# Final draft ETSI EN 300 708 V1.2.1 (2002-12)

*European Standard (Telecommunications series)*

## Television systems;
## Data transmission within Teletext

European Broadcasting Union

Union Européenne de Radio-Télévision

EBU·UER

Reference
REN/JTC-TTEXT-DT-R1

Keywords
broadcasting, data, Teletext, TV

***ETSI***

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

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

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

***Important notice***

Individual copies of the present document can be downloaded from:
http://www.etsi.org

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

# Contents


Intellectual Property Rights ........ 6

Foreword ........ 6

1 Scope ........ 7

2 References ........ 7

3 Definitions and abbreviations ........ 7
3.1 Definitions ........ 7
3.2 Abbreviations ........ 8

4 Page Format - Clear ........ 9
4.1 Introduction ........ 9
4.1.1 General points ........ 9
4.1.2 Advantages ........ 9
4.1.3 Disadvantages ........ 9
4.2 Data streams ........ 10
4.2.1 Application data blocks ........ 10
4.2.2 Structure Header ........ 10
4.2.3 Bundle Information structure ........ 11
4.2.3.1 Format ........ 11
4.2.3.2 Structure Header ........ 11
4.2.3.3 Checksum ........ 12
4.2.3.4 Number of Applications ........ 12
4.2.3.5 Application Type ........ 12
4.3 Coding of the Teletext page ........ 12
4.3.1 Page header ........ 12
4.3.1.1 Page number ........ 13
4.3.1.2 Subcode ........ 13
4.3.1.3 Control bits C4 to C14 ........ 14
4.3.1.4 Data Bytes ........ 14
4.3.2 Packets 1 to 25 ........ 14
4.3.2.1 Valid packets ........ 14
4.3.2.2 Packet encoding ........ 14
4.3.2.3 Block Separator ........ 15
4.3.2.4 Block Pointer ........ 15
4.3.2.5 Filler Byte ........ 15
4.3.2.6 Mapping ........ 15
4.3.2.7 Byte coding ........ 16
4.3.3 Packet 28 ........ 16
4.3a Page Transmission ........ 16
4.4 Coding of the packets ........ 17
4.4.1 Page header ........ 17
4.4.2 Packets 1 to 25 ........ 19
4.4.2.1 Transmission order ........ 19
4.4.2.2 The Structure Header (SH) ........ 20
4.4.2.3 Packing the data into the pages ........ 20
4.4.3 Packet 28 ........ 21
4.5 Providing a service according to Page Format - Clear ........ 21
4.5.1 General points on Page Format - Clear ........ 21
4.5.2 Encoding scheme for electronic data transmission ........ 22

5 Page Format - CA ........ 22
5.1 General points ........ 22
5.2 Advantages ........ 22
5.3 Disadvantages ........ 22
5.4 Method of coding ........ 22
5.4.1 Teletext page data ........ 23

5.4.1.1 Scrambled Data Pages ....24
5.4.1.2 Pages containing Reformatted Data ....24
5.4.2 The Page Key Packet ....24
5.4.2.1 Service Mode bits ....25
5.4.2.2 Service Identification ....25
5.4.2.3 Continuity and Repeat Indicators ....26
5.4.2.4 Packet Flags ....26
5.4.2.5 Data Length ....26
5.4.2.6 Scrambling Method ....26
5.4.2.7 The Cyclic Redundancy Check (CRC) word ....27
5.4.3 Terminal Equipment Addressing Pages ....27
5.4.3.1 System Key Packet ....27
5.4.3.2 Shared User Packets ....28
5.4.3.3 Unique User Packets ....28
5.4.3.4 Service/Page Link packets ....29
5.4.3.5 Link to Independent Data Line Services ....29
5.5 Security of Page Format - CA ....29
5.5.1 Cipher feedback algorithm ....30
5.5.2 One-way function ....30
5.5.3 Text scrambling ....31

6 Independent Data Lines (IDL) ....31
6.1 General points ....31
6.2 Advantages ....31
6.3 Disadvantages ....31
6.4 Methods of coding ....32
6.4.1 Designation code ....32
6.4.1.1 Transmission multiplexing ....32
6.4.2 Data Channel addressing ....32
6.5 IDL Format A ....33
6.5.1 Format Type (FT) ....33
6.5.2 Interpretation and Address Length (IAL) ....33
6.5.3 Service Packet Address (SPA) ....34
6.5.4 Repeat Indicator (RI) ....34
6.5.5 Continuity Indicator (CI) ....34
6.5.6 Data Length (DL) byte ....34
6.5.7 User Data Group ....35
6.5.7.1 Dummy bytes ....35
6.5.8 Cyclic Redundancy Check (CRC) word ....35
6.5.8.1 Check word generation ....35
6.5.8.2 Check result ....35
6.5.9 Transmission sequence ....36
6.6 Datavideo format ....36
6.6.1 Packet address ....36
6.6.2 Control Bytes (CB) ....36
6.6.2.1 Packet Continuity Indicator (CI) ....36
6.6.3 Masking indicator ....36
6.6.4 Packet type indicator ....37
6.6.5 User data group ....37
6.6.6 Cyclic Redundancy Check (CRC) word ....37
6.6.6.1 Check word generation ....37
6.6.6.2 Check result ....37
6.7 Low bit-rate audio ....37
6.7.1 Method of coding ....38
6.7.1.1 Decoder action ....38
6.7.2 Programme-related audio service ....38
6.7.2.1 Service Byte (SB) ....38
6.7.2.2 Control Byte (CB) ....38
6.7.2.3 Audio data ....39
6.7.3 Programme independent audio service ....39
6.7.3.1 Service Byte (SB) ....39
6.7.3.2 Control Byte (CB) ....39

6.7.3.3 Audio data ....................................................................................................39
6.8 IDL Format B ....................................................................................................39
6.8.1 Packet structure ....................................................................................................39
6.8.1.1 Format Type ....................................................................................................40
6.8.1.2 Application Identifier ....................................................................................................40
6.8.1.3 Continuity Index ....................................................................................................40
6.8.1.4 User Data ....................................................................................................40
6.8.1.5 FEC ....................................................................................................40
6.8.2 Forward Error Correcting algorithm ....................................................................................................41
6.8.2.1 Introduction ....................................................................................................41
6.8.2.2 Encoding ....................................................................................................41
6.8.2.3 Calculating FEC suffix bytes ....................................................................................................41
6.8.2.4 Decoding ....................................................................................................42
6.8.2.5 Detecting and correcting single byte errors ....................................................................................................42
6.8.2.6 Correcting data blocks ....................................................................................................43
6.8.3 Effective data rate ....................................................................................................43

7 IDL - CA (type A) ....................................................................................................43
7.1 General points ....................................................................................................43
7.2 Advantages ....................................................................................................43
7.3 Disadvantages ....................................................................................................44
7.4 Methods of coding ....................................................................................................44
7.4.1 Block Separator ....................................................................................................44
7.4.2 Block Formats ....................................................................................................44
7.4.3 Block Format A ....................................................................................................44
7.4.3.1 Block Types ....................................................................................................44
7.4.3.2 Primary Block Key Messages ....................................................................................................45
7.4.3.3 Secondary Block Messages and Scrambled User Data ....................................................................................................45
7.4.3.4 System-Key Message Block ....................................................................................................45
7.4.3.5 Shared-User Message Block ....................................................................................................45
7.4.3.6 Unique User Message Block ....................................................................................................46
7.4.3.7 Service Address Message Block - Independent Data Service ....................................................................................................46
7.4.3.8 Service Address Message Block - Link to Page Format - CA ....................................................................................................46

**Annex A (informative): Additional information for Page Format - Clear ....................................................................................................47**

A.1 Use of Block Pointer, Block Separator and Block Length information ....................................................................................................47

History ....................................................................................................48

# Intellectual Property Rights

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

All published ETSI deliverables shall include information which directs the reader to the above source of information.

# Foreword

This European Standard (Telecommunications series) has been produced by Joint Technical Committee (JTC) Broadcast of the European Broadcasting Union (EBU), Comité Européen de Normalisation ELECtrotechnique (CENELEC) and the European Telecommunications Standards Institute (ETSI), and is now submitted for the ETSI standards One-step Approval Procedure.

NOTE: The EBU/ETSI JTC Broadcast was established in 1990 to co-ordinate the drafting of standards in the specific field of broadcasting and related fields. Since 1995 the JTC Broadcast became a tripartite body by including in the Memorandum of Understanding also CENELEC, which is responsible for the standardization of radio and television receivers. The EBU is a professional association of broadcasting organizations whose work includes the co-ordination of its members' activities in the technical, legal, programme-making and programme-exchange domains. The EBU has active members in about 60 countries in the European broadcasting area; its headquarters is in Geneva.

European Broadcasting Union
CH-1218 GRAND SACONNEX (Geneva)
Switzerland
Tel: +41 22 717 21 11
Fax: +41 22 717 24 81

| **Proposed national transposition dates** | |
|---|---|
| Date of latest announcement of this EN (doa): | 3 months after ETSI publication |
| Date of latest publication of new National Standard or endorsement of this EN (dop/e): | 6 months after doa |
| Date of withdrawal of any conflicting National Standard (dow): | 6 months after doa |

# 1 Scope

The present document describes the various ways in which Teletext may be used to carry non-Teletext services. It should be used in conjunction with EN 300 706 [1]. An example is fully described in EN 300 707 [2]. The present document includes additional practical information on implementing a data service of this type.

Hooks into existing Teletext services may be provided from within the application which is carried as a non-Teletext service. A data broadcast application may be pointed to from the Magazine Inventory Page (MIP) or it may be allocated a specific page number. A Code of Practice (CoP) shall ensure that services destined for the consumer may be found quickly by "low-end" Teletext decoders but this is outside the scope of the present document.

There are two methods available for carrying data services. The first method carries the data within Teletext pages. The data in these pages is not suitable for direct display by a Teletext decoder and shall normally be allocated a special page number and/or have the display inhibited. The second method carries the data within Independent Data Lines (IDL) and these are independent of the page service.

With both Page and IDL formats there exist versions which offer Conditional Access (CA).

There are other specific IDL data services which have been defined but it is beyond the scope of the present document to cover all of these.

# 2 References

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

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

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

[1] ETSI EN 300 706: "Enhanced Teletext specification".

[2] ETSI EN 300 707: "Electronic Programme Guide (EPG); Protocol for a TV Guide using electronic data transmission".

[3] ETSI EN 300 231: "Television systems; Specification of the domestic video Programme Delivery Control system (PDC)".

[4] ETSI TR 101 231: "Television systems; Register of Country and Network Identification (CNI) and of Video Programming System (VPS) codes".

# 3 Definitions and abbreviations

## 3.1 Definitions

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

**20 ms rule:** insertion of a one field period delay (20 ms) between the transmission of the Teletext page header and the next packet for that page

NOTE: The delay provides sufficient time for all existing decoders to implement a memory clear.

**application data:** raw data from the service provider prior to encoding in the format of the chosen transmission method

**bit numbering within bytes:** in common with earlier and existing Teletext specifications, the bits of a Teletext data byte are numbered 1 to 8 (LSB to MSB)

**data stream:** sequence of bytes carried in a uniquely addressable data service

**encryption:** process whereby a sequence of data is made secret

NOTE: See clause 5.5 for further information.

**Hamming 8/4:** coding scheme which adds four protection bits to four message bits such that a receiver can correct a single bit error and detect a double bit error in the resulting 8-bit byte

NOTE: Defined in full in EN 300 706 [1].

**network operator:** organization responsible for the compilation of the Teletext service for insertion into the available Vertical Blanking Interval (VBI) lines

**scrambling:** process whereby a sequence of data is made unintelligible

NOTE: See clause 5.5 for further information.

**service provider:** organization responsible for the creation and supply of the data service application

**transmitted bit order:** unless otherwise stated, the bits of a Teletext byte are transmitted least significant bit first

## 3.2 Abbreviations

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

| | |
|---|---|
| AUDETEL | AUdio DEscription of TELevision |
| BL | Block Length |
| BP | Block Pointer |
| BS | Block Separator |
| BT | Block Type |
| CA | Conditional Access |
| CI | Continuity Index |
| CoP | Code of Practice |
| EPG | Electronic Programme Guide |
| FB | Filler Bytes |
| IDL | Independent Data Lines |
| LSB | Least Significant Bit |
| MIP | Magazine Inventory Page |
| MSB | Most Significant Bit |
| RI | Repeat Indicator |
| SH | Structure Header |
| VBI | Vertical Blanking Interval |

# 4 Page Format - Clear

## 4.1 Introduction

### 4.1.1 General points

The application data is broadcast within uniquely addressable Teletext pages. Unlike the Page Format - CA method described in clause 5, any conditional access requirements must be included in the data prior to encoding.

It is possible to carry several independent applications within a page identified by a 3-digit page number. Up to 15 separate data streams can be accommodated within one such page, individual streams being uniquely addressable via part of the subcode component of the page header. Each data stream can carry multiple applications. Within each stream, blocks of application data are preceded by service identifier and length information. Pointers placed at the start of each packet indicate the location of block boundaries within the page.

Data streams can be broadcast with or without adherence to the 20 ms rule. It is possible to provide a total service, which is constructed from more than one data stream, requiring the receiver to handle more than one data page. An example is an electronic programme guide service as defined in EN 300 707 [2]. One stream, carrying "now and next" type information, is intended for reception by all receivers and is transmitted obeying the 20 ms rule. A second stream, providing details on future programmes, is aimed at more sophisticated receivers which do not require the 20 ms delay. The two streams can be interleaved implying a suitable receiver may have to acquire and process two or more separate data pages simultaneously.

### 4.1.2 Advantages

This method is appropriate for reception by all existing Teletext decoders. A default coding scheme is defined in order that services may be offered using standard Teletext systems. To allow for future applications, the default method can be overridden by including an additional packet 28 per page to define the coding scheme in use.

The network operator requires no knowledge of the application data he is transporting apart from knowing where each block commences and how long it is. Each of the application data bytes is assumed to contain 8 usable bits and it is up to the data service provider to determine how they are used. It is also left to the data service provider to apply access control or any additional protection to those parts of his service that require it.

It is possible to sub-divide a service into those parts which will be required by all decoders and those parts which will only be required by some, perhaps more sophisticated, decoders.

It is not necessary to synchronize the application data to the Teletext page format. Block level synchronization at the decoder is assured by means of a simple but robust method of dual pointing to each block start location.

### 4.1.3 Disadvantages

This method is not efficient for sending a small amount of data as there is always the overhead of including a page header and the optional, but useful, packet 28. Also, data transmission efficiency is further reduced if the pages are sent in a fragmented way in order to interleave the data service with the primary Teletext service.

## 4.2 Data streams

An application data block is prepared for transmission by the addition of service identifier and block length information known as a Structure Header. Ultimately such entities are concatenated to form a data stream that is then mapped to Teletext pages. It is also necessary to include additional data in the stream in the form of a Bundle Information Structure to identify the different applications that may be present. A representative data stream is shown in figure 1.

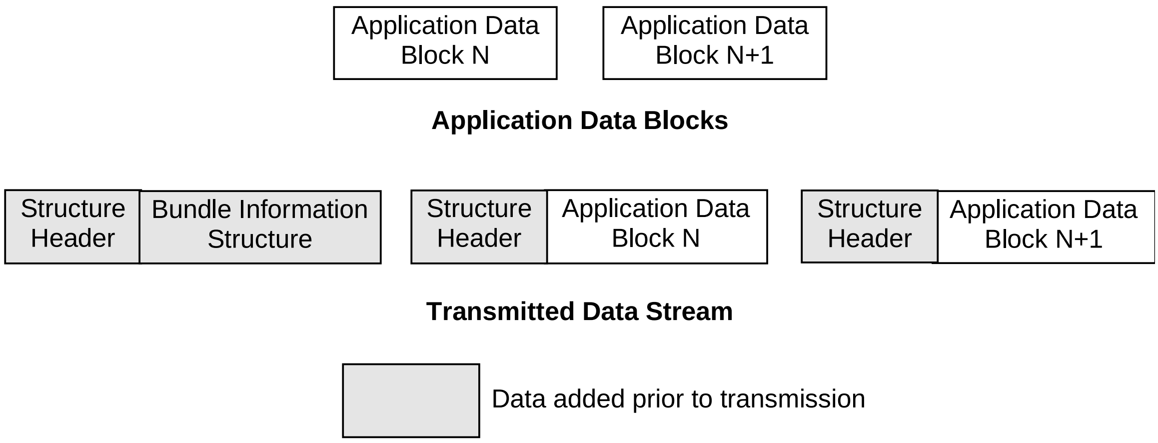



**Figure 1: Formation of a data stream from application data blocks**

### 4.2.1 Application data blocks

The application data is assumed to exist in the form of individual blocks of up to 2 047 bytes. The application data is broadcast without further error protection or access control data being added to the data stream during the encoding and transmission process.

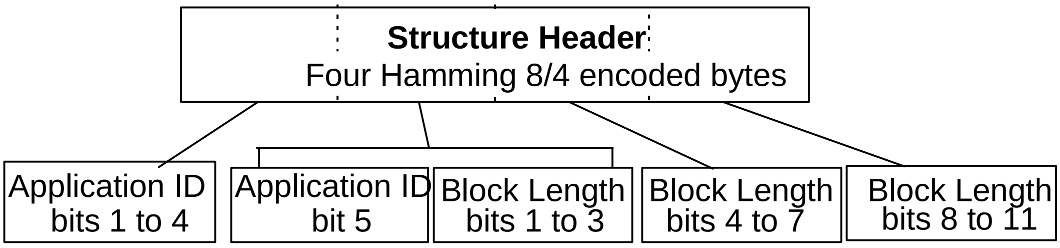
### 4.2.2 Structure Header

A 16-bit Structure Header (SH), figure 2, shall be added to the start of each application data block. The first 5 data bits provide the Application ID value. This allows up to 31 different services to be interleaved within a particular data stream identified by a 3 digit page number and a specific value for the S3 component of the page sub-code (see clause 4.3.1.2).

The actual value for the Application ID is allocated by the overall service provider. The Application ID value of 0 is reserved for the Bundle Information structure, clause 4.2.3, which provides the system with information about the services carried in this data stream.

The remaining 11 data bits of the Structure Header define the Block Length (BL). This indicates the size in bytes of the following block of application data. The maximum block length is 2 047 bytes.

The 16 bits of the Structure Header are Hamming 8/4 encoded prior to transmission.




NOTE: Fields shown in transmission order from left to right.
LSB transmitted first. Only the message bits of the Hamming 8/4 encoded bytes are shown.

**Figure 2: The Structure Header**

## 4.2.3 Bundle Information structure

A Bundle Information structure (BI) shall be inserted in the data stream at periodic intervals to identify the different applications present. For each application (except for itself) the structure defines an identifier and links it to a code that indicates the application type. Once a receiver has this information, it can identify the data blocks in the stream belonging to a particular application by the Application ID value carried in the Structure Header.

An application may specify a minimum rate at which BI structures should be inserted in the data stream.

### 4.2.3.1 Format

The format of a BI structure is shown in figure 3 and the syntax is described in table 1. All the data is Hamming 8/4 encoded before transmission. All bits are transmitted LSB first.

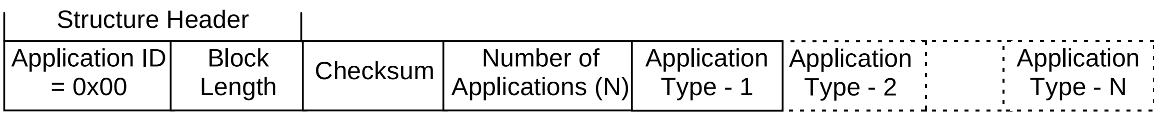



**Figure 3: Format of the Bundle Information structure**

**Table 1: Syntax for the Bundle Information structure**

| Bundle Information Structure | No. of data bits | No. of transmitted bytes |
|---|---|---|
| Application ID (= 0x00) } Structure<br>Block Length } Header | 5<br>11 | 4 |
| Checksum | 8 | 2 |
| Number of Applications | 8 | 2 |
| for (i = 1; i < = Number of Applications; i++) { | | |
| Application Type | 16 | 4 |
| } | | |
| NOTE: All bits are Hamming 8/4 encoded prior to transmission. | | |

### 4.2.3.2 Structure Header

As defined in clause 4.2.2. The Application ID value is fixed at 0x00.

#### 4.2.3.3 Checksum

The checksum shall be calculated before the data bits have been Hamming 8/4 encoded.

The checksum includes the contents of the Application ID, Block Length, Number of Applications and Application Type fields. It is calculated as follows: The data bits of these fields are concatenated and divided into 4-bit nibbles. The nibbles are then expanded into 8-bit bytes with the four most significant bits set to 0. These bytes are summed (modulo 256) and the checksum is (0x100 - sum of bytes) modulo 256.

The checksum is split into two nibbles and each is Hamming 8/4 encoded prior to transmission.

#### 4.2.3.4 Number of Applications

This Number of Applications field defines the number of applications carried in the data stream. The maximum number permitted is 31. The 3 most significant bits in this 8-bit field shall be set to 0. The field is split into two nibbles and each is Hamming 8/4 encoded prior to transmission.

#### 4.2.3.5 Application Type

The code for an application as allocated in TR 101 231 [4]. The Application ID in the Structure Headers for the data blocks belonging to this application is assigned the current loop index value, i, as defined in table 1. The field is split into two nibbles and each is Hamming 8/4 encoded prior to transmission.

## 4.3 Coding of the Teletext page

A data stream, as defined in clause 4.2, is transmitted by mapping it to successive versions of a Teletext page. The page consists of a page header, packets 1 up to 25 carrying the data and an optional packet 28.

### 4.3.1 Page header

The transmission of a page commences with a page header (packet 0), figure 4. The byte coding is fully defined in EN 300 706 [1]. The bytes are numbered 1 to 45. The page header is essential but it shall not contain any application data. Other than the page address and control bytes, it may be identical to all others in the same magazine.

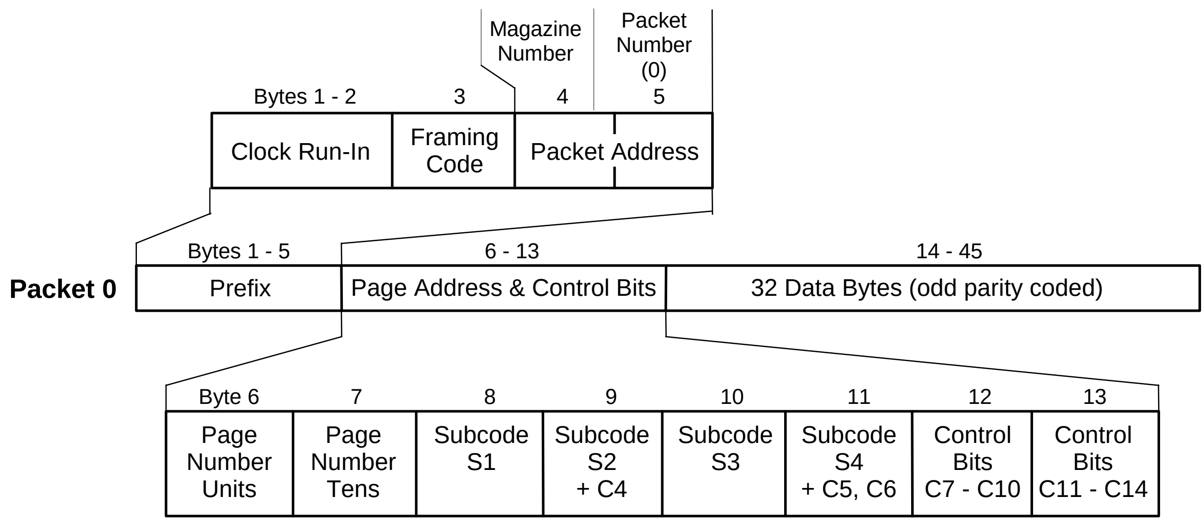



**Figure 4: Page Header**

### 4.3.1.1 Page number

Since a Page Format - Clear encoded page does not contain data intended for direct display by standard Teletext decoders, the use of a page number which includes at least one hexadecimal digit for Page Number Tens or Page Number Units is recommended. The reserved page number value of FF is not permitted for any data service.

Individual applications may suggest the use of a particular page number. However, it is recommended that the page number in use is always defined in the Magazine Inventory Page (MIP) for that magazine, see EN 300 706 [1].

### 4.3.1.2 Subcode

The subcode fields of the page header convey data stream identity, continuity and page length information, figure 5.

NOTE: The coding of the subcode fields given here is different from that defined in EN 300 706 [1] for both normal Teletext pages and enhancement data pages.

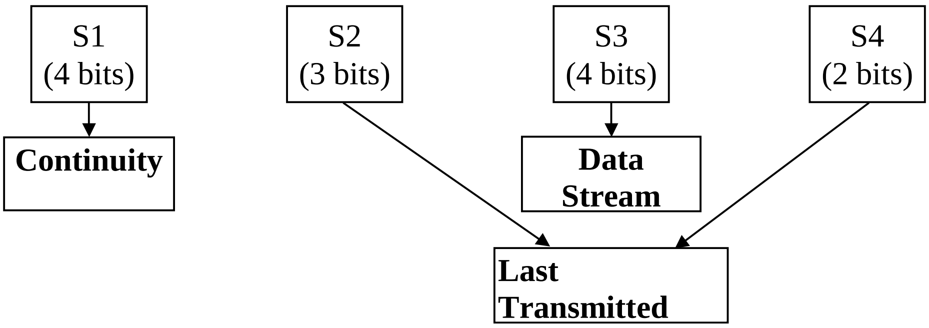



**Figure 5: Function of the Teletext page subcode bytes**

**S3 - Data Stream Identifier:** This 4-bit field allows up to 15 different data streams to share the same 3-digit page number, table 2. The value 0xF is reserved for future use. Thus the location of a particular data stream within the total Teletext transmission is defined uniquely by a 3-digit page number and a Data Stream Identifier value.

If the LSB is set to 0, the 20 ms transmission rule (see EN 300 706 [1]) must be applied before any further packets belonging to the page can be transmitted. If the LSB is set to 1, the next packet for the page may be transmitted on the next TV line.

**Table 2: Values of S3 - Data Stream Identifier**

| Hex value of S3 | Interpretation |
|---|---|
| 0 | 20 ms rule applies - data stream 1 |
| 1 | No 20 ms rule - data stream 2 |
| 2 | 20 ms rule applies - data stream 3 |
| 3 | No 20 ms rule - data stream 4 |
| 4, 6, 8, A, C and E | 20 ms rule applies - data stream 5, 7, 9, 11, 13 and 15 respectively |
| 5,7,9,B and D | No 20 ms rule - data stream 6, 8, 10, 12 and 14 respectively |
| F | Reserved for future use, no 20 ms rule applies |

**S1 - Continuity Index (CI):** This 4-bit field increments modulo 16 for each page which contains different application data. The value is data stream specific and must be sequential in the range 0 to F, with no missing values. If a page is transmitted in fragments (see clause 4.4), the same CI value shall be used for each page header required.

**S2 and S4 - Last Transmitted Packet:** The S2 and S4 fields are combined to make a 5-bit field (S2 providing the three LSBs) which indicates the last packet that will be transmitted as part of the page defined by the current page header. The encoded value shall represent a packet address in the range 1 to 25. A receiver may use this information to aid "end-of-page" detection.

If the page is transmitted in fragments, the same Last Transmitted Packet value shall be used for each page header required.

The presence of a packet 28 has no effect on the calculation of the Last Transmitted Packet value.

NOTE: As the highest packet number is 25, the individual S2 and S4 fields can never have values of 7 and 3 simultaneously. Thus the reserved subcode value of 3F7F defined in EN 300 706 [1] can never occur.

#### 4.3.1.3 Control bits C4 to C14

The C4 control bit (Erase Page) shall be set to 1 the first time the page header is sent with a new Continuity Index value. If the page is transmitted in fragments, subsequent page headers for the same data stream with the same CI value shall have set C4 to 0.

The C5 (Newsflash) and C6 (Subtitle) bits shall be set to 0 to avoid problems with existing decoders which look explicitly for pages with either of these bits set.

It is recommended that the C7 (Suppress Header), C9 (Interrupted Sequence) and C10 (Inhibit Display) control bits are all set to 1 to inhibit the inadvertent display of the header or the body of the page.

The use of the C8 (Update) bit by Page Format - Clear pages is reserved for future use. For transmissions designed to this specification it should be set to 0.

The C11 control bit (Magazine Serial) shall be set to match all the other page headers in the complete Teletext service.

The C12, C13 and C14 (National Option Character Subset) control bits have a display-related function according to EN 300 706 [1]. Their use is reserved.

#### 4.3.1.4 Data Bytes

The contents of bytes 14 to 37 of the page header should normally match those of the other pages in the same magazine. This is to avoid possible display problems with viewers of the normal Teletext service that shares this Magazine.

The last 8 bytes (38 to 45) of the header shall be used in a way that takes account of the rest of the Teletext service. Where a real-time clock can be inserted, it shall be presented in the same format as used on the normal Teletext service. Where this is not possible, these 8 bytes should be filled with spaces. In the case where there is no other Teletext service then the local time shall be placed in these 8 bytes according to the EN 300 706 [1].

### 4.3.2 Packets 1 to 25

Packets 1 up to 25 carry the application data streams.

#### 4.3.2.1 Valid packets

If a packet 28 is included in the page then packets 1 to 25 can be used to carry the data. The implied default condition in the absence of a packet 28 is that packets 1 up to 23 are used.

A code of practice for a particular application may state its own upper limit ($\leq 25$) in which case packets with higher addresses shall not be used. Where multiple services are combined in one data stream, the lowest maximum value shall determine the highest packet address that can be used in that data stream.

#### 4.3.2.2 Packet encoding

The packets are encoded in ascending address order up to and including the last packet indicated by the S2 and S4 components of the page subcode. There shall be no missing packets within this range.

Each page is constructed in the manner shown in figure 6. For clarity, the packet identity and addressing bytes, numbered 1 to 5, are not shown.

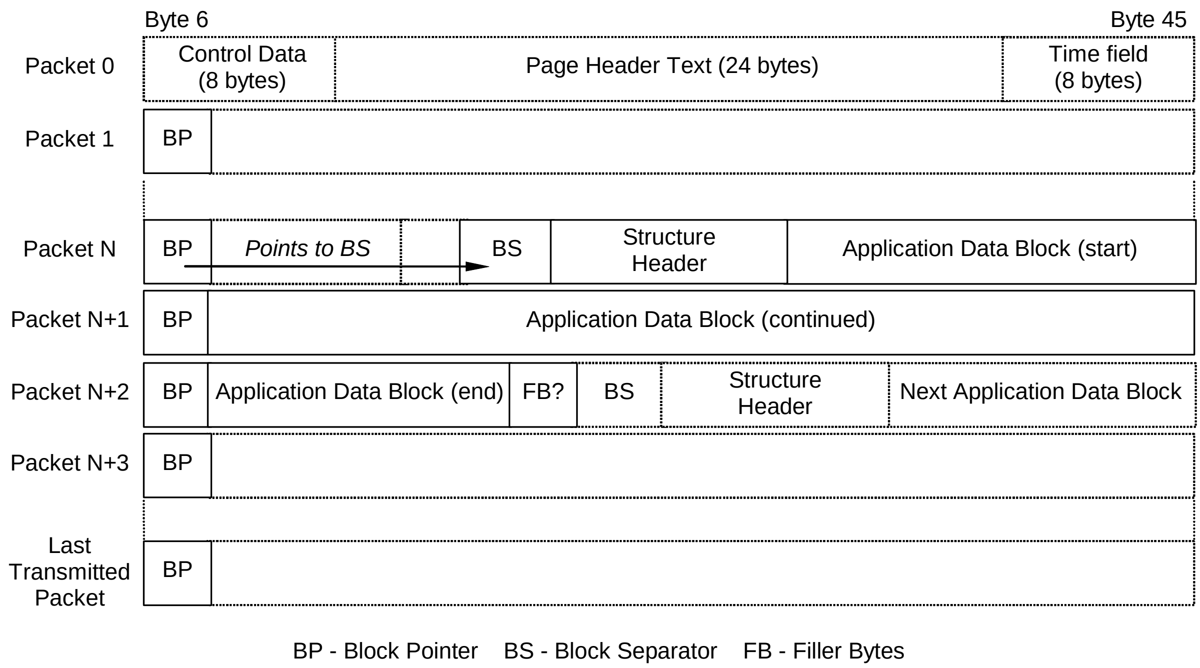



**Figure 6: Example of a page according to Page Format - Clear**

### 4.3.2.3 Block Separator

A Block Separator (BS) byte shall be inserted in the data stream immediately before each Structure Header. The BS transmission code is 0xA1 (i.e. the Hamming 8/4 encoded value of 0x0C).

### 4.3.2.4 Block Pointer

Byte 6 of every packet is reserved as a Block Pointer (BP). This byte shall be Hamming 8/4 encoded and the four message bits shall indicate the position of the first Block Separator code within the packet, if any. The coding for BP is given in table 3.

**Table 3: Values for Block Pointer (BP)**

| Hex value for BP | Interpretation |
|---|---|
| 0 to C | Points to BS in byte position (BP x 3) + 7 |
| D | No BS in this packet |
| E and F | Reserved for future use |
| NOTE: Byte positions are numbered from 1 to 45. | |

### 4.3.2.5 Filler Byte

Filler Bytes (FB) shall be inserted in the data stream to occupy any unused bytes between the end of an application data block and the next Block Separator. The FB transmission code is 0x5E (i.e. the Hamming 8/4 encoded value of 0x03).

### 4.3.2.6 Mapping

According to table 3, the Block Pointer may point to 1 of 13 possible byte positions in the packet. The application data stream shall be mapped such that the Block Separator codes are placed at locations that are addressable in this way. At the start of the transmission the first Block Separator shall occur at the byte 7 of packet 1.

With very short data blocks it is possible that more than one block may commence in the same packet. Where this occurs, the Block Pointer shall point to the first Block Separator. Subsequent Block Separators shall be placed immediately after the end of the preceding data block.

If the first Block Separator within a packet does not automatically occur at one of the 13 possible start locations, a sufficient number of Filler Bytes (clause 4.3.2.5) shall be inserted after the end of the previous data block until the Block Separator is aligned correctly.

Where there is a need to pad to the end of a packet or page, perhaps when there is no application data available or to simplify the encoding process, the Filler Byte code shall be used throughout.

NOTE: As data blocks can be up to 2 047 bytes long, a given block may start within one Teletext page and finish in a different one.

#### 4.3.2.7 Byte coding

As a result of the mapping process bytes 6 to 45 of a given packet will contain both 8-bit application data and Hamming 8/4 coded bytes inserted by the encoding equipment. Only byte 6 can be guaranteed to be coded the same way in every packet.

### 4.3.3 Packet 28

As an option a page may include a packet 28 to define the function and coding of the page. In the absence of the packet a receiver shall assume that packets 1 up to 23 are coded according to clause 4.3.2.

When present the packet 28 shall be transmitted after the page header and before any packets with addresses in the range 1 up to 25. Its presence does not affect the Last Transmitted Packet calculation.

The packet shall be coded as a packet 28/0, Format 1 for Data Broadcasting Applications as defined in EN 300 706 [1]. The designation code shall be set to 0000. The contents of the rest of the packet are given in table 4.

**Table 4: Coding of Packet 28 for Page Format - Clear**

| Triplet | Bits | Function |
|---|---|---|
| 1 | 1 - 4 | **Page Function**<br>These bits define the function of the data in packets 1 up to 25 of the current page. They shall be set to 0001 to indicate a Data Broadcasting Page (Page Format - Clear) defined according to this clause in the present document. |
| 1 | 5 - 7 | **Page Coding**<br>These bits define the coding of packets 1 to up to 25 of the current page. These bits shall be set to 001 to indicate the coding scheme defined in clause 4.3.2. All other values are reserved for future use. |
| 1 | 8 - 18 | Set to 11111111100 (bits 8 to 18). This value is chosen to ensure existing data broadcasting decoders, designed to clause 5 of the present document, ignore this type of page. |
| 2 - 13 | 1 - 18 | **Reserved for future use** |

## 4.3a Page Transmission

A page shall be transmitted obeying the 20 ms rule (EN 300 706 [1]) if the LSB of the Data Stream Identifier (S3) is set to 0.

Packets 1 up to 25 shall be sent in ascending address order up to and including the packet specified as Last Transmitted Packet within the page subcode. In this way a receiver can detect any lost packets.

A page in any stream may be transmitted in fragments. Each fragmented page contains a page header and some of the packets of the complete page. These packets shall be transmitted in ascending address order. The same Continuity Index and Last Transmitted Packet values shall appear in all the page headers that are required to transmit the fragments that make up the full page. The C4 (Erase Page) shall only be set to 1 for the page header of the first fragment. Fragmented pages with the LSB of the Data Stream Identifier (S3) set to 0 shall obey the 20 ms rule for each fragment.

The number of packets used per page is permitted to vary with each instance of a page. If there is insufficient application data to fill all the packets indicated by the Last Transmitted Packet value in the page header, Filler Byte codes shall be inserted in the remaining bytes and these "empty" packets shall still be transmitted.

An application may set a limit on the rate at which pages may be broadcast thus dictating the maximum overall data rate.

## 4.4 Coding of the packets

The transmission of a page commences with a page header, with the subcode bytes coded according to clause 4.4.1. An indication of the last packet in use for one particular instance of the page is contained in the S2 and S4 components. The S3 component defines the data stream. If the LSB is set to 0, the 20 ms rule must be applied before any further packets can be transmitted. As an option a page may include a packet 28 to define the function and coding of the page as defined in clause 4.4.3. If present, it is transmitted after the page header and before any packets with addresses in the range 1 to 25 inclusive. Packets 1 up to 25 carry the data and are coded according to clause 4.4.2. There can be restrictions on the highest numbered packet that can be used. The packets are encoded and transmitted in ascending address order up to and including the last packet indicated by the S2 and S4 components. There shall be no missing packets within this range.

In order to provide greater flexibility for the network operator it is not necessary to fill each data page. Several separated headers, each with some data packets, may be required to provide the data service and this technique is covered fully in EN 300 706 [1]. In the case of a data service it is possible to send full but fragmented pages as well as to simply send shorter pages.

Fragmented pages are usually interleaved with one or more other pages. Each fragmented page contains, in row order, parts of a full page. Fragmented pages transmitted in odd numbered streams shall obey the 20 ms rule for each fragment.

### 4.4.1 Page header

This is necessary but shall not generally contain any of the application data. It may be identical to all others in the same magazine. The use of the Teletext page sub-code is defined as shown in figure 7.

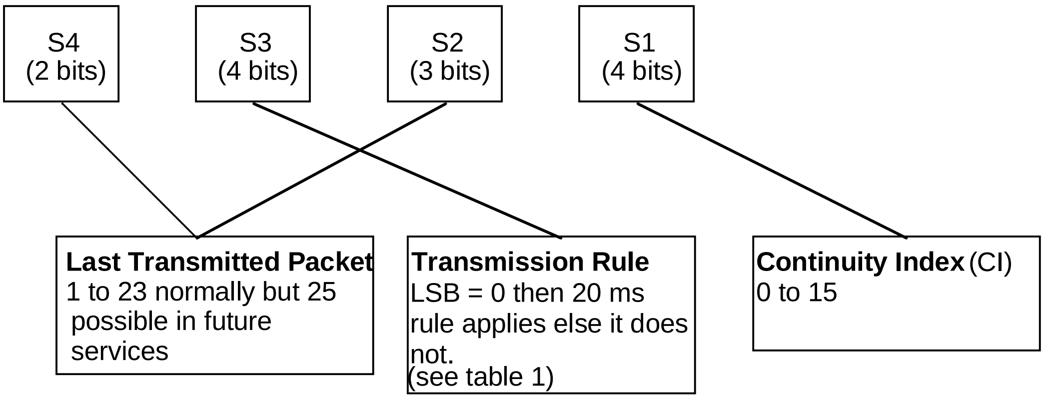



**Figure 7: The use of the Teletext page sub-code**

**The default valid range for the Last TransmittedPackets is 1 to 23** where this value provides the last packet which is being transmitted for the page with this particular value for the Continuity Index (CI). The presence or otherwise of a packet 28 has no affect on the calculation of the Last Transmitted Packet value. Where a new header is required to complete the page then this shall be provided with the same CI. The maximum number of packets per page is 25 and as a result S2 and S4 will never have values of 7 and 3 simultaneously. Thus the reserved sub-code value of 3F7F defined in EN 300 706 [1] can never occur. As expected the Least Significant Bit (LSB) of each parameter in the page sub-code is transmitted first. S2 contains least significant 3 bits of the "Last Transmitted Packet".

**The 20 ms rule** requires that the header packet 0 is sent during a previous field. This rule shall be followed when the LSB of S3 is set to 0. When this bit is set to 1 then the header may be followed immediately by the packets associated with this header. The following values have been defined in table 5.

**Table 5: Values of S3**

| Hex value of S3 | Interpretation |
|---|---|
| 0 | 20 ms rule applies - data stream 1 |
| 1 | no 20 ms rule - data stream 2 |
| 2 | 20 ms rule applies - data stream 3 |
| 3 | no 20 ms rule - data stream 4 |
| 4, 6, 8, A, C and E | 20 ms rule applies - data stream 5, 7, 9, 11, 13 and 15 respectively |
| 5,7,9,B and D | no 20 ms rule - data stream 6, 8, 10, 12 and 14 respectively |
| F | reserved for future use, no 20 ms rule applies |

In each stream the **CI value** increments modulo 16 for each page which contains different application data. The values used in each stream should be sequential in the range 0 to F, with no missing values.

**Page control bytes C4 to C14;** their action is as follows:

The C4 control bit (Erase Page) should be set to 1 the first time the header is sent with a new Continuity Index value. If the page is transmitted in fragments, subsequent page headers with the same CI value should use a C4 value of 0.

The values for C5, C6, C7, C8, C11, C12, C13 and C14 depend upon the type of service which the network provider is offering and have the same meaning as for normal Teletext pages. The following control bytes in table 6 have a recommended value.

**Table 6: Page Control Bits**

| C9 | C10 |
|---|---|
| Interrupted Sequence | Inhibit Display |
| 1 | 1 |

The next 24 bytes should normally match those of the other pages in the same magazine. This is to avoid possible display problems with viewers of the normal Teletext service which shares this Magazine.

The last 8 bytes of the header shall be used in a way which takes account of the rest of the Teletext service. Where the time can be the same as that on the normal Teletext service it shall be presented in the same format. Where this is not possible these 8 bytes should be filled with spaces. In the case where there is no other Teletext service then the local time shall be placed in these 8 bytes according to the Enhanced Teletext Specification EN 300 706 [1].

## 4.4.2 Packets 1 to 25

These packets may carry the data service. The page sub-code indicates to the decoder how many packets to expect. If a packet 28 is included then packets 1 to 25 can be used to carry the data. The default condition in the absence of a packet 28 is packets 1 to 23. A data service, e.g. EPG, may state its own upper limit in which case packets with higher addresses should not be used. Where multiple data services are combined in one data stream, the lowest maximum value determines the highest packet address that can be used in that data stream. Each Teletext packet is constructed in the manner shown in figure 8 which, for the sake of clarity, excludes the packet addressing information contained in bytes 1 to 5 of a Teletext packet coded according to EN 300 706 [1]. The Block Pointer byte (BP) is thus located in byte 6 of the Teletext packet.

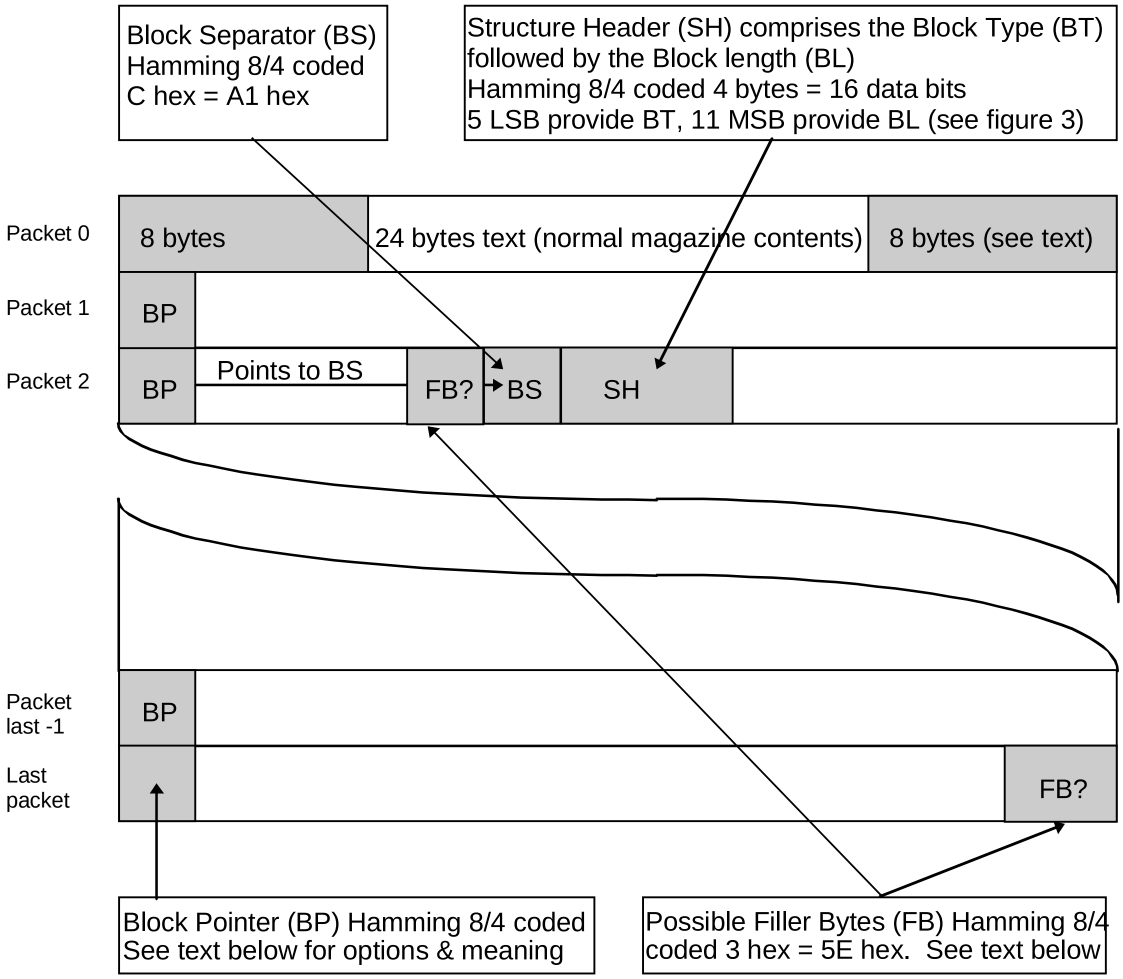



**Figure 8: Example of a page according to Page Format - Clear**

### 4.4.2.1 Transmission order

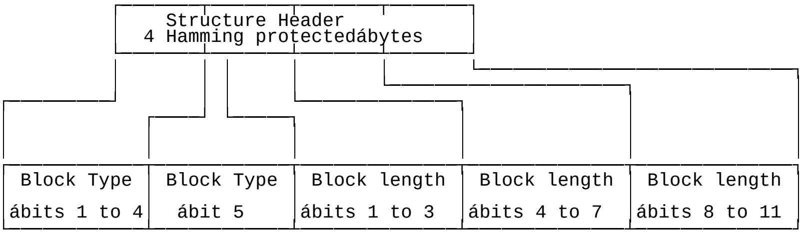
This follows the normal Teletext format where the LSB is transmitted first. For example in the Structure Header (SH) the least significant 4 bits of the Block Type shall be placed in the byte which is transmitted first. The most significant bit of BT shall be placed in the least significant position of the next byte. The Hamming 8/4 coding is carried out in the same way as fully described in EN 300 706 [1]. The bytes defined above are all transmitted in the same way. It is up to the data service provider how he codes the bytes which constitute the "user data". The values for BP are given in table 7.

**Table 7: Values for Block Pointer (BP)**

| Hex value for BP | Interpretation |
|---|---|
| 0 to C | point to BS in byte position (BP x 3) + 1 |
| D | means that there is no BS in this packet |
| E and F | reserved for future use |

### 4.4.2.2 The Structure Header (SH)

The SH comprises four Hamming 8/4 coded bytes. This is put in by the data service provider and the 16 available data bits are used in the following way.




NOTE: Shown in transmission order from left to right. LSB transmitted first. Only the data bytes are shown.

**Figure 9: The Structure Header**

The first 5 bits (BT) provide the Application ID. This gives the data application provider the possibility of providing up to 31 different categories of data within one page service. This is addition to the separation into a number of streams which is possible by means of the page sub-code value for S3. The actual value for the Application ID is up to the application provider. When the Application ID has the value 0 this is used to provide the system with information about which services are carried in this data channel. This is fully covered in EN 300 707 [2].

The remaining 11 bits define the Block Length. This indicates the number of bytes within the following data structure. This is the number of bytes defined by the application and it does not include the bytes required for the Block Type and Block Length. The maximum value for Block Length is 2 047 bytes.

### 4.4.2.3 Packing the data into the pages

**The network operator** packs the available data into the Teletext packets. He has to read the block lengths in order to put in the Block Pointer (BP) values. The BP may point to 1 of 13 locations in a Teletext packet according to table 7. Where a data block does not fill up to 1 of the 13 possible start locations the free bytes are filled with Filler Bytes (FB). Filler bytes are also used to pad out the end of a packet if the broadcaster has no data available to fill it. The benefit of this process is that each block start is not only pointed to from the previous block length but also from the BP. It provides a robust method of synchronizing the application data stream. Block Pointer (BP), Block Separator (BS) and Block Length (BL) are all Hamming 8/4 coded. Even with bit errors it should be possible for the reception device to have a high confidence that it is in correct synchronization. The BL may not point precisely to the next BS as these may only be placed in every third byte, when pointed to from the BP. The existence of the FB informs the decoder to look forward to find the next BS.

Where the Service Operator provides very short data blocks it is possible that more than one block may commence in the same Teletext Packet. Where this occurs the BP shall point to the first BS. The second BS shall be placed immediately after the end of the first block, as indicated by the BL in the previous SH. There is no requirement for Filler Bytes between the subsequent data blocks as the next BS will not be pointed to from a BP.

The Filler Byte (FB) (5E hex = 01011110 binary) has the inverted value of the Block Separator (BS) (A1 hex = 10100001 binary).

Where the broadcaster starts from no data then the first Block Separator (BS) shall occur at the start of the page. Data Blocks may start within one page and finish in a different one. The network operator requires no knowledge of the data which he is transporting apart from knowing where each block commences and how long it is. Each of the user data bytes contains 8 usable bits and it is up to the Data service provider to determine how he uses them. The packets shall be sent in ascending sequence through from 1 to n (given by the n value in the page sub-code provided by S4 and S2). In this way any lost packets can be detected.

The employment of a packet 28 permits these data bytes to be encoded in various different ways but currently only the one mode is defined. When the service provider requests the use of more than one data stream then the network operator shall treat each stream as an independent data service. In the case of an EPG there are some restrictions and these shall be checked in each circumstance as they will vary according to the capabilities of the reception equipment.

### 4.4.3 Packet 28

Packet 28 is optionally required to define unambiguously the format and purpose of the data contained in the following packets which comprise the page. A packet 28 shall be included if packets 24 and 25 are to be used to carry the data service. It should be sent, as soon as possible, following the header. Some Teletext decoders do not acquire the packet 28 and so it should be employed only for applications which are aimed at newer decoders. The designation code should be set to 0000 and the page function set to 0001. The contents of the packet are given in table 8.

**Table 8: The Packet 28 for Page Format - Clear**

| Triplet | Bits | Function |
|---|---|---|
| 1 | 1 ⇩ 4 | **Page Function**<br>These bits define the function of the data in packets 1 to 25 of the current page. They shall be set to 0001 to indicate a Data Broadcasting Page (Page Format - Clear) defined to this clause in the present document. |
| 1 | 5 ⇩ 7 | **Page Coding**<br>These bits define the coding of the data in packets 1 to 25 of the current page. These bits shall be set to 001 to indicate that there is no regular coding scheme and each byte should be regarded as 8-bit data. |
| 1 | 8 ⇩ 18 | Set to 11111111100 (bits 8 to 18). This value is chosen to ensure existing data broadcasting decoders, designed to clause 5 of the present document, ignore this type of page. |
| 2 ⇩ 13 | 1 ⇩ 18 | **Reserved for future use** |

## 4.5 Providing a service according to Page Format - Clear

The following lists lay down the conditions which are necessary for a satisfactory transmission and reception of a data broadcast application which employs this method for transporting data.

### 4.5.1 General points on Page Format - Clear

1) The 20 ms rule applies between transmission of the header and any following packets when the received data is required to be processed by all "low end" decoders. The value for the LSB of the S3 value in the page header shall be set to 0. Enhanced data which is destined for use by new "high end" decoders does not need to follow the 20 ms rule (LSB of S3 = 1).

2) The maximum number of packets following the header shall be limited to 23.

3) The data within the packets shall be treated by the Teletext decoder as 8-bit data.

4) The data shall all be transmitted using the method described in Page Format - Clear.

5) The user data shall not necessarily be synchronized to the page format.

6) The block structure shall be as indicated under Page Format - Clear.

7) Where the service is constructed from more than one data stream then each stream shall be dealt with as an independent data service. It is up to the application to make intelligent use of the data extracted from more than one stream.

8) The implication of 7) is that the decoder may have to simultaneously build up two separate data pages. This may happen if fragmented pages from two streams are interleaved with each other. This can be managed by a Code of Practice (CoP) document for the data service.

### 4.5.2 Encoding scheme for electronic data transmission

This clause shows how to take the application to the transport system. The main issues are:

1) The application data is encoded in a way appropriate to its purpose. Critical data is better protected. This function is left to the data service provider, or his agent, and is of no concern to the transportation layer. He may group together items or divide them but the end result shall be blocks of data which he wishes to deliver to the network operator.

2) The user data blocks are linked together into one data stream by means of the Block Separators (BS). The Data service provider puts in the Application ID the Block Type (BT) and the Block Lengths (BL) as defined in Page Format - Clear.

3) The resulting data stream is fitted into pages by the network operator who shall shift through the data in order to put in the Block Pointers (BP) and the Filler Bytes (FB). Apart from this the network operator has no interest in the data.

4) Where 2 Blocks start in the same Teletext Packet the network operator inserts the BP for the first one only as described in clause 4.4.2.3.

5) Where the data is destined to be transmitted in more than one data stream then the data for each stream shall be supplied separately and considered as an independent and separate service.

6) The network operator shall take account of the capabilities of the receiving equipment when he inserts the data streams into the Teletext service (see clause 4.5.1, point 8).

# 5 Page Format - CA

## 5.1 General points

This mode provides the means for subscription data services where the subscriber management centre is under the control of the network operator.

## 5.2 Advantages

This mode enables many service providers the means to transmit large amounts of data efficiently to subscriber groups. As the network operator is providing a value added service it potentially provides him with additional revenue. The service provider does not need to concern himself with the complex issues of Conditional Access (CA).

An efficient scheme has been defined which provides an effective way of managing large numbers of subscribers with varying requirements. It achieves this by employing a hierarchical approach to handling the required encryption keys.

## 5.3 Disadvantages

This method requires the use of special Teletext decoders which incorporate the appropriate CA sub-system. It is not a very efficient way to transmit short blocks of data in view of the essential overhead of the page header and the packet 28. Error checking, at the transport level, is only possible on a per page basis.

## 5.4 Method of coding

There may be an introductory page of non-scrambled text. When no introductory text for display is required, this page shall at least include the header packet with Y = 0 and a packet or packets with Y = 27 to provide links to the conditional access service. The links are provided in a packet with Y = 27, designation codes 0100 to 0111 (see EN 300 706 [1]). Figure 10 shows how the system works.

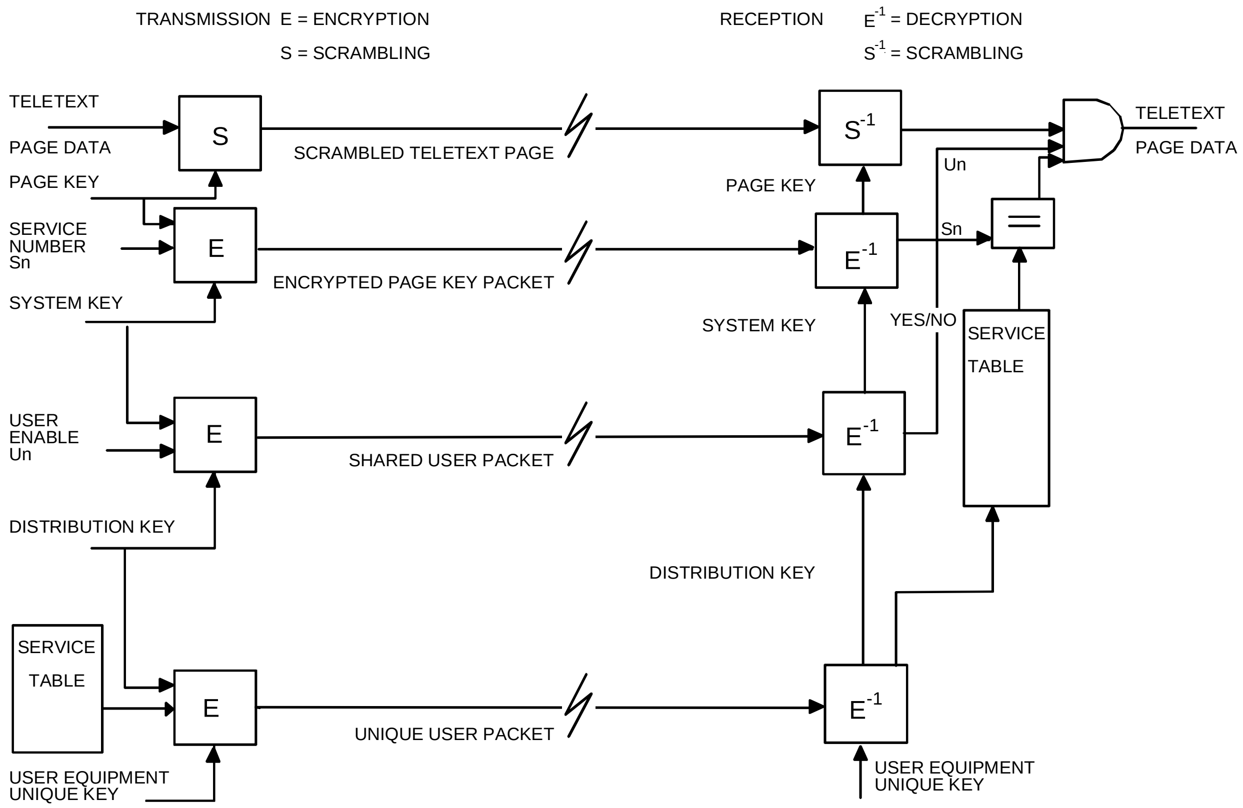



**Figure 10: Component parts required for the service**

Figure 10 helps to understand the operation of a service operating to the Page Format - CA standard.

## 5.4.1 Teletext page data

For the purpose of scrambling, two types of pages are defined.

1) **Scrambled Data Pages** are essentially normal Teletext pages but with each byte scrambled to make it unintelligible. The scrambling process is initialized at the start of each packet. Unused packets need not be transmitted. See clause 5.4.1.1.

2) **Pages containing Reformatted Data** shall be signalled by means of the packet X/28 as shown in table 9. In this case the scrambling process is initialized at the start of each page. See clause 5.4.1.2.

**Page Mode Definition** is achieved by means of a Packet X/28. The first 8 bits in the first triplet define the type of page data which follows. Table 9 provides the relevant values but for more detail see EN 300 706 [1]. Note that this table only refers to packets X/28 with designation codes of 0000 and 0010. Table 10 provides details on how to interpret bits 15 to 18 in this packet X/28.

**Table 9: Coding of Packets X/28 - interpretation of the following page**

| 8 7 6 5 4 3 2 1 | Bit Numbers |
|---|---|
| 1 0 0 0 0 1 0 0 | Reformatted Data |
| 1 0 0 0 0 1 0 1 | Terminal Equipment Addressing page |
| 0 * * 0 0 0 0 0 | Page has standard character position and row format |
| NOTE: * = do not care value. | |

**Table 10: Coding of Packets X/28 - data bit organization and protection**

| 18 17 16 15 | Bit Numbers |
|---|---|
| 0 0 0 0 | Data coded as 8-bit bytes, 7 bits data plus 1 parity bit |
| 0 0 0 1 | Data coded as 8-bit bytes, with 8 data bits each |
| 0 0 1 0 | Data coded as 3 groups of 8-bit bytes, 18 data bits Hamming protected 24/18 |
| 0 0 1 1 | Data coded as 8-bit bytes, 4 data bits Hamming protected 8/4 |

This interpretation is only defined when the remaining bits in this 18 bit group, bits 9 through to 14 all have the value 0.

#### 5.4.1.1 Scrambled Data Pages

The data for transmission in bytes 6 to 45 of packets with Y = 1 to Y = 25, plus the 18 data bits in each three byte data group for transmission in packets with Y = 26, is scrambled, using a suitable scrambling algorithm. Clause 5.5 provides an example of a suitable algorithm. The numbers of the packets included in a scrambled text page are specified as in clause 5.4.2.4.

**Parity protected 7-bit data**

To provide a sequence of complete bytes for scrambling when 7-bit data is used, a most significant bit is added. The resulting bytes are scrambled and the respective bit masked before the odd parity bit added.

**Data in Packets with Y = 26**

To provide a sequence of complete bytes for scrambling, the 18 data bits have the 6 most significant bits added. The resulting 3 bytes are scrambled and the respective 6 bits masked before Hamming protection bits are calculated and added.

#### 5.4.1.2 Pages containing Reformatted Data

The data for transmission in bytes 14 to 37 of packets with Y = 0 and bytes 6 to 45 of packets with Y = 1 to Y = 25 are scrambled using an encryption algorithm. The number of bytes included in the page of scrambled data is indicated by the Data Length parameter according to clause 5.4.2.5.

**Parity protected 7-bit data**

To provide a sequence of complete bytes for scrambling when 7-bit data is used, a most significant bit is added. The resulting bytes are scrambled and the respective bit masked before the odd parity bit is added.

### 5.4.2 The Page Key Packet

Descrambling of a Scrambled Teletext Page Data is by means of a Page Key contained in a packet with Y = 28 of a scrambled page. The Page Key Packet also provides details concerning the scrambled data which is carried by the Teletext page. Byte 6, 4 bits data plus 4 bits Hamming protection, carries the Designation Code with the data bits set to 0010. Bytes 7 to 45 are used as 13 groups of 18 data bits plus 6 bits Hamming protection as shown in figure 11.

```
 Clock run-in | Framing | Magazine and   | Designation | First  Three-
              | Code    | Packet Address | Code        | Byte Data Group

                                       DATA NOT ENCRYPTED

 P D P D P D P D  | P  P  D  P  D  D  D  P  D  D  D  D  D  D  D  P  D  D  D  D  D  D  D  P
 Designation Code | 1  2  1  3  2  3  4  4  5  6  7  8  9  10 11 5  12 13 14 15 16 17 18 6

 Service | Set     | Current    | Service        | Page    | Key Value
 Modes   | to '0'  | System Key | Identification | Key     | Credit Tokens
 2 Bits  | 6 Bits  | 56 Bits    | 8 Bits         | 56 Bits | 16 Bits
                    ENCRYPTED WITH CURRENT SYSTEM KEY

                                ------ 27ábits ------
 Continuity | Repeat    | Packet  | Set     | Scrambling | In Use | Set to | Cyclic
 Indicator  | Indicator | Flags   | to 0    | Method     | System | 0      | Redundancy
 7 Bits     | 1 Bit     | 26 Bits | 1 Bit   | 5 Bits     | Key    | 8 Bits | Check
            |           |   or    |   or    |            | Label  |        | 16 Bits
            |           | Data    | Set     |            | 8 Bits |        |
            |           | Length  | to 0    |            |        |        |
            |           | 10 Bits | 17 Bits |            |        |        |
                                   DATA NOT ENCRYPTED
```

**Figure 11: Format of Page Key Packet (X/28)**

The first group of 18 data bits is not encrypted and shall be set to designate the page type as provided in table 9 and table 10. The next 144 bits of data are encrypted with the Current System Key. The following clauses provide additional detail which is not included in figure 11.

### 5.4.2.1 Service Mode bits

**The two Service Modes bits** are interpreted as shown in table 11.

**Table 11: Service Mode Bits**

| Bit 2 | Bit 1 | Service Mode | Mode Description |
|---|---|---|---|
| 0 | 0 | 1 | 256 Services Non-Tiered |
| 0 | 1 | 2 | 64 Services Tiered |
| 1 | 0 | 3 | 256 Services with Credit Tokens |
| 1 | 1 | 4 | Not Assigned |

### 5.4.2.2 Service Identification

**The 9-bit Service Identification Number** is used in the following way dependent on the Service Mode:

Service Mode 1 or 3:8 bits define service number, 0 to 255;
or
Service Mode 2: 6 most significant bits of 8-bit group define service number 0 to 63.
2 least significant bits define the service tier as shown in table 12.

**Table 12: Service Tiers**

| Bit 2 | Bit 1 | Service Tier |
|---|---|---|
| 0 | 0 | Basic Tier |
| 0 | 1 | Basic + Premium Tier |
| 1 | 0 | Basic + Premium + Extra Tier |
| 1 | 1 | Not Assigned |

**The 16-bit Key Value Credit Tokens** are defined as shown below:

8 bits: fraction part;

8 bits: whole part.

The remaining 72 bits of data are not encrypted. The following clauses provide additional detail on these.

#### 5.4.2.3 Continuity and Repeat Indicators

**The Continuity Indicator** is incremented by 1, modulo 127, for each subsequent page with the same service number.

**The Repeat Indicator** is set to 1 when it is expected that the page will be repeated. It is set to 0 when no repeat is planned.

#### 5.4.2.4 Packet Flags

Packet Flags are used for Scrambled Data Pages. The Packet Flags are set to 1 if the packet is present. Each flag bit refers to one packet in the page starting with packet 1 and finishing with packet 26. This last flag may indicate a sequence of Packets with Y = 26. The least significant packet number is transmitted first.

#### 5.4.2.5 Data Length

The Data Length parameter is used for Pages containing Reformatted Data. The 10 bits permit data lengths up to 1 024 bytes which is the maximum possible with 24 bytes in packet 0 and $25 \times 40$ in packets 1 through 25.

#### 5.4.2.6 Scrambling Method

The Scrambling Method is selected from those listed in table 13.

**Table 13: Register of Scrambling Methods**

| Method number | Bits 1 to 5 in transmission order | Scrambling Method |
|---|---|---|
| 1 | 00000 | Variable Length Algorithm using a One Way Function Method A (see figures 9 and 10) |
| 2 | 10000 | Block Enciphering Algorithm using Differential Code Book or Output Feedback (see note 1) |
| 3 | 01000 | Variable Length Algorithm using a One Way Function Method B (see note 2) |
| 4 | 11000 | MAC Scrambler (see note 3) |
| 32 | 11111 | Method not specified |
| NOTE 1: Differential Code Book. It is not covered further here.<br>NOTE 2: Proposed for use in connection with non-Teletext data packets. It is not covered further here.<br>NOTE 3: Proposed for use in connection with non-Teletext data packets. It is not covered further here. | | |

#### 5.4.2.7 The Cyclic Redundancy Check (CRC) word

This is shown in figure 12. This is not the same as a classical CRC word generator and is only defined for use here.

**16-bit Shift Register**

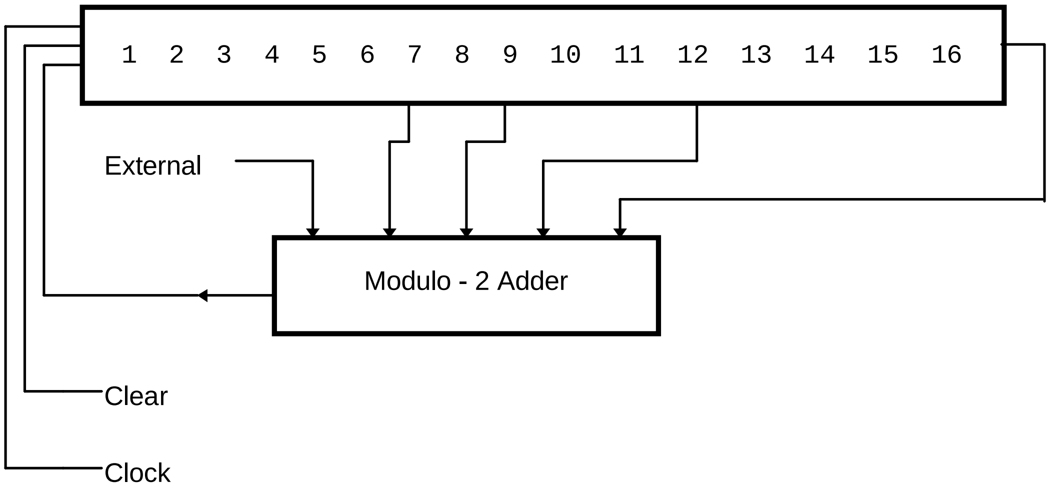



**Figure 12: Example of suitable Check Word Generator**

In the example shown a 16-bit shift register has as input the modulo-2 sum of an external input and the contents of the 7th, 9th, 12th and 16th stages of the register. Initially the register is cleared to "all zeros". During a sequence of 8 192 clock pulses, the first 24 character bytes (192 bits) of the page header packet and the following character bytes of packets with Y up to 25, in conventional transmission order, form the input. Any absent packets are considered to contain the character SPACE (2/0) throughout. In each byte, the bit order is b8 to b1 inclusive. This order, that is the reverse of that used in the transmission sequence, is to facilitate decoder operation where the data used is stored in the page memory.

At the transmitting end of the generating process the contents of the register are the basic page check word and it is transmitted along the register beginning with the bit held in the first stage.

The transmission order for the two byte group resulting from the 16 bit cyclic redundancy check on the page is bits 9 to 16 followed by bits 1 to 8 inclusive.

For Scrambled Data Pages the 24 character positions bytes 14 to 37 in packets with Y = 0 are assumed to contain the character SPACE (2/0). For Reformatted Data, the check word is calculated over the specified data length.

## 5.4.3 Terminal Equipment Addressing Pages

Access to the Page Key contained in a packet with Y = 28 of a scrambled page, is by means of the Current and New System Keys. These pages shall contain:

1) A System Key Packet encrypted with the New System Key;

2) Shared User Data Packets encrypted with the Shared Distribution Key;

3) Unique User Data Packets encrypted with a key that is unique to the user's equipment.

The Shared User Data Packets are transmitted relatively frequently and the Unique User Data Packets relatively infrequently. The System Key Packet is provided in the packet X/28 but in practice this will only be changed occasionally at the discretion of the network operator, perhaps once a month.

### 5.4.3.1 System Key Packet

There shall be a packet with Y = 28, with the designation code in byte 6 set to 0010. The first group of 18 data bits shall be set to define the page as a Terminal Equipment Addressing Page, see table 9. For the interpretation of remaining data bits in this packet see figure 13 and the supporting text. The remaining bytes 10 to 45 are used as 12 groups of 19 bits data plus 6 bits Hamming protection. See figure 13.

| Clock run-in | Framing Code | Magazine and Packet Address | Designation Code | First Three-Byte Data Group |
|---|---|---|---|---|

| P D P D P D P D Designation Code | P 1 | P 2 | D 1 | P 3 | D 2 | D 3 | D 4 | P 4 | D 5 | D 6 | D 7 | D 8 | D 9 | D 10 | D 11 | P 5 | D 12 | D 13 | D 14 | D 15 | D 16 | D 17 | D 18 | P 6 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

| Encryption Method 8 Bits | Set to 0 8 Bits | New System Key 56 Bits | Current System Key 56 Bits | Set to 0 16 Bits | New System Key Label 8 Bits | Current System Key Label 8 Bits | Set to 0 56 Bits |
|---|---|---|---|---|---|---|---|
| NOT ENCRYPTED | | ENCRYPTED WITH NEW SYSTEM KEY | | | NOT ENCRYPTED | | |

**Figure 13: System Key Packet X/28 for Terminal Equipment Addressing**

The Encryption Method is selected from the register of Encryption Methods provided in table 14.

**Table 14: Register of Encryption Methods**

| Method number | Bits 1 to 8 in transmission order | Encryption Method |
|---|---|---|
| 1 | 00000000 | Variable Length Algorithm using a One Way Function (see figures 9 and 10) |
| 2 | 10000000 | Block Enciphering Algorithm using Differential Code Book or Output Feedback (see note) |
| x | 11111111 | Method Not Specified |
| NOTE: Differential Code Book. This is beyond the scope of the present document. | | |

## 5.4.3.2 Shared User Packets

These may be carried in any packet with Y = 1 to Y = 25. The designation code is provided in byte 6 (4 bits data plus 4 bits Hamming protection). The Data bits are set to 0000. Bytes 7 to 45 are used as 13 groups of 18 bits data plus 6 bits Hamming protection (see figure 14). Service is enabled when the enabling bit is set to 1 and disabled when the bit is set to 0.

## 5.4.3.3 Unique User Packets

See figure 14. The following provide additional detail.

**The 2 Service Modes bits** are interpreted as shown in table 15.

**Table 15: Interpretation of the Service Mode bits**

| Bit 2 | Bit 1 | Service Mode | Interpretation | | Number of bits |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 7 Service Numbers of 8 bits | | 56 |
| 0 | 1 | 2 | 7 Service Numbers of 8 bits | | 56 |
| 1 | 0 | 3 | Total credit tokens purchased for 2 services | | |
| | | | 1st Service Number 8 bits | Credit Tokens Purchased 20 bits | 28 |
| | | | 2nd Service Number 8 bits | Credit Tokens Purchased 20 bits | 28 |
| 1 | 1 | 4 | Not Assigned | | |

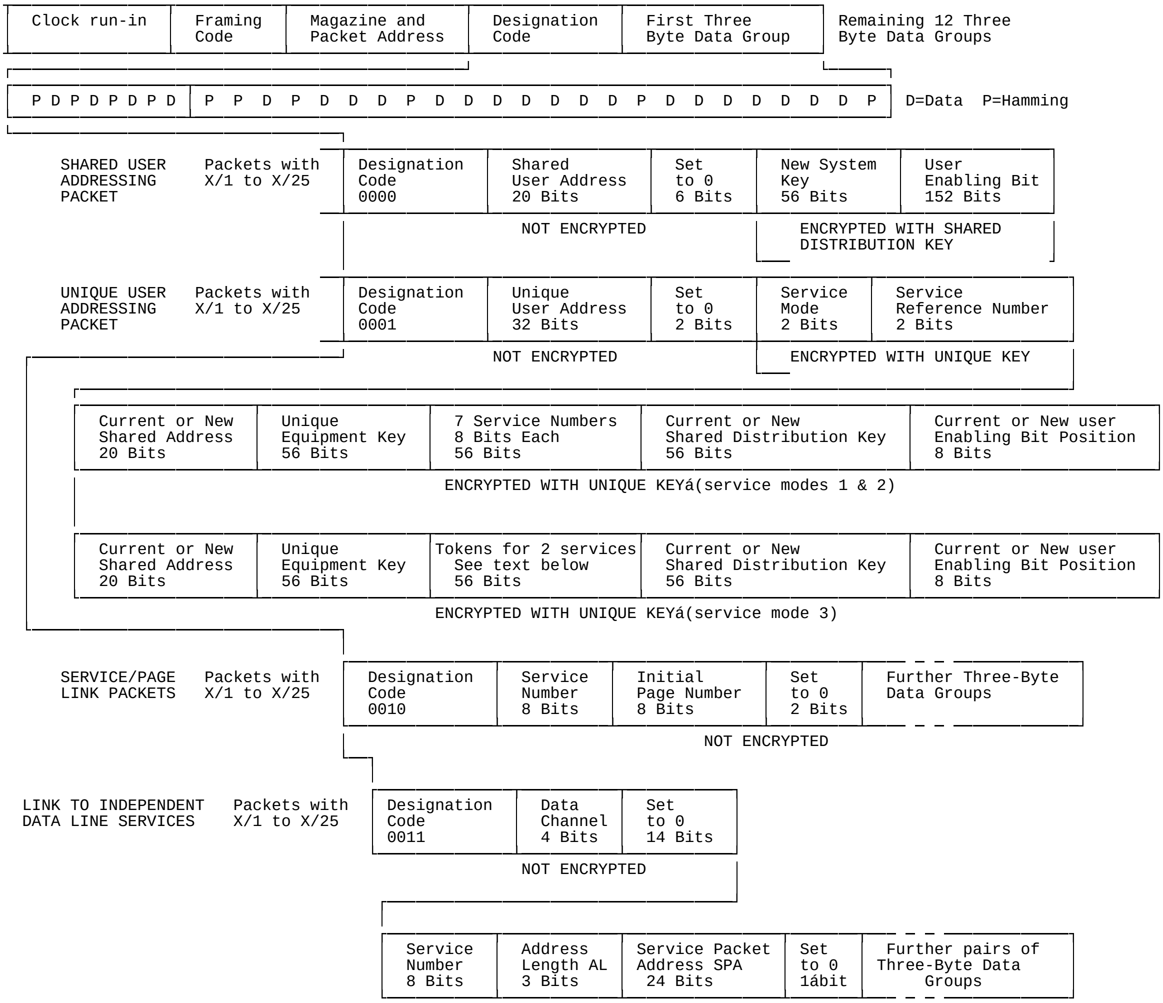



**Figure 14: Terminal Equipment Addressing packets**

### 5.4.3.4 Service/Page Link packets

Packets with Y = 1 to Y = 25. Byte 6, 4 bits data plus 4 bits Hamming protection. Designation Code Data bits are set to 0010. Each further three byte group is used in the same way to provide Service Number, Initial Page Number and the 2 bits cleared to 0. Any unused groups in a packet are set to 0.

### 5.4.3.5 Link to Independent Data Line Services

The Link to Independent Data Line Services is provided as shown in figure 14. Packets with Y = 1 to Y = 25. Byte 6, 4 bits data plus 4 bits Hamming protection. Designation Code Data bits are set to 0011. The first Three Byte Data Group defines the Data Channel. The following three byte groups are paired to provide 36 bits each and thus provide the Service Number and Service Packet Address details. Each further pair of three byte groups is used in the same way. Any unused groups in a packet are set to 0.

## 5.5 Security of Page Format - CA

The security of the user addressing process may be optimized by the use of "error extension" techniques. Should any bit of the ciphertext have been changed, this causes the message to be totally corrupted when decrypted with the correct key.

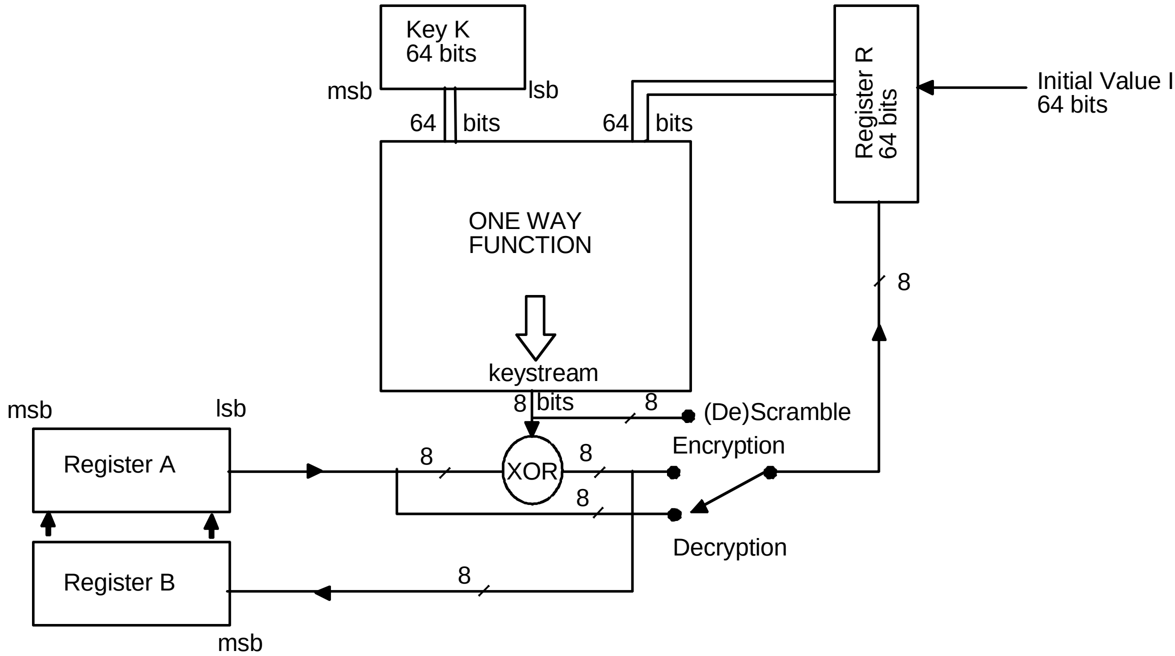



**Figure 15: Variable length encryption algorithm**

In order to provide the property of "error extension" a cipher feedback technique containing a "one-way function" is used with a multiple of enciphering called "rounds". Each round reverses the order of the previous ciphertext bytes, as shown in figure 15. A typical one-way function having good security is shown in figure 16.

## 5.5.1 Cipher feedback algorithm

The secret key K is loaded into the 64-bit key register K. The 64-bit key is derived from a 56-bit key that forms the 56 least significant bits. For reformatted data pages and data not in page format, the 8 least significant bits of the 56 are used to look up a corresponding 8-bit value from look-up-tables (LUT) in figure 16. This value is used as the 8 most significant bits of the 64-bit key. For textual page data, the 8 most significant bits are the magazine and row address of each packet forming the page.

The register R is first loaded with a 64-bit secret initial condition I, that is constant for the particular security device in the equipment. It is a random number having an impulse auto-correlation function. This data word I is loaded into the register R at the beginning of each round of the encryption or decryption process.

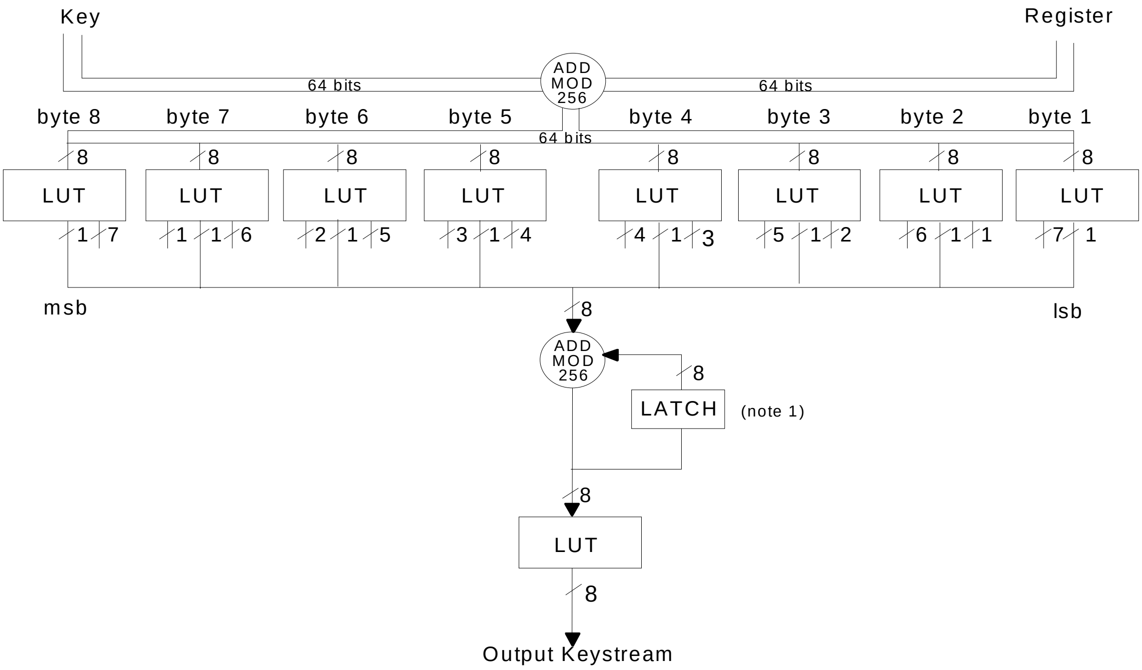
The encryption and decryption processes are represented in figure 15, the switch being placed in the appropriate position. The message to be encrypted or decrypted is placed in register A and after the appropriate number of rounds appears in register B. The data in register A is taken, byte by byte and the EXCLUSIVE-OR function with the keystream is performed. It is then placed in the B register and the output from the switch is placed in the R register. The previous contents of the B and R registers are shifted along, byte by byte until all the bytes appear in the B register. This process constitutes one round. The next round starts by placing the contents of the B register in the A register but in reverse byte order. At least three rounds are required to produce good ciphertext.

## 5.5.2 One-way function

A suitable one-way function is shown in figure 16. It has a $256 \times 8$-bit look-up-table (LUT) in nine positions. This table contains truly random "ones" and "zeros".

The 64-bit key and the 64-bit contents of register R are added modulo 256. The resulting 64-bit value is applied to 8 identical look-up tables. A different 1-bit output is taken from each table and these form an 8-bit value. This 8-bit value is applied to a modulo 256 accumulator. This causes each output byte to be influenced by the previous bytes generated during each round of the main algorithm.

The accumulator memory is cleared to zero at the start of each round. The output of the accumulator is applied to a ninth lookup table, identical to the others. Its output forms the key stream of figure 15.




NOTE 1: Latch cleared to 00 hexadecimal at the start of each round.
NOTE 2: Each byte in the figure has the most significant bit on the left
NOTE 3: Latch cleared to 00 hexadecimal at the start of each round.
NOTE 4: Each byte in the figure has the most significant bit on the left.

**Figure 16: Example of a suitable one-way function**

### 5.5.3 Text scrambling

The algorithm can also be used to scramble or descramble the user data by placing the switch in figure 15 in the appropriate position. Only one round is required. To perform either function, the input data is placed in register A and the result appears in register B.

# 6 Independent Data Lines (IDL)

## 6.1 General points

The Independent Data Lines (IDL) provide the means for service providers to transmit a data service completely independently of any accompanying Teletext service organized as magazines of pages. A number of different coding schemes are defined here.

## 6.2 Advantages

The technique enables multiple service providers to transmit data to their customers in a flexible way. It is an effective way to transmit data blocks of almost any length. With some methods error checking can be easily done on each packet and therefore lost blocks are relatively small compared with the page format. The IDLs can readily be inserted at any point in the Teletext data stream at the convenience of the broadcaster. Packets can occupy the otherwise unused data lines in a Teletext transmission obeying the 20 ms rule.

## 6.3 Disadvantages

The technique requires the use of special Teletext decoders that incorporate the means to recognize and decode IDLs. There is little else against this format.

## 6.4 Methods of coding

The general form of an IDL packet is shown in figure 17. The run-in, bytes 1 and 2, and framing code, byte 3, are the same as those for any Teletext packet coded to EN 300 706 [1]. Bytes 4 and 5 are used to identify the service unambiguously as an IDL service as described in the following clauses. All bytes are transmitted least significant bit first unless otherwise stated.

| | Bytes 1 - 2 | 3 | 4 | 5 | 6 ... 45 |
|---|---|---|---|---|---|
| **IDL** | Clock Run-in | Framing Code | Data Channel | Designation Code = 1111 | Format Specific Bytes |

**Figure 17: General form of an IDL**

## 6.4.1 Designation code

Byte 5 is Hamming 8/4 encoded according to EN 300 706 [1]. The four message bits set to 1111 designate that this is an Independent Data Service packet.

NOTE: With reference to EN 300 706 [1], the Designation Code value of 1111 fixes the packet address as either 30 or 31, depending on the most significant message bit of byte 4.

### 6.4.1.1 Transmission multiplexing

The data lines carrying these packets may be interleaved with the data lines of a page-based Teletext service or may be transmitted on otherwise unused lines. These data lines may always be added at any point in the transmission chain, provided that a new data channel is used (see also clause 6.4.2).

## 6.4.2 Data Channel addressing

Byte 4 is Hamming 8/4 encoded according to EN 300 706 [1]. The sixteen possible message values define the Data Channel. These are numbered 0 to 15 as shown in table 16.

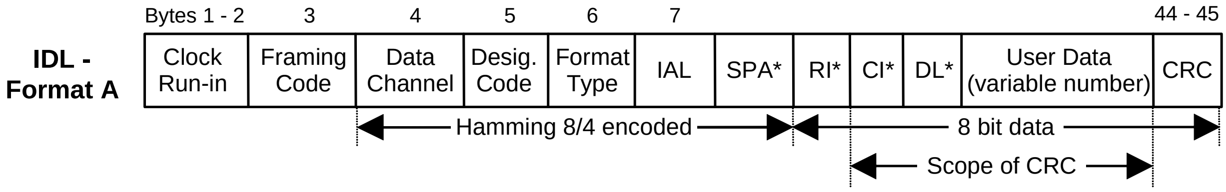
**Table 16: The IDL Data Channels**

| Data Channel number | Service name | Reference | Byte 4 Data (Transmission Order) |
|---|---|---|---|
| 0 | Packet 8/30 | Broadcast service data packets, see EN 300 706 [1] for format 1 and EN 300 231 [3] for format 2 | 0000 |
| 1 | Packet 1/30 | | 1000 |
| 2 | Packet 2/30 | | 0100 |
| 3 | Packet 3/30 | | 1100 |
| 4 | Low bit rate audio | Clause 6.7 | 0010 |
| 5 | Datavideo | Clause 6.6 | 1010 |
| 6 | Datavideo | Clause 6.6 | 1010 |
| 7 | Packet 7/30 | | 1110 |
| 8 | IDL Format A or B | Clauses 6.5 or 6.8 | 0001 |
| 9 | IDL Format A or B | Clauses 6.5 or 6.8 | 1001 |
| 10 | IDL Format A or B | Clauses 6.5 or 6.8 | 0101 |
| 11 | IDL Format A or B | Clauses 6.5 or 6.8 | 1101 |
| 12 | Low bit rate audio | Clause 6.7 | 0011 |
| 13 | Datavideo | Clause 6.6 | 1011 |
| 14 | Datavideo | Clause 6.6 | 0111 |
| 15 | IDL Format B | Clause 6.8 | 1111 |

IDLs Formats A and B are designed so that they can coexist in the same Data Channel.

## 6.5 IDL Format A

Figure 18 shows the structure of a Teletext packet conforming to the IDL Format A definition.




The presence and contents of the fields marked * depend upon the contents of the Format Type and IAL fields.

**Figure 18: Structure of an IDL according to Format A**

The Data Channel (byte 4) and Designation Code (byte 5) are as defined in clause 6.4. Four data channels are allocated for IDL Format A, as shown in table 16.

All bytes after the framing code and up to and including the Service Packet Address (SPA) are Hamming 8/4 encoded. The remaining fields are constructed from bytes containing 8 data bits.

### 6.5.1 Format Type (FT)

Byte 6 is Hamming 8/4 encoded and contains Format Type (FT) information. The least significant message bit set to 0 defines the IDL to be of type Format A. The other message bits indicate the other fields present in the packet before the start of the User Data as indicated in table 17. This byte is always present.

**Table 17: Interpretation of the four message bits in the Format Type byte**

| Bit | Value | Meaning | Include | Bytes required |
|---|---|---|---|---|
| Bit 1 | 0 | Format A - Bits 2 to 4 as defined below | | |
| Bit 1 | 1 | Other IDL format | | |
| Bit 2 | 0 | No repeat facility | | |
| Bit 2 | 1 | Repeat packet facility applies | RI | 1 |
| Bit 3 | 0 | Continuity indicator is implicit | | |
| Bit 3 | 1 | Explicit continuity indicator included | CI | 1 |
| Bit 4 | 0 | Data Length Byte not in use | | |
| Bit 4 | 1 | Data Length Byte in use | DL | 1 |

### 6.5.2 Interpretation and Address Length (IAL)

Byte 7 is Hamming 8/4 encoded and contains the Interpretation and Address Length (IAL) information, table 18. This byte is always present.

**Table 18: Interpretation of the four messages in the IAL byte**

| Value (transmission order) bit 1 2 3 4 | Service Packet Address (SPA) length (bits) | Bytes required for SPA |
|---|---|---|
| 000X | None | 0 |
| 100X | 4 | 1 |
| 010X | 8 | 2 |
| 110X | 12 | 3 |
| 001X | 16 | 4 |
| 101X | 20 | 5 |
| 011X | 24 | 6 |
| 111X | Future use | |

In table 18, the three least significant message bits define the number of bytes in the Service Packet Address (SPA) field. This interpretation does not apply if all three bits are set to 1 as this value is reserved for future extensions.

Bit 4 set to 0 defines the User Data as independent of the contents of any other channel or service packet address. Bit 4 set to 1 indicates that interpretation of the User Data may require the use of data in other channels or with other service packet addresses as defined by the application. This interpretation does not apply if the 3 least significant bits are all set to 1 as this value is reserved for future extensions.

## 6.5.3 Service Packet Address (SPA)

The presence and size of the Service Packet Address field (SPA) is determined by the contents of the IAL byte. When present this field occupies byte 8 up to byte 13 and each byte is Hamming 8/4 encoded.

When differentiated by the appropriate service packet address length data in the IAL, byte 7, the less significant bytes of an SPA may constitute another complete SPA in the same data channel. Thus, for example, the 24-bit service packet address ABC123 can coexist with the 20-bit service packet address BC123 and the 8-bit service packet address 23.

## 6.5.4 Repeat Indicator (RI)

The location of the optional Repeat Indicator (RI) byte within the packet depends upon the size of the SPA field.

RI is coded as 8-bit data. The bits 1 to 4 are set to 0 when a new packet of that service data channel is first transmitted and shall be incremented modulo 16 on subsequent repeats. The bits 5 to 7 are reserved for future extensions. Bit 8 is set to 0 to indicate that no further repeats of the current packet should be expected. Bit 8 shall be set to 1 when a further repeat is to be expected.

Message bit 2 of the FT byte shall be set to 1 if the RI byte is included (see clause 6.5.1).

## 6.5.5 Continuity Indicator (CI)

The location of the optional Continuity Indicator (CI) byte within the packet depends upon the size of the SPA field and presence of the RI byte.

CI is coded as 8-bit data. It represents an 8-bit number that is incremented modulo 256 with each new packet of the same service packet address on the same data channel. It is not incremented on repeated transmissions of the same packet.

Message bit 3 of the FT byte shall be set to 1 if the CI byte is included (see clause 6.5.1).

The CI byte, if present, is included in the CRC calculation (see clause 6.5.8).

NOTE: When an explicit CI value is not included, the CRC generation method is modified so that two identical 8-bit CI values are calculated by a decoder if the packet is received without errors (see clause 6.5.8.1).

## 6.5.6 Data Length (DL) byte

The location of the optional Data Length (DL) byte within the packet depends upon the size of the SPA field and the presence of the RI and CI bytes.

The DL byte is included when it is necessary to send an incompletely filled packet. DL is coded as 8-bit data. The two most significant bits (7 and 8) are not defined. The remaining six bits define the number of 8-bit bytes of user data intended to be delivered to the user by this packet. The count is taken from the start of the User Data Byte Group and includes any dummy bytes (see clause 6.5.7.1). Any remaining bytes of the user data group are not defined but are subject to the CRC (see clause 6.5.8).

The six message bits defining the data byte length may be set to 0, to keep a data service channel open when there is no data for delivery to the user.

Message bit 4 of the FT byte shall be set to 1 if the DL byte is included (see clause 6.5.1).

The DL byte, if present, is included in the CRC calculation (see clause 6.5.8).

## 6.5.7 User Data Group

The remaining data bytes in the packet, excepting the last two, constitute the data carried for users of the service bearing that Service Packet Address (SPA) on that data channel. The number of bytes available and the location of the first byte depends upon the size of the SPA field and the presence of the RI, CI and DL bytes. Thus there are between 27 and 36 data bytes available.

### 6.5.7.1 Dummy bytes

Certain forms of coding may give rise to long strings of 0s or 1s. It is desirable to remove these from the transmitted data field to ensure reliable operation of all equipment that may process the signal. When within any user data group a sequence of eight consecutive bytes containing all 0s or eight consecutive bytes containing all 1s occurs, taken together with its CI byte if present, a following dummy byte shall be inserted. This dummy byte is included in the calculation of the CRC, (see clause 6.5.8) but is otherwise ignored by the decoder.

Decoders shall be designed to recognize these dummy bytes. However their inclusion may in the future no longer be necessary and it would be desirable to omit them to increase efficiency. It is therefore recommended that decoders should be capable of convenient modification or adjustment when this occurs.

## 6.5.8 Cyclic Redundancy Check (CRC) word

The last two bytes contain a Cyclic Redundancy Check on the User Data Group (see clause 6.5.7) and on any Continuity Indicator or Data Length bytes if present (see clauses 6.5.5 and 6.5.6).

### 6.5.8.1 Check word generation

The data to be checked is considered as a polynomial in x with the highest degree term transmitted first and the term of degree zero last. This is divided, using modulo 2 arithmetic by the polynomial:

$$x^{16} + x^9 + x^7 + x^4 + x^0$$

The remainder from this process, with the highest term transmitted first, is the CRC.

When an implicit continuity indicator is signalled by the third message bit of the FT byte, the transmitted CRC is modified such that the checking process at the receiver results in the register containing the 8-bit continuity indicator byte twice, with the least significant bit at the right-hand end, assuming the relevant bytes of the packet are received without error.

### 6.5.8.2 Check result

The register of figure 19 is set to 0s. The serial data followed by the CRC is then entered.

If an explicit CI is indicated by the FT byte, the check is satisfied if the register again contains all 0s.

If an implicit CI is indicated by the FT byte, the register will contain two identical bytes representing an 8-bit continuity indicator, with the least significant bit at the right-hand end, assuming the relevant bytes of the packet are received without error.

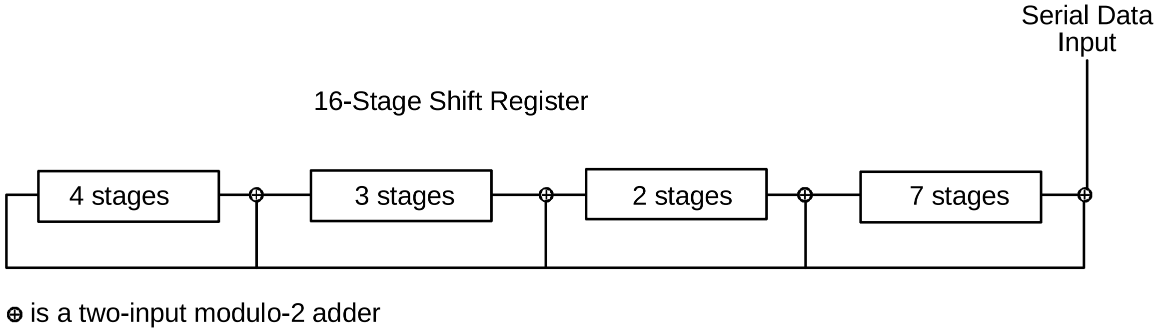



**Figure 19: Cyclic Redundancy Check (Format A)**

### 6.5.9 Transmission sequence

For any Service Packet Address (SPA), the corresponding serial data stream is divided into User Data Groups. These shall be transmitted in the correct sequence, which is monitored by the Continuity Indicator. Provided that the Repeat Indicator is used, each group may be repeated any number of times before the next is sent. There may or may not be an interval between consecutive data-lines with the same Service Packet Address.

Data lines carrying different Service Packet Addresses may be combined in any order to form a Data Channel, provided that the sequence for each Service Packet Address is not disturbed.

Data-lines from different Data Channels may be combined in any order provided that the sequence within each contributing source is not disturbed.

## 6.6 Datavideo format

This protocol is based closely upon the specification given in clause 6.5 but with the variations indicated in the following clauses and in figure 20. The Data Channel (byte 4) and Designation Code (byte 5) are as defined in clause 6.4. Four channels are allocated for this format, as shown in table 16.

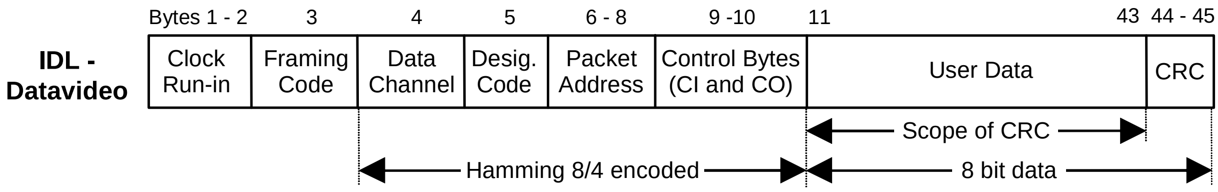



**Figure 20: Structure of an IDL according to the Datavideo format**

### 6.6.1 Packet address

Bytes 6, 7 and 8 are each Hamming 8/4 encoded and overall carry a 12-bit Packet Address.

### 6.6.2 Control Bytes (CB)

Bytes 9 (CI) and 10 (CO) are both Hamming 8/4 encoded.

#### 6.6.2.1 Packet Continuity Indicator (CI)

The CI is represented by a 6-bit number that is incremented modulo 64 with each new packet of the same address on the same data channel. It is not incremented on repeated transmissions of the same packet. Repeated packets may be interleaved with newly transmitted ones. In order to distinguish such repeated packets from new packets, in case there is a discontinuity due to packet-losses, the interleaving space shall not be greater than 32.

The CI is transmitted in byte 9 and the first two data bits of byte 10; the least significant bit is transmitted first.

### 6.6.3 Masking indicator

Bit M, which is the third data bit of byte 10, indicates the presence of "masking": i.e. when it has value 1, this means that bytes 11 to 45 (useful data + CRC) are EXOR-ed with the sequence (in hexadecimal notation):

| | |
|---|---|
| Mask | AF AA 81 4A F2 EE 07 3A 4F 5D 44 86 70 9D 83 43 BC 3F EO F7 C5 CC 82 53 B4 79 F3 62 A4 71 B5 71 31 10 08 |
| Byte No. | 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 |

It may be common practice to transmit the same packets once masked and once unmasked: this removes the adverse effects of critical bit combinations. Of course, masked and unmasked packets, when the information content is the same, are transmitted with the same continuity index.

## 6.6.4 Packet type indicator

Bit L, which is the fourth data bit of byte 10, defines the packet type. It is used by higher levels of the protocol and therefore it may be defined in a different way, according to the type of protocol used. In one version of the protocol, this bit, when it is set to 1, indicates the presence, in the User Data Group, of a "packet length indicator". This is contained in byte 11, and indicates in binary code (least significant bit transmitted first, the last three bits reserved for future use) the number of following useful bytes in the data body. In another version of the protocol, Bit L allows to distinguish between packets containing useful data (L = 0) and "control packets", i.e. packets containing service information such as access control information. Higher levels of the protocol define the structure of such packets, and it is not part of the present document.

## 6.6.5 User data group

The remaining data bytes in the line, except the last two, (byte 11 to 43) constitute the data carried for users of the service, bearing that Packet Address in that Data Channel. There are 33 data bytes available.

## 6.6.6 Cyclic Redundancy Check (CRC) word

The last two bytes contain a Cyclic Redundancy Check on the User Data Group.

### 6.6.6.1 Check word generation

The data to be checked is considered as a polynomial in x with the highest degree term transmitted first and the term of degree zero last.

This is divided, using modulo 2 arithmetic, by the polynomial:

$$x^{16} + x^{12} + x^5 + x^0$$

The remainder from this process, with the highest term transmitted first, is the CRC. Before transmission the bits are complemented (i.e. XOR-ed with FF hex).

### 6.6.6.2 Check result

The register of figure 21 is set to 0s. The serial data followed by the CRC is then entered. The check is satisfied if the register again contains all 0s.

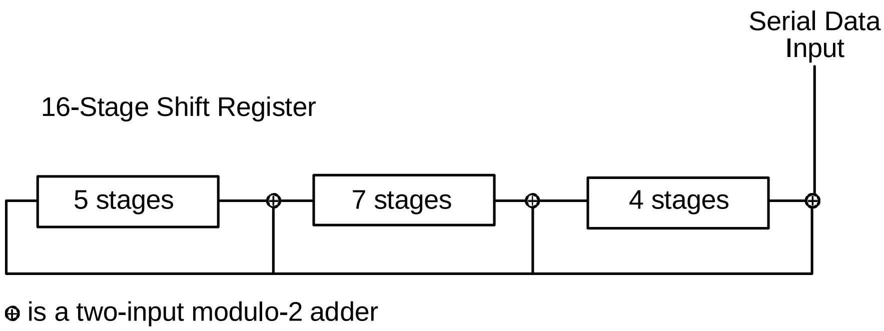



**Figure 21: Cyclic Redundancy Check (Datavideo)**

## 6.7 Low bit-rate audio

A number of applications have arisen where there is a requirement to carry low bit-rate audio signals within the Teletext signal. Examples are programme-related material such as audio description and programme-independent material such as "Talkback".

### 6.7.1 Method of coding

The general format of a low bit-rate audio IDL is shown in figure 22. The Data Channel (byte 4) and Designation Code (byte 5) are as defined in clause 6.4. Two channels have been allocated for this purpose as shown in table 16.

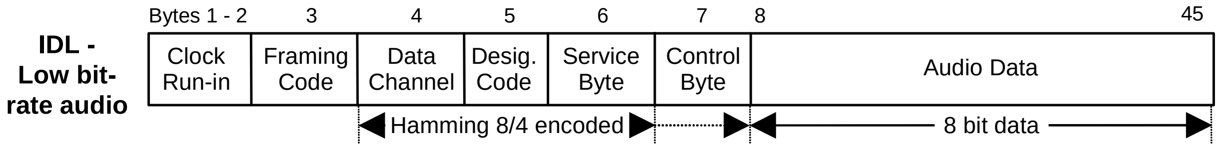



**Figure 22: Structure of an IDL for low bit-rate audio**

#### 6.7.1.1 Decoder action

Decoders shall decode the whole of bytes 6 and 7 to define a particular service. Low bit-rate audio Teletext packets may be multiplexed with other packets in the Teletext stream, thus the decoder shall mute the audio when there are no packets of the appropriate service being received within a time defined in that services session protocol.

### 6.7.2 Programme-related audio service

Programme-related audio services are allocated Data Channel 4. They are used to carry audio information related to the programme video. An example of such a service is AUDETEL.

#### 6.7.2.1 Service Byte (SB)

The Service Byte, byte 6, is Hamming 8/4 encoded.

The message bits are set to 0000 to designate an AUDETEL service. All other bit patterns are reserved.

It is envisaged that the reserved states will be used for future services and decoders should recognize all bit states.

#### 6.7.2.2 Control Byte (CB)

The Control Byte, byte 7, is available for control functions.

In an AUDETEL service this byte is designated the Fade byte and is Hamming 8/4 encoded.

The three least significant message bits, 1 - 3, are used to control the programme sound at the receiver as shown in table 19.

**Table 19: Fade byte**

| B4 | B3 | B2 | B1 | Interpretation |
|---|---|---|---|---|
| x | 0 | 0 | 0 | 0 dB attenuation |
| x | 0 | 0 | 1 | 4 dB attenuation |
| x | 0 | 1 | 0 | 8 dB attenuation |
| x | 0 | 1 | 1 | 12 dB attenuation |
| x | 1 | 0 | 0 | 18 dB attenuation |
| x | 1 | 0 | 1 | 24 dB attenuation |
| x | 1 | 1 | 0 | 30 dB attenuation |
| x | 1 | 1 | 1 | 0 dB attenuation |

The most significant message bit, B4, controls muting of the speech audio output, as given in table 20.

**Table 20: The meaning of bit B4**

| Bit 4 | Interpretation |
|---|---|
| 0 | Speech audio output unmuted |
| 1 | Speech audio output muted |

#### 6.7.2.3 Audio data

Bytes 8 to 45 contain audio data. In the AUDETEL format, bytes 8 to 26 contain the first audio frame and bytes 27 to 45 the second audio frame.

### 6.7.3 Programme independent audio service

Programme-independent audio services are allocated Data Channel 12.

#### 6.7.3.1 Service Byte (SB)

The Service Byte (SB), byte 6, is Hamming 8/4 encoded. The interpretation of the four message bits is shown in table 21. The SB is used to define whether the system is operating in single- or two-channel mode and how many VBI lines are being used for the service. In the two-channel mode each audio channel is totally independent of the other audio channel. In single-channel mode there are 38 bytes of audio data available. In two-channel mode there are 2 groups of 19 audio data bytes.

**Table 21: Service byte for independent audio service**

| B4 | B3 | B2 | B1 | Interpretation |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Single-channel mode using 1 VBI line per frame |
| 0 | 0 | 0 | 1 | Single-channel mode using 2 VBI lines per frame |
| 0 | 0 | 1 | 0 | Single-channel mode using 3 VBI lines per frame |
| 0 | 0 | 1 | 1 | Single-channel mode using 4 VBI lines per frame |
| 0 | 1 | 0 | 0 | Mute Channel 1 |
| 0 | 1 | 0 | 1 | Two-channel Mode using 2 VBI lines per frame |
| 0 | 1 | 1 | 0 | Mute Channel 2 |
| 0 | 1 | 1 | 1 | Two-channel Mode using 4 VBI lines per frame |
| 1 | x | x | x | User-defined service |

In the single-channel mode, the channel shall mute in the absence of packets with the appropriate data channel address and Service byte. In the two-channel mode muting of individual channels is achieved by setting the appropriate bit states as shown in table 21. Both channels shall mute in the absence of packets with the appropriate channel address and service byte. In the case where bit 4 is set for a user-defined service then bits 1, 2 and 3 are user-defined.

#### 6.7.3.2 Control Byte (CB)

The Control Byte (CB), byte 7, is used for the definition of control functions. If message bit 4 in the Service Byte (SB) is set to 0 then all the bit states in the Control Byte are reserved and shall be set to 0. If message bit 4 in the Service Byte is set to 1 then all the bit states in the Control Byte are user defined.

The use of Hamming protection for the Control Byte depends upon the application.

#### 6.7.3.3 Audio data

Bytes 8 to 45 contain audio data. In the two-channel format bytes 8 to 26 shall form Channel 1 and bytes 27 to 45 shall form Channel 2.

## 6.8 IDL Format B

### 6.8.1 Packet structure

The structure of a Teletext packet conforming to the IDL Format B definition is shown in figure 23. The electrical and timing characteristics of a Teletext packet plus the definition of the Clock Run-In and Framing Code fields are defined in [2]. The component fields are transmitted in the order Clock Run-in to User Data, and the bits of each field are transmitted least significant bit first.

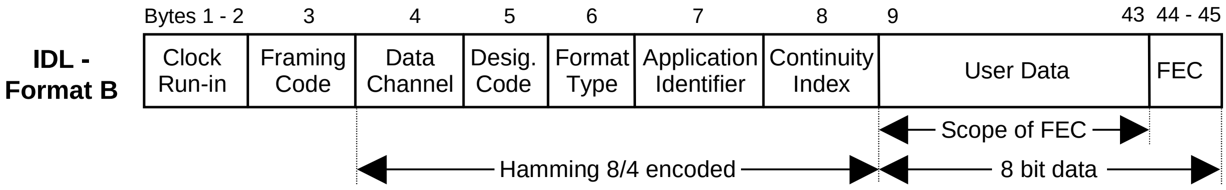



**Figure 23: Structure of an IDL Format B packet**

The Data Channel (byte 4) and Designation Code (byte 5) are as defined in clause 6.4. The coding of the Format Type byte allows a Format B packet to be distinguished from the Format A packet defined in clause 6.5. The allocation of Data Channels is shown in table 16.

All fields after the framing code and up to and including the Continuity Index byte (CI) are Hamming 8/4 encoded according to clause 8.2 in [2]. The remaining fields are constructed from bytes containing eight data bits.

### 6.8.1.1 Format Type

The Format Type (FT) byte, byte 6, is Hamming 8/4 encoded. Message bit 1 is set to 1 and message bit 2 is set to 0 to identify the packet type uniquely as Format B. Table 22 shows how a Format B IDL can be distinguished from a Format A IDL.

**Table 22: Coding of the four message bits in the Format Type byte**

| B4 | B3 | B2 | B1 | IDL Format Type |
|---|---|---|---|---|
| DL | CI | RI | 0 | Format A |
| AN1 | AN0 | 0 | 1 | Format B |
| X | X | 1 | 1 | Reserved for future use |

The two most significant message bits, AN1 and AN0, define a 2-bit Application Number value. The function of this is application dependent. One possible use is to distinguish between four separate services of a given application type in the same data channel.

### 6.8.1.2 Application Identifier

The Application Identifier (AI) byte, byte 7, is Hamming 8/4 encoded. It carries an Application Identifier, allowing up to 16 different applications to share the same Data Channel.

If the AN0 and AN1 bits in the Format Type byte are used to distinguish individual services, one Data Channel can carry a maximum of 64 services.

A register of Application Identifier values is maintained by the EBU and published in TR 101 231 [4].

### 6.8.1.3 Continuity Index

The Continuity Index (CI) byte, byte 8, is Hamming 8/4 encoded. This 4-bit value is incremented modulo 16 for each new packet for a particular service, i.e. for a given Data Channel, Application Identifier value and AN1, AN0 setting. Discontinuities in the sequence are not allowed to be transmitted so that a receiver can determine if any packets have been lost. The CI value is not incremented if a packet is repeated.

### 6.8.1.4 User Data

The User Data (bytes 9 to 43) carries the data of the application. These bytes shall be coded as 8 bit data at the transport layer.

### 6.8.1.5 FEC

The Forward Error Correction (FEC) bytes are carried in bytes 44 and 45. The algorithm is described in clause 6.8.2.

# 6.8.2 Forward Error Correcting algorithm

## 6.8.2.1 Introduction

Due to the unidirectional nature of VBI data transport, a forward error correction scheme is usually needed to ensure the integrity of the data at the receiver. The FEC scheme described here is based on industry standard t = 1 Reed Solomon encoding and decoding techniques, popular in CD ROM systems. t = 1 implies there are two protection bytes per unit block of data (in this case a single Teletext packet). Bytes 44 and 45 of the packet convey the error suffix bytes, figure 23. An algorithm is applied to the remaining 35 User Data bytes of a packet to determine the values for the two suffix bytes.

## 6.8.2.2 Encoding

The FEC scheme is applied to a "bundle" of 490 application bytes, i.e. the total payload of the User Data fields of 14 packets, excluding the FEC suffix bytes. Each bundle is transmitted via 16 instances of the packet shown in figure 23. These packets are considered to be arranged as a block as shown in figure 24. Thus the FEC works within a block of 16 rows and 37 columns, containing 14 rows and 35 columns worth of application data. The packets required to transmit a block are identified uniquely through their Continuity Index (CI) bytes. The actual value of the CI byte in a given packet is determined by the position of the packet within the block. The first packet has a count of 0, and the last a count of 15.

Packets with CI values of 14 and 15 contribute vertical protection to the bundle. The values inserted at each column position in the User Data field are calculated by applying the FEC algorithms described in clause 6.8.2.3 to the 14 application bytes at the same column position in the packets with CI values 0 to 13 inclusive.

Each packet has two FEC suffix bytes at column positions 44 and 45. These provide horizontal protection. For each of the 16 packets, the values for the suffix bytes are calculated by applying the FEC algorithms described in clause 5.3 to the 35 bytes in the User Data field.

| No. of Bytes: | 2 | 1 | 1 | 1 | 1 | 1 | 1 | | 35 | | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CR | FC | DC | 1111 | FT | AI | CI = 0 | $D_0$ $D_1$ | $D_n$ | $D_{33}$ $D_{34}$ | $S_0$ | $S_1$ |
| | CR | FC | DC | 1111 | FT | AI | CI = 1 | $D_0$ $D_1$ | $D_n$ | $D_{33}$ $D_{34}$ | $S_0$ | $S_1$ |
| | CR | FC | DC | 1111 | FT | AI | CI = 2 | $D_0$ $D_1$ | $D_n$ | $D_{33}$ $D_{34}$ | $S_0$ | $S_1$ |
| 16 Packets | | | | | | | | | | | | |
| | CR | FC | DC | 1111 | FT | AI | CI = 13 | $D_0$ $D_1$ | $D_n$ | $D_{33}$ $D_{34}$ | $S_0$ | $S_1$ |
| | CR | FC | DC | 1111 | FT | AI | CI = 14 | $P_0$ $P_1$ | $P_n$ | $P_{33}$ $P_{34}$ | $S_0$ | $S_1$ |
| | CR | FC | DC | 1111 | FT | AI | CI = 15 | $Q_0$ $Q_1$ | $Q_n$ | $Q_{33}$ $Q_{34}$ | $S_0$ | $S_1$ |

$D_x$ = User Data byte (490 in total), $P_x$ = Protection byte, $Q_x$ = Protection byte, $S_x$ = Suffix byte

**Figure 24: Transmitted Block of FEC Protected Data**

## 6.8.2.3 Calculating FEC suffix bytes

For a given packet, the two suffix bytes, $S_0$ and $S_1$, are calculated such that:

$$D_0 + D_1 + D_2 + D_3 + .... + D_{31} + D_{32} + D_{33} + D_{34} = S_0 + S_1 = X \quad (1)$$

$$a^{36} \text{ x } D_0 + a^{35} \text{ x } D_1 + a^{34} \text{ x } D_2 + .... + a^4 \text{ x } D_{32} + a^3 \text{ x } D_{33} + a^2 \text{ x } D_{34} = a \text{ x } S_0 + S_1 = Y \quad (2)$$

where the 35 User Data Bytes in the packet are represented by $D_0$ to $D_{34}$ (or $P_0$ to $P_{34}$ for the packet with CI = 14 or $Q_0$ to $Q_{34}$ for the packet with CI = 15) and **a** is the primitive element (value = 0x02) of the GF($2^8$) Galois Field.

It is necessary to have a definition of the process of multiplication by the primitive element. The primitive polynomial used here is :

$$X^8 + X^4 + X^3 + X^2 + 1$$

which effectively defines "multiplication" of a value by **a** as being equivalent to loading that value into a feedback shift register with the structure shown in figure 25 and clocking it once.

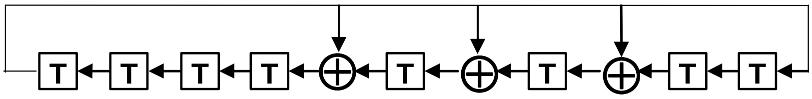



**Figure 25: Primitive Polynomial Shift Register**

From equations (1) and (2) above, $S_0 + S_1 = X$ and $a \times S_0 + S_1 = Y$, leading to $X + Y = (1 + a) \times S_0$

Thus $S_0 = (X + Y) / (1 + a)$ and $S_1 = S_0 + X$

where 1 + a = 0x03, log_a[j] - log_a[k] is used to perform the division, and X + Y implies X XOR Y.

There is an exception case to account for when X = Y. If this occurs, X + Y = 0 and the log_a[0] cannot be taken. Then by definition $S_0$ must be zero, otherwise $X = S_0 + S_1$ and $Y = a \times S_0 + S_1$ cannot be true simultaneously.

Thus when X = Y, $\mathbf{S_0 = 0}$ and $\mathbf{S_1 = X}$**.**

The suffix bytes are calculated for each of the first 14 packets, i.e. those carrying the User Data. $S_0$ is inserted at column position 44, and $S_1$ at column position 45.

The contents of the 35 User Data bytes in the last two packets of the block are generated by applying the same algorithm in the vertical direction to the 14 bytes at the same column position in packets with CI values 0 to 13, bearing in mind that there are only 14 data bytes instead of 35. The $S_0$ value is inserted in packets with CI = 14 ($P_0$ to $P_{34}$), and the $S_1$ value in packets with CI = 15 ($Q_0$ to $Q_{34}$). The process is symmetrical in that the suffix bytes for packets with CI = 14 or 15 are identical whether they are calculated vertically or horizontally.

### 6.8.2.4 Decoding

A decoder assembles the 16 packets into an array and removes all the data other than the User Data and suffix bytes. The decoding algorithm is applied to each row of the array in turn and it is possible to detect and correct a single byte error within each packet. If a row contains more than one error then it is corrupted beyond the correction capability of the horizontal code. The algorithm in then applied in the vertical direction and it may be possible to correct the erroneous bytes, depending on the number and location of the errors. Iterative processing techniques can be used to allow correction of a wider range of error patterns than is possible with a single pass in both the horizontal and vertical directions. Since the horizontal and vertical FEC are the same (except for the number of bytes of data to be protected), the algorithms here will work in both directions.

It should be noted that if a complete packet is lost from the block, it is sometimes possible to reconstruct the missing data. This is certainly true if there are no other errors within the block.

### 6.8.2.5 Detecting and correcting single byte errors

Equations [1] and [2] in clause 6.8.2.3 define the encoding algorithm for a packet such that:

$$D_0 + D_1 + D_2 + D_3 + .... + D_{31} + D_{32} + D_{33} + D_{34} + S_0 + S_1 = 0$$

$$a^{36} \times D_0 + a^{35} \times D_1 + a^{34} \times D_2 + .... + a^4 \times D_{32} + a^3 \times D_{33} + a^2 \times D_{34} + a \times S_0 + S_1 = 0$$

On reception the following calculations are made for a given packet:

$$X = R_0 + R_1 + R_2 + R_3 + .... + R_{31} + R_{32} + R_{33} + R_{34} + R_{35} + R_{36}$$

$$Y = a^{36} \times R_0 + a^{35} \times R_1 + a^{34} \times R_2 + .... + a^4 \times R_{32} + a^3 \times R_{33} + a^2 \times R_{34} + a \times R_{35} + R_{36}$$

where $R_0$ to $R_{34}$ are the received bytes in the User Data field, and $R_{35}$ and $R_{36}$ are the received versions of the suffix bytes.

If X = Y = 0, there are no errors; otherwise

$$X = E_i \text{ and } Y = a^{36-i} \text{ x } E_i.$$

The error value, $E_i$, is given directly by the value X, and the error position (p) is calculated using a table of log_a[n].

If the p > 34 then there are at least two errors in that row.

The same algorithm is used vertically for each column, bearing in mind that there are only 14 data bytes instead of 35.

#### 6.8.2.6 Correcting data blocks

The results from checking the received data is interpreted as follows:

1) With no errors in a row or column either:

   - The FEC bytes match; there are probably no errors,
   - The FEC bytes fail to match in a way which could be caused by a single-byte error; in this case the location and value of such an error can be found,
   - The FEC bytes fail to match in a way that indicates that there are at least two bytes in error.

2) With one error in a row or column, it can be corrected and then either:

   - The FEC bytes now match; there are probably no other errors,
   - The FEC bytes do not match; there is at least one more error that cannot be corrected.

3) The capability to detect errors is lost once there is a double byte error within a row or column. Given the location of two errors within the whole block, it is possible to correct them assuming all the other bytes are correct. It is possible to use this FEC to correct double errors by marking errors and performing multiple passes in both the horizontal and vertical directions.

### 6.8.3 Effective data rate

Using the FEC scheme defined above, the payload per packet is 35 bytes. The FEC scheme requires the exclusive use of two packets in every 16. Therefore, the effective data rate in bits per second per TV line for 625 line systems is given by:

35 (bytes) x 8 (bits/byte) x 50 (packets/s) x 14/16 = 12 250 bit/s

In a full channel application using 300 packets per field, the effective rate is 3,675 Mbit/s.

# 7 IDL - CA (type A)

## 7.1 General points

This method employs the same CA system as in the Page Format with CA. It is possible with this method for the broadcaster to switch between IDL - CA and Page Format - CA. Data in the User Data Group (see clause 6.5.7) of a number of Independent Data Service Packets are linked to form Data Blocks. They may contain messages concerning access that are not Encrypted, Encrypted Messages concerning access and User Data to be communicated.

## 7.2 Advantages

As with the Page Format - CA it is the network operator who provides the CA system as a service to the provider of the user data. All the user data and the CA information is supplied under one service packet address. This method is optimum for short data packets. It is possible for the network operator to switch between Page Format - CA and IDL - CA mode. The broadcaster is adding value.

# 7.3 Disadvantages

A special decoder is required with the appropriate CA system.

# 7.4 Methods of coding

The data is packed into Teletext packets as described in clause 6. Within the user data, in any position, the following data block structure is included. There is no alignment necessary between the Teletext packet structure and the structures described below.

## 7.4.1 Block Separator

Byte 1 of a data block.

Blocks of data are separated by the transmission of the character hex 10 (DLE). If the character DLE occurs within the Data Block it shall be repeated to indicate that it is not a block separator.

## 7.4.2 Block Formats

Block Format A is described in clause 7.4.3. There is scope for defining other Block Formats.

## 7.4.3 Block Format A

| | | | | |
|---|---|---|---|---|
| DLE | Block Type 04/0X | Message Not Encrypted | Encrypted Message | User Data |

**Figure 26: General Form of Data Block - Block Format A**

"User Data" is the data stream originated by a sender and intended for delivery to a specified recipient or group of recipients. "Messages" are groups of data concerned with the access to User Data and are for controlling the decoder or placing it in operation. A suitable encryption algorithm is shown in figure 15. The general form of a Data Block is as shown in figure 26.

### 7.4.3.1 Block Types

Byte 2 of a data block defines the block type.

In Format A this byte is of the form 04/0x where x may have a value of 0 to 7, as shown in table 23.

**Table 23: Block Types defined for Block Format A**

| Block Type | Interpretation |
|---|---|
| 04/00 | Block contains User Data not Scrambled (see figure 13) |
| 04/01 | Block contains Key Message not Encrypted, and the Encrypted Key Message |
| 04/02 | Service Numbers and Sequence Numbers, not encrypted and Scrambled User Data |
| 04/03 | System Key Message Block |
| 04/04 | Shared User Message Block |
| 04/05 | Unique User Message Block |
| 04/06 | Service Address Message Block for Independent Data Services |
| 04/07 | Service Address Message Block for Page Format Services |

### 7.4.3.2 Primary Block Key Messages

Block Type Code 04/01 (see figure 27).

| DLE | Block Type | In Use System | Service | Set | Current | Service | Block |
|---|---|---|---|---|---|---|---|
| | 04/01 | Key Label | Modes | to 0 | System Key | Identification | Key |
| 8ábits | 8ábits | 8ábits | 2ábits | 6ábits | 56ábits | 8ábits | 56ábits |
| DATA NOT ENCRYPTED | | | ENCRYPTED WITH CURRENT SYSTEM KEY | | | | |

| Key Value | Scrambling | Set |
|---|---|---|
| in Credit Units | Method | to 0 |
| 16ábits | 5 Bits | 3 Bits |
| ENCRYPTED WITH CURRENT SYSTEM KEY | DATA NOT ENCRYPTED | |

**Figure 27: Block Type 04/01 - Primary Block Key Messages**

Refer to Page Format - CA for further details on the various elements of the blocks (see clause 5).

### 7.4.3.3 Secondary Block Messages and Scrambled User Data

Block Type Code 04/02 (see figure 28).

| DLE | Block Type | Service | Sequence | Scrambled |
|---|---|---|---|---|
| | 04/02 | Identification Number | Number | User Data |
| 8ábits | 8ábits | 8ábits | 8ábits | |
| DATA NOT ENCRYPTED | | | | |

**Figure 28: Block type 04/02 - Secondary Block Messages and Scrambled Data**

The Sequence Number may be used as the cipher initial variable. The rest of this data block contains the scrambled data (maximum 1 024 bytes) intended for receipt by the addressee(s).

### 7.4.3.4 System-Key Message Block

Block Type Code 04/03 (see figure 29).

| DLE | Block Type | Encryption | New System | Current System | New System | Current System |
|---|---|---|---|---|---|---|
| | 03/03 | Method | Key Label | Key Label | Key | Key |
| 8ábits | 8ábits | 8ábits | 8ábits | 8ábits | 56ábits | 56ábits |
| DATA NOT ENCRYPTED | | | | | ENCRYPTED WITH NEW SYSTEM KEY | |

**Figure 29: Block Type 04/03 System Key Block Messages**

### 7.4.3.5 Shared-User Message Block

Block Type Code 04/04 (see figure 30).

| DLE | Block Type | Shared User | Set | New System | User Enabling |
|---|---|---|---|---|---|
| | 04/04 | Address | to 0 | Key | Bits |
| 8ábits | 8ábits | 20ábits | 4ábits | 56ábits | 152ábits |
| DATA NOT ENCRYPTED | | | | ENCRYPTED WITH SHARED DISTRIBUTION KEY | |

**Figure 30: Block Type 04/04 Shared User Message Block**

### 7.4.3.6 Unique User Message Block

Block Type Code 04/05 (see figure 31).

| DLE | Block Type | Unique User | Service | Service |
|---|---|---|---|---|
| | 04/05 | Address | Mode | Reference Number |
| 8ábits | 8ábits | 32ábits | 2ábits | 2ábits |
| NOT ENCRYPTED | | | ENCRYPTED WITH UNIQUE KEY | |

| Current or New Shared Address | Unique Equipment Key | 7 Service Numbers 8 Bits Each | Current or New Shared Distribution Key | Current or New User Enabling Bit Position |
|---|---|---|---|---|
| 20ábits | 56ábits | 56ábits | 56ábits | 8ábits |
| ENCRYPTED WITH UNIQUE KEY | | | | |

**Figure 31: Block Type 04/05 Unique User Message Block**

### 7.4.3.7 Service Address Message Block - Independent Data Service

Block Type Code 04/06 (see figure 32).

| DLE | Block Type 04/06 | Service Number | Address Length | Set to 0 | Service Address | Group Repeated for Each Service Number |
|---|---|---|---|---|---|---|
| 8ábits | 8ábits | 8ábits | 3ábits | 5ábits | up to 24ábits | |
| DATA NOT ENCRYPTED | | | | | | |

**Figure 32: Block Type 04/06 Service Address Block Independent Data Services**

### 7.4.3.8 Service Address Message Block - Link to Page Format - CA

Block Type Code 04/07 (see figure 33).

| DLE | Block Type 04/07 | Magazine Number | Set to 0 | Service Number | Initial Page Number | Groups Repeated for Each Service Number |
|---|---|---|---|---|---|---|
| 8ábits | 8ábits | 3ábits | 5ábits | 8ábits | 8ábits | |
| DATA NOT ENCRYPTED | | | | | | |

**Figure 33: Block Type 04/07 Service Address Message Block - Link to Page Format - CA**

# Annex A (informative): Additional information for Page Format - Clear

## A.1 Use of Block Pointer, Block Separator and Block Length information

The Block Pointer, Block Separator and Block Length information are intended to assist a receiver in establishing and maintaining block synchronization with the application data stream. Starting from a position of no synchronization, the Block Pointer bytes should be checked to find the position in a packet where a Block Separator code should be present. Assuming the correct code is found where indicated, a receiver can interpret the following Block Length information to ascertain the expected position of the end of the data block. This should be followed by either a Block Separator or, since a Block Separator can only be placed every third byte, one or more Filler Bytes before the next Block Separator. The Block Pointer for that packet will also confirm the location of the next Block Separator. As all three elements are Hamming 8/4 coded, even with bit errors it should be possible for the receiver to have a high confidence that it has achieved correct synchronization.

It should be noted that in the event of a very short application data block which finishes within the packet in which it started, the Block Separator for the next application block is allowed to be inserted immediately following the end of the previous block. Consequentially, the start of the second block will not be indicated by any Block Pointer.

# History

| Document history | | |
|---|---|---|
| Edition 1 | March 1997 | Publication as ETS 300 708 |
| V1.2.1 | December 2002 | One-step Approval Procedure OAP 20030411: 2002-12-11 to 2003-04-11 |
| | | |
| | | |
| | | |