ATSC Standard: Dedicated Return Channel for ATSC 3.0 (A/323)

Doc. A/323:2018
7 December 2018

Advanced Television Systems Committee
1776 K Street, N.W.
Washington, D.C. 20006
202-872-9160

The Advanced Television Systems Committee, Inc., is an international, non-profit organization developing voluntary standards and recommended practices for digital television. ATSC member organizations represent the broadcast, broadcast equipment, motion picture, consumer electronics, computer, cable, satellite, and semiconductor industries. ATSC also develops digital television implementation strategies and supports educational activities on ATSC standards. ATSC was formed in 1983 by the member organizations of the Joint Committee on Inter-society Coordination (JCIC): the Electronic Industries Association (EIA), the Institute of Electrical and Electronic Engineers (IEEE), the National Association of Broadcasters (NAB), the National Cable Telecommunications Association (NCTA), and the Society of Motion Picture and Television Engineers (SMPTE). For more information visit www.atsc.org.

Note: The user's attention is called to the possibility that compliance with this standard may require use of an invention covered by patent rights. By publication of this standard, no position is taken with respect to the validity of this claim or of any patent rights in connection therewith. One or more patent holders have, however, filed a statement regarding the terms on which such patent holder(s) may be willing to grant a license under these rights to individuals or entities desiring to obtain such a license. Details may be obtained from the ATSC Secretary and the patent holder.

Implementers with feedback, comments, or potential bug reports relating to this document may contact ATSC at https://www.atsc.org/feedback/.

Revision History

Version	Date
Candidate Standard approved Updated CS approved	2 November 2017 30 May 2018
Standard approved	7 December 2018

Table of Contents

Index of Figures and Tables

ATSC Standard:
Dedicated Return Channel for ATSC 3.0

1. SCOPE

This standard specifies the physical layer and the Medium Access Control (MAC) layer for the ATSC 3.0 Dedicated Return Channel (DRC). The DRC physical layer specifies the uplink framing, baseband signal generation, random access, and downlink synchronization scheme. The DRC MAC layer specifies the MAC procedures, MAC PDU formats, and signaling schemes between the ATSC 3.0 downlink and DRC.

1.1 Introduction and Background

The radical shift towards mobile screens and wireless rich media has posed a pressing need for innovative broadcasting services and a new generation of enabling technologies. To date, terrestrial broadcasting remains one of the most efficient means to deliver massive amounts of information to large numbers of users. On the other hand, conventional linear TV services alone (albeit ultra-high-definition) may not be sufficient to sustain the terrestrial broadcasting business which requires a large amount of highly coveted spectrum resources. Intelligent media delivery and flexible service models that maximize the network Return on Investment (ROI) is of paramount importance to the broadcasting industry in the new era.

Recent studies have shown that interactivity between media customers and service providers and between users themselves will be one of the most important features in the next-generation media service. In this document, this unique opportunity is addressed by defining a Dedicated Return Channel (DRC) system for the next-generation broadcasting system.

In this document, both the physical layer and MAC (Media Access Control) layer for the ATSC 3.0 DRC (a.k.a uplink) are specified.

1.2 Organization

This document is organized as follows:

- Section 1 – Outlines the scope of this document and provides a general introduction.
- Section 2 – Lists references and applicable documents.
- Section 3 – Provides definitions of terms, acronyms, and abbreviations for this document.
- Section 4 – System overview
- Section 5 – PHY specification
- Section 6 – MAC specification
- Section 7 – Definition of DRCT
- Annex A – Description of PRACH
- Annex B – Synchronization error analysis
- Annex C – Lists the physical layer parameters of PLP-R
- Annex D – Signaling overhead

2. REFERENCES

All referenced documents are subject to revision. Users of this Standard are cautioned that newer editions might or might not be compatible.

2.1 Normative References

The following documents, in whole or in part, as referenced in this document, contain specific provisions that are to be followed strictly in order to implement a provision of this Standard.

[1] IEEE: "Use of the International Systems of Units (SI): The Modern Metric System," Doc. SI 10-2002, Institute of Electrical and Electronics Engineers, New York, N.Y.

[2] ATSC: "ATSC 3.0 Standard: Physical Layer Protocol," Doc. A/322:2017, Advanced Television Systems Committee, Washington, D.C., 6 June 2017.

[3] ATSC: "ATSC3.0 Standard: Scheduler and Studio-Transmitter Link," Doc. A/324:2018, Advanced Television System Committee, Washington, D.C., 5 January 2018.

[4] ATSC: "ATSC 3.0 Standard: Signaling, Delivery, Synchronization, and Error Protection," Doc. A/331:2017, Advanced Television System Committee, Washington, D.C., 6 December 2017.

[5] ATSC: "ATSC 3.0 Standard: ATSC 3.0 Security and Service Protection," Doc. A/360:2018, Advanced Television System Committee, Washington, D.C., 9 January 2018.

[6] ATSC: "ATSC 3.0 Standard: Link-Layer Protocol," Doc. A/330:2016, Advanced Television System Committee, Washington, D.C., 19 September 2016.

[7] ATSC: "ATSC 3.0 Standard: A/321, System Discovery and Signaling," Doc. A/321:2016, Advanced Television System Committee, Washington, D.C., 23 March 2016.

[8] IETF: RFC 3986, "Uniform Resource Identifier (URI): Generic Syntax," Internet Engineering Task Force, Reston, VA, January, 2005. http://tools.ietf.org/html/rfc3986

[9] W3C: "XML Schema Part 2: Datatypes Second Edition," W3C Recommendation, Worldwide Web Consortium, 28 October 2004. https://www.w3.org/TR/xmlschema-2/

[10] IETF: RFC 6726, "FLUTE – File Delivery over Unidirectional Transport," Internet Engineering Task Force, Reston, VA, November, 2012. http://tools.ietf.org/html/rfc6726

[11] "Universal Mobile Telecommunications System (UMTS); LTE; Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs (3GPP TS 26.346 version 13.3.0 Release 13)," Doc. ETSI TS 126 346 v13.3.0 (2016-01), European Telecommunications Standards Institute, 2014.

2.2 Informative References

The following documents contain information that may be helpful in applying this Standard.

[12] ATSC: "ATSC 3.0 Standard: ATSC 3.0 System," Doc. A/300:2017, Advanced Television System Committee, Washington, D.C., 19 October 2017.

[13] ATSC: "ATSC 1.0 Standard: Interactive Services Standard", Doc. A/105:2015, Advanced Television Systems Committee, Washington, D.C., 29 October 2015.

[14] Digital Video Broadcasting (DVB): "Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)", ETSI EN 302 755 V1.4.1, July 2015.

[15] Digital Video Broadcasting (DVB): "Interaction channel for Digital Terrestrial Television (RCT) incorporating Multiple Access OFDM," ETSI EN 301 958 V1.1.1, March 2002.

[16] ATSC: "Code Point Registry", Advanced Television System Committee, Washington, D.C., https://www.atsc.org/techdoc/code-point-registry/.

3. DEFINITION OF TERMS

With respect to definition of terms, abbreviations, and units, the practice of the Institute of Electrical and Electronics Engineers (IEEE) as outlined in the Institute's published standards [1] shall be used. Where an abbreviation is not covered by IEEE practice or industry practice differs from IEEE practice, the abbreviation in question will be described in Section 3.3 of this document.

3.1 Compliance Notation

This section defines compliance terms for use by this document:

shall – This word indicates specific provisions that are to be followed strictly (no deviation is permitted).

shall not – This phrase indicates specific provisions that are absolutely prohibited.

should – This word indicates that a certain course of action is preferred but not necessarily required.

should not – This phrase means a certain possibility or course of action is undesirable but not prohibited.

3.2 Treatment of Syntactic Elements

This document contains symbolic references to syntactic elements used in the audio, video, and transport coding subsystems. These references are typographically distinguished by the use of a different font (e.g., `restricted`), may contain the underscore character (e.g., `sequence_end_code`) and may consist of character strings that are not English words (e.g., `dynrng`).

3.2.1 Reserved Elements

One or more reserved bits, symbols, fields, or ranges of values (i.e., elements) may be present in this document. These are used primarily to enable adding new values to a syntactical structure without altering its syntax or causing a problem with backwards compatibility, but they also can be used for other reasons.

The ATSC default value for reserved bits is '1'. There is no default value for other reserved elements. Use of reserved elements except as defined in ATSC Standards or by an industry standards body is not permitted. See individual element semantics for mandatory settings and any additional use constraints. As currently-reserved elements may be assigned values and meanings in future versions of this Standard, receiving devices built to this version are expected to ignore all values appearing in currently-reserved elements to avoid possible future failure to function as intended.

3.3 Acronyms and Abbreviations

The following acronyms and abbreviations are used within this document.

16QAM	16-ary Quadrature Amplitude Modulation
3GPP	3rd Generation Partnership Project
ACK	Acknowledgement
ALP	ATSC Link-layer Protocol
AMC	Adaptive Modulation and Coding
ARQ	Automatic Repeat-reQuest
ATSC	Advanced Television Systems Committee
BAT	Broadcast Access Terminal

BBP	BaseBand Packet
BCI	Broadcast Control Information
BEB	Binary Exponential Backoff
bps	bits per second
BPSK	Binary Phase Shift Keying
BSID	Broadcast Stream ID
BTS	Broadcast Television Station
CDL	Convolutional Delay Line
CID	Connection ID
CP	Cyclic Prefix
CRC	Cyclic Redundancy Check
CTI	Convolutional Time Interleaver
CTC	Convolutional Turbo Code
dB	decibel
dBm	decibel relative to 1 milliwatt
DC	Direct Current
DFT	Discrete Fourier Transform
DRC	Dedicated Return Channel
DRCT	Dedicated Return Channel Table
FDT	File Delivery Table
FEC	Forward Error Correction
FFT	Fast Fourier Transform
GBR	Guaranteed Bit Rate
GP	Guard Period
GT	Guard Time
HTI	Hybrid Time Interleaver
ID	Identification
IDFT	Inverse Discrete Fourier Transform
IETF	Internet Engineering Task Force
IFFT	Inverse Fast Fourier Transform
IP	Internet Protocol
LDM	Layered Division Multiplexing
LDPC	Low Density Parity Check
LLS	Low Level Signaling
LSB	Least Significant Bit
MAC	Media Access Control
MBMS	Multimedia Broadcast/Multicast Services
MRC	Maximum Retransmission Count
MSB	Most Significant Bit
MTU	Maximum Transmission Unit
NACK	Negative Acknowledgement
OFDM	Orthogonal Frequency Division Multiplexing

OFDMA	Orthogonal Frequency Division Multiple Access
PDU	Protocol Data Unit
PHY	Physical Layer
PLP	Physical Layer Pipe
PLP-R	Physical Layer Pipe for Return Channel
ppm	parts per million
PRACH	Physical Random Access Channel
PUSCH	Physical Uplink Shared Channel
QCI	QoS Class Identifier
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
RFC	Request For Comments
RNTI	Radio Network Temporary Identity
RTC	Retransmission Count
RTP	Real-time Transport Protocol
SC-FDMA	Single Carrier Frequency Division Multiple Access
SINR	Signal to Interference plus Noise Ratio
SN	Sequence Number
SNR	Signal to Noise Ratio
STL	Studio to Transmitter Link
TB	Transport Block
TBI	Twisted Block Interleaver
TS	Transport Stream
TUID	Temporary User ID
UDP	User Datagram Protocol
uimsbf	unsigned integer, most significant bit first
UL-MAP	Uplink Resource MAP
W3C	World Wide Web Consortium
XML	Extensible Markup Language
ZC	Zadoff-Chu

3.4 Terms

The following terms are used within this document.

BAT – An ATSC 3.0 receiver with a DRC terminal module in it, or equivalently a DRC-enabled ATSC 3.0 receiver.

BTS – An ATSC 3.0 transmitter with a DRC base station module in it, or equivalently a DRC-enabled ATSC 3.0 transmitter.

Cell – One pair of I/Q components representing a modulated symbol [2].

DRC downlink – Downlink signaling and downlink data transmission of DRC-related information through the PLP for Return Channel (PLP-R).

DRC uplink – Uplink signaling and uplink data transmission of DRC through the Physical layer and MAC layer specifications defined in this document.

Packet – A collection of data sent as a unit, including a header to identify and indicate other properties of the data, and a payload comprising the data actually to be sent, either having a fixed known length or having means to indicate either its length or its end.

Paging – The BTS pages for a single terminal in the broadcast network.

Protocol Data Unit – The protocol data unit encapsulated in the DRC Uplink MAC layer. The maximum size of a DRC uplink MAC PDU is limited to 2048 bytes.

Random Access – User terminals try to access the system.

reserved – Set aside for future use.

Resource block – A resource block is composed of 3 resource tiles that are contiguous in frequency-time dimension or time-frequency dimension.

Resource tile – Basic unit in an uplink physical frame. One resource tile occupies 20 contiguous subcarriers in the frequency dimension and 2 contiguous symbols in the time dimension, which is equal to 40 cells. Among the cells in one resource tile, 8 of them are used for pilots and 32 are used for data transmission. Each resource tile has an index to represent it, which is numbered according to the position of the resource tile in the uplink frame.

Transport Block – The minimum channel coding block for uplink communications. The size of a transport block is determined by the number of allocated tiles for a user and the coding scheme. Referring to Table 5.5, the maximum number of information bits transmitted by a transport block can be 96, 144, 192, 288, 384, 432, 576, or 864. When the size of a MAC PDU is less than or larger than this limitation, padding or segmentation, respectively, shall be used.

4. SYSTEM OVERVIEW

Dedicated Return Channel (DRC) supports interactive services in ATSC 3.0 without dependence on other non-ATSC 3.0 network infrastructure. In ATSC 3.0, downlink broadcast channel and Dedicated Return Channel for interactive services use different RF frequencies (i.e. Frequency Division Duplexing). The PHY layer and Media Access Control (MAC) layer for DRC are defined in this specification.

4.1 Typical Application

When a line of sight path between BATs and BTS exists, the coverage range of DRC uplink can be up to 100 km (see Annex A). However, in urban areas the coverage range is reduced due to larger path loss. Taking different path losses into account, DRC uplink with relays is viable in DRC system operation.

In the case without using relays, all BATs shall communicate directly with the BTS.

In the case with using relays, the downlink broadcast shall be transmitted directly from the BTS to all BATs. However, the DRC uplink shall be transmitted from BATs to relays, and then relay stations shall forward the received signal to the BTS through high-speed wired or wireless networks with low latency. Relay stations shall be transparent to all BATs. The forwarded signal from the relay to the BTS should be the raw data coming from Analog-to-Digital conversion or the decoded MAC PDU. The number of relay stations, the scheme by which the relay station forwards signal to the BTS and the scheme by which the BTS utilizes the signal are determined by the network operator. In both cases for BATs with and without using relays, the BTS shall respond to all BATs through the downlink PLP-R only once, even when the BTS receives multiple signal copies for a BAT from multiple relays or from both relays and the BAT itself. How the BTS responds to the BAT is described in Section 4.4.

Two scenarios with and without relays between BTS and BATs are shown in Figure 4.1 and Figure 4.2, respectively.

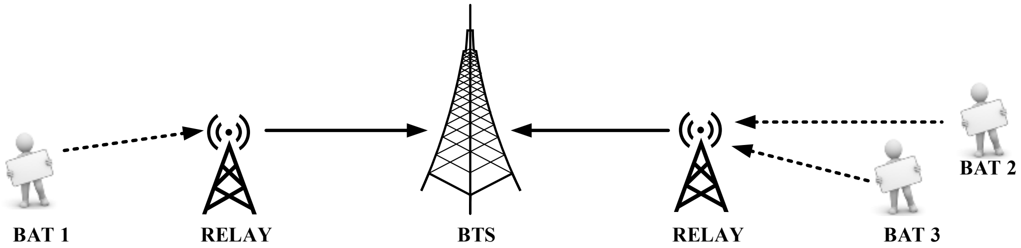

Figure 4.1 Uplink transmission from BATs to BTS through relays.

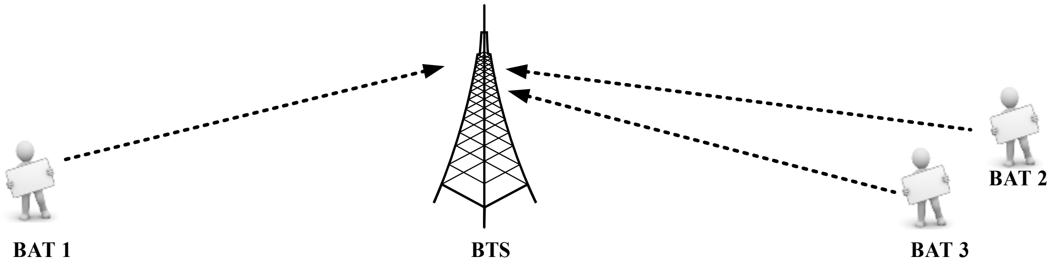

Figure 4.2 Direct uplink transmission from BATs to BTS.

4.2 System Architecture

The system architecture of ATSC 3.0 with DRC is shown in Figure 4.3. A BTS and BATs communicate through wireless channels. A BTS transmits downlink payloads on frequency f_0 and receives DRC uplink payloads on frequency f_1. In contrast, a BAT receives downlink payloads on frequency f_0 and transmits DRC uplink payloads on frequency f_1.

In a BTS, there is a link from the studio(s) to the ATSC 3.0 transmitter through an ATSC 3.0 Downlink Gateway [3]. This Gateway encapsulates ATSC Link-layer Protocol (ALP) packets from studios into BaseBand Packets (BBPs) and sends them to the ATSC 3.0 transmitter. Each PLP configured at the Gateway is mapped to a different port of the Internet Protocol (IP) connection. DRC downlink signaling and data are also transferred to the Gateway in ALP packets with a specific IP port, and are mapped to the designated PLP-R for Return channel application.

The ATSC 3.0 receiver in a BAT receives PLPs and separates DRC synchronization, signaling and DRC related data from traditional broadcast service data. DRC synchronization and signaling data are sent to the DRC uplink gateway for processing and maintaining system operation.

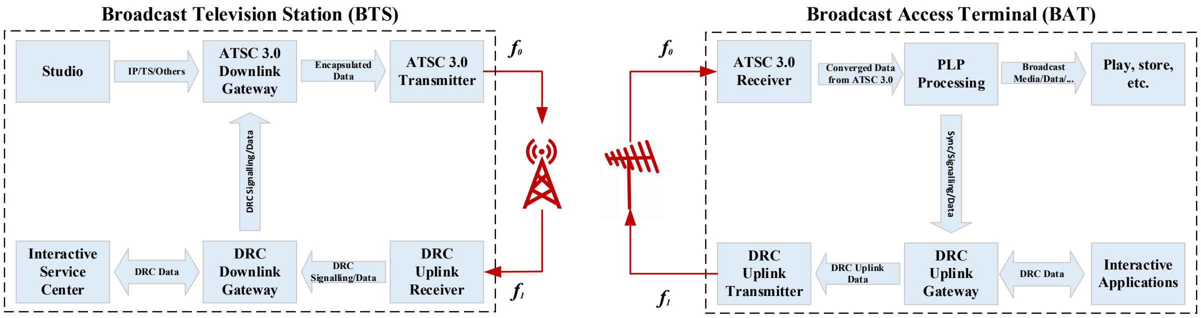

Figure 4.3 System Architecture of ATSC 3.0 with DRC.

4.3 DRC Uplink

The system architecture of the DRC uplink terminal is shown in Figure 4.4. The following modules are included in the DRC uplink terminal: ARQ, Link Adaption and Random Access. The function of each module is described below.

The Automatic Repeat-reQuest (ARQ) module is used for retransmission of lost packets and is defined in Section 6.2.10.

The link adaptation module is used for adaptive modulation and channel coding of DRC uplink.

The random access module is used for the initial access of the DRC uplink terminal when the DRC uplink terminal does not have an established connection with the BTS. This process is also initiated when BATs lose uplink synchronization to the BTS.

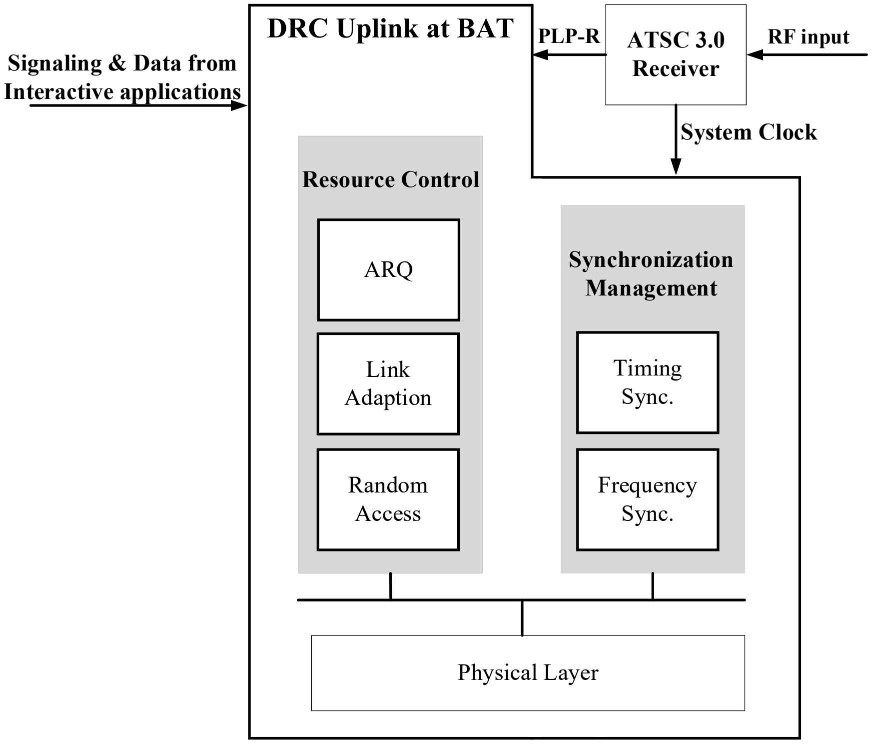

Figure 4.4 System Architecture of the DRC uplink terminal.

Other functions of the DRC uplink system use the existing ATSC 3.0 standards. The transport protocol shall be as specified in [4]. The encryption protocol shall be as specified in [5].

The DRC uplink transmitter uses Single Carrier Frequency Division Multiple Access (SC-FDMA) as the multiple access scheme. SC-FDMA is similar to Orthogonal Frequency Division Multiple Access (OFDMA) except for a Discrete Fourier Transform (DFT) operation performed before the Inverse Fast Fourier Transform (IFFT).

Data flows are input to the uplink transmitter and modulated into complex symbols after going through the Cyclic Redundancy Check (CRC) block and Convolutional Turbo Code (CTC) encoder. BPSK, QPSK and 16QAM are the modulation modes used at the uplink transmitter. N_{DFT} modulated symbols are grouped into symbol blocks. Then an N_{DFT}-point DFT is performed on each symbol block. Subcarrier mapping module maps N_{DFT}-point DFT output symbols to N_{IFFT} orthogonal subcarriers, where N_{IFFT} is the total number of orthogonal subcarriers in the frequency domain.

4.4 Interaction between Broadcast and DRC

To be compatible with the ATSC 3.0 traditional broadcast system without DRC, an indication bit (**L1B_return_channel_flag**) of whether a DRC system is associated with the current downlink broadcast system shall be included in the downlink physical layer (L1) signalling [2] Section 9.2.1. By default **L1B_return_channel_flag** = 0 to indicate that DRC is not supported in the current downlink frame of the current frequency band and current broadcast network. When **L1B_return_channel_flag** = 1, it indicates that DRC is supported in the current downlink frame of the current frequency band and current broadcast network.

Synchronization and resource scheduling among all users are required to realize multiple access. Furthermore, the downlink broadcast service and the DRC in ATSC 3.0 use different RF frequencies through Frequency Division Duplexing.

One special downlink PLP type, named PLP-R, is defined to carry downlink data and signalling that support operation of the Dedicated Return Channel. PLP-R in a downlink frame is allocated by the System Scheduler according to A/324 [3]. Data in PLP-R is fed into the scheduler in the same way as other ATSC 3.0 downlink traffic. Formats of the data payload in PLP-R are defined in Section 6. Other higher-layer headers for data transfer, such as IP header and UDP header, shall conform to A/331 [4].

The physical layer parameters of PLP-R shall be carefully selected to guarantee that all BATs in the expected coverage of the BTS can receive and decode PLP-R correctly. Informative guidelines for selection of physical layer parameters of PLP-R are given in Annex C.

Only one PLP-R should be used in DRC. The mapping of one PLP with type PLP-R is shown in Figure 4.5. It has been stated in Section 5.1.1 of A/322 [2] that ‘The maximum number of PLPs in each RF channel (6, 7 or 8 MHz) shall be 64. The minimum number of PLPs in an RF channel shall be one. The maximum number of PLPs in a frame carrying content requiring simultaneous recovery to assemble a single delivered product shall be four, subject to the constraints described in Section 7.1.2’. When DRC is used, PLP-R shall be considered to be one of the PLPs requiring simultaneous recovery to assemble any single delivered product within the downlink RF channel.

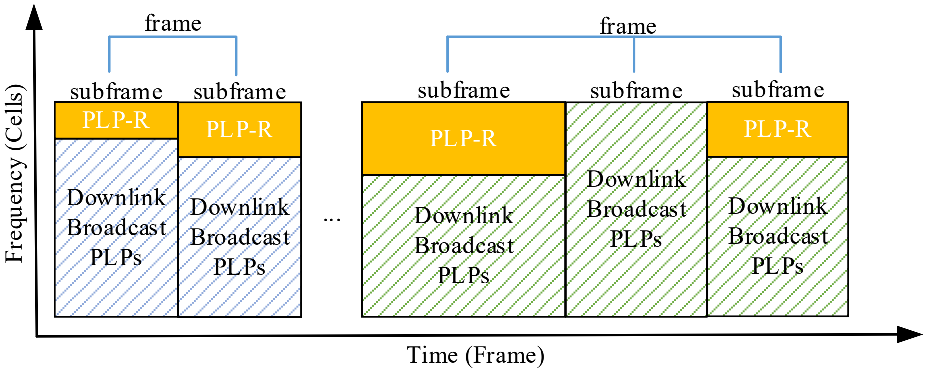

Figure 4.5 Mapping of PLP-R.

An example resource allocation scheme for PLP-R is shown in Figure 4.6. Three kinds of information may be sent in PLP-R, i.e., Broadcast Control Information (BCI), DRC Downlink MAC Protocol Data Units (PDUs) and Uplink MAP (UL-MAP).

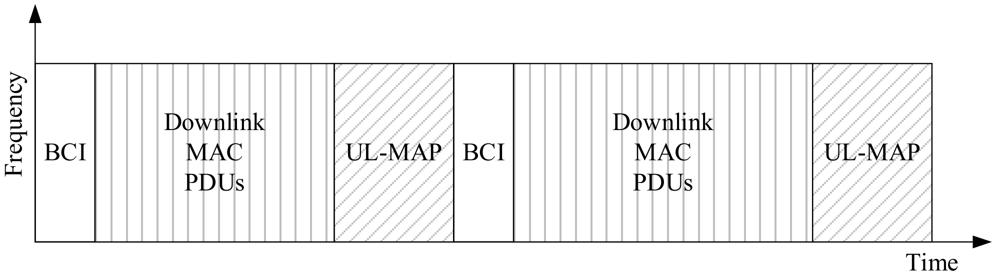

Figure 4.6 Illustration of resource allocation in PLP-R.

Considering the tradeoff between resource granularity and scheduling overhead, the uplink frame length for the DRC uplink is defined as 10ms. The response time from the BTS following an uplink transmission by a BAT depends on how PLP-R is reserved in the ATSC 3.0 transmitter. When the resource for PLP-R is reserved along the entire frame, BATs can obtain a response quickly. Otherwise, when PLP-R is only allocated in a subset of the subframes in a frame, the response time is at least the time duration between two successive subframes with allocated PLP-R.

The structure of the Studio to Transmitter Link (STL) and the physical layer of the downlink system with DRC is the same as that without DRC. The only difference is that, with DRC, a DRC-related downlink signaling is fed into the ATSC Link-layer Protocol (ALP) module beside existing traditional broadcasting service data sources. The high-level overview of STL with DRC is shown in Figure 4.7.

As explained in Section 5, the physical resource of DRC is organized in uplink frames with frame size equal to 10ms. The control signaling of uplink resource allocation in UL-MAP is identified by uplink frame index, referring to Section 6.5.2. Since control signaling of DRC is carried in PLP-R, it shall be guaranteed that the control signaling arrives at specific BATs before the uplink frames designated in the control signaling start.

To guarantee the arrival time at the receiver, the propagation delay between the DRC service center and the exciter needs to be known. Let d and δ denote the average and the variance of the propagation delay between the DRC service center and the exciter; let t denote the current time; then only uplink frames starting later than $t + d + \alpha\delta$ can be included in the UL-MAP, where $\alpha\delta$ is the time redundancy taking delay variation into consideration.

The values of d and δ can be tested offline or online depending on the system operator. When d and δ are tested offline, a larger α should be used because the delay offset may become larger with time. When d and δ are tested online, a smaller α should be used because the delay error can be compensated as time progresses. The exact value of α depends on different systems and shall be determined by the system operator.

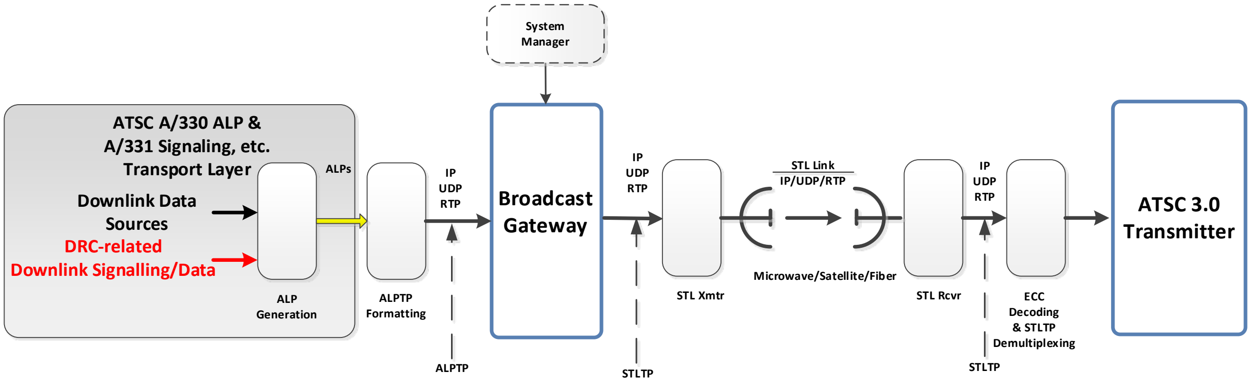

Figure 4.7 High-level overview of Downlink STL with DRC.

The identification of the service type carried in a PLP (**LLS_table_id**) is contained in Low Level Signalling (LLS) table in A/331 [4]. In [16], a service type for DRC, i.e., Dedicated Return Channel Table (**DRCT**) is defined as case 0x07. The syntax of a **DRCT** table is defined in Section 7 of this document.

When a PLP-R is present in an ATSC 3.0 system, a DRC-enabled receiver shall identify the PLP-R and then process it by the receiver component for DRC, while a conventional receiver without DRC function shall ignore PLP-R.

5. PHY SPECIFICATION

In this section, the framing, channel coding, modulation, DFT transform, resource mapping, pilot mapping, signal generation, random access, synchronization, and mapping from MAC PDU to Transport Block (TB) of DRC uplink are defined.

5.1 DRC Uplink Frame and Modulation

Single-Carrier Frequency Division Multiple Access (SC-FDMA) with Cyclic Prefix (CP) shall be used for DRC uplink. The signal generation of SC-FDMA is depicted in Figure 5.1, where P/S means parallel-to-serial conversion of the complex values resulting from the IFFT (note that this is not a parallel-to-serial bit conversion).

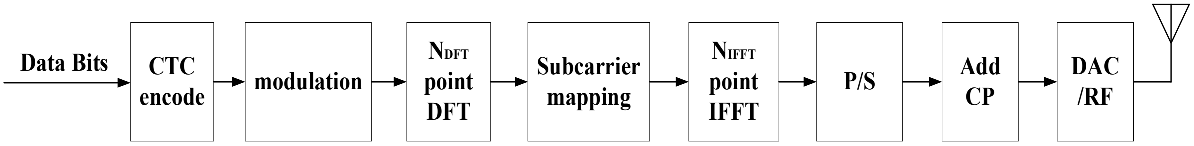

Figure 5.1 SC-FDMA signal generation structure

The uplink frame structure is shown in Figure 5.2. Each uplink frame shall have a time length of $T_F = 10\ msec$ and shall consist of 44 SC-FDMA symbols followed by one guard period (GP). A BAT shall not transmit on the DRC during a GP.

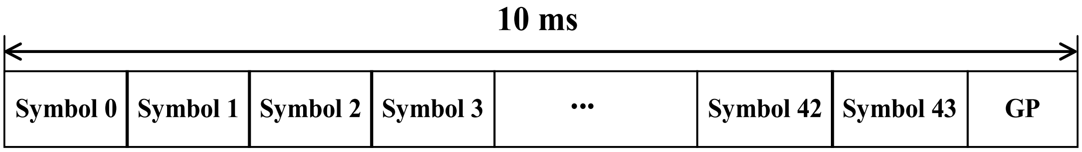

Figure 5.2 Uplink frame structure in time domain.

In the frequency dimension, there are 2048 subcarriers, which are indexed from 0 to 2047. Distribution and indexing of subcarriers in the frequency dimension shall be as illustrated in Figure 5.3. The virtual Direct Current (DC) subcarrier shall be located at the indicated central frequency of the DRC midway between subcarrier 0 and subcarrier 2047.

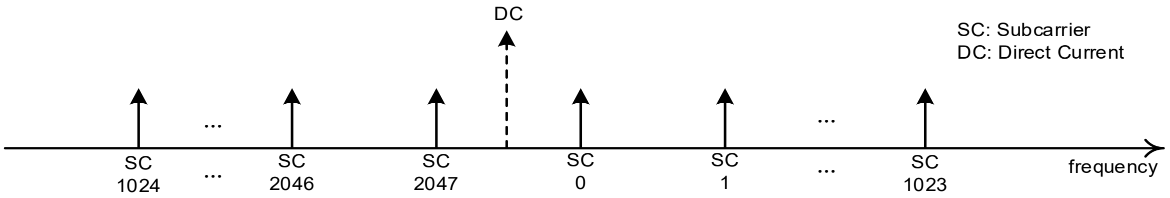

Figure 5.3 Distribution and indexing of DRC subcarriers.

Different bandwidths are supported in DRC uplink. The number of active subcarriers, N_A, is used to adjust DRC uplink bandwidth.

A resource unit consisting of one SC-FDMA symbol in the time dimension and one subcarrier in the frequency dimension is defined as a cell. A cell is uniquely indexed by the pair (k, l), where k and l are the indices in the frequency and time dimensions, respectively. The resource map structure of a DRC uplink frame with an arbitrary number of active subcarriers N_A is shown in Figure 5.4.

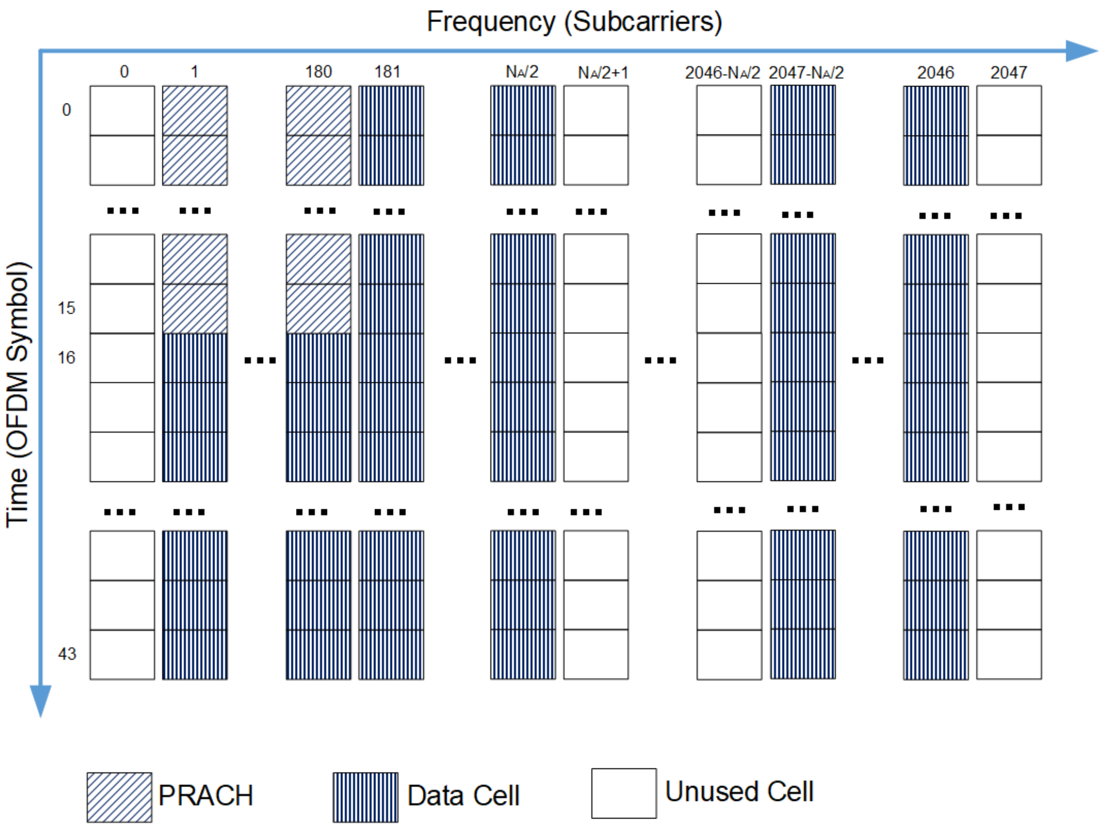

Figure 5.4 Resource map in DRC uplink frame.

Cells with subcarrier indices from 1 to 180 in the frequency dimension and SC-FDMA symbol indices from 0 to 15 in the time dimension in each frame shall be reserved as Physical Random Access Channel (PRACH) resources for random access. The other valid cells within a DRC uplink frame shall be available to the Physical Uplink Shared Channel (PUSCH) for normal transmission.

A tile is defined as 40 cells with $N_{symb} = 2$ SC-FDMA symbols in the time dimension and $N_{SC} = 20$ subcarriers in the frequency dimension. Among the N_{SC} subcarriers in each SC-FDMA symbol, there are $N_{SC\_data} = 16$ data cells and 4 pilot cells. The distribution of the data cells and pilot cells is shown in Figure 5.5.

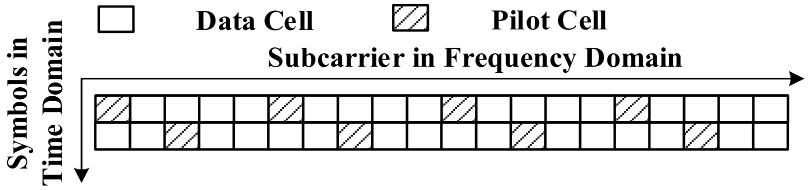

Figure 5.5 Tile pattern.

The physical resource parameters for the DRC, as a function of system bandwidth, shall be as given in Table 5.1 and Table 5.2.

Table 5.1 PHY Parameters for the DRC as a Function of System Bandwidth (1)

Parameter	**1MHz**	**2MHz**	**3MHz**	**4MHz**
Number of active subcarriers(N_A)	200	400	600	800
Occupied Bandwidth(MHz)	0.9863	1.9629	2.9395	3.9160
Active subcarrier indices	[1,100]∪ [1947,2046]	[1,200]∪ [1847,2046]	[1,300]∪ [1747,2046]	[1,400]∪ [1647,2046]
Total number of Tiles in the first 16 SC-FDMA symbols(NT_1)	8	88	168	248
Total number of Tiles in the following 28 SC-FDMA symbols(NT_2)	140	280	420	560
Total number of Tiles(NT_T)	148	368	588	808
Number of Tiles per symbol pair in the first 16 SC-FDMA symbols($NTPPS_1$)	1	11	21	31
Number of Tiles per symbol pair in the following 28 SC-FDMA symbols($NTPPS_2$)	10	20	30	40

Table 5.2 PHY Parameters for the DRC as a Function of System Bandwidth (2)

Parameter	**5MHz**	**6MHz**	**7MHz**	**8MHz**
Number of active subcarriers(N_A)	1000	1200	1400	1600
Occupied Bandwidth(MHz)	4.8926	5.8691	6.8457	7.8223
Active subcarrier indices	[1,500]∪ [1547,2046]	[1,600]∪ [1447,2046]	[1,700]∪ [1347,2046]	[1,800]∪ [1247,2046]
Total number of Tiles in the first 16 SC-FDMA symbols(NT_1)	328	408	488	568
Total number of Tiles in the following 28 SC-FDMA symbols(NT_2)	700	840	980	1120
Total number of Tiles(NT_T)	1028	1248	1468	1688
Number of Tiles per symbol pair in the first 16 SC-FDMA symbols($NTPPS_1$)	41	51	61	71
Number of Tiles per symbol pair in the following 28 SC-FDMA symbols($NTPPS_2$)	50	60	70	80

Tiles in an uplink frame shall be indexed from 0 to $NT_T - 1$. Let $0 \leq q_{tile} < NT_T$ denote the index of a tile. The resource allocated to a BAT is identified by the index of the start tile q_{tile} and the length of the allocated resource in tiles, l_{tile}. The allocated tiles shall be those with indices ranging from q_{tile} to $q_{tile} + l_{tile} - 1$ in ascending order. The tiles can be allocated in one of two configurable ways, i.e., time-frequency dimension or frequency-time dimension.

Let n_{symb} and m_{subc} denote the starting SC-FDMA symbol and the starting subcarrier index of a tile, respectively. When the resource is allocated in the time-frequency dimension, the starting position of the tile with index q_{tile} shall be:

$$n_{symb} = \begin{cases} \lfloor q_{tile} / NTPPS_1 \rfloor \times 2, & 0 \leq q_{tile} < NT_1 \\ 16 + \lfloor (q_{tile} - NT_1) / NTPPS_2 \rfloor \times 2, & NT_1 \leq q_{tile} < NT_T \end{cases} \tag{5.1}$$

$$m_{subc} = \begin{cases} 181 + \mathrm{mod}\left(q_{tile}, NTPPS_1\right) \times 20, & \text{condition } C1 \\ 2047 - N_A/2 + \left[\mathrm{mod}\left(q_{tile}, NTPPS_1\right) - NTPPS_2/2 + 9\right] \times 20, & \text{condition } C2 \\ \mathrm{mod}(q_{tile} - NT_1, NTPPS_2) \times 20 + 1, & \text{condition } C3 \\ 2047 - N_A/2 + [\mathrm{mod}(q_{tile} - NT_1, NTPPS_2) - NTPPS_2/2] \times 20, & \text{condition } C4 \end{cases} \qquad (5.2)$$

The four conditions in Equation (5.2) shall be defined as follows:

$$\begin{aligned} &C1: \left(0 \le q_{tile} < NT_1\right) \; and \left[\mathrm{mod}\left(q_{tile}, NTPPS_1\right) < NTPPS_2/2 - 9\right] \\ &C2: \left(0 \le q_{tile} < NT_1\right) \; and \left[\mathrm{mod}\left(q_{tile}, NTPPS_1\right) \ge NTPPS_2/2 - 9\right] \\ &C3: \left(NT_1 \le q_{tile} < NT_T\right) \; and \left[\mathrm{mod}(q_{tile} - NT_1, NTPPS_2) < NTPPS_2/2\right] \\ &C4: \left(NT_1 \le q_{tile} < NT_T\right) \; and \left[\mathrm{mod}(q_{tile} - NT_1, NTPPS_2) \ge NTPPS_2/2\right] \end{aligned} \qquad (5.3)$$

where $\lfloor x \rfloor$ means truncation of x toward zero, and $\mathrm{mod}(x, y)$ means x modulo y.

When tiles are indicated as being allocated in the frequency-time dimension, the starting position of the tile with index q_{tile} shall be:

$$n_{symb} = \begin{cases} \mathrm{mod}\left(q_{tile}, 14\right) \times 2 + 16, \quad 0 \le q_{tile} < 126 \\ \mathrm{mod}\left(q_{tile} - 126, 22\right) \times 2, 126 \le q_{tile} < NT_T \end{cases} \qquad (5.4)$$

$$m_{subc} = \begin{cases} \lfloor q_{tile}/14 \rfloor \times 20 + 1, & \text{condition } C5 \\ \lfloor \left(q_{tile} - 126\right)/22 \rfloor \times 20 + 1, & \text{condition } C6 \\ 2047 - N_A/2 + \lfloor \left(q_{tile} - NTPPS_2 \times 11 + 72\right)/22 \rfloor \times 20, & \text{condition } C7 \end{cases} \qquad (5.5)$$

The three conditions in Equation (5.5) shall be defined as follows:

$$\begin{aligned} &C5: q_{tile} < 126 \\ &C6: 126 \le q_{tile} < NTPPS_2 \times 11 - 72 \\ &C7: q_{tile} \ge NTPPS_2 \times 11 - 72 \end{aligned} \qquad (5.6)$$

5.2 Channel Coding

Convolutional Turbo Coding (CTC) is used for error correction and channel coding in the DRC uplink.

5.2.1 CTC Encoding

The input bit sequence to the CTC Encoder is written as $c_0, c_1, c_2, c_3, \dots c_{L_O - 1}$, where L_O is the length of the input sequence.

5.2.1.1 CTC Encoder

The input bit sequence shall be divided into two parts A and B.

$$\begin{aligned} A_i &= c_{2i}, \quad i = 0,1,2,\cdots,L_O/2-1, \\ B_i &= c_{2i+1}, i = 0,1,2,\cdots,L_O/2-1. \end{aligned} \qquad (5.7)$$

Every bit pair, (A_i,B_i), constitutes a couple and shall be fed into the Convolutional Turbo Coding (CTC) module, which shall generate outputs as shown in Figure 5.6. The CTC encoder uses Circular Recursive Systematic Convolutional Codes (CRSC) as component codes with double-binary input. As required by CRSC, the length of the input bit sequence, L_O, shall be even (i.e., $L_O = 2 \times N_{CTC\_in}$ where N_{CTC_in} is the integer number of input couples).

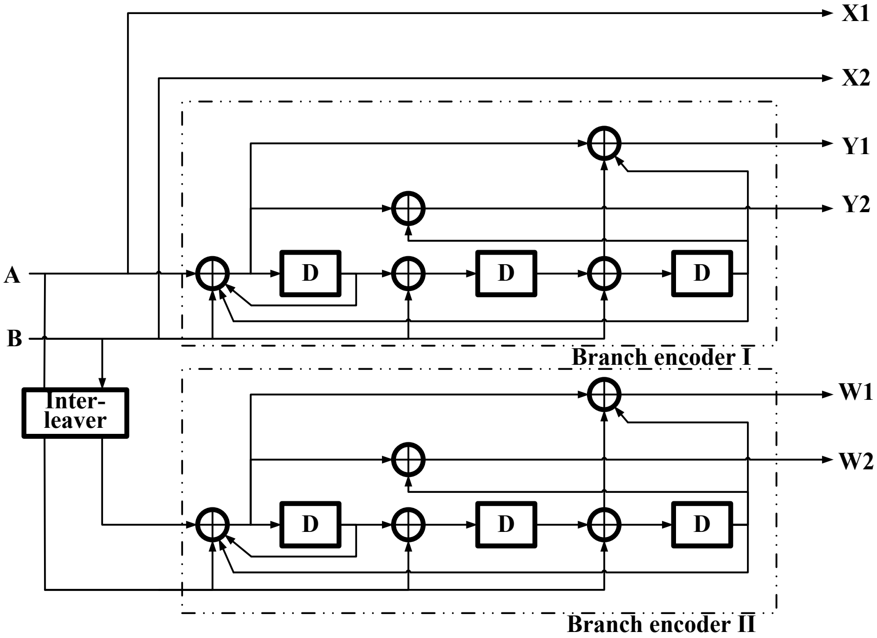

Figure 5.6 Duo-binary Convolution Turbo Coding (CTC).

As shown in Figure 5.6, the CTC encoder generates six output bits for each pair of input bits.

Suppose the input sequences A and B are written as $A_0, A_1, A_2, A_3, \dots, A_{N_{CTC\_in}-1}$ and $B_0, B_1, B_2, B_3, \dots, B_{N_{CTC\_in}-1}$, respectively. The output sequences of the CTC are $X1$, $X2$, $Y1$, $Y2$, $W1$, and $W2$. $X1$ and $X2$ correspond exactly to the input sequences A and B, respectively (i.e. $X1$ and $X2$ represent the systematic bits). $Y1$ and $Y2$ represent the output of branch encoder I. $W1$ and $W2$ represent the output of branch encoder II. Sequences $Y1$ and $Y2$ can be written as $Y1_0, Y1_1, Y1_2, Y1_3, \dots, Y1_{N_{CTC\_in}-1}$ and $Y2_0, Y2_1, Y2_2, Y2_3, \dots, Y2_{N_{CTC\_in}-1}$, respectively. Sequences $W1$ and $W2$ can be written as $W1_0, W1_1, W1_2, W1_3, \dots, W1_{N_{CTC\_in}-1}$ and $W2_0, W2_1, W2_2, W2_3, \dots, W2_{N_{CTC\_in}-1}$, respectively.

Based on the CTC encoder given in Figure 5.6, the final output sequence of the CTC encoder can be written as $z_0, z_1, z_2, z_3, \dots, z_{6N_{CTC\_in}-1}$, where:

$$z_i = \begin{cases} A_i, & i = 0,1,2,\cdots,N_{CTC\_in}-1, \\ B_{i-N_{CTC\_in}}, & i = N_{CTC\_in}, N_{CTC\_in}+1,\cdots,2N_{CTC\_in}-1, \\ Y1_{i-2N_{CTC\_in}}, & i = 2N_{CTC\_in}, 2N_{CTC\_in}+1,\cdots,3N_{CTC\_in}-1, \\ Y2_{i-3N_{CTC\_in}}, & i = 3N_{CTC\_in}, 3N_{CTC\_in}+1,\cdots,4N_{CTC\_in}-1, \\ W1_{i-4N_{CTC\_in}}, & i = 4N_{CTC\_in}, 4N_{CTC\_in}+1,\cdots,5N_{CTC\_in}-1, \\ W2_{i-5N_{CTC\_in}}, & i = 5N_{CTC\_in}, 5N_{CTC\_in}+1,\cdots,6N_{CTC\_in}-1. \end{cases} \tag{5.8}$$

5.2.1.2 Branch Encoder Generator Polynomials

For the feedback branch, i.e., $Y1$ and $W1$, the generator polynomial shall be $G_1(D) = [D^3 + D + 1]$. For the parity bits of $Y2$ and $W2$, the generator polynomial shall be $G_2(D) = [D^3 + D^2 + 1]$.

The initial value of each encoder register shall be set to zero prior to the start of each code block.

5.2.1.3 CTC interleaver

The CTC interleaver consists of two permutation steps. The first step is a permutation on the level of each couple individually, and the second step is on the level of the sequence of all couples.

The first step is defined as switching the elements of alternate couples.

For the input sequence of all couples written as $[(A_0, B_0), (A_1, B_1), (A_2, B_2) \ldots]$, the output of the first step shall be $[(A_0, B_0), (B_1, A_1), (A_2, B_2) \ldots]$. That is, the two elements of couple (A_i, B_i) shall maintain their order when i is even and shall be swapped with each other when i is odd. This operation shall be repeated for the entire block.

The second step provides the interleaved address j of couple i. Given the permutation parameters as P_0, P_1, P_2 and P_3, the second step of the interleaver shall be defined as:

$$P(i) = \begin{cases} 0, & i \bmod 4 = 0, \\ N_{CTC\_in}/2 + P_1, & i \bmod 4 = 1, \\ P_2, & i \bmod 4 = 2, \\ N_{CTC\_in}/2 + P_3, & i \bmod 4 = 3. \end{cases} \tag{5.9}$$

$$j = \left[P_0 \times i + P(i) + 1\right] \bmod N_{CTC\_in} \tag{5.10}$$

where i is the index of an input couple to the interleaver, j is the index of the corresponding output couple after interleaving, and mod is the modulo operation taking the remainder after division. Parameters P_0, P_1, P_2 and P_3 depend on the length of the sequence to be encoded. The CTC interleaver parameters for the different modulation modes and code rates allowed for the DRC shall be as listed in Table 5.3.

Table 5.3 CTC Parameters for Different Modulation Modes and Code Rates

Modulation Mode	Code Rate	L_O	N_{CTC_in}	P_0	P_1	P_2	P_3
BPSK	1/3	96	48	13	24	0	24
QPSK	1/2	96	48	13	24	0	24
		192	96	7	48	24	72
		288	144	17	74	72	2
	3/4	144	72	11	6	0	6
		288	144	17	74	72	2
		432	216	13	108	0	108
16QAM	1/2	192	96	7	48	24	72
		384	192	11	96	48	144
		576	288	23	50	188	50
	3/4	288	144	17	74	72	2
		576	288	23	50	188	50
		864	432	13	0	4	8

5.2.1.4 CTC Puncturing

In order to generate different CTC code rates, parity bits shall be punctured from the encoding output. The puncturing method shall be as shown in Table 5.4 for the configured code rate.

Table 5.4 CTC Puncturing Method

Code Rate	Retained Bits
1/3	$z_0, z_1, z_2, z_3, \dots, z_{6N_{CTC\_in}-1}$
1/2	$z_0, z_1, z_2, z_3, \dots, z_{4N_{CTC\_in}-1}$
3/4	$z_0, z_1, z_2, z_3, \dots, z_{2N_{CTC_{in}}-1},$ $z_{2N_{CTC\_in}}, z_{2N_{CTC\_in}+3}, z_{2N_{CTC\_in}+6}, z_{2N_{CTC\_in}+9}, \dots, z_{4N_{CTC\_in}-6}, z_{4N_{CTC\_in}-3}$

In summary, CTC parameters for different modulation and code rate combinations are listed in Table 5.5. L_{CTC_out} is the number of bits after encoding. L_{punc_out} is the number of bits after puncturing. L_{symb} is the number of modulation symbols. L_{tile} is the number of tiles occupied with the corresponding modulation-coding scheme. Given L_{tile}, the number of resource blocks can be calculated as $L_{tile}/3$.

Table 5.5 Summary of CTC Parameters for Different Modulation and Code Rate Combinations

Modulation Mode	**Code Rate**	L_O	L_{CTC_out}	L_{punc_out}	L_{symb}	L_{tile}	**AMC Type**
BPSK	1/3	96	288	288	288	9	0
QPSK	1/2	96	288	192	96	3	1
		192	576	384	192	6	2
		288	864	576	288	9	3
	3/4	144	432	192	96	3	4
		288	864	384	192	6	5
		432	1296	576	288	9	6
16QAM	1/2	192	576	384	96	3	7
		384	1152	768	192	6	8
		576	1728	1152	288	9	9
	3/4	288	864	384	96	3	10
		576	1728	768	192	6	11
		864	2592	1152	288	9	12

5.3 Modulation

There are three kinds of modulation modes supported in DRC: BPSK, QPSK, and 16QAM. Gray coding is used for mapping binary bits to modulation symbols. The input to the modulation block shall be the output of the CTC encoder after puncturing. Assume the bit streams of the output of the CTC encoder after puncturing are $v_0, v_1, v_2, \dots, v_n$, and according to the configured modulation mode, the encoded and punctured bits enter serially to the constellation mapper as described in Table 5.6, Table 5.7 and Table 5.8, respectively. The resulting output sequence consists of L_{symb} complex modulated symbols $d(is) = I + j \times Q; is = 0,1, \dots, L_{symb} - 1$.

5.3.1 BPSK

For BPSK modulation, each input bit v_i shall be mapped to a complex modulated symbol according to Table 5.6.

Table 5.6 BPSK Modulation Mapping

v_i	I	Q
0	$1/\sqrt{2}$	$1/\sqrt{2}$
1	$-1/\sqrt{2}$	$-1/\sqrt{2}$

5.3.2 QPSK

For QPSK modulation, each pair of input bits v_{2i}, v_{2i+1} shall be mapped to a complex modulated symbol according to Table 5.7.

Table 5.7 QPSK Modulation Mapping

v_{2i}, v_{2i+1}	I	Q
0,0	$1/\sqrt{2}$	$1/\sqrt{2}$
0,1	$1/\sqrt{2}$	$-1/\sqrt{2}$
1,0	$-1/\sqrt{2}$	$1/\sqrt{2}$
1,1	$-1/\sqrt{2}$	$-1/\sqrt{2}$

5.3.3 16QAM

For 16QAM, each set of four input bits $v_{4i}, v_{4i+1}, v_{4i+2}, v_{4i+3}$ shall be mapped to a complex modulated symbol according to Table 5.8.

Table 5.8 16QAM Modulation Mapping

$v_{4i}, v_{4i+1}, v_{4i+2}, v_{4i+3}$	I	Q
0,0,0,0	$1/\sqrt{10}$	$1/\sqrt{10}$
0,0,0,1	$1/\sqrt{10}$	$3/\sqrt{10}$
0,0,1,0	$3/\sqrt{10}$	$1/\sqrt{10}$
0,0,1,1	$3/\sqrt{10}$	$3/\sqrt{10}$
0,1,0,0	$1/\sqrt{10}$	$-1/\sqrt{10}$
0,1,0,1	$1/\sqrt{10}$	$-3/\sqrt{10}$
0,1,1,0	$3/\sqrt{10}$	$-1/\sqrt{10}$
0,1,1,1	$3/\sqrt{10}$	$-3/\sqrt{10}$
1,0,0,0	$-1/\sqrt{10}$	$1/\sqrt{10}$
1,0,0,1	$-1/\sqrt{10}$	$3/\sqrt{10}$
1,0,1,0	$-3/\sqrt{10}$	$1/\sqrt{10}$
1,0,1,1	$-3/\sqrt{10}$	$3/\sqrt{10}$
1,1,0,0	$-1/\sqrt{10}$	$-1/\sqrt{10}$
1,1,0,1	$-1/\sqrt{10}$	$-3/\sqrt{10}$
1,1,1,0	$-3/\sqrt{10}$	$-1/\sqrt{10}$
1,1,1,1	$-3/\sqrt{10}$	$-3/\sqrt{10}$

5.4 DFT Transform

DFT transform shall be applied to the block of modulated symbols $d(0), d(1), \dots, d(L_{symb} - 1)$ according to Equation (5.11) below, where N_{sc_data} is the number of data subcarriers in one SC-FDMA symbol of a tile as defined in Section 5.1.

$$
\begin{aligned}
& x_{DFT}\left(l \cdot N_{sc\_data} + k\right) = \frac{1}{\sqrt{N_{sc\_data}}} \sum_{i=0}^{N_{sc\_data}-1} d\left(l \cdot N_{sc\_data} + i\right) e^{-j2\pi ik / N_{sc\_data}} \\
& k = 0, \cdots, N_{sc\_data} - 1 \qquad (5.11) \\
& l = 0, \cdots, 2L_{tile} - 1
\end{aligned}
$$

5.5 Physical Resource Mapping

The complex-valued symbols $x_{DFT}(i)$ $(0 \leq i < L_{symb} - 1)$ shall be mapped in sequence, starting with $x_{DFT}(0)$, to the data cells of the assigned resource tiles. Within the tiles assigned for transmission, the mapping of $x_{DFT}(i)$ to cells shall be in increasing order of subcarrier index within each SC-FDMA symbol of a tile, SC-FDMA symbol index within each tile and then tile index successively. That is, mapping shall begin with the first data subcarrier of the first SC-FDMA symbol of the first assigned tile, where the first data subcarrier of a tile shall be defined as the data subcarrier with the lowest subcarrier index for that tile. Mapping shall continue with the remaining data subcarriers, in increasing order, of the first SC-FDMA symbol of the first assigned tile, before moving on to the first data subcarrier of the second SC-FDMA symbol of the first assigned tile. When each assigned tile has been completely filled and complex-valued symbols $x_{DFT}(i)$ still remain to be mapped, mapping shall move on to the first data subcarrier of the first SC-FDMA symbol of the next assigned tile.

Tiles to be used for transmission shall be assigned by BTS. The corresponding tile allocation is broadcast through BCI.

5.6 Pilot Mapping

Scattered pilots shall be used in the DRC uplink.

5.6.1 Locations of the Pilots

Cell (k, l) (where k is the subcarrier index and l is the SC-FDMA symbol index) of each tile that has been assigned to a BAT shall be a scattered pilot when one of the conditions given in Equation (5.12) below is satisfied:

$$\begin{aligned}&\mathrm{mod}(k+4,5)=0, 1\le k\le N_A/2, l \text{ is even.}\\&\mathrm{mod}(k+3,5)=0, 2047-N_A/2\le k\le 2046, l \text{ is even.}\\&\mathrm{mod}(k+2,5)=0, 1\le k\le N_A/2, l \text{ is odd.}\\&\mathrm{mod}(k+1,5)=0, 2047-N_A/2\le k\le 2046, l \text{ is odd.}\end{aligned} \tag{5.12}$$

A BAT shall transmit scattered pilots only in the tiles that have been allocated to that BAT.

5.6.2 Generation of the Pilot Sequence

The pilot sequence is generated from a frequency-domain Zadoff-Chu sequence. The complex value for a pilot on subcarrier k shall be given by Equation (5.13) below.

$$x_p(k)=\begin{cases}\sqrt{2}\exp\left[-j\pi\dfrac{k(k+1)}{N_A+1}\right], & 1\le k\le N_A/2\\ \sqrt{2}\exp\left[-j\pi\dfrac{(k-846)(k-845)}{N_A+1}\right], & 2047-N_A/2\le k\le 2046\end{cases} \tag{5.13}$$

5.7 SC-FDMA Baseband Signal Generation

Each SC-FDMA symbol is composed of two parts: a useful part with time duration $N_{IFFT}T_S$ and a cyclic prefix with time duration $N_{CP}T_S$ preceding the useful part. The cyclic prefix shall be formed by copying the last N_{CP} values from the SC-FDMA symbol's useful part and prepending those values immediately before the SC-FDMA symbol's useful part. Each uplink frame shall contain a guard period with time duration $N_{GP}T_S$ at the end of the frame, during which time nothing shall be transmitted. The system parameters of a DRC uplink frame shall be as summarized in Table 5.9.

Table 5.9 System Parameters for SC-FDMA in DRC

Parameter	Value
IFFT point (N_{IFFT})	2048
DFT point (N_{DFT})	16
Number of active subcarriers	200(1MHz Bandwidth)
	400(2MHz Bandwidth)
	600(3MHz Bandwidth)
	800(4MHz Bandwidth)
	1000(5MHz Bandwidth)
	1200(6MHz Bandwidth)
	1400(7MHz Bandwidth)
	1600(8MHz Bandwidth)
Sampling rate	10 MHz
Time period between samples (T_s)	0.1 µsec
Cyclic prefix (N_{CP})	210
Guard period (N_{GP})	648
Subcarrier spacing	4.883 kHz
Actual occupied bandwidth	0.9863MHz(1MHz Bandwidth)
	1.9629MHz(2MHz Bandwidth)
	2.9395MHz(3MHz Bandwidth)
	3.9160MHz(4MHz Bandwidth)
	4.8926MHz(5MHz Bandwidth)
	5.8691MHz(6MHz Bandwidth)
	6.8457MHz(7MHz Bandwidth)
	7.8223MHz(8MHz Bandwidth)

The time-continuous signal $s(t)$ for SC-FDMA symbol l shall be defined by:

$$s_l\left(t\right) = \sum_{k=-1024}^{1023} x_{\hat{k},l} e^{j2\pi\left(k+1/2\right)\Delta f\left(t-N_{CP}T_s\right)} \tag{5.14}$$

where $0 \le t < (N_{IFFT} + N_{CP}) \times T_s$, $\hat{k} = (k + 2048) \bmod 2048,\ \Delta f = 4.883\ kHz$ and $x_{k,l}$ is the modulated symbol value of cell (k, l).

The SC-FDMA symbols in an uplink frame shall be transmitted in increasing order of l.

5.8 Random Access

Random access procedure shall be used to initialize the communication between the BTS and BATs. Random access may be used in the following cases: (i) Registration Request; (ii) Deregistration Request; (iii) Bandwidth Allocation Request; (iv) Connection Release Request; (v) Paging Response.

In random access procedure, the BAT first sends a random access sequence in PRACH to the BTS. The random access sequence shall be randomly chosen by the BAT in Registration Request, Deregistration Request, Bandwidth Allocation Request and Connection Release Request cases, and shall be assigned by the BTS in the Paging case.

When the BTS correctly receives the random access sequence, referring to Section 6.4.1, the BTS shall send a Random Access Response to the BAT with time advance, power offset, and resource allocated for subsequent message transmission.

A random access sequence shall be generated from the following N_{ZC}-point Zadoff-Chu (ZC) sequence:

$$Z(k) = \exp\left[-j\frac{\pi\mu k(k+1)}{N_{ZC}}\right], 0 \le k \le 1776 \tag{5.15}$$

where $N_{ZC} = 1777$ and μ is a positive integer with $0 < \mu < N_{ZC}$. Due to the feature of ZC sequences, a cyclic shift of a ZC sequence will generate another orthogonal sequence, which can be used as a random access sequence. To avoid time ambiguity, as explained in Annex A.4, only the cyclic shifts of 0 and N_{CS}=888 shall be used in DRC. It means that each root μ can generate two random access sequences. Therefore, there shall be 3552 distinct random access sequences in DRC.

All random access sequences are indexed by n_{zc} ranging from 0 to 3551. The random access sequence with index n_{zc} shall be defined by Equation (5.16) below:

$$Z_{n_{zc}}(k) = \begin{cases} \exp\left[-j\frac{\pi\left(n_{zc}+1\right)k\left(k+1\right)}{N_{ZC}}\right], & 0 \le n_{zc} < N_{ZC}-1 \\ \exp\left[-j\frac{\pi\left(n_{zc}-N_{ZC}+1\right)\left(k+N_{cs}-1\right)\left(k+N_{cs}\right)}{N_{ZC}}\right], & N_{ZC}-1 \le n_{zc} \le 2N_{ZC}-3 \end{cases} \tag{5.16}$$

The ZC sequence $Z_{n_{zc}}(k)$ shall be mapped to PRACH subcarriers in the frequency domain as shown below:

$$X_{n_{zc}}\left(k\right) = \begin{cases} Z_{n_{zc}}\left(k\right), & 0 \le k \le 888, \\ 0, & 889 \le k \le 1159, \\ Z_{n_{zc}}\left(k-271\right), & 1160 \le k \le 2047. \end{cases} \tag{5.17}$$

The random access sequence then passes through a 2048-point IFFT to generate the PRACH sequence in the time domain. The space between PRACH subcarriers is one tenth of that between data subcarriers. The time-continuous random access signal $x_{n_{zc}}(t)$ with cyclic prefix shall be:

$$x_{n_{zc}}\left(t\right) = \sum_{k=-1024}^{1023} X_{n_{zc}}\left(\hat{k}\right) e^{j2\pi\left(k+905\right)\Delta f^{RA}\left(t-N_{CP}^{RA}T_S^{RA}\right)} \tag{5.18}$$

where $0 \le t < (N_{IFFT} + N_{CP}^{RA}) \times T_S^{RA}$, $\hat{k} = (k + 2048) \bmod 2048$ and $\Delta f^{RA} = 488.3\ Hz$.

The Guard Time (GT) shall be padded with zeros at the end of the random access sequence in the time domain. The structure of a random access sequence shall be as shown in Figure 5.7.

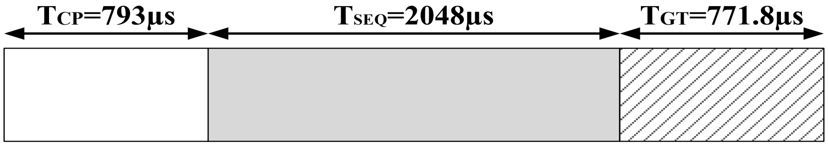

Figure 5.7 The structure of a random access sequence.

All of the parameters for random access shall be as summarized in Table 5.10.

Table 5.10 PRACH Parameters

Parameter	**Value**
Sampling rate (f_S^{RA})	1 MHz
Time period between samples (T_S^{RA})	1 μsec
PRACH subcarrier spacing (Δf^{RA})	488.3 Hz
Type of Random Access Sequence	ZC sequence
Length of Random Access Sequence	1777 points
PRACH Cyclic prefix (N_{CP}^{RA})	$793T_S^{RA}$
PRACH Cyclic prefix duration	793 μsec
PRACH Sequence $\left(N_{SEQ}^{RA}\right)$	$2048T_S^{RA}$
PRACH Sequence duration	2048 μsec
Guard Time (GT) duration	771.8 μsec
Total duration of PRACH	3612.8 μsec
PRACH occupied bandwidth (BW^{RA})	878.9 kHz

5.9 Synchronization Procedure

The DRC system uses a synchronization procedure to ensure that a BTS synchronously receives all uplink frames transmitted by different BATs. This is achieved by compensating for propagation delays and aligning all uplink frames to the BTS time reference.

The DRC system works with uplink frame length equal to $T_F = 10msec$. A BTS should set up its time reference by Global Navigation Satellite System (GNSS) or Precision Time Protocol (PTP) [3]. The time reference shall be time-aligned with 1pps timing pulse and shall be divided into 100 uplink frames for each second. Every set of 1000 uplink frames shall begin at an integer multiple of ten seconds and the uplink frames within each set shall be sequentially indexed with I_{itvl} $(0 \leq I_{itvl} < 1000)$.

In order to minimize interference, the following process shall be followed in the DRC system:

1) For the ith downlink frame, a BTS shall include the **L1D_time_sec**, **L1D_time_msec**, **L1D_time_usec**, and **L1D_time_nsec** fields in that frame's preamble. The start time of the bootstrap's first sample for the ith frame referenced to the BTS is $t_{bootstrap}(i)$. The start time of the closest 10ms uplink frame that begins prior to or coincident with the bootstrap of the ith downlink frame referenced to the BTS is $t_{relativetime}(i)$. The elapsed time is defined as $T_{L1}(i) = t_{bootstrap}(i) - t_{relativetime}(i)$.
2) A BAT will detect the first sample of the bootstrap belonging to the ith downlink frame at time $t_{sync}(i) = t_{bootstrap}(i) + T_{delay}$, where T_{delay} is the propagation delay from the BTS to the BAT. The BAT shall decode L1-Detail to obtain **L1D_time_sec**, **L1D_time_msec**, **L1D_time_usec**, **L1D_time_nsec**. The BAT shall then obtain $t_{bootstrap}(i) = \textbf{L1D\_time\_sec} +$

$\mathbf{L1D\_time\_msec} \times 10^{-3} + \mathbf{L1D\_time\_usec} \times 10^{-6} + \mathbf{L1D\_time\_nsec} \times 10^{-9}$ and $t_{relativetime}(i) = \text{floor}(t_{bootstrap}(i) \times 100)/100$; then $T_{L1}(i)$ can be calculated.

3) The BAT shall reconstruct a time reference with 10ms intervals using an internal 10 MHz clock. The start time of the reconstructed time reference shall be $t_{sync}(i) - T_{L1}(i)$. Each 10ms interval is a DRC uplink frame without synchronization between the BTS and the BAT.
4) The BAT shall randomly select a random access sequence using a uniform distribution and then shall transmit it in the PRACH channel of the uplink frame right after the end of the backoff. The rule of backoff is defined in Section 6.2.2. The BAT shall record I_{itvl} of the uplink frame for receipt of the Random Access Response from the BTS.
5) When the BTS detects an uplink PRACH signal, the BTS shall calculate T_{delay} and transmit it to the BAT via PLP-R. The BAT shall then adjust its reconstructed time reference to compensate for T_{delay} during its next transmission.

The synchronization procedure is shown in Figure 5.8.

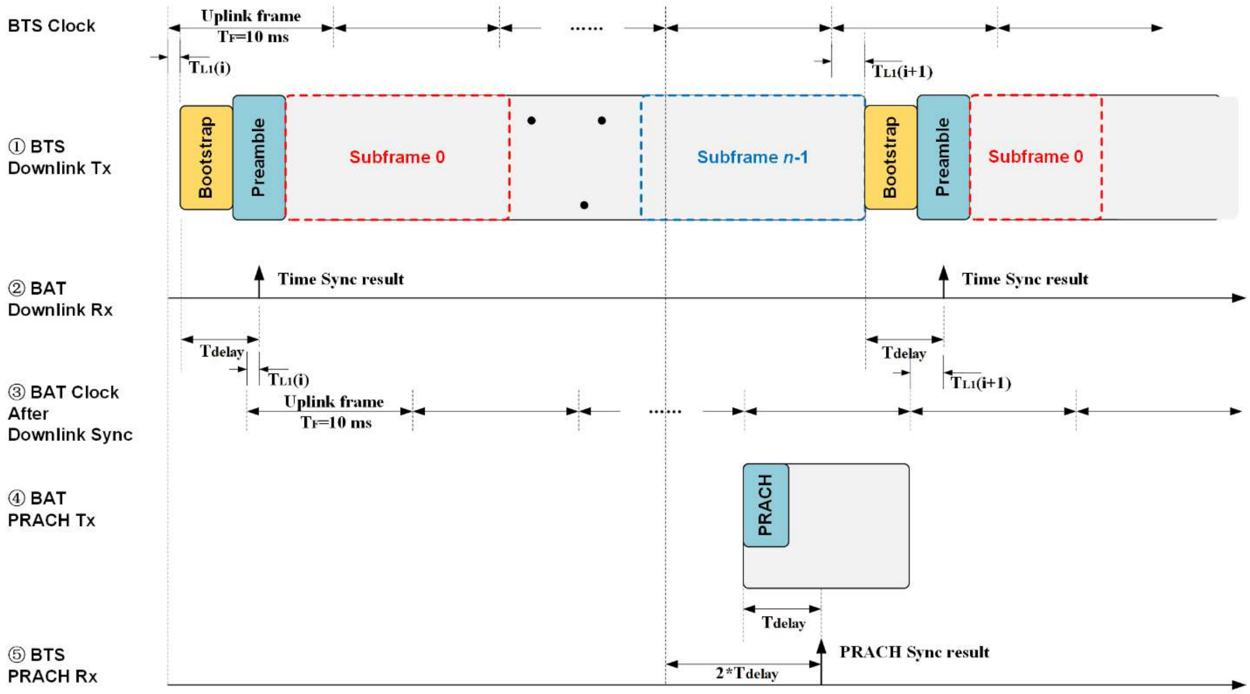

Figure 5.8 Synchronization procedure.

5.10 Mapping from DRC Uplink PDU to Transport Block (TB)

The CTC coding block is defined as a Transport Block (TB). The size of a TB shall be equal to the number of bits input to the CTC block defined in Section 5.2. As shown in Figure 5.9, a TB is composed of three parts: TB header, TB payload and padding as required at the end of TB. The TB header is defined further below. The TB payload is the container of MAC PDU. Padding shall be used when the size of a MAC PDU or a MAC PDU segment plus the size of the TB header is smaller than the size of the TB.

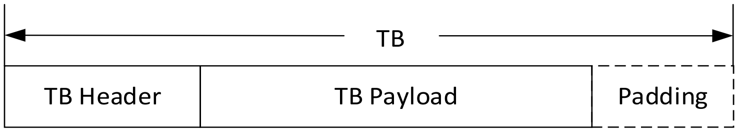

Figure 5.9 Transport Block (TB).

The structure of the TB header shall be as shown in Figure 5.10. SI, E, TBSN, C, R, and PL are abbreviations of **segment_indication**, **extension**, **tb_sequence_number**, **continuation**, **reserved** and **padding_length**, respectively. The syntax of the transport_block() header shall be as specified in Table 5.11. The semantics of the fields in the transport_block() shall be as given immediately below the table.

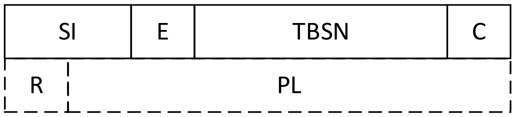

Figure 5.10 TB Header.

Table 5.11 Transport Block Header Syntax

Syntax	No. of Bits	Format
transport_block () {		
segment_indication	2	uimsbf
extension	1	uimsbf
tb_sequence_number	4	uimsbf
continuation	1	uimsbf
if (extension == 1){		
reserved	1	‘1’
padding_length	7	uimsbf
}		
}		

segment_indication – This field shall indicate the transport block payload segment status of a MAC PDU.

When the payload in this transport block is a whole MAC PDU, **segment_indication** shall be set to 0.

When the payload in this transport block is the first segment of a MAC PDU, **segment_indication** shall be set to 0x1.

When the payload in this transport block is the last segment of a MAC PDU, **segment_indication** shall be set to 0x2.

When the payload in this transport block is one segment of a MAC PDU but neither the first nor the last segment of that MAC PDU, **segment_indication** shall be set to 0x3.

extension – This field shall indicate the transport block payload start point.

When the payload in this transport block starts from the second byte of the transport block and the R (**reserved**) and PL (**padding_length**) fields are not included in the TB Header, **extension** shall be set to 0.

When the payload in this transport block starts from the third byte of the transport block and the R (**reserved**) and PL (**padding_length**) fields are included in the TB Header, **extension** shall be set to 1.

tb_sequence_number – This field shall indicate the transport block sequence number.

When a MAC PDU is not segmented, **tb_sequence_number** shall always be set to 0. When a MAC PDU is segmented, **tb_sequence_number** of the first segment shall be set to 1 and **tb_sequence_number** for each successively following segment of the same MAC PDU shall be incremented by 1. After **tb_sequence_number** reaches 15, it shall be wrapped to 1. A **tb_sequence_number** of 0 shall not be used for any segment of a segmented MAC PDU.

continuation – This field shall indicate whether the current transport block payload is a continuation of the same segmented DRC uplink MAC PDU from the previous DRC uplink TB.

This field shall be considered valid only when **tb_sequence_number** is greater than 0. When **tb_sequence_number** = 1 and the current transport block is the first segment of the MAC PDU, **continuation** shall be set to 0. When **tb_sequence_number** is not 0 and the payload in this transport block is a continuation of the same segmented DRC uplink MAC PDU, **continuation** shall be set to 1. When **tb_sequence_number** = 0, **continuation** shall be set to 0.

padding_length – This field shall indicate the number of padding bytes.

When there are multiple uplink PDUs corresponding to multiple TBs in one uplink frame, the PDUs shall be mapped to TBs in sequence.

A DRC uplink MAC PDU is composed of one or more different subheaders and their respective payloads (refer to Section 6.3). The size of an uplink MAC PDU may be larger than the size of a TB payload, in which case the MAC PDU shall be segmented into multiple TBs for transmission. However, one TB shall contain at most one MAC PDU or one MAC PDU segment. Concatenation of multiple MAC PDUs and/or MAC PDU segments into a TB shall not be allowed in the DRC uplink.

5.11 Radio Link Failure

Any of the following conditions shall trigger DRC radio link failure at a BAT:

1) The BAT is unable to detect any bootstraps and/or decode any preambles over a consecutive time period equal to 60 seconds;
2) Inability to successfully decode any FEC Blocks of PLP-R over a consecutive time period equal to 60 seconds;
3) Three consecutive failures of the random access procedure;
4) The maximum number of retransmissions of a Type I MAC PDU in the uplink channel has been exceeded.

When a BAT detects DRC radio link failure, the BAT shall release any **tuid** and **cid(s)** that have been assigned to that BAT and shall reset the PHY and MAC layer. Then the BAT shall reselect the uplink channel (see Section 6.2.1) and shall restart the registration procedure if needed.

When the BTS receives a registration request from a BAT with an existing registered connection, the BTS shall release the **tuid** and **cid(s)** that were assigned to that BAT and shall clear any corresponding status variables and/or timers. At the same time, the BTS shall accept the new registration from the BAT and shall allocate a new **tuid** to the BAT.

6. MAC SPECIFICATION

6.1 Introduction

This chapter defines the MAC (Media Access Control) protocol of the DRC, which specifies functions for resource sharing in the uplink channel and the control signaling required in the downlink.

6.1.1 Services

The MAC layer services are defined as:

1) Services provided to upper layers
   a) Packet Data;
   b) Reconfiguration of the uplink wireless resources.
2) Services required from the physical layer
   a) Received Data after error correction;
   b) Report of channel status information.

6.1.2 Functions

1) Multiplexing of data packets from different connections to MAC PDU;
2) De-multiplexing of packets from MAC PDUs;
3) Resource scheduling;
4) Retransmission of MAC PDUs as required.

6.2 MAC Procedures

6.2.1 Uplink Channel Selection

When there is more than one uplink channel indicated by BCI, a BAT shall select one uplink channel to register and transmit uplink data. A BAT shall select the uplink channel with the least load as indicated in BCI. When there is more than one uplink channel with the same least load, a BAT shall randomly select one of those uplink channels with a uniform probability distribution. A BAT shall not transmit in more than one uplink channel at the same time.

6.2.2 Random Access

The random access procedure describes how a Broadcast Access Terminal (BAT) initiates an access request through the random access channel, and establishes a wireless connection between the BAT and a BTS (Broadcast Television Station).

A BTS shall notify all BATs of the network configuration through downlink Broadcast Control Information (BCI). When needed, a BAT initiates random access based upon information obtained from BCI. The random access procedure includes the following six steps:

1) The BAT shall acquire downlink frequency and time synchronization from the downlink bootstrap as specified in Section 5.9.
2) The BAT shall acquire the system information from the downlink broadcast control information (BCI). The BAT shall randomly select, with a uniform probability distribution, a Group A random access sequence. Then the BAT shall transmit the selected random access sequence in the PRACH channel of the next uplink frame and start ***Timer_Random_Access_Response***. The duration of ***Timer_Random_Access_Response*** shall be equal to $10N_{RR}$ msec, where N_{RR} is the number of uplink frames to wait after sending a random access sequence and the duration of an uplink frame is 10 msec. The value of N_{RR} shall be equal to **frame_wait_count** as broadcast in BCI by the BTS.

3) When the selected random access sequence is successfully received by the BTS, the BTS may transmit a random access response in the downlink channel, including the random access sequence index, time adjustment, power adjustment, and scheduled resource for further message transmission. The **uplink_channel_id** field of the downlink MAC PDU containing a random access response shall correspond to the uplink channel in which the random access sequence was received by the BTS.
4) When a random access response with a matching uplink channel and random access sequence index is correctly received by the BAT, the BAT shall stop ***Timer_Random_Access_Response***, adjust the BAT's time and transmission power according to the received random access response, send Message A to the BTS using the resources allocated in step 3, and start ***Timer_Contention*** and ***Timer_Uplink_Synchronized***. The duration of ***Timer_Contention*** shall be equal to $10N_{RR}$ msec.
5) After receiving a Message A, the BTS shall transmit the response Message B to the BAT.
6) When a BAT does not receive a matching random access response before the timeout of ***Timer_Random_Access_Response*** or does not receive Message B before the timeout of ***Timer_Contention***, the BAT shall declare a failure in this particular random access, stop ***Timer_Uplink_Synchronized*** if it is running and initiate a backoff. When the backoff ends, the BAT shall schedule another random access with updated transmission power according to Equation (6.1) until the maximum number of random access attempts is reached.

The random access procedure can be used for registration request, deregistration request, bandwidth allocation request, connection release request or paging response.

Therefore, the contents of Message A and Message B depend on different applications of the random access procedure. When the random access procedure is used for registration, Message A and Message B are Registration Request and Registration Confirmation, respectively. When the random access procedure is used for deregistration, Message A and Message B are Deregistration Request and Deregistration Confirmation, respectively. When the random access procedure is used for bandwidth allocation request, Message A and Message B are Bandwidth Allocation Request and Bandwidth Allocation Response, respectively. When the random access procedure is used for connection release request, Message A and Message B are Connection Release Request and Connection Release Confirmation, respectively.

Note that a paging response corresponds to a Bandwidth Allocation Request as defined in Section 6.2.9.

The flow chart of a successful random access procedure is illustrated in Figure 6.1.

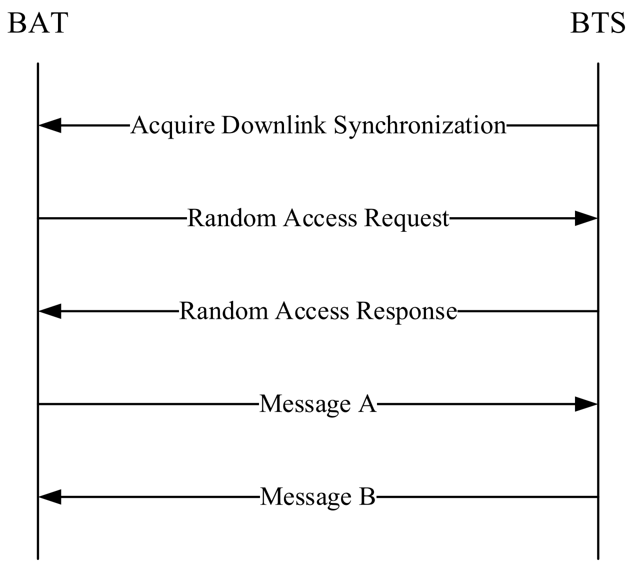

Figure 6.1 Random access procedure.

The Binary Exponential Backoff (BEB) mechanism shall be used in DRC. The user's initial backoff window size, **random_access_backoff_init_window**, and the maximum backoff window size, **random_access_backoff_max_window**, are signaled in the downlink BCI by BTS.

When a random access attempt of a BAT fails and the maximum number of random access retransmissions, **retrans_count_max**, for this random access procedure has not been reached, the BAT shall perform a backoff. When a BAT retransmits a random access sequence, the BAT shall perform the random access backoff as follows.

Let W represent the backoff window size for a BAT. When the BAT determines that it needs to retry random access, it shall randomly choose an integer value y from the interval $[0, W-1]$ with uniform distribution, reselect a Group A random access sequence, and transmit the selected random access sequence in the uplink frame which occurs y uplink frames after the current uplink frame. After the retransmission, the current backoff window size shall be doubled until the maximum window size is reached.

When the maximum number of random access attempts is reached for a random access procedure, the BAT shall redo the uplink channel selection procedure and perform a new random access procedure. When the random access procedure fails 3 times continuously, the uplink channel selection and random access procedure shall be stopped and the event shall be reported to upper layers.

In the random access procedure, the BAT's transmission power shall be set according to the following equation:

$$P_{tx} = P_{rx}^{\min} + L_{dl} + (N_{trans} - 1) \times P_{step} \tag{6.1}$$

where P_{tx} is the transmission power; L_{dl} is the downlink path loss, which is estimated from **downlink_transmission_power** indicated in BCI minus the measured received power at the BAT; N_{trans} is the number of transmissions (including the current transmission), which shall be reset to 1 at the beginning of each random access procedure; P_{rx}^{min} is the minimum required receiving power at the BTS for correct decoding of data; and P_{step} is the power ramp step after a transmission failure. Both P_{rx}^{min} and P_{step} are broadcast by the BTS in the downlink BCI.

The flow charts for the random access procedure at the BAT and the BTS sides are shown in Figure 6.2 and Figure 6.3, respectively.

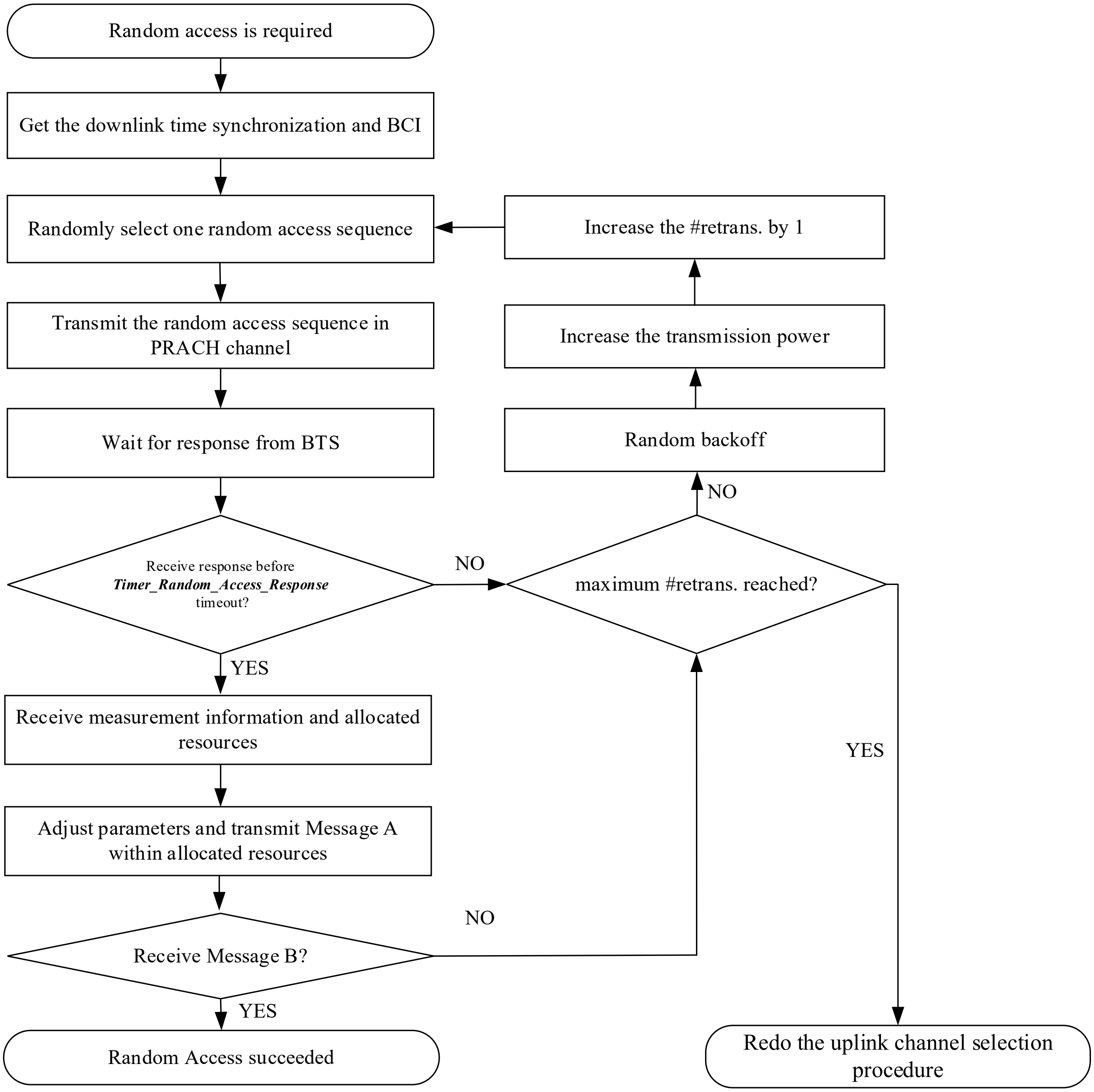

Figure 6.2 Random access procedure at the BAT.

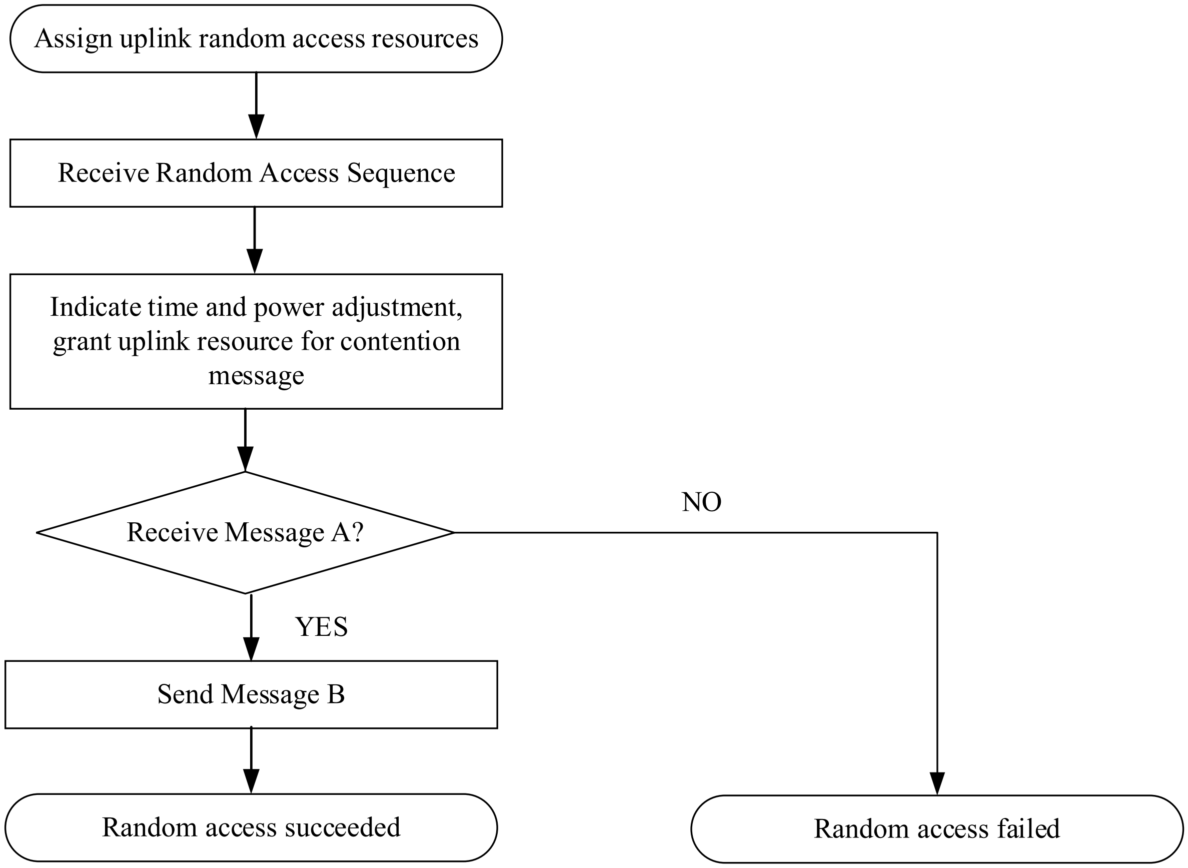

Figure 6.3 Random access procedure at the BTS.

6.2.3 Registration

The registration request procedure is included in the random access procedure. Definitions of Registration Request and Registration Confirmation are given in Sections 6.4.1 and 6.7.7, respectively.

The unique identity of a BAT is the MAC address. The length of a MAC address is 48 bits, which can accommodate an extremely large number of users. However, the maximum number of users covered by a broadcast tower is much more limited. Consequently, a shorter Temporary User ID (**tuid**) is used to identify the BAT in the Registration Confirmation message to reduce the overhead of user ID. The length of TUID is defined as 18 bits. Therefore, the maximum number of simultaneously active users supported by a BTS is $2^{18} = 262,144$.

When **tuid** is allocated to a BAT, both the BAT and the BTS shall start ***Timer_tuid_Release***.

6.2.4 Deregistration

Deregistration is a necessary procedure for soft shutdown of a BAT to release the allocated identifications and resources, and to notify the BTS that the BAT has been shut down. When a BAT is shut down abruptly without deregistration, the allocated identifications shall be released immediately by the BAT and shall be released by the BTS upon timeout of the corresponding ***Timer_tuid_Release*** at the BTS.

The deregistration request procedure shall be triggered when a BAT is powered down. When the BAT is powered down, it shall send Deregistration Request to the BTS and start

Timer_Deregistration_Request. The duration of ***Timer_Deregistration_Request*** shall be 30 seconds whenever this timer is started.

When there are allocated resources for the BAT, the BAT shall transmit a Deregistration Request subheader in the allocated resources. Otherwise, the BAT shall start a random access procedure and send the Deregistration Request in Message A of the random access procedure. When the Deregistration Request has been sent, the BAT shall start ***Timer_Deregistration_Request***.

When the BTS correctly receives a Deregistration Request from a BAT, and the indicated **tuid** is active at the BTS, the BTS shall respond with Deregistration Confirmation subheader to the BAT, release all the connections between the BAT and BTS and release the allocated **tuid** for that BAT. When the BTS correctly receives a Deregistration Request from a BAT, and the indicated **tuid** has already been deregistered at the BTS, the BTS shall discard the Deregistration Request.

When the BAT correctly receives Deregistration Confirmation from the BTS before the timeout of ***Timer_Deregistration_Request***, the BAT shall stop ***Timer_Deregistration_Request***, release all connections, release the **tuid** and finish the deregistration procedure. When the timeout of ***Timer_Deregistration_Request*** happens at the BAT before receiving the Deregistration Confirmation from the BTS, the BAT shall release all connections, release the **tuid** and finish the deregistration procedure.

If the BAT receives Deregistration Confirmation after the deregistration procedure has already finished at the BAT, the BAT shall discard the Deregistration Confirmation.

6.2.5 Connection Release

When a BAT wants to release a connection, the BAT shall send a Connection Release Request subheader to the BTS to release the connection.

When there are allocated resources for the BAT, the BAT shall transmit a Connection Release Request subheader in the allocated resources. Otherwise, the BAT shall start a random access procedure and send the Connection Release Request in Message A of the random access procedure. When the Connection Release Request subheader has been sent, the BAT shall start ***Timer_Connection_Release_Response.*** The value of ***Timer_Connection_Release_Response*** is broadcast in BCI.

When the BTS receives a Connection Release Request subheader and the corresponding connection is active at the BTS, the BTS shall respond with Connection Release Confirmation subheader and release the connection. When the BTS receives a Connection Release Request subheader and the corresponding connection has already been released at the BTS, the BTS shall discard the Connection Release Request subheader.

When the BAT receives the Connection Release Confirmation subheader from the BTS before ***Timer_Connection_Release_Response*** expires, the connection shall be released and ***Timer_Connection_Release_Response*** shall be stopped. When ***Timer_Connection_Release_Response*** expires at the BAT, the connection shall be released by the BAT.

When ***Timer_Connection_Release*** expires at the BTS, the connection shall be released immediately by the BTS. When ***Timer_Connection_Release*** expires at a BAT, the connection shall be released immediately by the BAT.

When a connection has been released at either BTS or a BAT, any subsequent received message related to this connection shall be discarded.

6.2.6 Bandwidth Allocation Request

When a BAT needs resources for uplink transmission, the BAT shall start a bandwidth allocation request procedure, in which the BAT shall send a Bandwidth Allocation Request subheader to the BTS.

When the BAT needs more bandwidth for an existing connection and there are allocated resources for the connection, the BAT shall transmit a Bandwidth Allocation Request subheader in the allocated resources.

When the BAT needs more bandwidth for an existing connection and there are no allocated resources for the connection, the BAT shall start a random access procedure and send a Bandwidth Allocation Request subheader with the **cid** of the connection in Message A of the random access procedure.

When the BAT needs to establish a new connection and request bandwidth for that connection, the BAT shall start a random access procedure and send a Bandwidth Allocation Request subheader with **cid**=0 in Message A of the random access procedure. When a new **cid** is allocated to a BAT, both the BAT and the BTS shall start ***Timer_Connection_Release*** for that **cid**. When the BTS receives any data from the BAT for a particular **cid**, ***Timer_Connection_Release*** for that **cid** shall be restarted at the BTS. When the BAT receives any data from the BTS for a particular **cid**, ***Timer_Connection_Release*** for that **cid** shall be restarted at the BAT. The value of ***Timer_Connection_Release*** is broadcast in BCI.

6.2.7 Status Report Request

When a BTS needs the status of a BAT, the BTS shall start a status report request procedure, in which the BTS shall send a Status Report Request subheader to the BAT. The status report request procedure is only used for a BAT with allocated resources for an active connection.

When a BAT receives a Status Report Request from the BTS, the BAT shall transmit a Status Report subheader in the allocated resources to the BTS.

6.2.8 Online Adjustment

To keep BATs synchronized to the BTS, when the BTS receives uplink data from BATs, the BTS shall calculate updated values for the time advance offset and received power offset and shall transmit, as appropriate, Online Adjustment Subheader to a specific BAT with ID **tuid**.

When a BAT receives a time adjustment from the BTS, the BAT shall apply the time adjustment and restart ***Timer_Uplink_Synchronized*** as defined in Section 6.2.2. When ***Timer_Uplink_Synchronized*** expires, it indicates the BAT has lost uplink synchronization and the BAT shall execute random access procedure to regain synchronization if the BAT wants to send data.

6.2.9 Paging

When a BTS needs to contact a specific BAT that is powered on and registered, but the BAT has no active uplink connection, the BTS uses paging for rapid access. Idle BATs that are completely turned off or just monitoring the bootstrap for Emergency Alert wake-ups cannot be contacted via paging. The paging information, which includes the **tuid** of the paged BAT and the uplink random access sequence allocated to that BAT, is broadcast in the downlink BCI. When a BAT receives a paging request intended for that BAT, the BAT shall start a random access procedure.

Each powered-on and registered BAT without an active uplink connection shall monitor the downlink MAC PDUs in the PLP-R for Paging Request subheaders intended for that BAT. When a monitoring BAT receives a Paging Request with an assigned random access sequence and an assigned PRACH, the BAT shall transmit the assigned random access sequence in the corresponding PRACH and shall then wait for a response from the BTS. The BAT shall transmit the assigned random access sequence in the uplink channel corresponding to the **uplink_channel_id** field of the downlink MAC PDU which contained the Paging Request. The transmission power of the assigned random access sequence depends on the number of repetitions of the paging request with the same assigned random access sequence as received by the BAT. The transmission power shall be calculated conforming to Equation (6.1), in which N_{trans} shall be set equal to the number of repetitions of the paging request as received by the BAT. When the BAT receives a paging request with a random access sequence different from the last one, N_{trans} shall be reset to 1.

When a response is received, the BAT shall adjust its timing and power according to the measurement information, and shall transmit a Bandwidth Allocation Request message with **cid** = 0 and **qci** = 5. Paging succeeds when the BAT receives a Bandwidth Allocation Response message from the BTS before ***Timer_Random_Access_Response*** expires. The flow chart of paging at the BAT side is shown in Figure 6.4.

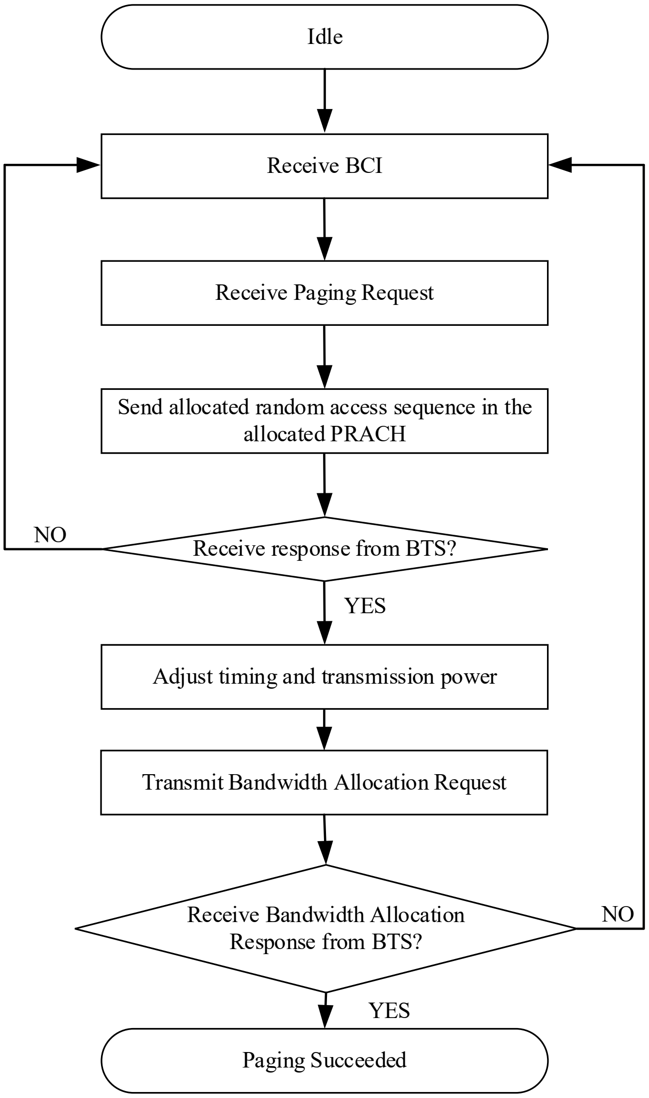

Figure 6.4 Flow chart of paging at the BAT side.

The flow chart of paging at the BTS side is shown in Figure 6.5. A BTS sends Paging Requests to BATs as requested by upper layers. For one paging procedure to a specific BAT, a Group B random access sequence shall be randomly selected and reused in the following retransmission of the paging procedure if needed. To avoid confusion between two consecutive but different paging procedures, the random access sequences assigned for the two consecutive paging procedures to a specific BAT shall be different. When the BTS successfully receives the random access sequence from a paged BAT, the BTS shall send a Random Access Response to the corresponding BAT. Otherwise, the BTS shall retransmit the Paging Request if the maximum number of paging request transmission attempts has not been reached. The maximum number of paging request transmission attempts shall be the same as the maximum number of random access attempts.

When the BTS correctly receives a Bandwidth Allocation Request from the BAT in the allocated resource in the Random Access Response message before the maximum number of paging request transmission attempts is reached, the BTS shall transmit a Bandwidth Allocation Response message with allocated **cid** and the allocated resource. Then the paging shall be considered to have succeeded.

When the maximum number of paging request transmission attempts is reached and no random access request has been received from the paged BAT, the paging shall be considered to have failed. The event shall be notified to upper layers.

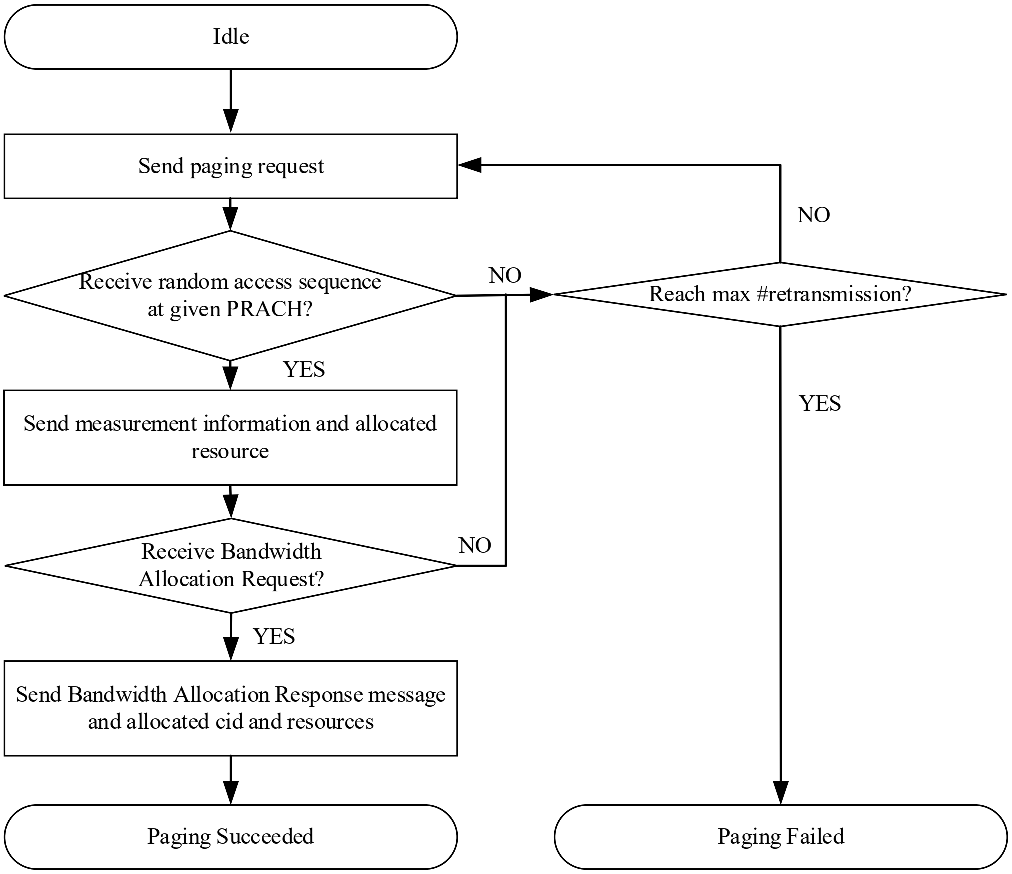

Figure 6.5 Flow chart of paging at the BTS side.

6.2.10 ARQ procedures

A window-based selective ARQ procedure shall be used only for DRC Type I MAC PDU. Type II MAC PDUs are not subject to ARQ and shall not be retransmitted. Within this section, a MAC PDU refers to either a DRC uplink MAC PDU or a DRC downlink MAC PDU. Both DRC uplink and DRC downlink use the same ARQ scheme defined in Section 6.2.10.

A separate ARQ entity shall be maintained for each distinct **tuid** that is active. A BAT shall have an ARQ entity for the **tuid** connecting the BAT to the BTS. The BTS shall maintain a separate ARQ for each **tuid** that is active between the BTS and a BAT.

Referring to Sections 6.3 and 6.6, an 8-bit sequence number (SN) field is contained in the header of each MAC PDU to identify the specific MAC PDU. The way to use SN to identify the specific MAC PDU is defined in Section 6.3 for DRC uplink MAC PDU and Section 6.6 for DRC downlink MAC PDU, respectively. All arithmetic operations and comparisons performed upon sequence numbers shall be calculated modulo 0xFE.

6.2.10.1 Parameter Definitions

Three parameters, TxW, RxW and MRC, are defined for the window-based selective ARQ.

a) TxW: Transmission Window Size, which defines the maximum allowed difference between the sequence number of the newest Type I MAC PDU that is allowed to be transmitted and the sequence number of the oldest transmitted Type I MAC PDU that has not yet been positively acknowledged.

b) RxW: Receiving Window Size, which defines the maximum allowed difference between the sequence number of the next Type I MAC PDU waiting to be received and the sequence number of the oldest Type I MAC PDU that has not yet been correctly received.

c) MRC: Maximum Retransmission Count, which defines the maximum retransmission count of a Type I MAC PDU.

6.2.10.2 Transmitter-Side Variables

Four transmitter-side variables, $V(S)$, $V(AS)$, $V(MS)$, and $V(TC)$, are defined for the window-based selective ARQ as follows:

a) $V(S)$: Sequence Number for Sending, which is an unsigned integer and defines the sequence number of the next Type I MAC PDU to be transmitted for the first time (i.e. excluding Type I MAC PDU(s) requiring retransmission). It shall be incremented after the aforementioned Type I MAC PDU is transmitted. The initial value of this variable shall be 0.

b) $V(AS)$: Sequence Number waiting for Acknowledgement, which defines the sequence number of the Type I MAC PDU immediately after the final in-sequence positively acknowledged Type I MAC PDU at the transmitter side. It indicates the lower edge of the transmission window. It shall be updated based on the receipt of an ACK Message including a **first_sequence_number** that implicitly or explicitly positively acknowledges the Type I MAC PDU with sequence number equal to $V(AS)$. The initial value of this variable shall be 0.

c) $V(MS)$: Maximum Sequence Number for Sending, which defines the upper limit on the sequence number of a new Type I MAC PDU to be transmitted. A transmitter shall not transmit a Type I MAC PDU with a sequence number greater than or equal to $V(MS)$. $V(MS) = (V(AS) + TxW)$. $V(MS)$ shall be updated when $V(AS)$ is updated. The initial value of $V(MS)$ shall be TxW.

d) $V(TC)$: Transmission Counter for Sending, which defines the number of times a Type I MAC PDU has been transmitted. There shall be one $V(TC)$ for each Type I MAC PDU.

6.2.10.3 Receiver-Side Variables

Three receiver-side variables, *V(AR), V(R)* and *V(MR),* are defined for the window-based selective ARQ as follows:

a) $V(AR)$: Sequence Number of the Type I MAC PDU for Receiving, which defines the sequence number immediately after that of the final in-sequence Type I MAC PDU that has been correctly received. It shall be updated upon the successful reception of the Type I MAC PDU with sequence number equal to *V(AR)*. The initial value of *V(AR)* shall be 0.

b) $V(R)$: Sequence Number of the Type I MAC PDU for Receiving, which defines the sequence number of the Type I MAC PDU immediately after the highest sequence number among all correctly received Type I MAC PDUs. Let x denote the highest sequence number of the Type I MAC PDUs that have been correctly received. If $V(R) \leq x < V(MR)$, then the new value of $V(R)$ shall be set to $(x + 1)$. The initial value of $V(R)$ shall be 0.

c) $V(MR)$: Maximum sequence number for Receiving, which defines the upper limit of the sequence number of received Type I MAC PDUs that shall be accepted. $V(MR) = (V(AR) + RxW)$. When a Type I MAC PDU with sequence number $V(MR) \leq x < V(AR)$ is received, the Type I MAC PDU shall be discarded by the receiver. The initial value of $V(MR)$ shall be RxW. When $V(AR)$ is updated, $V(MR)$ shall also be updated.

6.2.10.4 Timer Definition

Two timers are defined for the window-based selective ARQ:

a) ***Timer_ACK***: Timer for ACK at the transmitter, which shall be started for each specific Type I MAC PDU when the Type I MAC PDU is submitted to the physical layer for either initial transmission or retransmission. There shall be one ***Timer_ACK*** for each Type I MAC PDU. The initial value of ***Timer_ACK*** shall be configured by BCI. When the timer expires, the Type I MAC PDU associated with the timer shall be considered for retransmission.

b) ***Timer_ACK_Periodic***: Timer for ACK at the receiver, which is used to trigger an ACK Message at the receiver. When the timer expires, an ACK Message shall be triggered and the timer shall be restarted if *V(AR)* is not equal to *V(R)*. The value of ***Timer_ACK_Periodic*** shall be configured by BCI.

6.2.10.5 Transmitter Operation

The transmitter shall maintain a transmitting window according to transmitter-side variables $V(AS)$ and $V(MS)$. The transmitter shall not deliver to the physical layer any Type I MAC PDU whose sequence number falls outside of the transmitting window.

When a new Type I MAC PDU is generated, the transmitter shall deliver it to the physical layer and:

- Set the sequence number of the Type I MAC PDU to $V(S)$, and increment $V(S)$ by 1;
- Set $V(TC)$ associated with the Type I MAC PDU to 0 and start ***Timer_ACK*** for the Type I MAC PDU;

When a new Type II MAC PDU is generated, the transmitter shall deliver it to the physical layer directly.

Upon receiving an ACK message and letting x represent the **first_sequence_number** in the ACK message referring to Section 6.4.2, the transmitter shall perform the following operations:

- Set $V(AS)$ to $(x+1)$, and set $V(MS)$ to $(V(AS)+TxW)$;
- Stop ***Timer_ACK*** for any Type I MAC PDU whose sequence number is explicitly or implicitly referenced in the ACK message;
- Type I MAC PDU(s) that have been negatively acknowledged in the ACK message shall be considered for retransmission.

Upon expiration of ***Timer_ACK***, the transmitter shall consider the corresponding Type I MAC PDU for retransmission. The transmitter shall prioritize the retransmission of Type I MAC PDUs over the transmission of new Type I MAC PDUs.

When a Type I MAC PDU is considered for retransmission, the transmitter shall:

- Increment the value of $V(TC)$ by 1 for the Type I MAC PDU;
- if $V(TC) > MRC$, then:
  - Indicate to upper layers that the maximum number of retransmissions has been exceeded;
- else:
  - submit the Type I MAC PDU to the physical layer;
  - start ***Timer_ACK*** for the Type I MAC PDU.

6.2.10.6 Receiver operation

When a MAC PDU is correctly received and x represents the SN of the received MAC PDU, the receiver shall perform the following operations:

- if x equals 0xFF, i.e., the MAC PDU is a Type II MAC PDU, then:
  - decode the signaling subheaders assembled in the Type II MAC PDU.
- else:
  - if x is outside the interval $V(AR) \leq x < V(MR)$, then:
    - the Receiver shall discard the Type I MAC PDU;
  - else:
    - if ***Timer_ACK_Periodic*** is not running, then:
      - start ***Timer_ACK_Periodic***;
    - if a Type I MAC PDU with sequence number x has already been correctly received:
      - the Receiver shall discard the Type I MAC PDU and consider the Type I MAC PDU with this sequence number x as having been correctly received when constructing the next ACK Message to be transmitted;
    - else:
      - decode the Type I MAC PDU, get the signaling subheaders for further processing, and put the Data Payload and the SN of the Type I MAC PDU in the reception buffer;
      - update V(AR), V(R) and V(MR) according to the received Type I MAC PDU (see Section 6.2.10.3).

The Receiver transmits an ACK Message to the Transmitter in order to tell the Transmitter about which Type I MAC PDUs have or have not been correctly received. The following conditions shall trigger the transmission of an ACK Message: ***Timer_ACK_Periodic*** expires. When multiple ACK Messages have been triggered before an ACK Message can be transmitted, then only one ACK Message shall be transmitted and all currently pending triggers of ACK Messages shall be cleared at the time that ACK Message is generated.

When an ACK Message is triggered, the Receiver shall generate an ACK Message (see Section 6.4.2 and 6.7.4) at the next opportunity (i.e. when the next new MAC PDU is generated) according to the Receiver state variables and shall submit the ACK Message to the physical layer when the payload of the ACK Message is not empty.

When *V(AR)* is equal to *V(R)* and the Type I MAC PDU corresponding to *V(R)*-1 has previously been positively acknowledged, then the ACK Message shall have an empty payload. Otherwise, the value of **first_sequence_number** in the ACK Message shall be set equal to *V(AR)*-1, and the ACK Message shall explicitly acknowledge, either positively or negatively, each Type I MAC PDU with a sequence number from *V(AR)*-1 to *V(R)*-1, inclusive.

All successfully received Type I MAC PDUs shall be acknowledged explicitly or implicitly at least once by the Receiver in a triggered ACK Message. In the implicit acknowledgement case, all Type I MAC PDUs with sequence numbers less than that given by the value of **first_sequence_number** in the ACK Message shall be considered to have been confirmed implicitly.

When a Type I MAC PDU with sequence number x greater than the current value of $V(R)$ is correctly received, all Type I MAC PDUs with sequence numbers y in the range $V(R) \le y < x$ shall be considered to be missing. Any Type I MAC PDU that has been detected to be missing shall be negatively acknowledged by the receiver in any ACK Message that is triggered.

When $V(AR)$ is updated, the receiver shall check the Data Payload(s) in the reception buffer and release the appropriate Data Payload(s) in-sequence to upper layers.

6.3 Uplink MAC PDU Format

The structure of an uplink MAC PDU (Protocol Data Unit) is shown in Figure 6.6. It includes four parts, i.e., MAC Header, Subheader MAC Payloads, Data Payload and Cyclic Redundancy Check (CRC). The CRC for an uplink MAC PDU shall be calculated over the contents of the MAC Header, the Subheader MAC Payloads and the Data Payload (when present) for that MAC PDU using the CRC defined in Section 6.3.3.

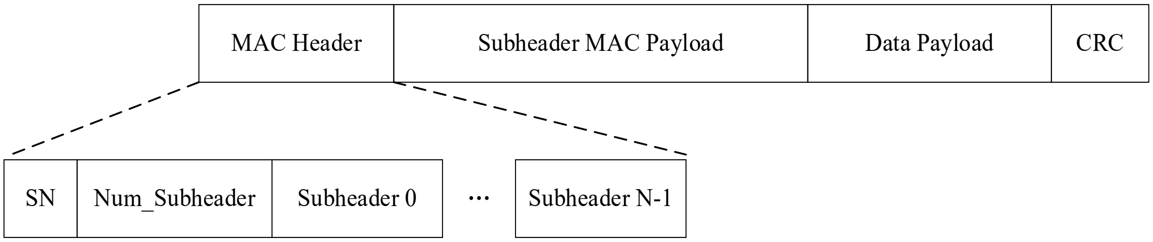

Figure 6.6 Structure of Uplink MAC PDU.

The syntax of an `uplink_mac_header()` shall conform to Table 6.1. The semantics of the fields in the `uplink_mac_header()` shall be as given immediately below the table.

Depending on whether there is Uplink Data Subheader or not, uplink MAC PDUs are divided into two types. Type I uplink MAC PDU shall contain at least one Uplink Data subheader. Type II uplink MAC PDU shall not contain any Uplink Data Subheaders.

For Type I uplink MAC PDU, the sequence number (SN) in an uplink MAC Header shall start indexing from 0x00, and shall increment by 1 whenever a new Type 1 uplink MAC PDU is generated. After reaching 0xFE, the SN shall wrap to 0x00.

For Type II uplink MAC PDU, the value 0xFF of SN shall always be used.

Each uplink MAC PDU shall be byte-aligned, i.e., its length shall be an integer multiple of 8 bits. The lengths of the MAC header and the payload are variable according to different situations.

Table 6.1 DRC Uplink MAC Header Syntax

Syntax	No. of Bits	Format
uplink_mac_header() {		
sequence_number_up	8	uimsbf
num_subheader_up	4	uimsbf
reserved	4	'1111'
for (i=0; i< **num_subheader_up**; i++) {		
subheader_type_up	4	uimsbf (Refer to Table 6.2)
reserved	4	'1111'
}		
}		

sequence_number_up – An 8-bit unsigned integer field that shall signal the sequence number of the current uplink MAC PDU.

num_subheader_up – A 4-bit unsigned integer field that shall signal the number of uplink subheaders included in the uplink MAC PDU.

subheader_type_up – A 4-bit unsigned integer field that identifies the type of the corresponding uplink subheader included in the uplink MAC PDU.

6.3.1 Definition of Uplink Subheader Types

Uplink subheader types shall be as defined in Table 6.2. The DRC uplink MAC Subheaders with subheader type values ranging from 0 to 5 are signaling subheaders. The DRC uplink MAC Subheader with subheader type value 6 is data subheader.

The priorities defined in Table 6.2 are used during the generation of a DRC uplink MAC PDU as specified in Section 6.3.2. A lower priority value shall correspond to a higher priority subheader type.

Table 6.2 DRC Uplink MAC Subheader Types

Value	Priority	Description
0	0	Registration Request
1	0	Uplink ACK Message
2	0	Connection Release Request
3	0	Deregistration Request
4	1	Bandwidth Allocation Request
5	1	Status Report Message
6	2	Uplink Data
7~15		reserved

6.3.2 Assembly of DRC Uplink MAC PDU

The size of an uplink MAC PDU is determined by the resources allocated to the BAT by the BTS in an UL-MAP packet as defined in Section 6.5.2. A BAT shall not construct a new uplink MAC PDU unless the BTS has allocated transmission resources to the BAT for that uplink MAC PDU.

An uplink MAC PDU shall be constructed according to the subheader priorities defined in Table 6.2. Applicable subheaders with higher priority shall be added first to the uplink MAC PDU. Applicable subheaders with lower priority shall then be added to the uplink MAC PDU, when space is available. When the priorities of multiple applicable subheaders are the same, the subheader with the smallest subheader type value shall be included first.

When there is data from upper layer, if the difference between the sequence number for sending, $V(S)$, and the maximum sequence number for sending, $V(MS)$, is smaller than **MAC_PDU_types_threshold**, data subheaders and signaling subheaders shall be assembled into Type I and Type II uplink MAC PDUs separately. In other cases, the assembly of MAC PDU is determined by the system operator. When there is no data from upper layer, signaling subheaders shall be assembled into Type II uplink MAC PDU. When a Type II uplink MAC PDU is present, it shall have priority over the transmission or retransmission of a Type I uplink MAC PDU.

6.3.3 CRC Code Generation

The CRC can be computed using a shift register circuit as illustrated in Figure 6.7.

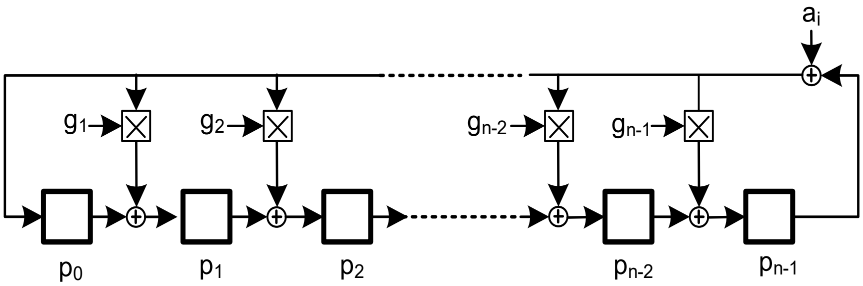

Figure 6.7 Shift register for CRC.

The generator polynomial of an *n* bit CRC can be expressed as:

$$G_{crc-n}(D) = D^n + g_{n-1}D^{n-1} + g_{n-2}D^{n-2} + \ldots.. + g_2 D^2 + g_1 D + 1 \tag{6.2}$$

16-bit CRC coding shall be used in DRC uplink. The CRC code generator polynomial shall be as presented in Equation (6.3) below:

$$G_{CRC-16}(D) = \left[D^{16} + D^{12} + D^5 + 1\right] \tag{6.3}$$

All coefficients of the 16-bit CRC generator polynomial are 0 except g_5 and g_{12}.

The input bit sequence to the CRC operation is written as $a_0, a_1, a_2, a_3, \ldots, a_{L_I-1}$, where L_I is the length of the input sequence. The check bits are written as $p_0, p_1, p_2, p_3, \ldots, p_{L_c-1}$ as shown in Figure 6.7, where $L_c = 16$ is the length of the CRC check. The combined output of the CRC operation is written as $c_0, c_1, c_2, c_3, \ldots c_{L_O-1}$, where $L_O = L_I + L_C$ and:

$$c_k = \begin{cases} a_k, & k = 0,1,2,\cdots,L_I - 1, \\ p_{k-L_I}, & k = L_I, L_I + 1, \cdots, L_O - 1. \end{cases} \tag{6.4}$$

At the receiver side, when the CRC check of a MAC PDU fails, the MAC PDU shall be discarded and the transmitter shall be notified via an ACK message when the MAC PDU is a Type I MAC PDU. The transmitter may retransmit a lost Type I MAC PDU.

6.4 Definitions of DRC Uplink Subheader Payloads

6.4.1 Payload of Registration Request Subheader

The syntax of the Registration Request subheader shall conform to Table 6.3. The semantics of the fields in the registration_request_subheader() shall be as given immediately below the table.

Table 6.3 Registration Request Subheader Syntax

Syntax	No. of Bits	Format
registration_request_subheader() {		
user_id	48	uimsbf
qci	3	uimsbf (See Table 6.4)
reserved	5	'11111'
}		

user_id – A 48-bit unsigned integer field that shall signal the MAC address of the BAT.

Table 6.4 QoS Class Identifier

QCI	Service type	Priority	Packet Delay Budget	Packet Error Rate	Example Services
0	GBR	1	50 msec	10^{-3}	Real Time Gaming
1	GBR	2	100 msec	10^{-2}	Conversational voice
2	GBR	3	150 msec	10^{-3}	Conversational Video (Live Streaming)
3	GBR	4	300 msec	10^{-6}	Non-Conversational Video
4	Non-GBR	5	10 sec	10^{-6}	Voting, comments, program preference
5	Non-specific	6	/	/	Paging
6~7	Reserved				

6.4.2 Payload of Uplink ACK Message Subheader

The syntax of the Uplink ACK Message subheader shall conform to Table 6.5. The semantics of the fields in the ack_message_subheader() shall be as given immediately below the table.

Table 6.5 ACK Message Subheader Syntax

Syntax	No. of Bits	Format
ack_message_subheader() {		
bitmap_length	8	uimsbf
first_sequence_number	8	uimsbf
bitmap	variable	uimsbf
padding	variable	uimsbf
}		

bitmap_length – An 8-bit unsigned integer field that shall specify one less than the length in bits of the bitmap.

first_sequence_number – An 8-bit unsigned integer field that shall signal the sequence number represented by the first bit in the bitmap. At the same time, it shall implicitly indicate that all the downlink Type I MAC PDUs with SN less than or equal to **first_sequence_number** have been correctly received.

bitmap – A variable-length unsigned integer field that shall signal the values of the bits in the bitmap. The meaning of the bit in bit_position ∈ [0, **bitmap_length** - 1] shall be:

0x0: The downlink Type I MAC PDU with SN = (**first_sequence_number** + bit_position) has not been correctly received.

0x1: The downlink Type I MAC PDU with SN = (**first_sequence_number** + bit_position) has been correctly received.

padding – A variable-length unsigned integer field that shall pad the subheader contents for byte alignment.

6.4.3 Payload of Connection Release Request Subheader

The syntax of the Connection Release Request subheader shall conform to Table 6.6. The semantics of the fields in the connection_release_request_subheader() shall be as given immediately below the table.

Table 6.6 Connection Release Request Subheader Syntax

Syntax	No. of Bits	Format
connection_release_request_subheader() {		
tuid	18	uimsbf
cid	5	uimsbf
reserved	1	‘1’
}		

tuid – An 18-bit unsigned integer field that shall specify the temporary user identification.

cid – A 5-bit unsigned integer field that shall specify the identification of the connection to be released. A value of ‘0’ is reserved for requesting a new connection and shall not be considered valid here.

6.4.4 Payload of Deregistration Request Subheader

When the MAC layer of a BAT receives a power down command from upper layer or detects a power down, the BAT shall transmit a Deregistration Request to the BTS. The syntax of the Deregistration Request subheader shall conform to **Table 6.7**.

Table 6.7 Deregistration Request Subheader Syntax

Syntax	No. of Bits	Format
deregistration_request_subheader() {		
tuid	18	uimsbf (See Section 6.4.3)
reserved	6	‘111111’
}		

6.4.5 Payload of Bandwidth Allocation Request Subheader

In the case of DRC uplink data being queued in a BAT’s uplink buffer but without scheduled transmission resources, the BAT shall transmit a Bandwidth Allocation Request subheader. When **cid** is ‘0’, the BAT requests for both a new connection and corresponding bandwidth allocation request for that new connection. Otherwise, the Bandwidth Allocation Request subheader requests resources for the connection with index of **cid.** The syntax of the Bandwidth Allocation Request subheader shall conform to Table 6.8.

Table 6.8 Bandwidth Allocation Request Subheader Syntax

Syntax	No. of Bits	Format
bandwidth_allocation_request_subheader() {		
tuid	18	uimsbf (See Section 6.4.3)
cid	5	uimsbf (See Section 6.4.3)
qci	3	uimsbf (See Table 6.4)
reserved	6	'111111'
}		

6.4.6 Payload of Status Report Subheader

When a BAT sends a Bandwidth Allocation Request subheader or a BAT is requested by the BTS via reception of a Status Report Request subheader, the BAT shall transmit a Status Report subheader. When sending a Bandwidth Allocation Request, the BAT shall send the Status Report subheader together with the Bandwidth Allocation Request subheader. When responding to a Status Report Request, the Status Report subheader shall be included in the BAT's next DRC uplink transmission, after receiving the Status Report Request from the BTS.

The syntax of the Status Report subheader shall conform to Table 6.9. The semantics of the fields in the status_request_subheader() shall be as given immediately below the table.

Table 6.9 Status Report Subheader Syntax

Syntax	No. of Bits	Format
status_request_subheader() {		
buffer_status	6	uimsbf (Refer to Table 6.10)
channel_estimation	6	uimsbf
transmission_power	6	uimsbf
num_connections	5	uimsbf
reserved	1	'1'
}		

channel_estimation – A 6-bit unsigned integer field that shall signal the measured SINR of the bootstrap in dB. The signaled value shall equal the actual value of the measured SINR plus 16. For example, signaled values of 0x00 and 0x3F would signal the measured SINR of the bootstrap as -16 dB and 47 dB, respectively.

transmission_power – A 6-bit unsigned integer field that shall signal the transmission power used by the BAT in dBm. The signaled value shall equal the transmission power in dBm plus 33. For example, 0x00 and 0x3F would signal actual transmission powers of -33 dBm and 30 dBm, respectively.

num_connections – A 5-bit unsigned integer field that shall signal the number of active connections.

Table 6.10 Buffer Status Definition

Index	Buffer Depth (BD) [bytes]	Index	Buffer Depth value [bytes]
0	$BD = 0$	32	$1132 < BD \le 1326$
1	$0 < BD \le 10$	33	$1326 < BD \le 1552$
2	$10 < BD \le 12$	34	$1552 < BD \le 1817$
3	$12 < BD \le 14$	35	$1817 < BD \le 2127$
4	$14 < BD \le 17$	36	$2127 < BD \le 2490$
5	$17 < BD \le 19$	37	$2490 < BD \le 2915$
6	$19 < BD \le 22$	38	$2915 < BD \le 3413$
7	$22 < BD \le 26$	39	$3413 < BD \le 3995$
8	$26 < BD \le 31$	40	$3995 < BD \le 4677$
9	$31 < BD \le 36$	41	$4677 < BD \le 5476$
10	$36 < BD \le 42$	42	$5476 < BD \le 6411$
11	$42 < BD \le 49$	43	$6411 < BD \le 7505$
12	$49 < BD \le 57$	44	$7505 < BD \le 8787$
13	$57 < BD \le 67$	45	$8787 < BD \le 10287$
14	$67 < BD \le 78$	46	$10287 < BD \le 12043$
15	$78 < BD \le 91$	47	$12043 < BD \le 14099$
16	$91 < BD \le 107$	48	$14099 < BD \le 16507$
17	$107 < BD \le 125$	49	$16507 < BD \le 19325$
18	$125 < BD \le 146$	50	$19325 < BD \le 22624$
19	$146 < BD \le 171$	51	$22624 < BD \le 26487$
20	$171 < BD \le 200$	52	$26487 < BD \le 31009$
21	$200 < BD \le 234$	53	$31009 < BD \le 36304$
22	$234 < BD \le 274$	54	$36304 < BD \le 42502$
23	$274 < BD \le 321$	55	$42502 < BD \le 49759$
24	$321 < BD \le 376$	56	$49759 < BD \le 58255$
25	$376 < BD \le 440$	57	$58255 < BD \le 68201$
26	$440 < BD \le 515$	58	$68201 < BD \le 79846$
27	$515 < BD \le 603$	59	$79846 < BD \le 93479$
28	$603 < BD \le 706$	60	$93479 < BD \le 109439$
29	$706 < BD \le 826$	61	$109439 < BD \le 128125$
30	$826 < BD \le 967$	62	$128125 < BD \le 150000$
31	$967 < BD \le 1132$	63	$150000 < BD$

6.4.7 Payload of Uplink Data Subheader

The syntax of the Uplink Data subheader shall conform to Table 6.11. The semantics of the fields in the uplink_data_subheader() shall be as given immediately below the table.

Table 6.11 Uplink Data Subheader Syntax

Syntax	No. of Bits	Format
uplink_data_subheader() {		
cid	5	uimsbf (See Section 6.4.3)
payload_length	11	uimsbf
}		

payload_length – An 11-bit unsigned integer field that shall signal one less than the number of data payload bytes. An uplink data payload shall have a minimum length of 1 byte, and a maximum length of 1500 bytes, where 1500 bytes is the Maximum Transmission Unit (MTU) of Ethernet. Values of **payload_length** between 0x5DC~0x7FF shall be reserved.

6.5 Downlink Broadcast Control Data Format

The downlink control of DRC includes two parts: Downlink Broadcast Control Information (BCI) and Uplink MAP (UL-MAP). In Downlink BCI, system information is broadcast to all BATs. UL-MAP specifies the uplink resource allocation for DRC. Each of the two sets of control information is encapsulated into a control packet for ATSC Link-layer Protocol (ALP) [6], and transferred from the DRC downlink gateway to the ATSC 3.0 gateway through RTP/UDP/IP. At the beginning of each downlink packet, the packet type (PT) shall be indicated by the first 3 bits of the packet and shall be as defined in Table 6.12.

Table 6.12 Downlink Packet Type (PT)

Value	Description
0x0	Downlink Broadcast Control Information (BCI)
0x1	Uplink Resource MAP (UL-MAP)
0x2	Downlink MAC PDU
0x3 ~ 0x7	reserved

6.5.1 Downlink Broadcast Control Information (BCI)

The syntax for a Downlink Broadcast Control Information packet shall conform to Table 6.13. The semantics of the fields in the downlink_broadcast_control_info() packet shall be as given immediately below the table.

Table 6.13 Downlink Broadcast Control Information Packet Syntax

Syntax	No. of Bits	Format
downlink_broadcast_control_info() {		
packet_type	3	uimsbf (Refer to Table 6.12)
downlink_transmission_power	6	uimsbf
num_uplink_channels	4	uimsbf
reserved	3	'111'
for (i=0;i≤ **num_uplink_channels**;i++) {		
uplink_channel_id	4	uimsbf
uplink_channel_bandwidth	4	uimsbf
load_status	2	uimsbf
frequency	22	uimsbf
fft_size	2	uimbsf
max_uplink_transmission_power	6	uimbsf
tile_configuration	1	uimbsf
MAC_PDU_types_threshold	8	uimbsf
random_access_sequence_partition	12	uimbsf
frame_wait_count	9	uimsbf
random_access_min_power	6	uimsbf
random_access_max_power	6	uimsbf
random_access_power_ramp_step	2	uimsbf
random_access_backoff_init_window	2	uimsbf
random_access_backoff_max_window	2	uimsbf
transmission_window_arq	8	uimsbf
receiving_window_arq	8	uimsbf
retrans_count_max	2	uimsbf
timer_ack	13	uimsbf
timer_ack_periodic	13	uimsbf
timer_connection_release	6	uimsbf
timer_connection_release_response	8	uimsbf
timer_uplink_synchronized	8	uimsbf
timer_tuid_release	5	uimsbf
reserved	9	'111111111'
}		
}		

downlink_transmission_power – A 6-bit unsigned integer field that shall signal the downlink transmission power in dBm with values between 0x00 to 0x3C. 0x3D, 0x3E and 0x3F shall be reserved for future use. The downlink transmission power shall be between 0dBm and 60dBm.

num_uplink_channels – A 4-bit unsigned integer field that shall signal one less than the number of supported DRC uplink channels.

uplink_channel_id – A 4-bit unsigned integer field that shall signal the identification of the uplink channel.

uplink_channel_bandwidth – A 4-bit unsigned integer field that shall signal the bandwidth of the DRC uplink channel in MHz with values between 0x01 to 0x08. Bandwidths from 1 MHz to 8 MHz are supported. Values 0x00 and from 0x09 to 0x0F shall be reserved.

load_status – A 2-bit unsigned integer field that shall signal the load status of the uplink channel. Indicated values shall be interpreted as follows:

0x0: idle
0x1: moderate load
0x2: heavy load
0x3: full

frequency – An 22-bit unsigned integer field that shall signal the center frequency of the current DRC uplink channel in kHz with values between 0x0186A0 to 0x3D0900. Frequencies from 100 MHz to 4 GHz are supported. Values from 0x000000 to 0x01869F and from 0x3D0901 to 0x3FFFFF shall be reserved.

fft_size – A 2-bit unsigned integer field that shall signal the size of the FFT to use for DRC transmissions. A value of '0' shall indicate a 2048 subcarrier FFT. Values of '1' to '3' shall be reserved.

max_uplink_transmission_power – A 6-bit unsigned integer field that shall signal the maximum allowed uplink transmission power in dBm with values between 0x00 and 0x28. Values between 0x29 and 0x3F shall be reserved. The maximum allowed transmission power shall be between 0dBm and 40dBm.

tile_configuration – A 1-bit unsigned integer field that shall indicate the tile configuration for uplink allocations. A value of '0' shall indicate tiles are allocated in time-frequency dimension. A value of '1' shall indicate tiles are allocated in frequency-time dimension. Definitions of the tile allocation schemes are in Section 5.1.

MAC_PDU_types_threshold – An 8-bit unsigned integer field that shall indicate the threshold for choosing Type II MAC PDU. The value of **MAC_PDU_types_threshold** shall be between 0x00 and 0xFE. **MAC_PDU_types_threshold** indicates that signaling subheaders shall be assembled only into Type II MAC PDUs if the gap between the sequence number for sending, $V(S)$, and the maximum sequence number for sending, $V(MS)$, is smaller than **MAC_PDU_types_threshold**. **MAC_PDU_types_threshold** = 0xFE indicates that signaling subheaders and data subheaders shall always be assembled into Type II and Type I MAC PDUs separately.

random_access_sequence_partition – A 12-bit unsigned integer field that shall signal the partition of random access sequences. It partitions all random access sequences into two groups. All random access sequences with indices less than **random_access_sequence_partition** shall form Group A, and shall be used for random access procedures. The other set of random access sequences shall form Group B and shall be used for paging. Let x represent the indicated value. Random access sequences with indices from 0 to x-1 shall belong to Group A, and random access sequences with indices from x to 0xDDF shall belong to Group B. Random access sequences that are reserved for paging shall only be used by a BAT when specifically assigned by a BTS to that BAT. Values between 0xDE0~0xFFF shall be reserved.

frame_wait_count – A 9-bit unsigned integer field that shall signal one less than the number of uplink frames for waiting after random access (N_{RR}).

random_access_min_power – A 6-bit unsigned integer field that shall indicate the minimum transmission power to use for random access. Indicated values shall have the range:

0x00~0x32: The minimum random access power, in dBm, shall equal the indicated value minus 50. For example, 0x00 and 0x32 would indicate -50dBm and 0dBm, respectively.
0x33~0x3F: reserved.

random_access_max_power – A 6-bit unsigned integer field that shall indicate the maximum allowable transmission power for random access. Indicated values shall have the range:

0x00~0x32: The maximum random access power, in dBm, shall equal the indicated value minus 20. For example, 0x00 and 0x32 would indicate -20dBm and 30dBm, respectively.

0x33~0x3F: reserved.

random_access_power_ramp_step – A 2-bit unsigned integer field that shall signal the power ramp step to use for random access. Indicated values shall be interpreted as follows:

0x0: 3dB
0x1: 6dB
0x2: reserved
0x3: reserved

random_access_backoff_init_window – A 2-bit unsigned integer field that shall signal the backoff window size for the first retransmission, in units of 10 ms uplink frames. Indicated values shall be interpreted as follows:

0x0: 16
0x1: 32
0x2: 64
0x3: reserved

random_access_backoff_max_window – A 2-bit unsigned integer field that shall signal the maximum backoff window size for random access, in units of 10 ms uplink frames. Indicated values shall be interpreted as follows:

0x0: 1024
0x1: 2048
0x2: 4096
0x3: reserved

transmission_window_arq – An 8-bit unsigned integer field that shall signal one less than the transmission window size for ARQ (*TxW*).

receiving_window_arq – An 8-bit unsigned integer field that shall signal one less than the receiving window size for ARQ (*RxW*).

retrans_count_max – A 2-bit unsigned integer field that shall signal the maximum retransmission count for ARQ (*MRC*). Indicated values shall be interpreted as follows:

0x0: 4
0x1: 8
0x2: 16
0x3: reserved

timer_ack – A 13-bit unsigned integer field that shall signal one less than the value of ***Timer_ACK***, in units of milliseconds. The maximum possible value for ***Timer_ACK*** is 8192 milliseconds.

timer_ack_periodic – A 13-bit unsigned integer field that shall signal one less than the value of ***Timer_ACK_Periodic***, in units of milliseconds. The maximum possible value for ***Timer_ACK_Periodic*** is 8192 milliseconds.

timer_connection_release – A 6-bit unsigned integer field that signals the maximum idle time of a connection in seconds. When the idle time of a connection is larger than **timer_connection_release**, the connection will be released. The maximum valid value shall be

0x3F seconds, and the minimum valid value shall be 0x05 seconds. Values from 0x00 to 0x04 shall be reserved.

timer_connection_release_response – An 8-bit unsigned integer field that signals one less than the maximum time waiting for connection release confirmation transmitted by BTS. The value of ***Timer_Connection_Release_Response*** shall be equal to (**timer_connection_release_response** + 1) seconds. The maximum valid value of **timer_connection_release_response** shall be 0xFF, and the minimum valid value of **timer_connection_release_response** shall be 0x03. Values from 0x00 to 0x02 shall be reserved.

timer_uplink_synchronized – An 8-bit unsigned integer field that signals one less than the maximum time waiting for timing adjustment transmitted by BTS. The value of ***Timer_Uplink_Synchronized*** shall be equal to (**timer_uplink_synchronized** + 1) seconds. The maximum valid value of **timer_uplink_synchronized** shall be 0xFF, and the minimum valid value of **timer_uplink_synchronized** shall be 0x01. Values from 0x00 shall be reserved.

timer_tuid_release – A 5-bit unsigned integer field that signals the maximum idle time of a **tuid** in hours. When the idle time of a **tuid** is larger than **timer_ tuid _release,** the **tuid** shall be released and the BAT shall be deregistered. The maximum valid value is 0x18 hours, and the minimum valid value shall be 0x01 hours. Values of 0x00 and from 0x19 to 0x1F shall be reserved.

6.5.2 Uplink Resource Map Information

The syntax for Uplink Resource MAP packets shall conform to Table 6.14. The semantics of the fields in the uplink_resource_map() packet shall be as given immediately below the table.

Table 6.14 Uplink Resource Map Packet Syntax

Syntax	No. of Bits	Format
uplink_resource_map() {		
packet_type	3	uimsbf (Refer to Table 6.12)
uplink_channel_id	4	uimsbf (See Section 6.5.1)
first_uplink_frame_index	10	uimsbf
num_uplink_frames	9	uimsbf
reserved	6	'111111'
for (i=0;i< **num_uplink_frames**;i++) {		
num_users	10	uimsbf
reserved	6	'111111'
for (i=0;i< **num_users**;i++) {		
tuid	18	uimbsf (See Section 6.4.3)
num_connections	5	uimbsf
reserved	1	'1'
for (i=0;i< **num_connections**;i++) {		
cid	5	uimsbf (See Section 6.4.3)
uplink_start_tile	11	uimsbf
resource_size	10	uimsbf
amc_type	4	uimsbf
reserved	2	'11'
}		
}		
}		
}		

first_uplink_frame_index – A 10-bit unsigned integer field that shall signal the first uplink frame index in the uplink resource map. Values from 0x3E8 to 0x3FF shall be reserved.

num_uplink_frames – A 9-bit unsigned integer field that shall signal the number of uplink frames of the uplink channel in the uplink resource map. 0 shall be a reserved value.

num_users – A 10-bit unsigned integer field that shall signal the number of users in the current uplink frame. The minimum number of users is zero, and the maximum number of users is different for different uplink channel bandwidths as listed in Table **6.15**.

Table 6.15 Maximum Number of Users Per Uplink Frame for Different Bandwidths

Bandwidth (MHz)	Maximum Number of Users	Description
1	0x31	Values between 0x032~0x3FF shall be reserved
2	0x7A	Values between 0x07B~0x3FF shall be reserved
3	0xC4	Values between 0x0C5~0x3FF shall be reserved
4	0x10D	Values between 0x10E~0x3FF shall be reserved
5	0x156	Values between 0x157~0x3FF shall be reserved
6	0x1A0	Values between 0x1A1~0x3FF shall be reserved
7	0x1E9	Values between 0x1EA~0x3FF shall be reserved
8	0x232	Values between 0x233~0x3FF shall be reserved

num_connections – A 5-bit unsigned integer field that shall signal the number of connections with allocated resources for the BAT identified by **tuid**.

uplink_start_tile – An 11-bit unsigned integer field that shall signal the index of the start tile allocated to the connection with identity **cid** of the BAT with identity **tuid**.

resource_size – An 10-bit unsigned integer field that shall signal one less than the number of transport blocks allocated to the connection with identity **cid** of the BAT with identity **tuid.** The number of tiles allocated shall be equal to (indicated value + 1) × N, where N represents the number of tiles in one transport block according to **amc_type**. The indices of tiles in a transport block shall be consecutive.

amc_type – A 3-bit unsigned integer field that shall signal the modulation and code rate to use for the uplink transmission. Indicated values shall be interpreted according to Table 5.5.

6.6 Downlink MAC PDU Format

The structure of a Downlink MAC PDU is shown in Figure 6.8. It includes four parts, i.e., MAC Header, Subheader MAC Payloads, Data Payload and Cyclic Redundancy Check (CRC). The CRC for a downlink MAC PDU shall be calculated over the contents of MAC header, Subheader MAC Payloads and Data Payload of the MAC PDU using the CRC defined in Section 6.3.3.

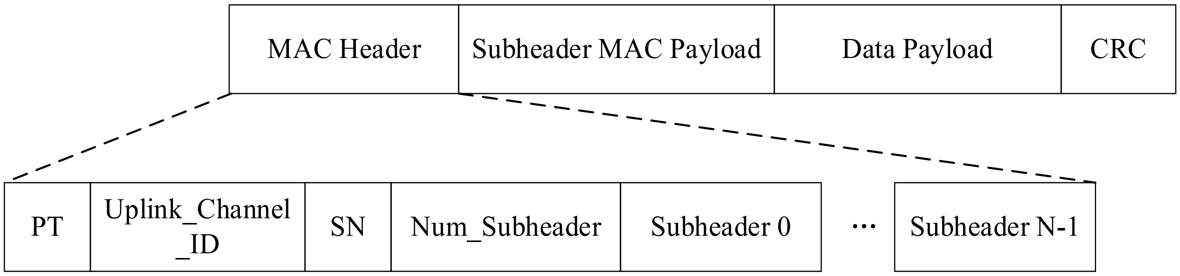

Figure 6.8 Structure of downlink MAC PDU.

The syntax of a downlink MAC header shall conform to Table 6.16. The semantics of the fields in the downlink_pdu_subheader() packet shall be as given immediately below the table.

Depending on whether there is Downlink Data Subheader or not, downlink MAC PDUs are divided into two types. Type I downlink MAC PDU shall contain at least one Downlink Data subheader. Type II downlink MAC PDU shall not contain any Downlink Data subheaders.

For Type I downlink MAC PDU, the sequence number (SN) indexing in the downlink MAC Header shall start from 0x00, and shall increment by 1 whenever a new downlink MAC PDU is transferred. After reaching 0xFE, the SN shall be wrapped to 0x00.

For Type II downlink MAC PDU, the value 0xFF of SN shall always be used.

Each downlink MAC PDU shall be byte-aligned, i.e., its length shall be an integer multiple of 8 bits. The lengths of the MAC header and the payload are variable according to different situations.

Table 6.16 Downlink MAC PDU Header Syntax

Syntax	No. of Bits	Format
downlink_pdu_subheader() {		
packet_type	3	uimsbf (Refer to Table 6.12)
uplink_channel_id	4	uimsbf (See Section 6.5.1)
sequence_number_down	8	uimsbf
num_subheader_down	15	uimsbf
reserved	2	'11'
for (i=0; i≤ **num_subheader_down**; i++) {		
downlink_subheader_type	4	uimsbf (Refer to Table 6.17)
reserved	4	'1111'
}		
}		

sequence_number_down – An 8-bit unsigned integer field that shall signal the sequence number of the current downlink MAC PDU.

num_subheader_down – A 15-bit unsigned integer field that shall signal one less than the number of downlink subheaders included in the current downlink MAC PDU.

6.6.1 Definition of Downlink Subheader Types

Downlink MAC PDU subheader types, **downlink_subheader_type**, shall be as defined in Table 6.17. The downlink MAC Subheaders with subheader type values ranging from 0 to 8 are signaling subheaders. The downlink MAC PDU with subheader type value 9 is data subheader. A lower priority value shall correspond to a higher priority subheader type.

Table 6.17 Types of Downlink MAC PDU Subheaders

Value	Priority	Description
0	0	Random Access Response
1	0	Bandwidth Allocation Response
2	0	Status Report Request
3	0	Downlink ACK Message
4	0	Online Adjustment Message
5	1	Paging Request
6	1	Registration Confirmation
7	1	Connection Release Confirmation
8	1	Deregistration Confirmation
9	2	Downlink Data
0xA~0xF		reserved

6.6.2 Assembly of the DRC Downlink MAC PDU

A downlink MAC PDU shall be constructed according to the subheader priorities defined in Table 6.17. Applicable subheaders with higher priority shall be added first to the downlink MAC PDU. When the priorities of multiple applicable subheaders are the same, the subheader with the smallest subheader type value shall be included first.

When there is data from upper layer, if the difference between the sequence number for sending, $V(S)$, and the maximum sequence number for sending, $V(MS)$, is smaller than **MAC_PDU_types_threshold**, data subheaders and signaling subheaders shall be assembled into Type I and Type II downlink MAC PDUs separately. In other cases, the assembly of MAC PDU is determined by the system operator. When there is no data from upper layer, signaling subheaders shall be assembled into Type II downlink MAC PDU. When a Type II downlink MAC PDU is present, it shall have priority over the transmission or retransmission of a Type I downlink MAC PDU.

6.7 Definition of Downlink MAC Subheaders

6.7.1 Payload of Random Access Response Subheader

When the BTS receives a random access request identified by a specific random access sequence in an uplink frame, the BTS shall respond with a Random Access Response subheader. The syntax of the Random Access Response subheader shall conform to Table 6.18. The semantics of the fields in the downlink_random_access_response_subheader() packet shall be as given immediately below the table.

Table 6.18 Downlink Random Access Response Subheader Syntax

Syntax	No. of Bits	Format
downlink_random_access_response_subheader() {		
frame_index	10	uimsbf
rand_access_seq_index	12	uimsbf
time_advance	9	uimsbf
power_offset	7	uimsbf
uplink_start_tile	11	uimsbf
resource_size	2	uimsbf
amc_type	4	uimsbf
reserved	1	‘1’
}		

frame_index – A 10-bit unsigned integer field that shall signal the index of the DRC uplink frame in which the random access sequence was received. Values between 0x3E8~0x3FF shall be reserved.

rand_access_seq_index – A 12-bit unsigned integer field that shall signal the index of the received random access sequence. Values between 0xDE0~0xFFF shall be reserved.

time_advance – A 9-bit unsigned integer field that shall specify the required time advance to be applied by the BAT. Let N_{TA} denote the signaled value. The actual time advance shall be equal to $T_A = 16 \times T_S \times N_{TA}$, where T_S is defined in Table 5.9. A positive time advance indicates that the BAT needs to advance its transmission time earlier to compensate for the propagation delay of the uplink transmission. The maximum time advance that can be signaled is 817.6 µsec.

power_offset – A 7-bit unsigned integer field that shall signal the BAT offset power (in dB) to be applied to the BAT’s future uplink transmissions. The step size shall be 1 dB. The exact power offset to be applied by the BAT shall be equal to the indicated value minus 64. For example, 0x00 and 0x7F would signal offsets of -64dB and +63dB, respectively. A positive exact power offset shall indicate that the BAT shall increase its transmission power by the specified amount, and a negative exact power offset shall indicate that the BAT shall decrease its transmission power by the specified amount.

uplink_start_tile – An 11-bit unsigned integer field that shall signal the index of the start tile allocated to the BAT.

resource_size – A 2-bit unsigned integer field that shall signal one less than the number of transport blocks allocated to the BAT. One BAT can be allocated 4 transport blocks at most in the payload of Random Access Response subheader. The number of tiles allocated shall be equal to (indicated value + 1) × N, where N represents the number of tiles in one transport block according to **amc_type**. The indices of the tiles in each transport block shall be consecutive.

amc_type – A 4-bit unsigned integer field that shall signal the modulation and coding scheme to be used for the uplink transmission. Indicated values shall be interpreted according to Table 5.5.

6.7.2 Payload of Bandwidth Allocation Response Subheader

When a BTS receives a bandwidth allocation request from a BAT, the BTS shall send a bandwidth allocation response to the BAT. The syntax of the Bandwidth Allocation Response subheader shall conform to Table 6.19. The semantics of the fields in the downlink_bw_allocation_subheader() packet shall be as given immediately below the table.

Table 6.19 Downlink Bandwidth Allocation Response Subheader Syntax

Syntax	No. of Bits	Format
downlink_bw_allocation_subheader() {		
tuid	18	uimsbf (See Section 6.4.3)
cid	5	uimsbf (See Section 6.4.3)
qci	3	uimsbf (See Table 6.4)
bw_allocation_result	2	uimsbf
reserved	4	‘1111’
}		

bw_allocation_result – A 2-bit unsigned integer field that shall specify the result of bandwidth allocation. ‘0x1’ shall indicate that the bandwidth has been allocated, and the allocation information will be given in the following UL-MAP. ‘0x0’ shall indicate that the bandwidth allocation request was rejected. ‘0x2’ and ‘0x3’ shall be reserved.

6.7.3 Payload of Status Report Request Subheader

When a BTS requires a BAT’s status for resource allocation or scheduling, the BTS shall send a Status Report Request subheader to the BAT. The syntax of the Status Report Request subheader shall conform to Table 6.20.

Table 6.20 Downlink Status Report Request Subheader Syntax

Syntax	No. of Bits	Format
downlink_status_report_request_subheader() {		
tuid	18	uimsbf (See Section 6.4.3)
reserved	6	‘111111’
}		

6.7.4 Payload of Downlink ACK Message Subheader

The syntax of a Downlink ACK Message subheader shall conform to Table 6.21. The semantics of the fields in the downlink_ack_message_subheader() packet shall be as given immediately below the table.

Table 6.21 Downlink ACK Message Subheader Syntax

Syntax	No. of Bits	Format
downlink_ack_message_subheader() {		
tuid	18	uimsbf (See Section 6.4.3)
bitmap_length	8	uimsbf
first_sequence_number	8	uimsbf
bitmap	variable	uimsbf
padding	variable	uimsbf
}		

bitmap_length – An 8-bit unsigned integer field that shall specify the length of the bitmap (in bits) minus one.

first_sequence_number – An 8-bit unsigned integer field that shall signal the sequence number (SN) represented by the first bit in the bitmap. At the same time, it shall implicitly indicate that all the uplink Type I MAC PDUs with SN less than or equal to **first_sequence_number** have been correctly received.

bitmap – A variable-length unsigned integer field that shall signal the values of the bits in the bitmap. The meaning of the bit in bit_position ∈ [0, **bitmap_length** - 1] shall be:

0x0: The uplink Type I MAC PDU with SN = (**first_sequence_number** + bit_position) has not been correctly received.

0x1: The uplink Type I MAC PDU with SN = (**first_sequence_number** + bit_position) has been correctly received.

padding – A variable-length unsigned integer field that shall pad the subheader contents for byte alignment.

6.7.5 Payload of Online Adjustment Subheader

The syntax of the Online Adjustment subheader shall conform to Table 6.22. The semantics of the fields in the online_adjust_subheader() packet shall be as given immediately below the table. When the time that a BAT receives an Online Adjustment subheader from the BTS is in the time period of DRC uplink frame n, the corresponding adjustment of the BAT's timing and power shall apply from the beginning of uplink frame $(n + 6)$ modulo 1000.

Table 6.22 Online Adjustment Subheader Syntax

Syntax	No. of Bits	Format
online_adjust_subheader() {		
tuid	18	uimsbf (See Section 6.4.3)
online_time_offset	6	uimsbf
online_power_offset	6	uimsbf
reserved	2	'11'
}		

online_time_offset – A 6-bit unsigned integer field that shall indicate the transmission time offset to be applied by the BAT. Let N_{TA} denote the indicated value. The exact time offset to be applied by the BAT shall be equal to $T_A = 16 \times T_S \times (N_{TA} - 32)$ where T_S is defined in Table 5.9. A positive T_A shall indicate that the uplink signal from the BAT is arriving too late at the BTS and that the BAT shall advance its transmission time earlier by the magnitude of T_A. A negative T_A shall indicate that the uplink signal from the BAT is arriving too early at the BTS and that the BAT shall delay its transmission time later by the magnitude of T_A.

online_power_offset – A 6-bit unsigned integer field that shall indicate the transmission power offset to be applied by the BAT. The step size shall be 1dB. The exact power offset to be applied, in units of dB, shall be equal to the indicated value minus 32. For example, 0x00 and 0x3F would indicate offsets of -32dB and +31dB, respectively. When a positive exact power offset is calculated, the BAT shall increase its transmission power by the magnitude of the exact power offset. When a negative exact power offset is calculated, the BAT shall decrease its transmission power by the magnitude of the exact power offset.

6.7.6 Payload of Paging Request Subheader

When a BTS wishes to contact a BAT that is powered on and registered but which currently has no active connection, the BTS shall transmit a Paging Request subheader. The syntax of the Paging Request subheader shall conform to Table 6.23. The semantics of the fields in the downlink_paging_request_subheader() packet shall be as given immediately below the table.

Table 6.23 Downlink Paging Request Subheader Syntax

Syntax	No. of Bits	Format
downlink_paging_request_subheader() {		
tuid	18	uimsbf (See Section 6.4.3)
rand_access_seq_paging	12	uimsbf
rand_access_PRACH_paging	10	uimsbf
reserved	8	‘11111111’
}		

rand_access_seq_paging – A 12-bit unsigned integer field that shall signal the index of the random access sequence to be used by the BAT in response to a Paging Request. Valid values for this index range from the **random_access_sequence_partition** as signaled in the Downlink Broadcast Control Information (see Table 6.13) to 0xDDF, inclusive.

rand_access_PRACH_paging – A 10-bit unsigned integer field that shall signal the uplink frame number of PRACH to be used by the BAT to transmit the assigned random access sequence.

6.7.7 Payload of Registration Confirmation Subheader

The BTS shall send a Registration Confirmation subheader within $5 \times N_{RR}$ msec after the time that a BTS receives a Registration Request from a BAT. The syntax of the Registration Confirmation subheader shall conform to Table 6.24. The semantics of the fields in the downlink_registration_confirmation_subheader() packet shall be as given immediately below the table.

Table 6.24 Downlink Registration Confirmation Subheader Syntax

Syntax	No. of Bits	Format
downlink_registration_confirmation_subheader() {		
user_id	48	uimsbf (See Section 6.4.1)
tuid	18	uimsbf (See Section 6.4.3)
cid	5	uimsbf (See Section 6.4.3)
bw_allocation_result	2	uimsbf
reserved	7	‘1111111’
}		

bw_allocation_result – A 2-bit unsigned integer field that shall specify the result of bandwidth allocation. ‘0x1’ shall indicate that the bandwidth has been allocated, and the allocation information will be given in the following UL-MAP. ‘0x0’ shall indicate that the bandwidth allocation request was rejected. ‘0x2’ and ‘0x3’ shall be reserved.

6.7.8 Payload of Connection Release Confirmation Subheader

When a BTS receives a Connection Release Request subheader from a BAT, the BTS shall send a Connection Release Confirmation subheader to the BAT and shall release the connection for that **cid**. The syntax of the Connection Release Confirmation subheader shall conform to Table 6.25.

Table 6.25 Downlink Connection Release Confirmation Subheader Syntax

Syntax	No. of Bits	Format
downlink_connection_release_confirmation_subheader() {		
tuid	18	uimsbf (See Section 6.4.3)
cid	5	uimsbf (See Section 6.4.3)
reserved	1	'1'
}		

6.7.9 Payload of Deregistration Confirmation Subheader

When a BTS receives a Deregistration Request subheader from a BAT, the BTS shall send a Deregistration Confirmation subheader to the BAT. The syntax of the Deregistration Confirmation subheader shall conform to Table 6.26.

Table 6.26 Downlink Deregistration Confirmation Subheader Syntax

Syntax	No. of Bits	Format
downlink_deregistration_confirmation_subheader() {		
tuid	18	uimsbf (See Section 6.4.3)
reserved	6	'111111'
}		

6.7.10 Payload of Downlink Data Subheader

The syntax of the Downlink Data subheader shall conform to Table 6.27. The semantics of the fields in the downlink_data_subheader() packet shall be as given immediately below the table.

Table 6.27 Downlink Data Subheader Syntax

Syntax	No. of Bits	Format
downlink_data_subheader() {		
tuid	18	uimsbf (See Section 6.4.3)
cid	5	uimsbf (See Section 6.4.3)
payload_length	11	uimsbf
reserved	6	'111111'
}		

payload_length – An 11-bit unsigned integer field that shall signal one less than the length of the corresponding downlink data payload, as measured in bytes. A downlink data payload shall have a minimum length of 1 byte, and a maximum length of 1500 bytes, where 1500 bytes is the Maximum Transmission Unit (MTU) of Ethernet. Values of **payload_length** between 0x5DC ~0x7FF shall be reserved.

7. DEFINITION OF DRCT

This specification defines a new Low Level Signaling (LLS) table (see A/331 [4]), the Dedicated Return Channel Table (DRCT). Its presence indicates that the payload carried in the PLP-R is for the Dedicated Return Channel. The DRCT is allocated the **LLS_table_id** 0x07 (see Code Point Registry [16]).

Each DRCT shall be included in the LLS in which it is transported in each ATSC 3.0 downlink physical layer frame.

7.1 XML Schema and Namespace

The DRCT shall be represented as an XML document containing a **DRCT** root element that conforms to the definitions in the XML schema that has namespace:

`tag:atsc.org,2016:XMLSchemas/ATSC3/DRC/DRCT/1.0/`

The definition of this schema is in an XML schema file, *DRCT-1.0-20170809.xsd* accompanying this standard. The XML schema xmlns short name shall be "`drct`". The XML schema document for the schemas defined in this document can be found at the ATSC website.

The sub-string part of namespaces between the right-most two '/' delimiters indicate major and minor version of the schemas. The schemas defined in this present document shall have version '1.0', which indicates major version 1 and minor version 0.

The namespace designator, "xs:", and many terms in the "Data Type" column of tables is a shorthand for datatypes defined in W3C XML Schema [9] and shall be as defined there.

In order to provide flexibility for future changes in the schema, decoders of XML documents with the namespaces defined in the present document shall ignore any elements or attributes they do not recognize, instead of treating them as errors.

All element groups and attribute groups are explicitly extensible with elements and attributes, respectively. Elements can only be extended from namespaces other than the target namespace. Attributes can be extended from both the target namespace and other namespaces. If the XML schema does not permit this for some element, that is an error in the schema.

XML schemas shall use **processContents**="strict" in order to reduce inadvertent typos in instance documents. Further, users are encouraged to modify the IETF FDT schema found in RFC 6726 [10] to change **processContents** to "strict". Similarly for the MBMS [11] 3GPP schemas.

In the event of any discrepancy between the XML schema definitions implied by the tables that appear in this document and those that appear in the XML schema definition files, those in the XML schema definition files shall be authoritative and take precedence.

The XML schema document for the schemas defined in this document can be found at the ATSC website: https://www.atsc.org/techdoc/code-point-registry/.

7.2 DRCT Syntax

While the indicated XML schema specifies the normative syntax of the DRCT element, informative Table 7.1 below describes the structure of the DRCT element in a more illustrative way. The specifications following the table give the semantics of the elements and attributes.

Table 7.1 DRCT XML Format

Element or Attribute Name	Use	Data Type	Description
DRCT			Root element of the DRCT.
`@bsid`	1	unsignedShort	Identifies the one or more Broadcast Streams comprising the Services.
`@DestinationIpAddress`	1	IPv4address	A string containing the dotted-IPv4 destination address of the packets carrying data for the Dedicated Return Channel.
`@DestinationUdpPort`	1	unsignedShort	Port number of the packets carrying data for the Dedicated Return Channel.
`@SourceIpAddress`	1	IPv4address	A string containing the dotted-IPv4 source address of the packets carrying data for the Dedicated Return Channel.

7.3 DRCT Semantics

The following text specifies the semantics of the elements and attributes in the DRCT.

DRCT – Root element of the DRCT.

`@bsid` – This list of one or more 16-bit unsigned integers shall identify the Broadcast Stream ID(s) of the original emission signal(s). The value of each `@bsid` shall be the same as the value signaled in **L1D_bsid** in L1-Detail Signaling in the physical layer (see A/322 [2]). In the case that the Service is delivered via channel bonding at the physical layer, the list shall include the BSID value of each RF emission involved in the bonding.

`@slsDestinationIpAddress` – A string containing the dotted-IPv4 destination address of the packets carrying data for the Dedicated Return Channel. The syntax shall be as defined in RFC 3986 [8] Section 3.2.2.

`@slsDestinationUdpPort` – Port number of the packets carrying data for the Dedicated Return Channel.

`@slsSourceIpAddress` – A string containing the dotted-IPv4 source address of the packets carrying data for the Dedicated Return Channel. The syntax shall be as defined in RFC 3986 [8] Section 3.2.2.

***Annex A*: PRACH Design (Informative)**

A.1 STRUCTURE OF RANDOM ACCESS SEQUENCE

As illustrated in Figure A.1.1, the structure of a random access sequence is similar to that of an OFDM symbol. The total duration of a random access sequence is 3.6128ms, including the Guard Time (GT) where nothing is transmitted. The Guard Time is reserved to compensate for propagation delay.

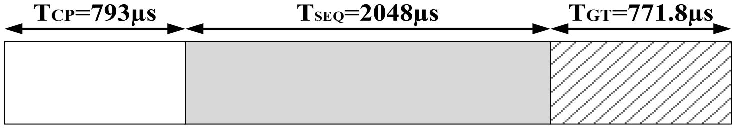

Figure A.1.1 The structure of a random access sequence.

The duration of the random access sequence was set based on the expected maximum round trip delay and the coverage performance. The details are given in the following sections.

A.2 MAXIMUM ROUND TRIP DELAY

In order to cover all users in a BTS's coverage range, the length of a random access sequence should be greater than the maximum round trip delay. In a cell with maximum target radius 100km, the length of a random access sequence T_{SEQ} should satisfy:

$$T_{SEQ} \geq \frac{100km \times 2}{c} = 666.67\mu s \quad \text{(A.1)}$$

where c represents the speed of light.

In order to minimize the loss of orthogonality between PRACH subcarriers and PUSCH (Physical Uplink Shared Channel) subcarriers, the subcarrier spacing Δf in PUSCH should be an integer multiple of the PRACH subcarrier spacing Δf_{RA} ,i.e.,

$$\Delta f_{RA} = \frac{1}{k}\Delta f \quad \text{(A.2)}$$

$$T_{SEQ} = kT_{SYM} \quad \text{(A.3)}$$

where T_{SEQ} stands for the time duration of PRACH, T_{SYM} is the duration of one PUSCH symbol used for data transmission, and k is the integer up-sampling rate in PRACH generator. The value of k is set to 10 in this specification.

The duration of the random access sequence can therefore be obtained as:

$$T_{SEQ} = kT_{SYM} = 2048\mu s \quad \text{(A.4)}$$

The total PRACH duration is the sum of T_{SEQ}, CP of random access sequence T_{CP}, and GT of PRACH T_{GT}. The PRACH duration is equal in length to the first 16 SC-FDMA symbols of an uplink frame. Each symbol has an FFT size of 2048 and a cyclic prefix length of 210 samples for a total length of 2258 samples. Multiplying by 16 symbols and using a sampling frequency of 10 MHz yields a total PRACH time length of 3.6128 ms.

A.3 CP AND GT DURATIONS

According to the random access sequence duration, the required CP (Cyclic Prefix) length of a random access sequence for a maximum cell radius of 100km can be calculated as:

$$T_{cp} = \frac{T_{PRA} - T_{SEQ}}{2} + \frac{d}{2} = \left(\frac{3612.8 - 2048}{2} + \frac{21}{2} \right) \mu s = 792.9 \approx 793 \mu s \tag{A.5}$$

where $d = 21\ \mu sec$ represents the CP length for an SC-FDMA symbol for the DRC uplink (see Table 5.9). The duration of the GT (Guard Time) is:

$$T_{GT} = T_{PRA} - T_{SEQ} - T_{cp} = 771.8 \mu s \tag{A.6}$$

A.4 THE CYCLIC SHIFT SIZE

The random access sequences are generated from a combination of ZC sequences with different root values and cyclic shifts applied to each ZC sequence.

The upper bound of N_{CS} is $\lfloor 1777/2 \rfloor$=888 due to the cyclic shift operation.

The lower bound of the cyclic-shift size, N_{CS}, should guarantee that the duration of the sample sequence is larger than the time delay spread and time uncertainty between asynchronous users.

Therefore, the lower bound of N_{CS} is:

$$N_{cs} \geq \left\lceil \left(\lambda r - \tau_{ds} \right) \frac{N_{zc}}{T_{SEQ}} \right\rceil = 563 \tag{A.7}$$

where the constant $\lambda = 20/3 \mu s/km$, $r \leq 100km$ represents the radius of the cell, $\tau_{ds} = 16.67\ \mu sec$ represents the maximum delay spread, and $N_{zc} = 1777$ is the ZC sequence length.

In this specification, $N_{CS} = 888$ is chosen as the size of cyclic-shift. When generating random access sequences, cyclic shifting is used to generate two random access sequences for each root value.

Annex B: Downlink Synchronization (Informative)

B.1 SYNCHRONIZATION ERROR ANALYSIS

The downlink broadcast operates frame by frame. BATs can acquire synchronization only at the beginning of a downlink frame. The synchronization error accumulates during the remaining time of the downlink frame. The maximum accumulated error is determined by the length of the downlink frame and the clock precision of a BAT.

According to A/322 [2], the maximum length of a downlink broadcast frame is 5 seconds. The typical precision of a crystal clock is 10ppm. Thus the maximum accumulated error can be bounded as:

$$5 \times 10 \times 10^{-6} s = 5 \mu s \quad \text{(B.1)}$$

Similarly, if the precision of a clock at a BAT is 50ppm, the maximum accumulated error will be 25 μs. This is tolerable by the DRC system.

Annex C: Physical Layer Parameters of PLP-R (Informative)

C.1 PHYSICAL LAYER PARAMETERS OF PLP-R

As noted in Section 4.4, PLP-R is one of the PLPs used for signalling and interactive data transfer from the BTS to BATs. Therefore, PLP-R is more delay-sensitive than the data carried through other PLPs. To reduce the end-to-end delay in PLP-R, a short FEC code length and a small time interleaver depth should be used in PLP-R. In contrast, a long FEC code length and a large time interleaver depth can be used in downlink broadcast for higher processing gain. This difference can result in the need for PLP-R to be configured with a more robust ModCod than that used for other PLPs in order to obtain an appropriate relative level of performance for a given receiver SNR.

To avoid this possible coverage imbalance between different PLPs, the physical layer parameters, i.e., the ModCod, of both PLP-R and other PLPs should be carefully chosen. The ModCod schemes of both PLP-R and other PLPs should be chosen to guarantee that all receivers in the desired coverage area of the BTS can correctly receive all PLPs including PLP-R.

Configuration of ModCod schemes for PLP-R and other PLPs is determined by the network operator. Some guidelines on the configuration of ModCod schemes for PLP-R are given in the following.

C.2 FORWARD ERROR CORRECTION (FEC) OF PLP-R

Since PLP-R is delay sensitive, the use of the Low Density Parity Check (LDPC) code with FEC Frame size N_{inner} =16200 bits is recommended for PLP-R configuration.

C.3 TIME INTERLEAVING OF PLP-R

There are three interleaving modes in A/322 [2], i.e., no time interleaver, Convolutional Time Interleaver (CTI) mode and Hybrid Time Interleaver (HTI) mode. The following information is noted in Section 7.1.1 in A/322 [2]:

> ‘When, as determined at the input to the time interleaver, a complete delivered product is composed of only a single constant-cell-rate PLP or is composed of a single constant-cell-rate Core PLP and one or more constant-cell-rate Enhanced PLPs layered division multiplexed with that Core PLP, the PLP(s) comprising that complete delivered product shall be configured with one of the following time interleaver modes: no time interleaving, CTI mode, or HTI mode.
>
> When, as determined at the input to the time interleaver, a complete delivered product is composed of PLPs having characteristics different from those described in the preceding paragraph, the PLPs comprising that complete delivered product shall be configured with one of the following time interleaver modes: no time interleaving or HTI mode.’

According to A/322, the CTI mode can only be configured when only a single constant-cell-rate PLP is used for a complete delivered product. If only DRC is implemented in an ATSC 3.0 system without any downlink broadcast, it is possible that there exists only one PLP, i.e., the PLP-R itself. In this case, either the CTI or HTI mode can be used. If DRC is implemented together with additional ATSC 3.0 downlink broadcast, at least two PLPs would need to be used. One of the PLPs would be PLP-R, and the others could be used for ATSC 3.0 downlink broadcast. In these cases, the HTI mode can be used.

The time interleaving for the HTI mode consists of an optional Cell Interleaver, a mandatory Twisted Block Interleaver (TBI), and an optional Convolutional Delay Line (CDL). Since the signalling is carried in PLP-R is more delay sensitive than the broadcast data in other PLPs, time interleaving with long latency should not be used for PLP-R. It is recommended that **L1D_plp_HTI_inter_subframe** be set 0 to bypass the CDL in HTI mode.

Annex D: Signaling Overhead Analysis (Informative)

D.1 SIGNALING OVERHEAD ANALYSIS FOR THE DOWNLINK

The number of signaling bits for each downlink MAC PDU subheader is shown in Table D.1.1. Assuming that there are N_{users} users in the cell, without considering the overhead of BCI and UL-MAP, the overall signaling overhead in the downlink system can be calculated as $N_{users} \times C_{ov}$, where C_{ov} is the average cost in bits.

According to the resource allocation scheme given in Section 5, taking 6Mhz of uplink channel bandwidth as an example, the maximum number of users that can be served within a DRC uplink frame is:

$$N_{\max} = \frac{1320 \text{ tiles} - 72\text{tiles}}{3 \text{ tiles}} = 416 \tag{D.1}$$

In extreme conditions, the maximum cost is $C_{ov} = 80$ bits and the maximum number of users per 10ms uplink frame is N_{max}=416. Thus, the theoretical maximum signaling overhead could reach 100×416×80=3.33 Mbps.

Assuming the ten subheaders listed in Table D.1.1 appear with equal probability, the average subheader length is 40 bits. Thus, the average signaling overhead in that scenario would be 1.66 Mbps, when the maximum number of users (416) appear in each uplink frame. The average signaling overhead would be 166 Kbps when an average of 41.6 users appear in each uplink frame.

Table D.1.1 Types and Lengths of Downlink Subheaders to Support DRC Operation

Subheader Index	Subheader	Total bits
0	Random Access Response	56
1	Bandwidth Allocation Response	32
2	Status Report Request	24
3	ACK Message	34 + variable
4	Online Adjustment Message	32
5	Paging Request	48
6	Registration Confirmation	80
7	Connection Release Confirmation	24
8	Deregistration Confirmation	24
9	Downlink Data	40

End of Document