC DLL using SCtoCC.PBRxFwd.

6.4.5.3 Interworking with VC

When SC DLL receives VCtoSC.ReqNegoVSCH from VC DLL, it shall perform subchannel negotiation according to ISO/IEC 4005-4:2023, 6.3.2. When SC DLL receives VCtoSC.PBTxReq from VC DLL, it generates TslotTxEvent and shall notify SCtoVC.PBTxNoti of success or failure of transmission. When the SC DLL receives a video channel related packet, it shall transmit it to the VC DLL using SCtoVC.PBRxFwd.

6.5 Synchronization

A unit shall know the date, hour, minute, second, slot number for synchronization. When the number of extra data bits in the broadcast slot is greater than or equal to 39, the A sync units shall transmit a parsing header '0x82' and an associated parsing field. A sync unit shall transmit a parsing header '0x83' and a related parsing field if the number of extra data bits in the broadcast slot is less than 39 and greater than or equal to 17.

The B sync unit shall know the date, hour, minute, second, slot number in order to occupy the slot.

If the B sync unit does not know the current slot number, transmission shall not be performed.

If the B synchronization state unit knows the current slot number, but lacks time information such as date, hour, minute, and second, only one-time transmission is possible. In this case, the B sync state unit can transmit a packet including the parsing header '0x84' requesting current time information to the nearby units. When receiving the parsing header '0x84', the units knowing all time information can transmit the parsing header '0x82' and an associated parsing field when transmitting the next data slot. However, this is not mandatory, but recommended.

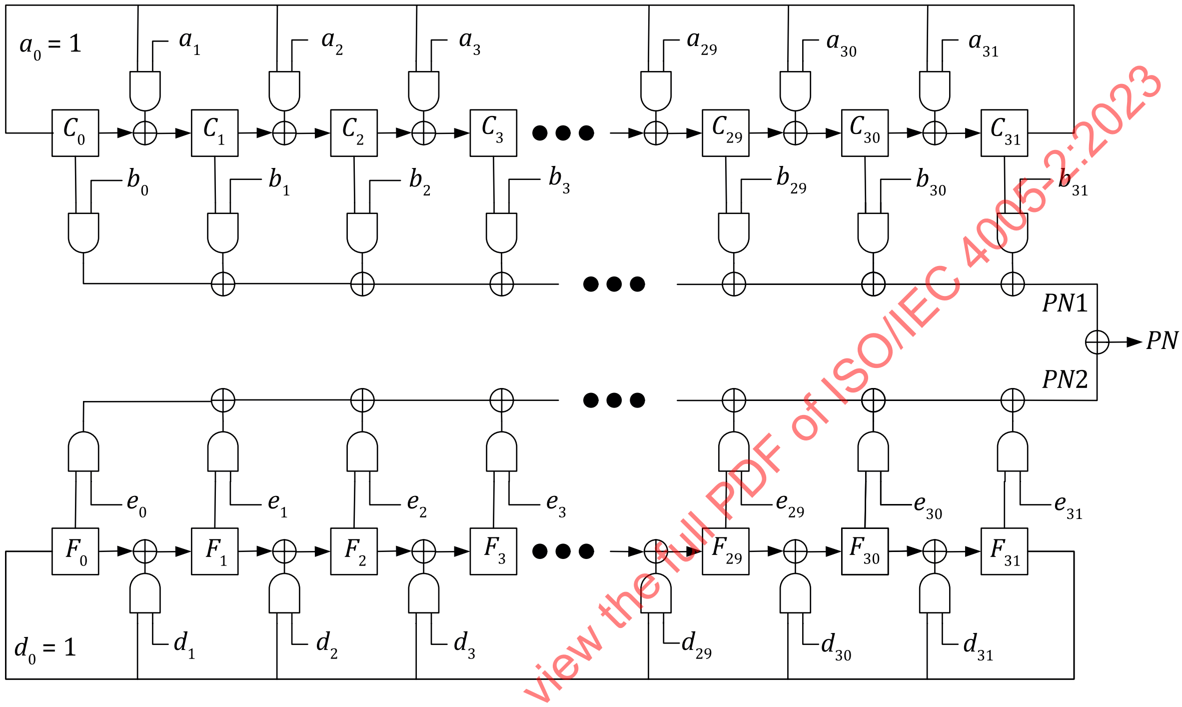
The C sync unit can broadcast data by using the broadcast slot occupied previously.

6.6 Data link layer security

In shared communication, units broadcast information or perform one-to-one communication. Generally, broadcasting does not require security, but security can be applied depending on the service in one-to-one communication.

If the upper layer requests the use of data link security, then the following security shall be used.

The data link layer receives the security key U from the upper layer. The data field, as shown in Figure 25, is scrambled by the PN code generated by the PN generator as follows.



Key

a_k	a k-th bit for AND operation with C_{31}
b_k	a k-th bit for AND operation with C_k
C_k	a k-th register of upper LFSR
F_k	a k-th register of lower LFSR
d_k	a k-th bit for AND operation with F_{31}
e_k	a k-th bit for AND operation with F_k
$PN1$	output of upper LFSR
$PN2$	output of lower LFSR
PN	output of PN code generator

Figure 25 — PN code generator used for data scrambling

$[a_{31}, a_{30}, ..., a_0] = [\mathrm{U}_{26}, \mathrm{U}_{25}, s_8, \mathrm{U}_{24}, ..., \mathrm{U}_{16}, s_6, \mathrm{U}_{15}, ..., \mathrm{U}_8, s_3, \mathrm{U}_7, ..., \mathrm{U}_0, s_0, 1]$

$[b_{31}, b_{30}, ..., b_0] = [\mathrm{U}_{55}, ..., \mathrm{U}_{47}, f_5, \mathrm{U}_{46}, ..., \mathrm{U}_{37}, f_3, \mathrm{U}_{36}, ..., \mathrm{U}_{28}, f_0, \mathrm{U}_{27}]$

$[C_{31}(0), C_{30}(0), ..., C_0(0)] = [\mathrm{U}_{77}, ..., \mathrm{U}_{73}, M_5, M_4, M_2, H_4, H_3, H_2, \mathrm{U}_{72}, ..., \mathrm{U}_{61}, Y_3, Y_1, W_3, W_1, \mathrm{U}_{60}, ..., \mathrm{U}_{56}]$

$[d_{31}, d_{30}, ..., d_0] = [\mathrm{U}_{103}, s_7, s_5, \mathrm{U}_{102}, ..., \mathrm{U}_{93}, s_4, \mathrm{U}_{92}, ..., \mathrm{U}_{85}, s_2, \mathrm{U}_{84}, ..., \mathrm{U}_{78}, s_1, 1]$

$[e_{31}, e_{30}, ..., e_0] = [\mathrm{U}_{132}, ..., \mathrm{U}_{126}, f_4, \mathrm{U}_{125}, ..., \mathrm{U}_{114}, f_2, \mathrm{U}_{113}, ..., \mathrm{U}_{105}, f_1, \mathrm{U}_{104}]$

$[F_{31}(0), F_{30}(0), ..., F_0(0)] =[\mathrm{U}_{154}, ..., \mathrm{U}_{150}, M_3, M_1, M_0, H_1, H_0, \mathrm{U}_{149}, ..., \mathrm{U}_{138}, Y_4, Y_2, Y_0, W_2, W_0, \mathrm{U}_{137}, ..., \mathrm{U}_{133}]$

$$PN1(clk) = (C_0(clk)\ \&\ b_0)\ \hat{}\ (C_1(clk)\ \&\ b_1)\ \hat{}\ \ldots\ \hat{}\ (C_{31}(clk)\ \&\ b_{31})$$

$$PN2(clk) = (F_0(clk)\ \&\ e_0)\ \hat{}\ (F_1(clk)\ \&\ e_1)\ \hat{}\ \ldots\ \hat{}\ (F_{31}(clk)\ \&\ e_{31})$$

$$PN(clk) = PN1(clk)\ \hat{}\ PN2(clk) \tag{25}$$

where

$C_x(clk)$	is a *x*-th bit of upper shift register *C* at *clk*;
$F_x(clk)$	is a *x*-th bit of lower shift register *F* at *clk*;
U_x	is a *x*-th bit of a security key provided by the upper layer;
f_x	is a *x*-th bit of frame number, the number of *f* bits is 6;
s_x	is a *x*-th bit of slot number, the number of *s* bits is 9;
M_x	is a *x*-th bit of minute, the number of *M* bits is 6;
H_x	is a *x*-th bit of hour for 24-hour clock, the number of *H* bits is 5 and *H* has the value from 0 to 23;
Y_x	is a *x*-th bit of date, with a value from 1 to 31, the number of *Y* bits is 5;
W_x	is a *x*-th bit of month, with a value from 1 to 12, the number of *W* bits is 4;
$PN(clk)$	is a pseudo random bit of PN code generator at *clk*, and *PN*(0) is the *PN* value when the initial value is loaded into the two PN code generation registers.

The first bit of the data field is scrambled with $PN(O_{\mathrm{offset}})$. Subsequent data bits are also scrambled sequentially, where O_{offset} is $[\mathrm{U}_{159}, \mathrm{U}_{158}, \ldots, \mathrm{U}_{155}]$.

The padding block in data field is not scrambled.

6.7 Interface with upper layer

6.7.1 General

The interface with the upper layer consists of '00' and the interface header and the following parameters as shown in Figure 26. The number of bits of the header is 8 bits, and the number of parameters and the number of bits of each parameter are different for each interface header.

00	Header	Parmeter1	Parmeter2	●●●	ParmeterN

Figure 26 — Interface packet structure with upper layer

6.7.2 Initialization interface

6.7.2.1 General

The initialization interfaces are used when the unit is initialized. The upper layer can use these interfaces shown in Table 20 after initialization if necessary.

Table 20 — Initialization interface list

Interface name	Header value	Direction
UPtoDL.InfoSharedChannel	0x01	to DL
UPtoDL.InfoSecurity	0x02	to DL
UPtoDL.InfoApprovedSlotMap	0x03	to DL
UPtoDL.InfoBslotMap	0x04	to DL
UPtoDL.InfoTslotMap	0x05	to DL
UPtoDL.InfoSuperFrameSlot	0x06	to DL
UPtoDL.InfoIndividualSlotTxPwr	0x07	to DL
UPtoDL.InfoDedicatedSlot	0x08	to DL

6.7.2.2 UPtoDL.InfoSharedChannel

Table 21 shows the parameters of UPtoDL.InfoSharedChannel.

Table 21 — UPtoDL.InfoSharedChannel Parameter

Parameter name	bits	Description
PRXtoneCompeteThre	5	Power threshold for carrier sensing, -115dBm(0) ~ -84dBm(31)
PRXcollsiontoneThre0	5	Power threshold for collision tone sensing, -115dBm(0) ~ -84dBm(31)
PRXcollsiontoneThre1	5	Power threshold for collision tone sensing, -115dBm(0) ~ -84dBm(31)
PRXcollsiontoneThre2	5	Power threshold for collision tone sensing, -115dBm(0) ~ -84dBm(31)
PRXnumAvgFrm	3	Number of frames for average in collision detection 1(0)~8(7)
nMax	3	Parameter for usable slot selection
kMax	5	Parameter for broadcast slot allocation
nMaxTalk	4	Parameter for broadcast slot allocation
PTXnormalSCH	5	Normal slot transmission power, -8dBm(0) ~ 23dBm(31)
PTX_SCHTCH_differ	4	Power difference data slot with tone slot, 0 dB(0) ~-15 dB(15)
TschReturn	4	When the return of the slot is determined, the unit returns the collision slot after the TschReturn frame from the current frame.
DataLen	10	Maximum length of data field
SrcAddrLen	6	The length of source address
SrcAddr	40	The unit's own address, Only the lower bits corresponding to the length of SrcAddrLen are meaningful.
DstAddr	40	The counterpart address, Only the lower bits corresponding to the length of SrcAddrLen are meaningful. In the case of a UA, this is the address of the controller, and in the case of a controller, this is the address of the UA. ‘0’ means no counterpart.

6.7.2.3 UPtoDL.InfoSecurity

Table 22 shows the parameters of UPtoDL.InfoSecurity.

Table 22 — UPtoDL.InfoSecurity Parameter

Parameter name	bits	Description
TrustOffset	5	Input offset used for calculating trust field
K	102	Secret key between the trust check system and the unit
U	160	Secret key for data field scrambling

6.7.2.4 UPtoDL.InfoApprovedSlotMap

Table 23 shows the parameters of UPtoDL.InfoApprovedSlotMap.

Table 23 — UPtoDL.InfoApprovedSlotMap Parameter

Parameter name	bits	Description
ApprovedSlotMap	500	'1' is usable and '0' is unusable. Default values are all '1'.

The parameter of UPtoDL.InfoApprovedSlotMap is ApprovedSlotMap. ApprovedSlotMap consists of 500 bits. Bit n indicates permission to use slot n. The data link can only allocate slots marked '1'. In the case of a super frame slot, if the corresponding bit is '1', all super frame slot resources of the corresponding slot number are available.

6.7.2.5 UPtoDL.InfoBslotMap

Table 24 shows the parameter of UPtoDL.InfoBslotMap.

Table 24 — UPtoDL.InfoBslotMap Parameter

Parameter name	bits	Description
BslotMap	500	'1' means broadcast slot and '0' means both broadcast slot and talk slot are possible. Default values are all '0'.

The parameter of UPtoDL.InfoBslotMap is BslotMap. BslotMap consists of 500 bits. Bit n represents the type of slot n. The data link recognizes slots marked with '1' as broadcast slots

6.7.2.6 UPtoDL.InfoTslotMap

Table 25 shows the parameter of UPtoDL.InfoTslotMap.

Table 25 — UPtoDL.InfoTslotMap Parameter

Parameter name	bits	Description
TslotMap	500	'1' means talk slot and '0' means both broadcast slot and talk slot are possible. Default values are all '0'.

The parameter of UPtoDL.InfoTslotMap is TslotMap. TslotMap consists of 500 bits. Bit n represents the type of slot n. The data link recognizes slots marked '1' as talk slots.

6.7.2.7 UPtoDL.InfoSuperFrameSlot

Table 26 shows the parameters of UPtoDL.InfoSuperFrameSlot.

Table 26 — UPtoDL.InfoSuperFrameSlot Parameter

Parameter name	bits	Description
Period	6	Possible values are 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60.
SlotNum	9	Slot number

The parameters of UPtoDL.InfoSuperFrameSlot are Period and SlotNum. SlotNum is a super frame slot number having a period of Period.

6.7.2.8 UPtoDL.InfoIndividualSlotTxPwr

Table 27 shows the parameters of UPtoDL.InfoIndividualSlotTxPwr.

Table 27 — UPtoDL.InfoIndividualSlotTxPwr Parameter

Parameter name	bits	Description
SlotNum	9	Slot number where power value is applied.
SlotTxPwr	6	(-20 + SlotTxPwr)dBm

The parameters of UPtoDL.InfoIndividualSlotTxPwr are SlotNum and SlotTxPwr. The transmit power of a slot SlotNum is (-20 + SlotTxPwr)dBm.

6.7.2.9 UPtoDL.InfoDedicatedSlot

Table 28 shows the parameters of UPtoDL. InfoDedicatedSlot.

Table 28 — UPtoDL.InfoDedicatedSlot Parameter

Parameter name	bits	Description
SlotNum	9	Dedicated slot number
SlotTxPwr	6	Dedicated slot transmission Power, (-20 + SlotTxPwr)dBm

The parameters of UPtoDL.InfoDedicatedSlot are SlotNum and SlotTxPwr. The transmit power of a dedicated slot SlotNum is (-20 + SlotTxPwr)dBm.

6.7.3 Dynamic Interface

6.7.3.1 General

The dynamic interfaces shown in Table 29 are used after unit initialization when a corresponding event occurs.

Table 29 — Dynamic interface list

Interface name	Header value	Direction
UPtoDL.ReqGetSCH	0x11	to DL
DLtoUP.NotiGetSCH	0x12	to UP
UPtoDL.ReqTxTslot	0x13	to DL
UPtoDL.ReqTxBslot	0x14	to DL
DLtoUP.NotiTxTslot	0x15	to UP

Table 29 *(continued)*

Interface name	Header value	Direction
DLtoUP.RsvSCHData	0x16	to UP
UPtoDL.ReqReturnSCH	0x17	to DL
UPtoDL.ResponseACK	0x18	to DL
DLtoUP.ResponseACK	0x19	to UP

6.7.3.2 UPtoDL ReqGetSCH

Table 30 shows the parameter of UPtoDL.ReqGetSCH.

Table 30 — UPtoDL.ReqGetSCH Parameter

Parameter name	bits	Description
IsOccupy	1	‘1’ means occupation, ‘0’ means one-time transmission.

The parameter of UPtoDL.ReqBslotAllocate is IsOccupy. If IsOccupy is '1', it means that slot clearing after slot allocation is performed to continuously occupy the slot, and if it is '0', it means one-time allocation.

6.7.3.3 DLtoUP.NotiGetSCH

Table 31 shows the parameters of DLtoUP.NotiGetSCH.

Table 31 — DLtoUP.NotiGetSCH Parameter

Parameter name	bits	Description
IsSuccess	1	‘1’ means success. ‘0’ means allocation fail or slot return by slot collision.
k	5	k value at the time this interface is reported
SlotNum	9	Allocated or return slot number, 0x1FF means one-time allocation success.

6.7.3.4 UPtoDL.ReqTxTslot

Table 32 shows the parameters of UPtoDL.ReqTxTslot.

Table 32 — UPtoDL.ReqTxTslot Parameter

Parameter name	bits	Description
DestAddr	40	The destination address, only the lower bits corresponding to the length of SrcAddrLen in UPtoDL.InfoSharedChannel are meaningful. ‘0’ means no counterpart.
SlotNum	9	Transmission slot number, 0x1FF means random slot number.
ResponseClearing	1	‘1’ – Performing response clearing ‘0’ – Not performing
DesignatedSlot	1	‘1’ – Transmission in designated slot ‘0’ – Transmission in normal slot
SlotNum	9	Response slot number, meaningful only when DesignatedSlot is ‘1’

Table 32 *(continued)*

Parameter name	bits	Description
DataSecurity	2	'00' – No security '01' – Data scrambling with PN code '10', '11' - reserved
DataLen	10	DataLen is the length of Tx data.
Data	DataLen	Tx data

6.7.3.5 UPtoDL.ReqTxBslot

Table 33 shows the parameters of UPtoDL.ReqTxBslot.

Table 33 — UPtoDL.ReqTxBslot Parameter

Parameter name	bits	Description
SlotNum	9	Slot number for data transmission
DataLen	10	The length of Tx data
Data	DataLen	Tx data

6.7.3.6 DLtoUP.NotiTxTslot

Table 34 shows the parameters of DLtoUP.NotiTxTslot.

Table 34 — DLtoUP.NotiTxTslot Parameter

Parameter name	bits	Description
IsSuccess	1	'1' means success. '0' means allocation fail.
SlotNum	9	Slot number where packet is transmitted

6.7.3.7 DLtoUP.RsvSCHData

Table 35 shows the parameters of DLtoUP.RsvSCHData.

Table 35 — DLtoUP.RsvSCHData Parameter

Parameter name	bits	Description
SrcAddr	40	Only the lower bits corresponding to the length of SrcAddrLen are meaningful. '0' means broadcast slot that has no SrcAddr.
DataLen	10	The length of Rx data.
Data	DataLen	Rx data

6.7.3.8 UPtoDL.ReqReturnSCH

Table 36 shows the parameter of UPtoDL.ReqReturnSCH.

Table 36 — UPtoDL.ReqReturnSCH Parameter

Parameter name	bits	Description
SlotNum	9	Slot number for return

6.7.3.9 UPtoDL.ResponseACK

Table 37 shows the parameters of UPtoDL.ResponseACK.

Table 37 — UPtoDL.ResponseACK Parameter

Parameter name	bits	Description
Ack	1	‘1’ is ACK, ‘0’ is NACK.
InterfaceHeader	8	Interface header received from DLL

When the DLL sends an interface packet to the upper layer, the upper layer can respond using UPtoDL. ResponseACK. The use of this interface is implementation dependent.

6.7.3.10 DLtoUP.ResponseACK

Table 38 shows the parameters of DLtoUP.ResponseACK.

Table 38 — DLtoUP.ResponseACK Parameter

Parameter name	bits	Description
Ack	1	‘1’ is ACK, ‘0’ is NACK.
InterfaceHeader	8	Interface header received from upper layer

When the upper layer sends an interface packet to the DLL, the DLL can respond with DLtoUP. ResponseACK. The use of this interface is implementation dependent.

6.8 Interface with other communication layer

6.8.1 General

This document describes the interface between SC DLL and CC DLL and VC DLL. This interface is described under the assumption that each DLL is independent. three DLLs can be integrated and operated as a single DLL. It is implementation dependent. When integrated into a single DLL, the use of the interface below is implementation dependent.

6.8.2 Interface with CC

6.8.2.1 General

The interfaces with CC DLL as shown in Table 39 are used for some reason like CSCH negotiation. The upper layer can designate interfaces for multiple purposes.

Table 39 — Interface list with CC

Interface name	Header value	Direction
CCtoSC.ReqNegoCSCH	0x40	to SC
SCtoCC.NotiNegoCSCH	0x41	to CC
CCtoSC.PBTxReq	0x42	to SC
SCtoCC.PBTxNoti	0x43	to CC
SCtoCC.PBRxFwd	0x44	to CC

6.8.2.2 CCtoSC.ReqNegoCSCH

CCtoSC.ReqNegoCSCH has four parameters as shown in Table 40, the meaning of each parameter is as follows.

Table 40 — Parameters of CCtoSC.ReqNegoCSCH

Parameter name	bits	Description
UsePB0x81	1	‘1’ means PB 0x81 use, ‘0’ means PB 0x81 not use.
RequestMethod	1	This value is used only when UsePB0x81 is ‘1’. ‘1’ – SC Bslot & Tslot use. ‘0’ – SC Bslot use.
m	3	The range of -4(0) ~ 3(7)
n	3	The range of -4(0) ~ 3(7)

6.8.2.3 SCtoCC.NotiNegoCSCH

SCtoCC.NotiNegoCCH has five parameters as shown in Table 41, the meaning of each parameter is as follows.

Table 41 — Parameters of SCtoCC.NotiNegoCCH

Parameter name	bits	Description
IsSuccess	1	‘1’ means success, ‘0’ means fail.
m	3	This value is meaningful when IsSuccess is ’0’. The value of m when negotiation is finished. The range of -4(0) ~ 3(7)
n	3	This value is meaningful when IsSuccess is ’0’. The value of m when negotiation is finished. The range of -4(0) ~ 3(7)
ChannelNum	5	This value is meaningful when IsSuccess is ’1’. Frequency channel number of an allocated channel
SubchannelNum	5	This value is meaningful when IsSuccess is ’1’. Subchannel number of the allocated ChannelNum

6.8.2.4 CCtoSC.PBTxReq

CCtoSC.PBTxReq has two parameters as shown in Table 42. the meaning of each parameter is as follows.

Table 42 — Parameters of CCtoSC.PBTxReq

Parameter name	bits	Description
PBLength	10	PB length
PB	PBLength	Parsing block for Tx

6.8.2.5 SCtoCC.PBTxNoti

SCtoCC.PBTxNoti has one parameter as shown in Table 43, the meaning of this parameter is as follows.

Table 43 — Parameters of CCtoSC.PBTxNoti

Parameter name	bits	Description
IsSuccess	1	‘1’ means PB transmission success, ‘0’ means PB transmission fail.

6.8.2.6 SCtoCC.PBRxFwd

SCtoCC.PBRxFwd has two parameters as shown in Table 44, the meaning of each parameter is as follows.

Table 44 — Parameters of SCtoCC.PBRxFwd

Parameter name	bits	Description
PBLength	10	PB length
PB	PBLength	Parsing block received in SC

6.8.3 Interface with VC

6.8.3.1 General

The interfaces with VC DLL as shown in Table 45 are used for some reason like VSCH negotiation. The upper layer can designate interfaces for multiple purposes.

Table 45 — Interface list with CC

Interface name	Header value	Direction
VCtoSC.ReqNegoVSCH	0x45	to SC
SCtoVC.NotiNegoVSCH	0x46	to VC
VCtoSC.PBTxReq	0x47	to SC
SCtoVC.PBTxNoti	0x48	to VC
SCtoVC.PBRxFwd	0x49	to VC

6.8.3.2 VCtoSC.ReqNegoVSCH

VCtoSC.ReqNegoVSCH has four parameters as shown in Table 46. The meaning of each parameter is as follows.

Table 46 — Parameters of VCtoSC.ReqNegoVSCH

Parameter name	bits	Description
UsePB0x81	1	‘1’ means PB 0x81 use, ‘0’ means PB 0x81 not use.
RequestMethod	1	This value is used only when UsePB0x81 is ‘1’. ‘1’ – SC Bslot & Tslot use. ‘0’ – SC Bslot use.
m	3	The range of -4(0) ~ 3(7)
n	3	The range of -4(0) ~ 3(7)

6.8.3.3 SCtoVC.NotiNegoVSCH

SCtoVC.NotiNegoVSCH has five parameters as shown in Table 47. The meaning of each parameter is as follows.

Table 47 — Parameters of SCtoVC.NotiNegoVSCH

Parameter name	bits	Description
IsSuccess	1	‘1’ means success, ‘0’ means fail.