| | |
|---|---|
| Project | **IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>** |
| Title | **IEEE 802.16 TG4 OFDM PHY Proposal for the 802.16b PHY Layer** |
| Date Submitted | **2001-03-04** |
| Source(s) | Yossi Segal<br>Dr. Zion Hadad<br>Itzik Kitroser<br><br>RunCom Technologies LTD. 14 Moshe Levi St., Rishon Lezion, Israel<br><br>Voice: 972-3-9528440<br>Fax: 972-3-9528805<br>mailto:zionh@runcom.co.il<br>yossis@runcom.co.il<br>itzikk@runcom.co.il<br><br>Olivier Seller<br><br>Sacet, Ker Lann, Bat Alliance, 35170 Bruz. France<br><br>Voice: 33-2-99571656<br>Fax: 33-2-99571617<br>mailto:olivier.seller@sacet.com<br><br>Dave Williams<br><br>Advanced Hardware Architecture<br><br>Voice: 1-509-334-1000<br>mailto:davew@aha.com |
| Re: | TG4 process of defining the PHY layer. |
| Abstract | This document contains a structure for the TG4 PHY development, it contains various PHY modes and describes the common structure between them. |
| Purpose | This proposal should be used for the PHY specification of the TG4. |
| Notice |  |
| Release |  |
| Patent Policy and Procedures |  |

# Table of Contents

# Table of Figures

# Table of Tables

# OFDM/OFDMA PHY proposal for TG4 PHY Development

## 1 Introduction

The PHY layer described in this clause is designated for operation in the unlicensed frequency bands of 5-6 GHz. The PHY parameters cover channel bandwidths of 5, 10, 20MHz.

The PHY is based on OFDM (Orthogonal Frequency Division Multiplex) modulation, supporting both TDMA (Time Division Multiple Access and OFDMA Orthogonal Frequency Division Multiple Access). In OFDM the information is imposed onto the medium by modulating multiple carriers transmitted in parallel. In TDMA, all carriers of an OFDM symbol are assigned to one transmitter. In OFDMA the carriers are divided into Sub-Channels. When the OFDMA concept is applied to the uplink, it allows users to operate with smaller power amplifiers, at expense of instantaneous data rate. On the other hand it allows allocating dynamically larger amounts of bandwidth to users capable of utilizing it in terms of their link budget. When applied to the downlink, OFDMA allows transmitting to multiple users in parallel with designated data streams, and allows improving the link budget of disadvantaged users by allocating to their Sub-Channels a larger fraction of their downlink transmit power.

The carrier spacing in frequency is dictated by the multipath characteristics of the channels in which the FWA system is designated to operate. As the channel propagation characteristics depend on the topography of the area and on the cell radius, the amount of Sub-Channels into which the channel is subdivided depends on the overall channel width and the carrier spacing. As the modulation is implemented using the FFT algorithm, the modes are designated by the FFT sizes 64 (optional), 256 or 1024 (mandatory). Another parameter controlling the multipath mitigation capability, at expense of overhead, is the time-domain “guard interval”. The size of the guard interval is programmable in the diapason of 1/32 up to 1/4 of the FFT interval duration.

Several options of dividing the carriers into Sub-Channels are supported. In one, the carriers are allocated contiguously; in other they are interleaved in a pseudorandom manner. The preference of one technique over other depends on implementation strategy and on deployment scenario.

## 2 Time Domain and Frequency Domain issues

In this section we focus on different aspects of the system Channel bandwidths, Regulatory masks, and a Time Domain and Frequency Domain description of the OFDM signal including the possible FFT and GI lengths.

### *2.1 Regulatory masks*

Channel bandwidths for frequencies of 5-6GHz use 5, 10, 20MHz channels. The spectral mask to be addressed should be compliant to:

[Here we should add the spectrum mask];

## 2.2 Supported FFT lengths and Guard Interval Lengths

The OFDM symbol duration, or the related carrier spacing in frequency, is the major design parameter of an OFDM system. The symbol duration is composed of the FFT interval and of the Guard Interval (GI). The Guard Interval, which constitutes an overhead, is closely related to the multipath delay spread parameter. In order to keep the overhead of the GI low, there is an intereste in increasing the FFT interval duration as much as possible.

The effective bandwidth of the transmitted signal is related to the carrier spacing and the number of carriers.
In order to calculate the sampling frequency for any bandwidth, we define the bandwidth efficiency using the next parameters:

$$BWEfficiency = \frac{F_s}{BW} \cdot \frac{53}{64} = \frac{\Delta f \cdot N_{used}}{BW}$$

$BW$ - denotes the channel bandwidth

$F_s$ - denotes the sampling frequency

$\Delta f$ - denotes the carrier spacing

$N_{used}$ - denotes the number of carriers used in the FFT

The Bandwidth efficiency should always be around 93-95%, in order to occupy the maximum usable bandwidth but still allow adequate RF filtering. From this notion we can extract the sampling frequency for each BW by:

$$F_s = BWEfficiency \cdot BW \cdot \frac{53}{64}$$

The conversion from carrier modulation values to time domain waveform is typically implemented by a FFT algorithm on blocks of size $2^n$. After the FFT, the time domain complex samples are transmitted at rate $F_s$. The carrier spacing is, therefore,

$$\Delta f = \frac{F_s}{N_{FFT}}$$

The number of carriers utilized is usually only about 83% of the FFT bins. For implementation reasons, this number is chosen to be about 83% of the nearest power of 2. This choice involves implementation aspects of antialiasing filters.

| Number of carriers used | FFT size |
|---|---|
| 53 | 64 |
| 213 | 256 |
| 849 | 1024 |

The number of active carriers on the US in all modes is chosen to be 2^n*53, for reasons to be detailed later.
** Note that the choice of FFT size is an artificial implementation parameter – the 212 carriers modulation can be implemented either with FFT of size 256, of with FFT of size 512 at double sampling rate. We will stick with the convention, in which OFDM modes are denoted by the "FFT size" which is the smallest power of two above the number of carriers.

The FFT interval duration is related to carrier spacing by

$$T_s = \frac{1}{\Delta f} = \frac{N_{FFT}}{F_s}$$

This specification allows for FFT sizes 64, 256, and 1024. A compliant device shall implement either 256 or 1024 FFT for any bandwidth (implementing more than one compliant FFT size or 64 FFT size is optional).

The following tables give some calculation of the Carrier Spacing, Symbol Duration and Guard Interval duration for different masks. These parameters fit the FFT 256 and 1K , were the sampling frequency is defined as: $F_s = \frac{8}{7 \cdot BW}$ (when using the 64 FFT sizes, the sampling rate is $F_s = \frac{1}{BW}$).

| Channel(MHz) | FFT size | 64 | 256 | 1024 |
|---|---|---|---|---|
| 5 | Sub-carrier (kHz) | 89.28571 | 22.32143 | 5.580357 |
| | Symbol (us) | 11.2 | 44.8 | 179.2 |
| | Guard =1/32 | 0.35 | 1.4 | 5.6 |
| | 1/16 | 0.7 | 2.8 | 11.2 |
| | 1/8 | 1.4 | 5.6 | 22.4 |
| | 1/4 | 2.8 | 11.2 | 44.8 |
| 10 | Sub-carrier (kHz) | 178.5714 | 44.64286 | 11.16071 |
| | Symbol (us) | 5.6 | 22.4 | 89.6 |
| | Guard =1/32 | 0.175 | 0.7 | 2.8 |
| | 1/16 | 0.35 | 1.4 | 5.6 |
| | 1/8 | 0.7 | 2.8 | 11.2 |
| | 1/4 | 1.4 | 5.6 | 22.4 |
| 20 | Sub-carrier (kHz) | 357.1429 | 89.28571 | 22.32143 |
| | Symbol (us) | 2.8 | 11.2 | 44.8 |
| | Guard =1/32 | 0.0875 | 0.35 | 1.4 |
| | 1/16 | 0.175 | 0.7 | 2.8 |
| | 1/8 | 0.35 | 1.4 | 5.6 |
| | 1/4 | 0.7 | 2.8 | 11.2 |

Table 1: Channelization vs. FFT size; carrier spacing, symbol duration and guard interval duration parameters

## 2.3 Time domain description.

Inverse-Fourier-transforming creates the OFDM waveform; this time duration is referred to as the useful symbol time ($T_u$). A copy of the last samples is inserted before the useful symbol time, and is called the Guard Interval (GI) its duration is denoted as a fraction of the useful symbol time as ($T_{gi}$). The two together are referred to as the symbol time ($T_s$). figure illustrates this structure:

## 2.4 Frequency Domain Description

The frequency domain description includes the basic symbol structure of an OFDM/OFDMA symbol.

### 2.4.1 OFDM description

An OFDM symbol is made up from carriers, the amount of carriers determines the FFT size used. There are several carrier types:

- Data carriers – for data transmission
- Pilot carriers – for different estimation purposes
- Null carriers – no transmission at all, for guard bands and DC carrier.

Figure 1 illustrates such a scheme:

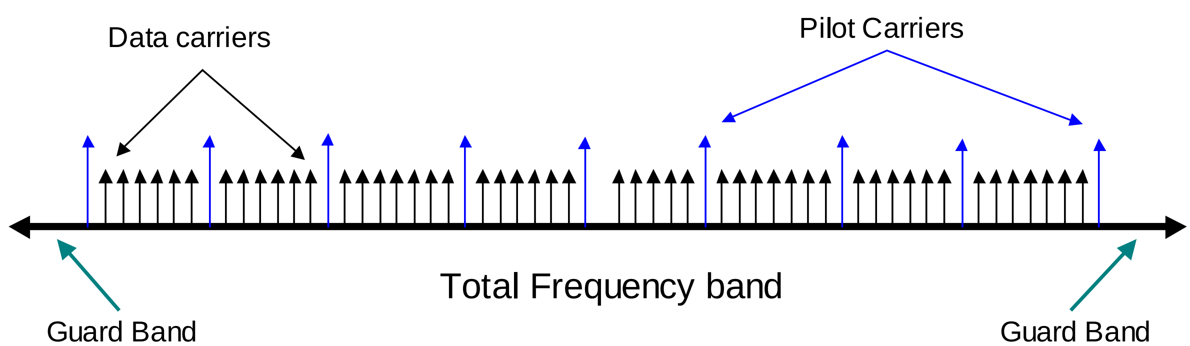



Figure 1: OFDM frequency description

The purpose of the guard bands is to enable the signal to naturally decay and create the FFT "brick Wall" shaping.

### 2.4.2 OFDMA description

Further enhancement to the OFDM symbol is added, when dividing it to enable OFDMA. OFDMA concept is based on the grouping of several carriers into one entity; this logical entity is called Sub-Channel Figure 2describes such a scheme:

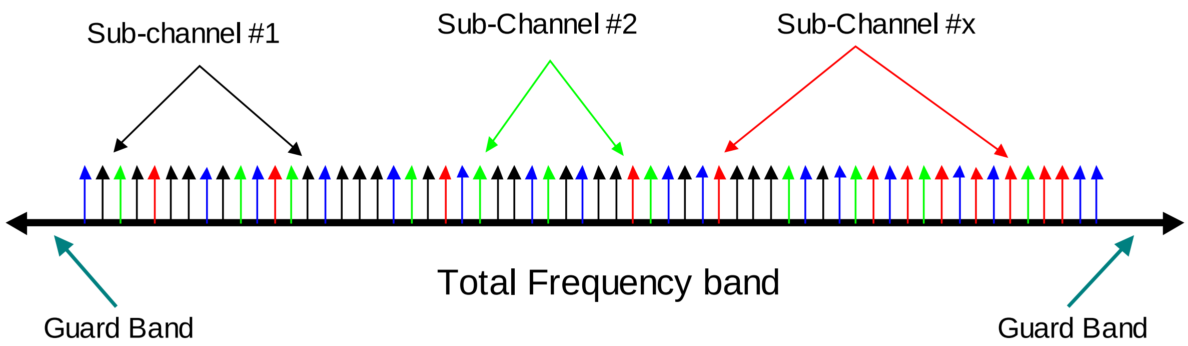



Figure 2: OFDMA frequency description

In order to use several FFT sizes but remain in the same block size (of data transmission), Common structure of Sub-Channel is defined for the US (see section 3.3.1). The same amount of data carriers (48) is used to define a Sub-Channel for the US is the appropriate modes of operation.

The 4096, 2048, 1024, 512 point FFT contains 64, 32, 16, 8 Sub-Channels respectively.
This partitioning gives several powerful added values, some of the important ones are:

- Frequency diversity – due to the spreading in the frequency band
- Power concentration – which allows the concentration of all the power some of the carrier (most usable on the users side)
- Forward Power Control – by allocating digitally different power amplification to the Sub-Channels (most usable on the Base-Station side)
- Interference spreading for each Sub-Channel – due to the frequency diversity

The symbol is divided into logical sub-channels to support scalability, multiple access, and advanced antenna array processing capabilities. The sub-channel structure will depend on the purpose for the sub-channelization. This structure can be described by a tone mapping function. For wideband processing, the mapping is based upon a special permutation code, which distributes consecutive symbols across the available bandwidth. For narrowband processing and advanced solutions employing antenna array processing, the tone mapping places consecutive symbols next to adjacent tones.

A schematic drawing of the Sub-Channel could be described as in Figure 3 (it describes only the logical Sub-Channels, regardless of the carriers dispersal in used carriers space):

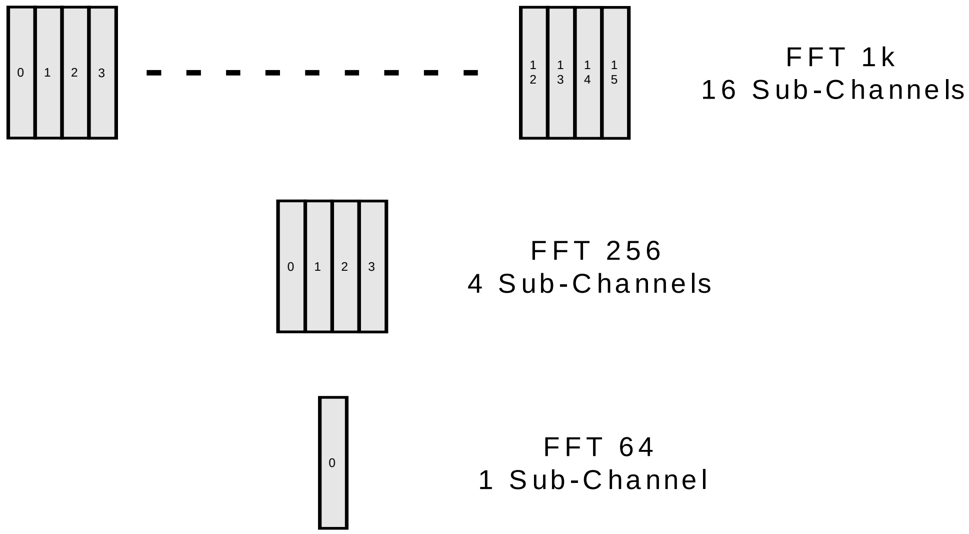



Figure 3: Sub-Channels schematic drawing within an OFDMA symbol

#### 2.4.2.1 Power Concentration and Adaptive Power Control

The OFDMA access in the downlink and uplink has many advantages. The biggest advantage beside the long symbol duration is the power concentration it enables. The power concentration is achieved due to power emission only on the Sub-Channels allocated. Therefore the energy of the user is transmitted only on selected carriers and not on the all-useable carries. By this technique users and Base Station can manipulate the amount of energy putted on different Sub-Channels. This power concentration can add up to **15dBb** per carrier when transmitting from the user, Comparing the power that could be emitted on all the bandwidth, for one Sub-Channel using 53 carriers, combined with a Backward APC (Automatic Power control) will give the optimum performance.

The Base Station can also regulate the amount of power on the different Sub-Channels and reach as much as 4dB concentrations gain. This technique is referred to as Forward APC (Automatic Power control), and is used in order to regulate the power to the users on the down stream.

This power concentration leads to several advantages:

- Better coverage
- Enable a larger APC range which is vital for larger cells
- Excellent Reuse factor
- Better channel availability
- Can use simpler and cheaper PA
- Can have better SNR for a transmitted signal
- Reach the distances specified for the system (better distances with the same EIRP).

Anti jamming advantages

## 2.5 Symbol Structuring Description

Conceptually, the PHY can be described in terms of upper and lower physical layers. As part of the upper physical layer, higher layer (data link, transport, session, etc.) information and PHY management data (e.g., training or synchronization) are mapped to symbols. The upper physical layer includes functions such as channel coding, bit interleaving, and modulation to form data symbols. As shown in Figure 4, the lower physical layer maps the data symbols to tones and forms OFDM symbols.

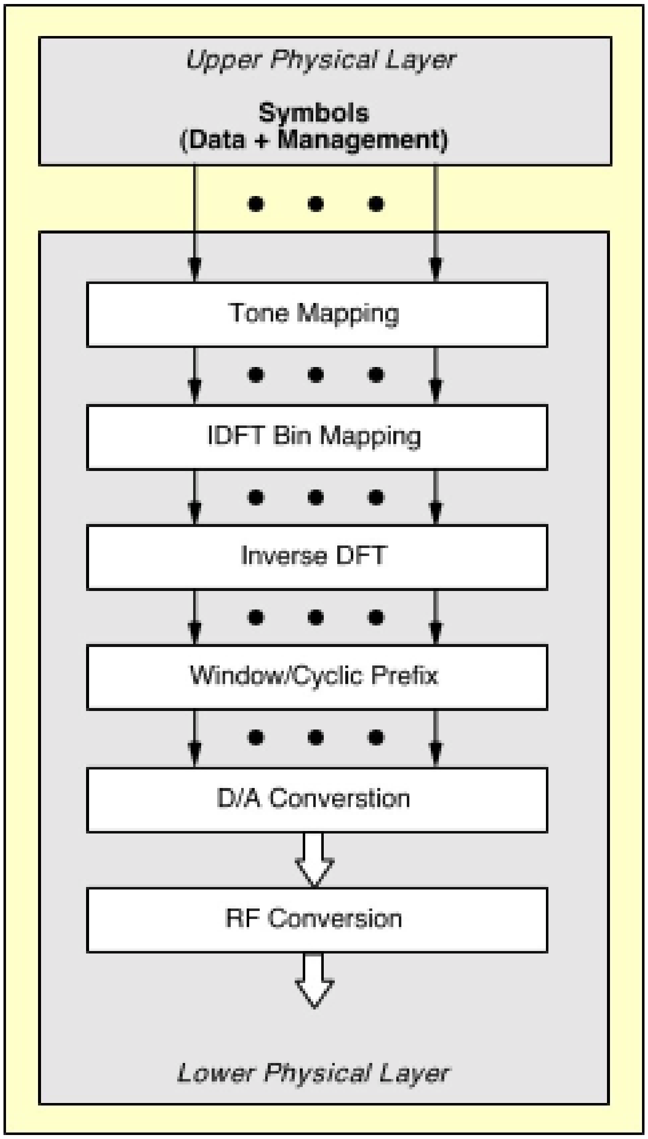



Figure 4: Lower Phyiscal Layer

The IDFT mapping function assigns usable carriers to specific DFT bin locations and zeros for guard bins. As shown in Figure 5, the tone mapping function assigns data symbols and PHY management data to specific carriers within a specific OFDM symbol.

In the case TDMA is used as access method, the mapping assigns all sub-carriers to a single transmitter.

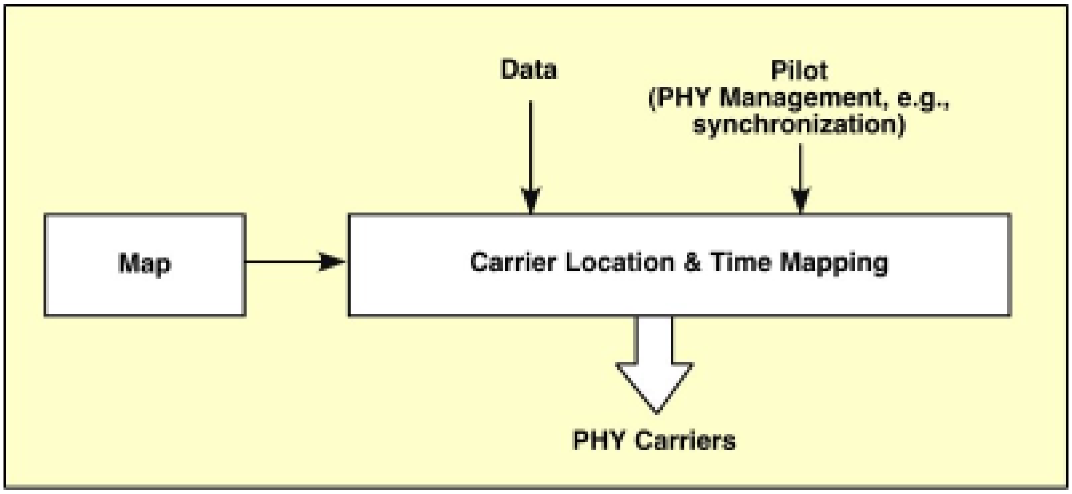



Figure 5: Tone Mapping Function

## 3 Multiple Access and Framing

The following description refers to the system access methods. There are two basic approaches for the system usage:

1. The first, where several OFDM symbols are used for data transmission (all Sub-Channels in the OFDM symbol are used for data only) and other OFDM symbols are used for synchronization (all Sub-Channels in the OFDM symbol are used for ranging only) this scheme is used by the FFT sizes 64 and 256.

   Figure 6 illustrates this structure:
   User Symbols: they include Sub-Channels where users transmit data
   Ranging symbols: they allow contention-based access
   Null Symbol: this optional symbol could be used to help allocate jamming and interferers

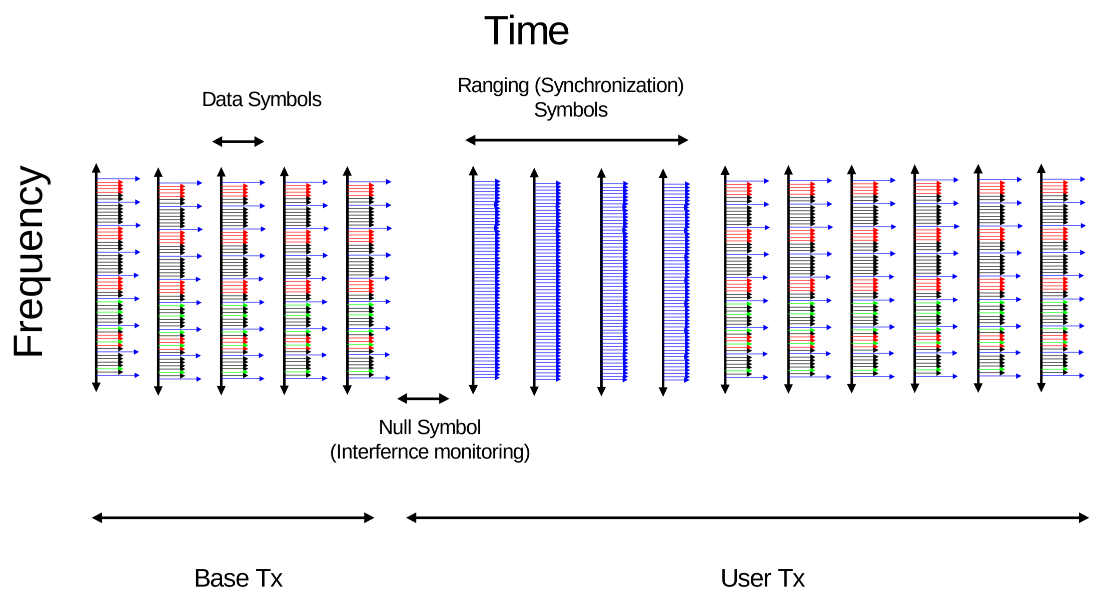



Figure 6: Conceptual Multiple Access Method for FFT size 64 and 256

2. The second uses all OFDM symbols to transmit both data and ranging signals (some Sub-Channels are used for data transmission and the other are used for ranging transmission), the number of symbols allocated to up stream and down stream are adaptable. This is used by the 1k FFT mode, Figure 7 illustrates this structure:

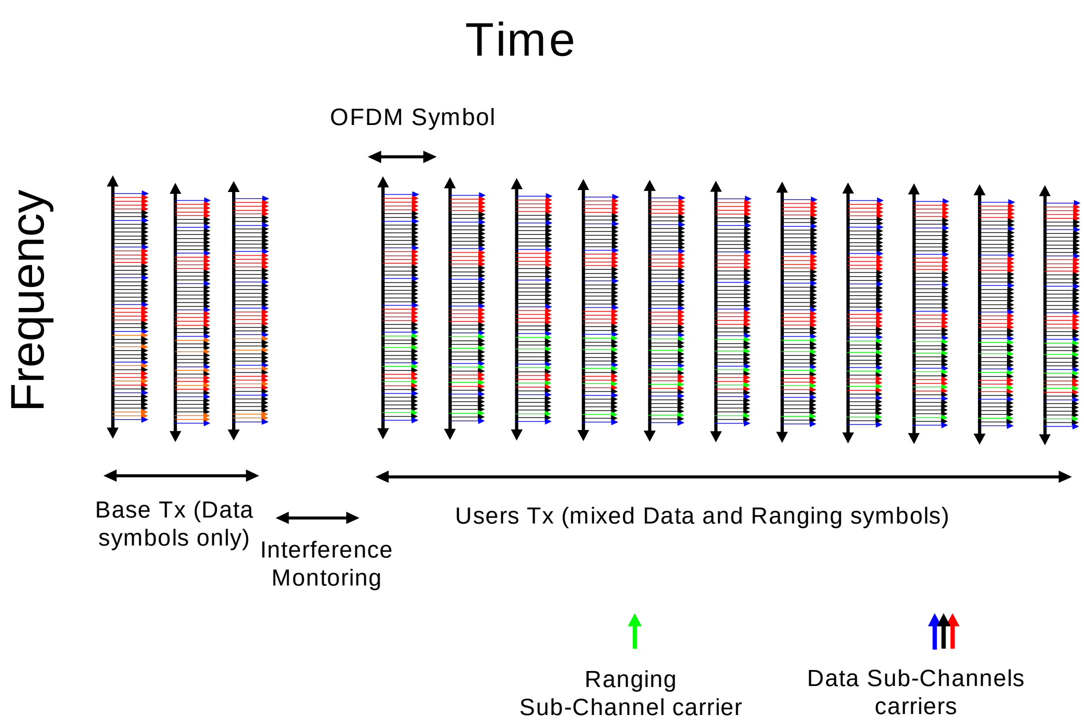



Figure 7: Conceptual Multiple Access Method for FFT size 1k

## *3.1 Duplexing*

Both frequency division duplex (FDD) and time division duplex (TDD) modes provide for bi-directional operation, and much of the PHY proposal is independent of the choice

## *3.2 Down-Stream*

### 3.2.1 Framing Structure

The framing structure used for the DS includes the transmission of a PHY control and US mapping, which is transmitted in the most robust coding and modulation of the system followed by transmission using modulation and coding schemes as defined in the PHY control. The MAC layer also defines the DS transmission frame length and the length of the different transmission parts. Figure 8 illustrates the DS framing:

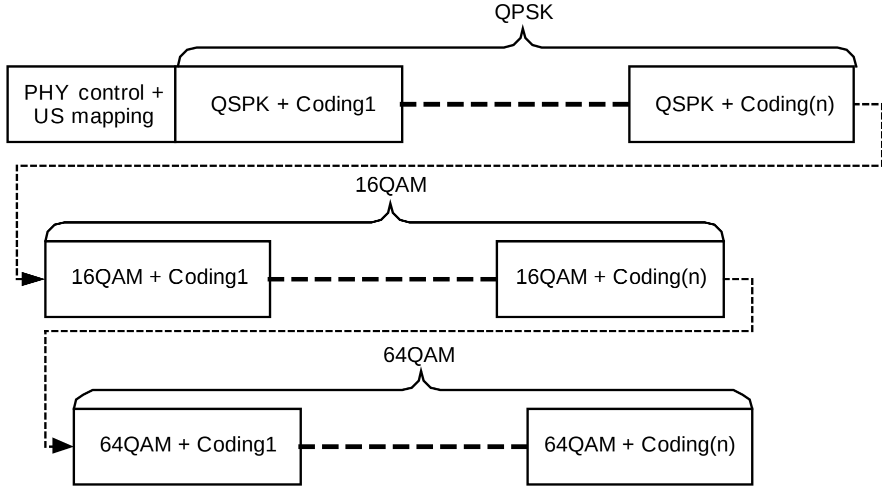



Figure 8: DS Framing

#### 3.2.1.1 For FFT size 64 and 256

The transmission of the DS is performed on the Sub-Channels of the OFDMA symbol, the amount of Sub-Channels needed for the different transmissions (modulation and coding) and their mapping is defined in the PHY control. The mapping of the Sub-Channels is performed in a two-dimensional grid, but the allocation of the Sub-channels is continues in the time-frequency, Figure 9 domain illustrates a possible two-dimensional transmission mapping (every color represents a different Modulation and coding scheme):

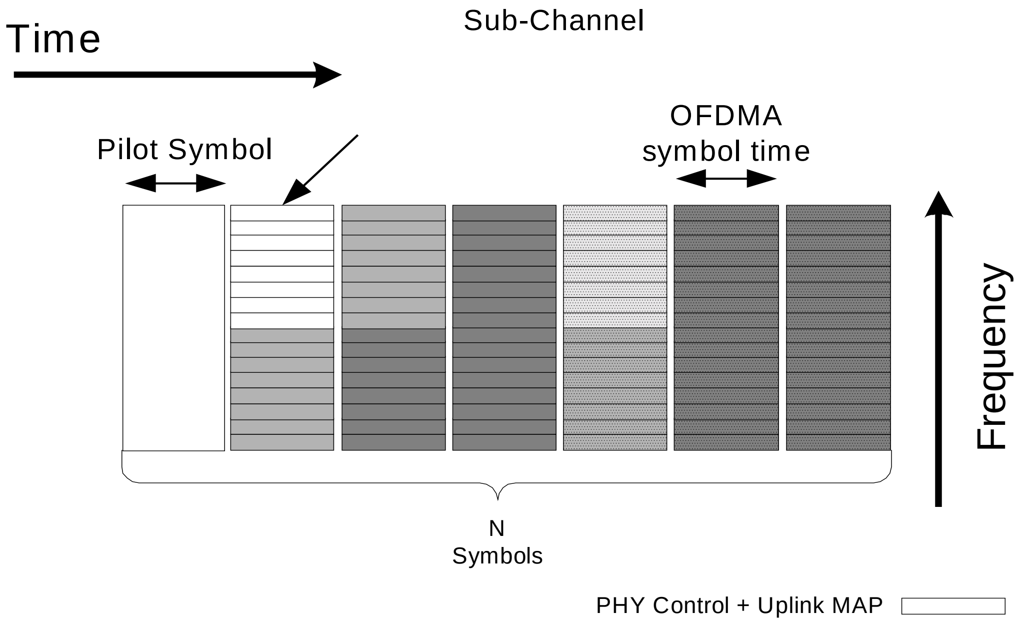



Figure 9: DS framing for FFT sizes of 64 and 256

### 3.2.1.2 For FFT 1k

The transmission of the DS is performed on the Sub-Channels of the OFDMA symbol, the amount of Sub-Channels needed for the different transmissions (modulation and coding) and their mapping is defined in the PHY control. The mapping of the Sub-Channels is performed in a two-dimensional grid, involving the Sub-Channels in the frequency domain and OFDMA symbols in the time domain. Figure 10 illustrates a possible two-dimensional transmission mapping (every color represents a different Modulation and coding scheme):

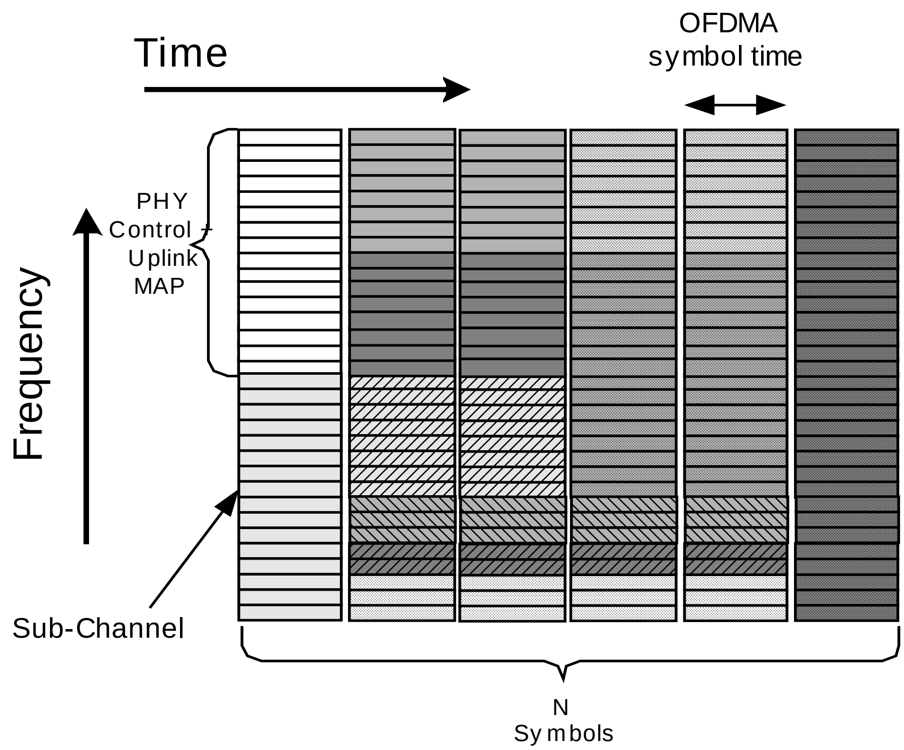



Figure 10: DS framing for FFT size of 1k

### 3.2.2 Symbol Structure

The symbol structure for the DS transmission is specified in the following sections 3.2.2.1, 3.2.2.2.

#### 3.2.2.1 FFT size 64 and 256

When using FFT size of 64 through 256, the symbol structure for is made up of constant and variable location pilots, which are spreaded all over the symbol, and from data carriers, which are divided into Sub-Channels. The amount of Sub-Channels differs between the different FFT sizes.

The data symbol structure is comprised of data carriers and pilot carriers. The data symbols are produced with a modulo 13 repetition (L denotes the modulo 13 index of the symbol), the location of the pilot symbols are shifting for every symbol produced, the first symbol (L=0) is produced after the all-pilot symbols (preamble/midamble). The basic pilot location differs between the different FFT sizes, we denote the first symbol variable pilot location - $\{BasicVariableLocationPilots_0\}$, then the pilot location as a function of L is - $\{BasicVariableLocationPilots_0 + L\}$ where L=0..12. Constant pilots are also defined in $\{BasicCon\tan tLocationPilots\}$ **Error! Reference source not found.** illustrates the symbol structure.

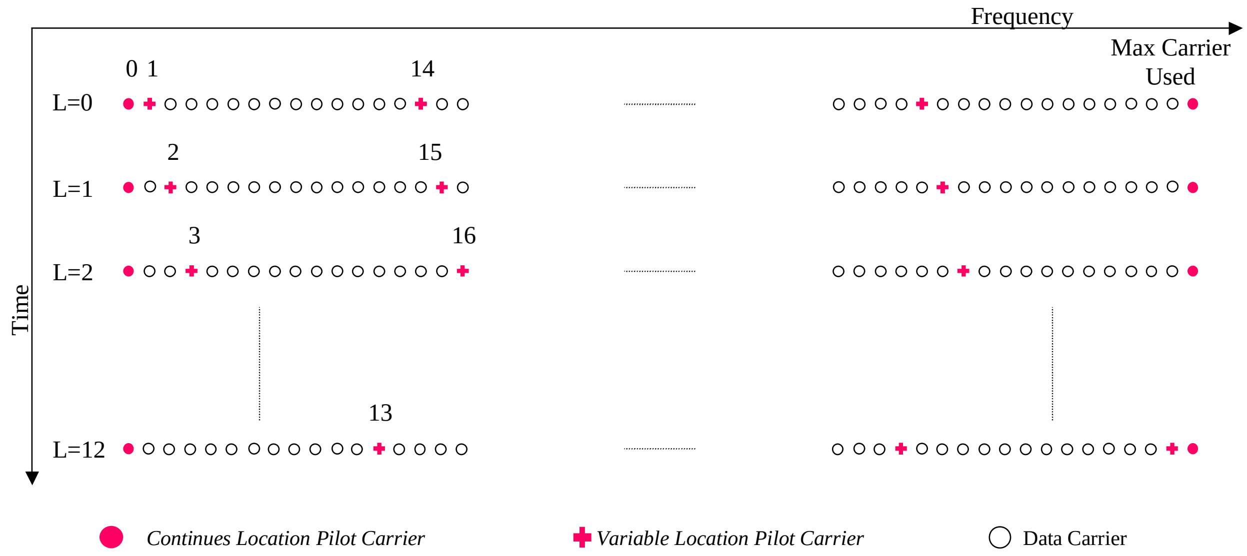



Figure 11: pilots and data carrier location the DS symbol using FFT 64 and 256

After mapping the pilots, the rest of the carriers (not including the DC carrier, which is not used) are data carriers scattered all over the usable spectrum (we should mention that the exact location of those carriers changes as a function of the symbol number which is modulo 13).

In order to achieve the DS Sub-Channels, the data carriers are grouped into one space (in acceding order of their indices) and then divided it into 48 basic groups. Each group containing a certain amount of carriers, and then special permutations are used to extract the Sub-Channels (each Sub-Channel is made up of 48 data carriers). Then using special permutations, we achieve the carrier location of each Sub-Channel.

**Error! Reference source not found.** illustrates the data carriers division into basic groups:

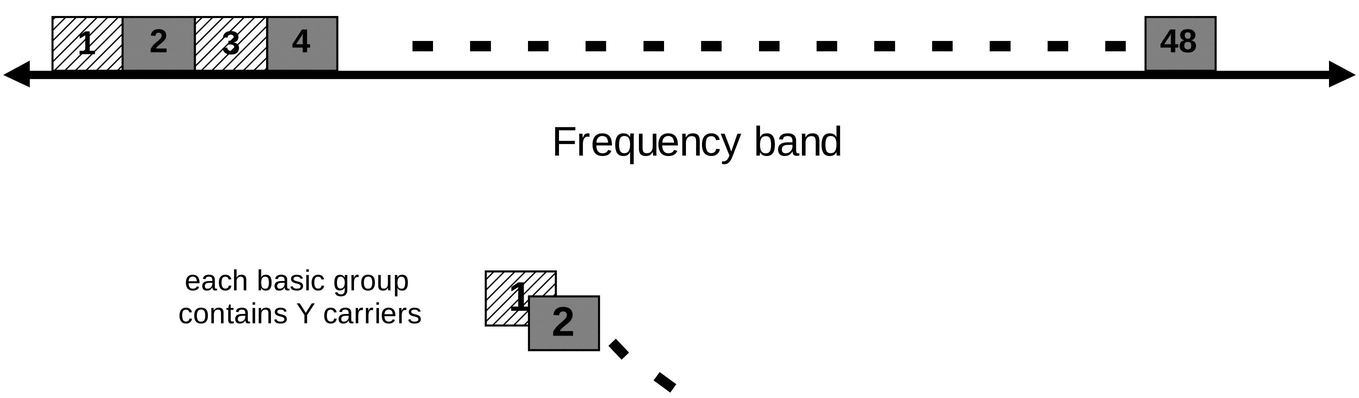



Figure 12: division of all carriers into basic groups for the DS OFDM symbol

#### 3.2.2.1.1 64 mode characterization

The parameters characterizing the 64 mode on the DS are as follow:

- Number of FFT points = 64

- Overall Used Carriers = 53
- Guard Bands = 6, 5 carriers on left and right sides of the spectrum respectively
- Number of Sub-Channels = 1

$\{BasicCon\tan tLocationPilots\}$ = None.

$\{BasicVariableLocationPilots_0\}$ = 0, 13, 27, 40

$\{PermutationBase\}$ = TBD

#### 3.2.2.1.2 256 mode characterization

The parameters characterizing the 256 mode on the DS are as follow:

- Number of FFT points = 256
- Overall Used Carriers = 213
- Guard Bands = 24, 23 carriers on left and right sides of the spectrum respectively
- Number of Sub-Channels = 4

$\{BasicCon\tan tLocationPilots\}$ = 0, 1, 211, 212.

$\{BasicVariableLocationPilots_0\}$ = 2, 15, 28, 41, 54, 67, 80, 93, 107, 120, 133, 146, 159, 172, 185, 198

$\{PermutationBase\}$ = TBD

### 3.2.2.2 FFT size 1k

When using FFT size of 1k the DS shall use OFDMA modulation technique (see section 2.4.2). The symbol structure for those FFT sizes is made up of constant and variable location pilots, which are spreaded all over the symbol, and from data carriers, which are divided into Sub-Channels. The amount of Sub-Channels differs between the different FFT sizes.

First allocating the pilots and then mapping the rest of the carriers to Sub-Channels construct the OFDMA symbol. There are two kinds of pilots in the OFDM symbol:

- Continues location pilots - which are transmitted every symbol
- Variable location pilots – which shift their location every symbol with a cyclic appearance of 4 symbols

The variable pilots are inserted in the locations defined by the next formula:

$$k = 3 * L + 12 * P_V$$

$k \in$ Indices from 0 to the number of Overall Usable Carriers

$L \in 0..3$ denotes the symbol number with a cyclic period of 4

$P_v \geq 0$ is an integer number

The pilot's locations are illustrated in Figure 13:

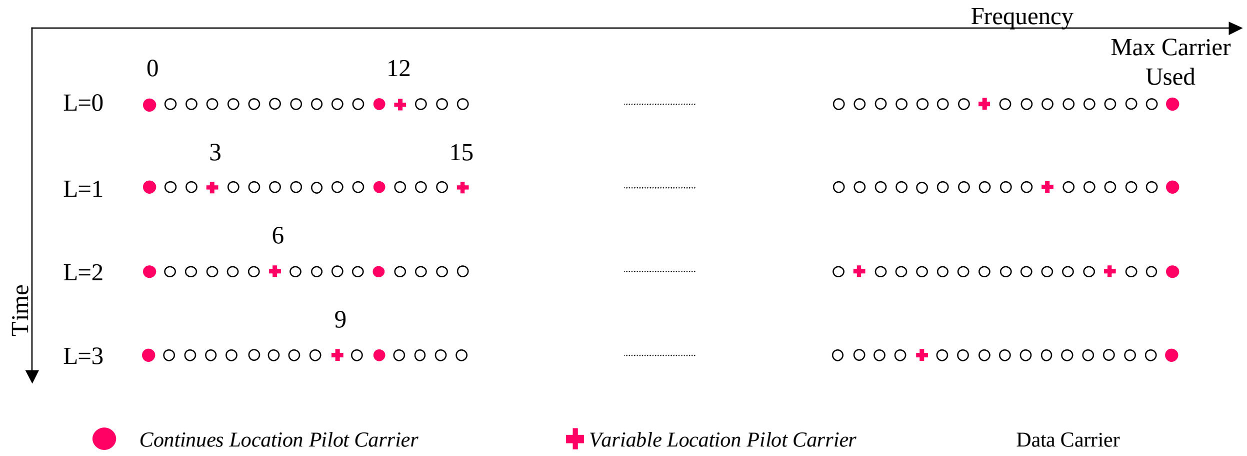



Figure 13: pilots and data carrier location the DS OFDMA symbol using FFT size of 1k

After mapping the pilots, the rest of the carriers (not including the DC carrier, which is not used) are data carriers scattered all over the usable spectrum (we should mention that the exact location of those carriers changes as a function of the symbol number which is modulo 4).

In order to achieve the DS Sub-Channels, the data carriers are grouped into one space (in acceding order of their indices) and then divided it into 48 basic groups. Each group containing a certain amount of carriers, and then special permutations are used to extract the Sub-Channels (each Sub-Channel is made up of 48 data carriers). Then using special permutations, we achieve the carrier location of each Sub-Channel.

Figure 14 illustrates the data carriers division into basic groups:

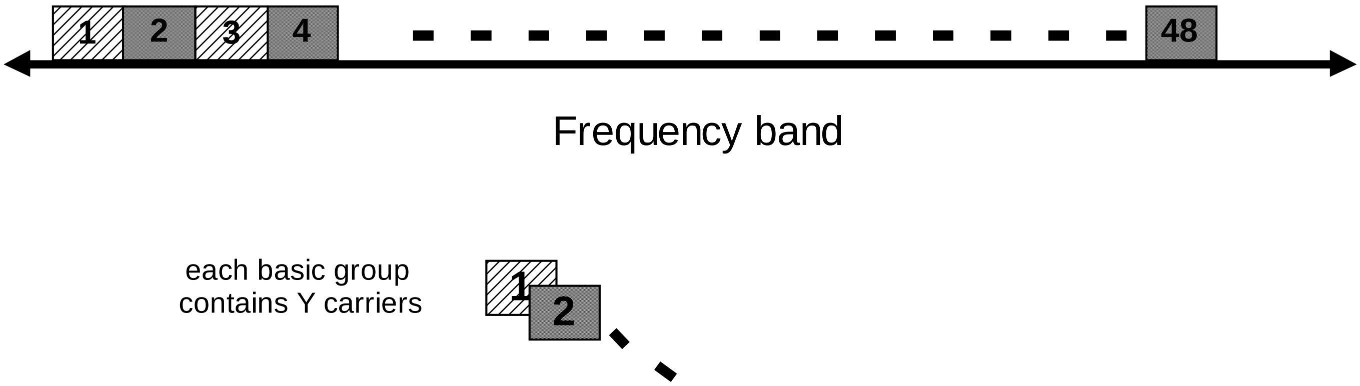



Figure 14: division of all carriers into basic groups for the DS OFDM symbol using FFT size of 1k

#### 3.2.2.2.1 1K mode characterization

The parameters characterizing the 1K mode on the DS are as follow:
- Number of FFT points = 1024 (1K)
- Overall Usable Carriers = 849
- Guard Bands = 87 carriers on both sides of the spectrum
- Number of Sub-Channels = 16

The Continues location pilots are located at the next indices **– TO COMPLETE**

The allocation of carriers to Sub-Channels are done by special permutation code which is based upon the following procedure:

1. The basic series of 16 numbers is 6, 14, 2, 3, 10, 8, 11, 15, 9, 1, 13, 12, 5, 7, 4, 0
2. In order to get 16 different permutation the series is rotated to the left (from no rotation at all up to 15 rotations), for the first permutation we get the following series: 14, 2, 3, 10, 8, 11, 15, 9, 1, 13, 12, 5, 7, 4, 0, 6
3. To get a 48 length series we concatenate the permutated series 5 times (to get a 64 length series) and take the first 48 numbers only, the concatenation depends on the cell Id (which characterizes the working cell and can range from 0 to 15) , the concatenated series is achieved by the next formula:

   (PermutatedSeries + CellId) mod 16; (PermutatedSeries + 2*CellId) mod 16; (PermutatedSeries + 3*CellId) mod 16; (PermutatedSeries + 4*CellId) mod 16;

   for example when using permutation 1 with CellId=2 we get the next concatenated series:

   0 ,4 ,5 , 12, 10, 13, 1, 11, 3, 15, 14, 7, 9, 6, 2, 8, 2, 6, 7, 14, 12, 15, 3, 13, 5, 1, 0, 9, 11, 8, 4, 10, 4, 8, 9, 0, 14, 1, 5, 15, 7, 3, 2, 11, 13, 10, 6, 12, 6, 10, 11, 2, 0, 3, 7, 1, 9, 5, 4, 13, 15, 12, 8, 14

   therefore the 48 length series is:
   0 ,4 ,5 , 12, 10, 13, 1, 11, 3, 15, 14, 7, 9, 6, 2, 8, 2, 6, 7, 14, 12, 15, 3, 13, 5, 1, 0, 9, 11, 8, 4, 10, 4, 8, 9, 0, 14, 1, 5, 15, 7, 3, 2, 11, 13, 10, 6, 12
4. The last step achieves the carrier numbers allocated for the specific Sub-Channel with the current Cell Id. Using the next formula we achieve the 48 carriers of the current permutation in the cell:

   $$Carrier\# = 16 * n + Index(n)$$

   where:
   $Carrier\#$ - denotes the carrier number for this Sub-Channel
   $n$ - Indices 0..47
   $Index(n)$ - denotes the number at index $n$ of the 48 length series

## 3.3 Up-Stream

### 3.3.1 US Sub-Channel description

The next section gives a description of the structure of a Sub-Channel. A Sub-Channel is made up of 48 usable carriers and 5 pilot carriers; this structure is used for all FFT modes, on the US transmission. The DS transmission for these modes is also made of Sub-Channel transmissions, but the Sub-Channel is made up of 48 data carriers only, while pilot carriers are spread all over the OFDMA symbol, to be used for channel estimation. The US sub-Channel structure is shown in Figure 15.

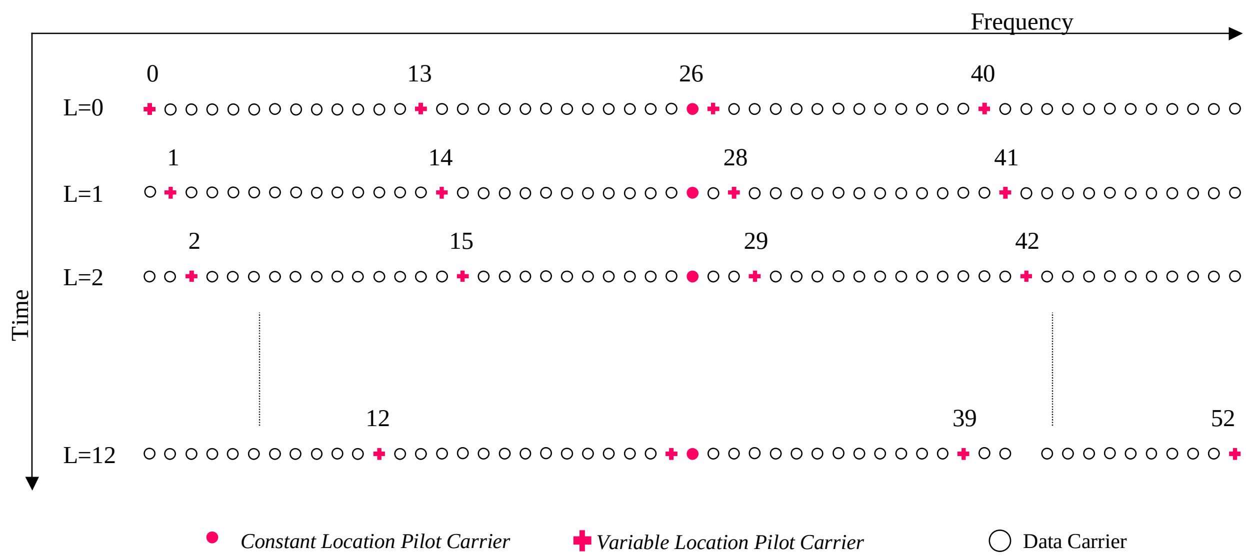



Figure 15: Allocation of data and pilot carriers for a US Sub-Channel

The US data symbol structure is comprised of data carriers and pilot carriers. The data symbols are produced with a modulo 13 repetition (L denotes the modulo 13 index of the symbol with indices 0..12), the location of the variable location pilots are shifted for every symbol produced, the first symbol (L=0) is produced after the all-pilot symbols (preamble). For L=0 the variable location pilots are positioned at indices: 0,13, 27,40 for other L these location vary by addition of L to those position, for example L=5 variable pilots location are: 5,18, 32, 45. the US Sub-Channel is also comprised of a constant pilot at the index 26. all other carriers (48) are data carriers, their location changes for every L.

### 3.3.2 US Allocation

The basic allocation for a user US transmission is made up of Sub-Channels, a basic user allocation is made up of one Sub-Channel over duration of 4 OFDMA symbols. The first is a preamble and remaining are used for data transmission, adding more data symbols or Sub-Channels increases the amount of data sent by the user, while preamble is repeated every X data Sub-Channels (in the time domain), this allocation is presented in Figure 16:

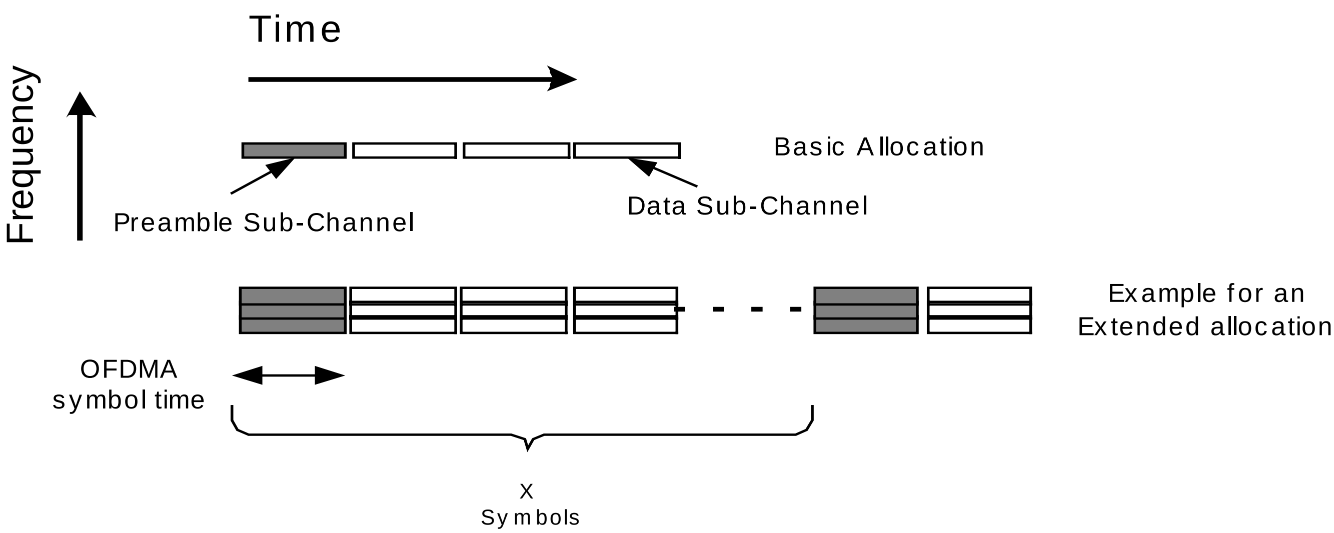



Figure 16: US User allocation

### 3.3.3 Framing Structure

The framing structure used for the US includes the transmission of a possible symbol for Jamming monitoring, an allocation for Ranging and an allocation for data transmission. The MAC sets the length of the US framing, and the US mapping.

#### 3.3.3.1.1 For FFT size 64 and 256

The basic allocation for a user US transmission is made up of a preamble and 3 OFDM data symbols, adding more data symbols prolongs the transmission, while preamble is repeated every X data symbols transmission. Therefore the US mapping is illustrated in Figure 17:

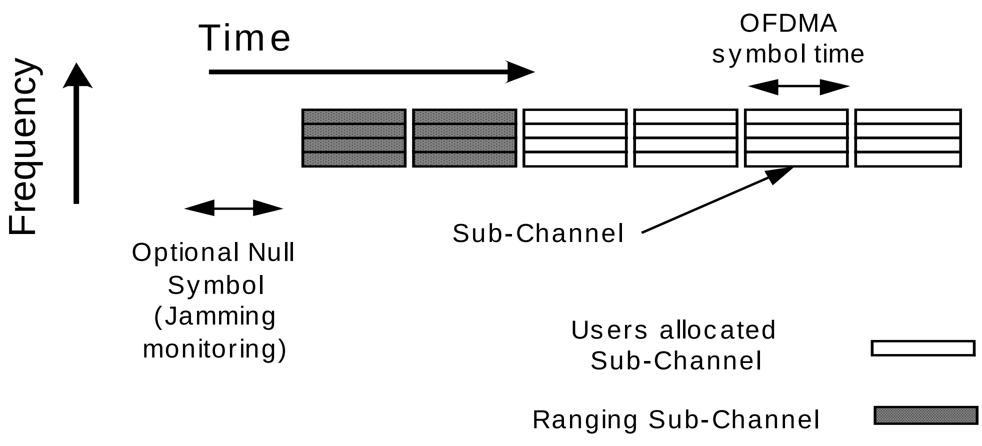



Figure 17: US framing for FFT sizes 64 and 256

#### 3.3.3.1.2 For FFT size of 1k

The framing for these modes involve the allocation of ranging Sub-Channels within the OFDMA symbols, while the rest of the Sub-channels are used for users transmission, the US mapping is illustrated in Figure 18:

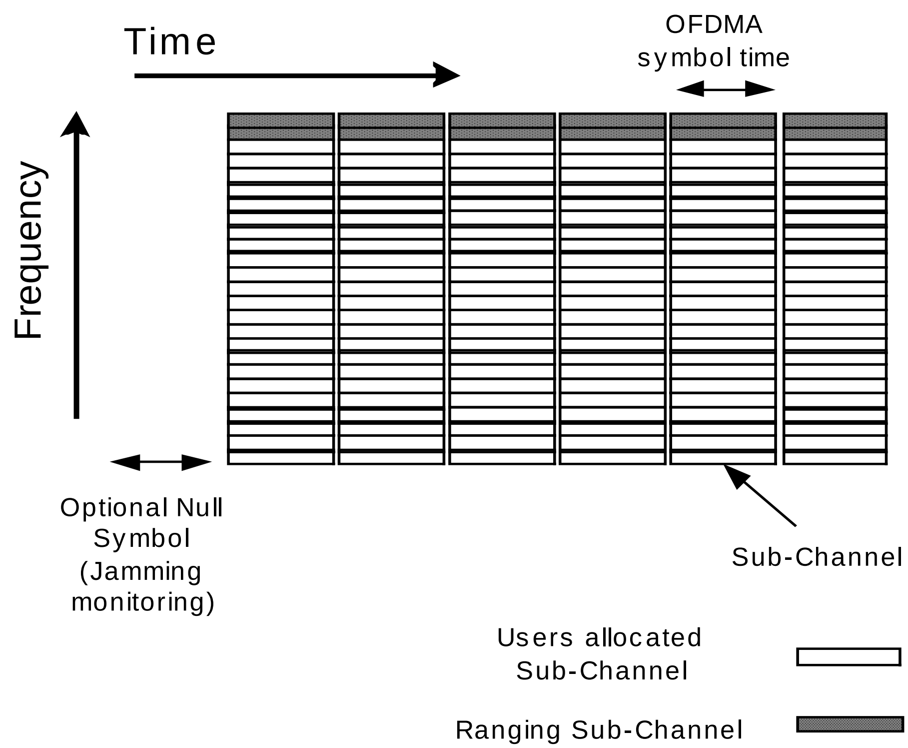
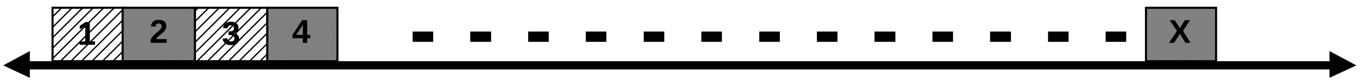



Figure 18: US framing for FFT size of 1k

### 3.3.4 Symbol Structure

The symbol structure for is made up of Sub-Channels, by their basic structure described in section 3.3.1. There are several methods splitting the whole US OFDMA symbol into Sub-Channels, the first two methods are performed by first dividing the used carriers into basic groups (not including the DC carrier, which is not used), each containing a certain amount of carriers (Figure 19 illustrates this principle)




each basic group contains Y carriers

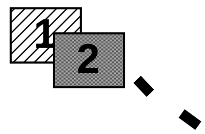



Figure 19: division of all carriers into basic groups for the US OFDMA symbol

Then the following methods exist:

1. The number of basic groups is 53 and they are allocated Y adjunct carriers, from the first usable carrier to the last, and then special permutations are used to extract the Sub-Channels.
2. Defining each basic group as a Sub-Channel, which implicate that the number of carriers Y=53 and that the carriers within the Sub-Channel are allocated adjunct. The carrier indices for each Sub-Channel is achieved using the next formula:

$$Carrier\# = 53 * n + I$$

where:

$Carrier\#$ - denotes the carrier number for Sub-Channel $n$

$n$ - Indices from 0 to the amount of Sub-Channels minus one

$I$ - Indices 0..52

The last method for defining the Sub-Channels involves programming by MAC message the carrier numbers for each Sub-Channel.

#### 3.3.4.1.1 64 mode characterization

The parameters characterizing the 64 mode on the DS are as follow:

- Number of FFT points = 64
- Overall Usable Carriers = 53
- Guard Bands = 6, 5 carriers on left and right sides of the spectrum respectively
- Number of Sub-Channels = 1

$\{PermutationBase\}$ = TBD

#### 3.3.4.1.2 256 mode characterization

The parameters characterizing the 256 mode on the DS are as follow:

- Number of FFT points = 256
- Overall Usable Carriers = 213
- Guard Bands = 24, 23 carriers on left and right sides of the spectrum respectively
- Number of Sub-Channels = 4

The parameters characterizing the Sub-Channels allocation:
Y – Number of carriers in each basic group = 4

$\{PermutationBase\}$ = TBD

#### 3.3.4.1.3 1k mode US characterization

The parameters characterizing the 1K mode are as follow:

- Number of FFT points = 1024 (1K)
- Overall Usable Carriers = 849
- Guard Bands = 88, 87 carriers on right an left side of the spectrum
- Number of Sub-Channels = 16

The parameters characterizing the Sub-Channels allocation:

- Y – Number of carriers in each basic group = 16

The allocation of carriers to Sub-Channels are done by special permutation code which is based upon the following procedure:

1. The basic series of 16 numbers is 6, 14, 2, 3, 10, 8, 11, 15, 9, 1, 13, 12, 5, 7, 4, 0
2. In order to get 16 different permutation the series is rotated to the left (from no rotation at all up to 15 rotations), for the first permutation we get the following series: 14, 2, 3, 10, 8, 11, 15, 9, 1, 13, 12, 5, 7, 4, 0, 6
3. To get a 53 length series we concatenate the permutated series 5 times (to get a 64 length series) and take the first 53 numbers only, the concatenation depends on the cell Id (which characterizes the working cell and can range from 0 to 15), the concatenated series is achieved by the next formula:

   (PermutatedSeries + CellId) mod 16; (PermutatedSeries + 2*CellId) mod 16; (PermutatedSeries + 3*CellId) mod 16; (PermutatedSeries + 4*CellId) mod 16;

   for example when using permutation 1 with CellId=2 we get the next concatenated series:

   0 ,4 ,5 , 12, 10, 13, 1, 11, 3, 15, 14, 7, 9, 6, 2, 8, 2, 6, 7, 14, 12, 15, 3, 13, 5, 1, 0, 9, 11, 8, 4, 10, 4, 8, 9, 0, 14, 1, 5, 15, 7, 3, 2, 11, 13, 10, 6, 12, 6, 10, 11, 2, 0, 3, 7, 1, 9, 5, 4, 13, 15, 12, 8, 14

   therefore the 53 length series is:
   0 ,4 ,5 , 12, 10, 13, 1, 11, 3, 15, 14, 7, 9, 6, 2, 8, 2, 6, 7, 14, 12, 15, 3, 13, 5, 1, 0, 9, 11, 8, 4, 10, 4, 8, 9, 0, 14, 1, 5, 15, 7, 3, 2, 11, 13, 10, 6, 12, 6, 10, 11, 2, 0, 3

1. The last step achieves the carrier indices allocated for the specific Sub-Channel with the current Cell Id. Using the next formula we achieve the 53 carriers of the current permutation in the cell:

$$Carrier\# = 16 * n + Index(n)$$

where:
$Carrier\#$ - denotes the carrier number for Sub-Channel $n$
$n$ - Index 0..52

$Index(n)$ - denotes the number at index $n$ of the 53 length series

# 4 Data Randomization

Data randomization is performed on data transmitted on the DS and US. The randomization is performed on each allocation (DS or US), which means that for each allocation of a data block (Sub-Channels on the frequency domain and OFDM symbols on the time domain) the randomizer shall be used independently. If the amount of data to transmit those not fit exactly the amount of data allocated, padding of FFx ('1' only) shall be added to the end of the transmission block, up to the amount of data allocated.

The shift-register of the randomizer shall be initialized for each new allocation or for every 1250 bytes passed through (if the allocation is larger then 1250 bytes).

The randomizer shall be initialized with the binary value: 100101010000000 (45200 in octal). Each data byte to be transmitted shall enter sequentially into the randomizer, MSB first.
The Pseudo Random Binary Sequence (PRBS) generator shall be $1+X^{14}+X^{15}$.

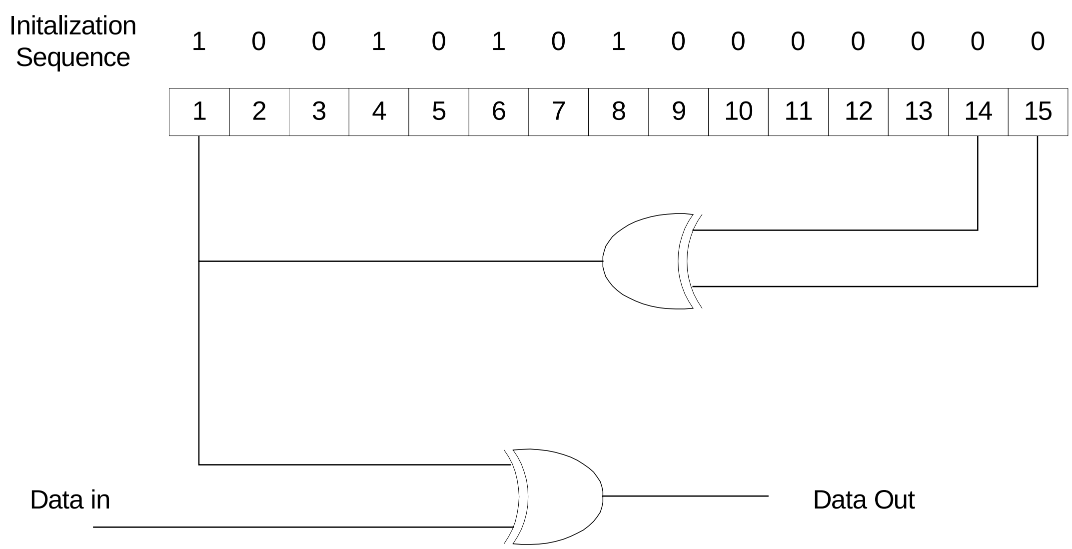



Figure 20: PRBS for data randomization

The bit issued from the randomizer shall be applied to the encoder.

# 5 Coding

Several ECC codes are defined for the system for DS transmissions and US:

1. Concatenated RS(N,K) (derived from the systematic RS(255,239,8)) concatenated with a convolutional tail biting coding (k=7, G1=171, G2=133)
2. Block Turbo Codes

## *5.1 Concatenated Reed Solomon and Convolutional Coding*

First passing the data through the RS encoder and then passing the data in block format to a tail biting convolutional encoder perform the encoding.

### 5.1.1 Reed Solomon encoding

The Reed Solomon encoding process shall use the systematic RS(255,239,8), with the possibility to make a variable RS(N,K,T), where:

- N - overall bytes, after encoding
- K - data bytes before encoding
- T - data bytes that can be fixed

The following polynomials are used for the systematic code:

- Code generator polynomial: $g(x)=(x+\lambda^0)(x+\lambda^1)(x+\lambda^2)...(x+\lambda^{2T-1}), \lambda = 02_{hex}$
- Field Generator polynomial: $p(x)=x^8+x^4+x^3+x^2+1$

The basic frame size for encoding includes 188 bytes of data using the RS(202,188,8), the data after passing the randomizer shall be divided into groups of 188 bytes and a remainder (if any), the remainder shall be encoded with a shortened code.

#### 5.1.1.1 Shortening the RS encoding

The shortening of the code is used whenever the data to be sent is smaller the 188 bytes or if a reminder of the data exists (the data to be sent is not a multiplication of 188 bytes).
The shortening is performed by setting the T parameter and using zero padding in order to bring the block to be encoded to 188 bytes, and after encoding the zero padding is discarded. For example if T=4 and the remainder is 40 bytes, the padding of 144 bytes will be added and 4 erasures are used to shorten the protection bytes.

### 5.1.2 Convolutional coding

Data bits issued from the Reed Solomon encoder, described in clause 5.1.1, shall feed the convolutional encoder depicted in Figure 21.

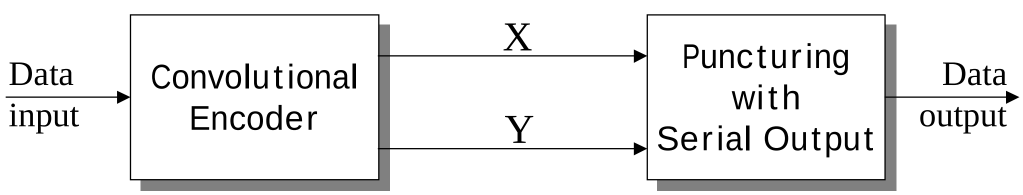



Figure 21: Convolutional encoder block diagram

The Convolutional encoder shall have a constraint length equal to k=7 and shall use the following mother codes:

$$G_1 = 171_{oct} \text{ For X}$$

$$G_1 = 133_{oct} \text{ For Y}$$

A basic convolutional encoding scheme, as depicted in Figure 22, shall be used.

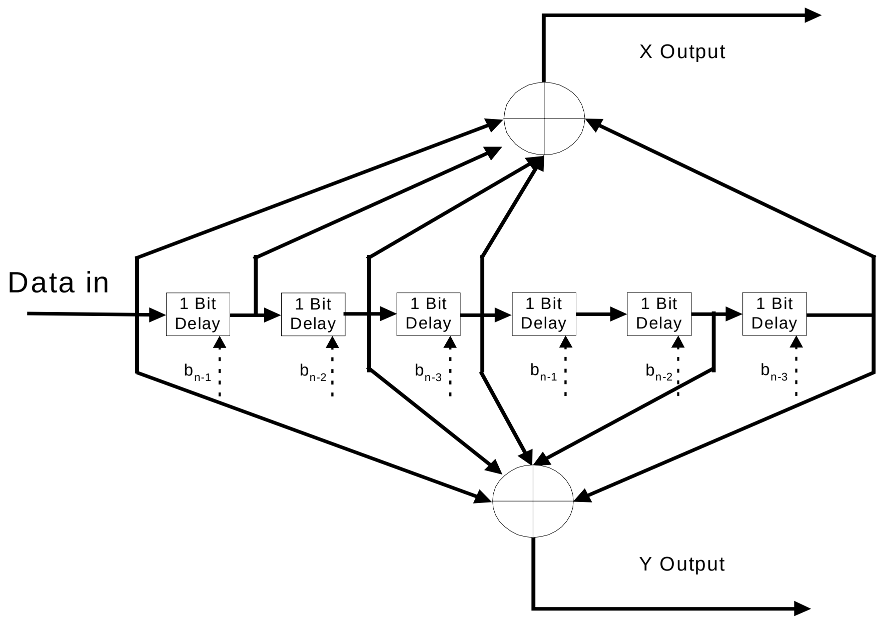



Figure 22: Convolutional encoder basic scheme

The puncturing pattern shall be as defined in Table 2.

| CC Code Rate | Puncturing Pattern | Transmitted Sequence (after parallel to serial conversion) |
|---|---|---|
| $1/2$ | X : 1<br>Y : 1 | $X_1Y_1$ |
| $2/3$ | X : 1 0<br>Y : 1 1 | $X_1Y_1Y_2$ |
| $3/4$ | X : 1 0 1<br>Y : 1 1 0 | $X_1Y_1Y_2X_3$ |

Table 2: Bit-Interleaver size as a function of the Modulation and Coding Rate

In order to achieve a tail biting convolutional encoding the memory of the convolutional encoder shall be initialized with the last data bits of the RS packet (the packet data bits are numbered $b_0..b_n$).

## 5.2 Block Turbo Coding

TBD This type of coding, based on the product of two or more simple component codes, is also called Turbo Product code, TPC. The decoding is based on the concept of Soft-in/Soft-out (SISO) iterative decoding (i.e, “Turbo decoding”). The component codes recommended for this proposal are binary extended Hamming codes or Parity check codes. The schemes supported follow the recommendation of the IEEE802.16.1 mode B. However, more flexibility in block size and code rates is enabled. The main benefits of using TPC mode are typically 2 dB better performance over the Concatenated RS, and shorter decoding delays. A detailed description of Turbo Product Codes is included as Appendix A. In this Section we present some particular turbo product codes that are perfectly matched for the proposed framing/modulation structure.

### 5.2.1 Block Turbo Constituent Codes

As mentioned in ANNEX A - Turbo Code Description, TPCs are constructed as a product of simple component codes. The complete constituent code set is defined in Table 3

| |
|---|
| (64, 57) Extended Hamming Code |
| (32,26) Extended Hamming Code |
| (16,11) Extended Hamming Code |
| (32,31) Parity Check Code |
| (16,15) Parity Check Code |
| (4,3) Parity Check Code |

Table 3: Constituent Block Turbo Code List

### 5.2.2 Overall Turbo Product Codes

The defined Turbo Product Codes are all multiples of 48 bits to facilitate integration into the framing structure. Table 4 lists the possible codes, rates and block size in bits. The block sizes are achieved by bit shortening as described in ANNEX A - Turbo Code Description.

| X-Code | Y-Code | Z-Code | Rate | Block Size |
|---|---|---|---|---|
| 56,49 | 55,48 | | 0.76 | 3072 |
| 25,19 | 25,19 | 5,4 | 0.46 | 3072 |
| 48,41 | 48,41 | | 0.72 | 2304 |
| 24,18 | 24,18 | 4,3 | 0.42 | 2304 |
| 64,57 | 21,20 | | 0.85 | 1296 |
| 41,34 | 32,26 | | 0.67 | 1296 |
| 64,57 | 12,11 | | 0.82 | 768 |
| 28,22 | 28,22 | | 0.62 | 768 |
| 54,47 | 9,8 | | 0.77 | 480 |
| 30,24 | 16,11 | | 0.55 | 480 |
| 32,26 | 6,5 | | 0.68 | 192 |
| 14,9 | 14,9 | | 0.41 | 192 |

Table 4: TPC Example Codes

The codes may be shortened using the method described in ANNEX A - Turbo Code Description. Shortening should only be performed in multiples of 48 bits.

When using Turbo Product Codes additional Bit Interleaver sizes are defined:

| X-Code | Y-Code | Z-Code | Bit Interleaver allocation |
|---|---|---|---|
| 56,49 | 55,48 | | 1024 |
| 25,19 | 25,19 | 5,4 | 1024 |
| 48,41 | 48,41 | | 768 |
| 24,18 | 24,18 | 4,3 | 768 |
| 64,57 | 21,20 | | 648 |
| 41,34 | 32,26 | | 648 |
| 64,57 | 12,11 | | 768 |
| 28,22 | 28,22 | | 768 |
| 54,47 | 9,8 | | 480 |
| 30,24 | 16,11 | | 480 |
| 32,26 | 6,5 | | 192 |
| 14,9 | 14,9 | | 192 |

Table 5: Optimal Bit Interleaver Sizes

Termination of allocations does not have to be the multiplications of the above bit interleaver sizes. Sizes in any multiple of 48, up to a multiple of 12 can be used to terminate the allocation.

# 6 Bit Interleaving

The data encoded is passed through a bit interleaver; the bit interleaver size is set to the size of 3 Sub-Channel allocation (the size depends on the modulation used). Termination of allocation does not have to be in multiplications of 3 Sub-Channels therefore if 1 or 2 Sub-Channel allocation are left at the end of the encoding, a shorter bit interleaver in the size of 1 Sub-Channel allocation is used (the size depends on the modulation used). Table 6 summarizes the bit-interleaver sizes as a function of the modulation and coding.

| Modulation | 3 Sub-Channel allocation | 1 Sub-Channel allocation |
|---|---|---|
| BPSK | 144 | 48 |
| QPSK | 288 | 96 |
| QAM16 | 576 | 192 |
| QAM64 | 864 | 288 |

Table 6: Bit-Interleaver size as a function of the Modulation

The interleaver scrambles the order of the input bits to produce the interleaved data, which is then fed to the mapper and the Sub-Channel allocation.

The PRBS generator depicted in Figure 23 is used to achieve the bit interleaver array, it is initialized with the binary value: 0001011010.
The PRBS generator produces an index value, which shall correspond to the new position of the input bit into the output interleaved data burst.
The interleaver shall use the following algorithm:

- The Interleaver indexes range from 1 to n (where n denotes the block size to be interleaved)
- For each input bit, the PRBS shall be rotated, the rotation produces a number, which is the value of the PRBS memory register.
- If the obtained number is bigger than n, it shall be discarded and the PRBS shall be rotated again. The rotation shall continue until an index between 1 to n is produced.
- The obtained index shall be used to address the position of the processed bit into the output interleaved data burst

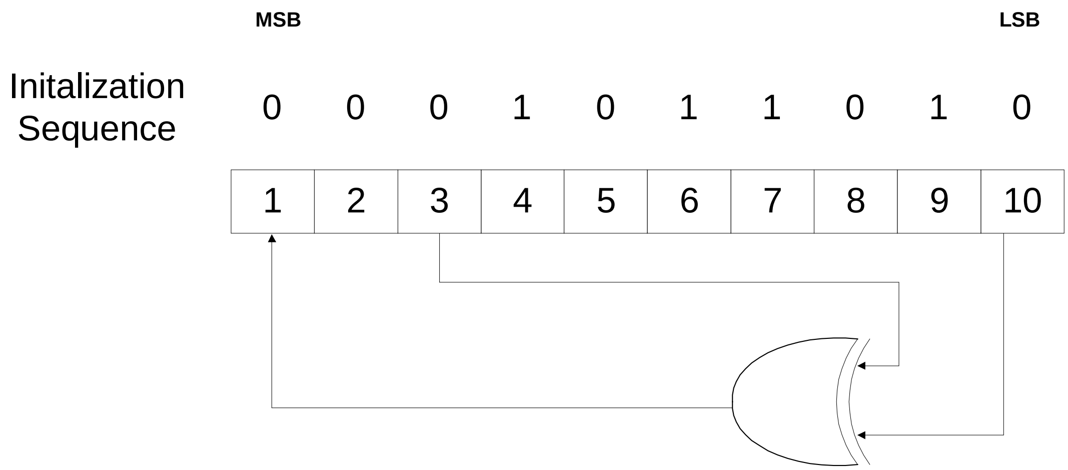



Figure 23: PRBS for Bit-Interleaver array

# 7 Ranging

## 7.1 Time and Power Ranging of the users

Time and Power ranging is performed by allocating several Sub-Channels to one Ranging Sub-Channel upon this Sub-Channels users are allowed to collide, each user randomly chooses a random code from a bank of codes. These codes are modulated by BPSK upon the contention Sub-Channel. The Base Station can then separate colliding codes and extract timing and power ranging information, in the process of user code detection the base station get the Channel Impulse Response (CIR) of the code, acquiring the base station vast information about the user channel and condition. The time and power ranging allows the system to compensate the far near user problems and the propagation delay caused by large cells. The Power ranging will handle a dynamic range of 45dB in the US (for the DS there should be a dynamic range of 30dB – controlled by Forward Automatic Power Control from the user). After time ranging the user should stabilize to a deviation of no more then 30% of the Guard Interval.

### 7.1.1 Using the Sub-Channels for ranging or fast bandwidth request purposes

The usage of the Sub-Channels for ranging or fast bandwidth request is done by the transmission of a Pseudo Noise (PN) code on the Sub-Channel allocated for ranging transmission. The code is always BPSK modulated and is produced by the PRBS described in Figure 24 (the PRBS polynomial generator shall be $1+X^4+X^7+X^{15}$):

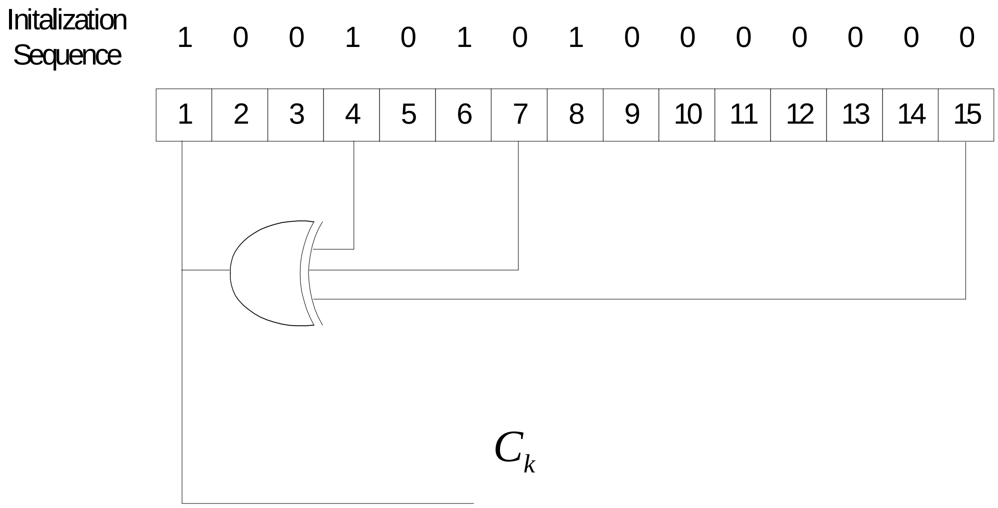



Figure 24: PRBS for ranging code production

Circulating through the PRBS (were each circulation produces one bit) produces the Ranging codes. The length of the ranging codes are multiples of 53 bits long, the codes produced are used for the next purposes:

- The first 16codes produced are for First Ranging, it shall be used by a new user entering the system.
- The next 16 codes produced are used for maintenance Ranging for users that are already entered the system.
- The last 16 codes produced are for users, already connected to the system, issuing bandwidth requests.

These 48 codes are denoted as Ranging Codes and are numbered 0..47.
The MAC sets the number of Sub-Channels allocated for Ranging, these ranging Sub-Channels could be used concatenates as orders by the MAC in order to achieve a desired length.

### 7.2 Frequency

The frequency accuracy of the Base-Station RF and Base-Band reference clocks shall be at least 2ppm. The user reference clock could be at a 20ppm accuracy, and the user should synchronize to the DS and extract his clock from it, after synchronization the RF frequency would be accurate to 1% of the carrier spacing.

### 7.3 Frame Timing

M slots shall define one frame. M shall coincide with the frame timing controlled by the MAC.

## 8 Constellation Mapping

The modulation used both for the US and DS data carrier is BPSK, QPSK, 16QAM and 64QAM. These modulations are used adaptively in the downlink and the uplink in order to achieve the maximum throughput for each link.

The modulation on the DS can be changed for each allocation, to best fit the modulation for a specific user/users. When using OFDMA the power of the modulated carrier can also vary by attenuation or busting of 4dB, this is used for the Forward APC.

For the up stream each user is allocated a modulation scheme, which is best suited for his needs.

The pilot carriers for the US and DS are mapped using a BPSK modulation.

## 8.1 Data Modulation

The data bits entering the mapper are after bit interleaving and they enter serially to the mapper, the mapping constellations are presented here after in Figure 25:

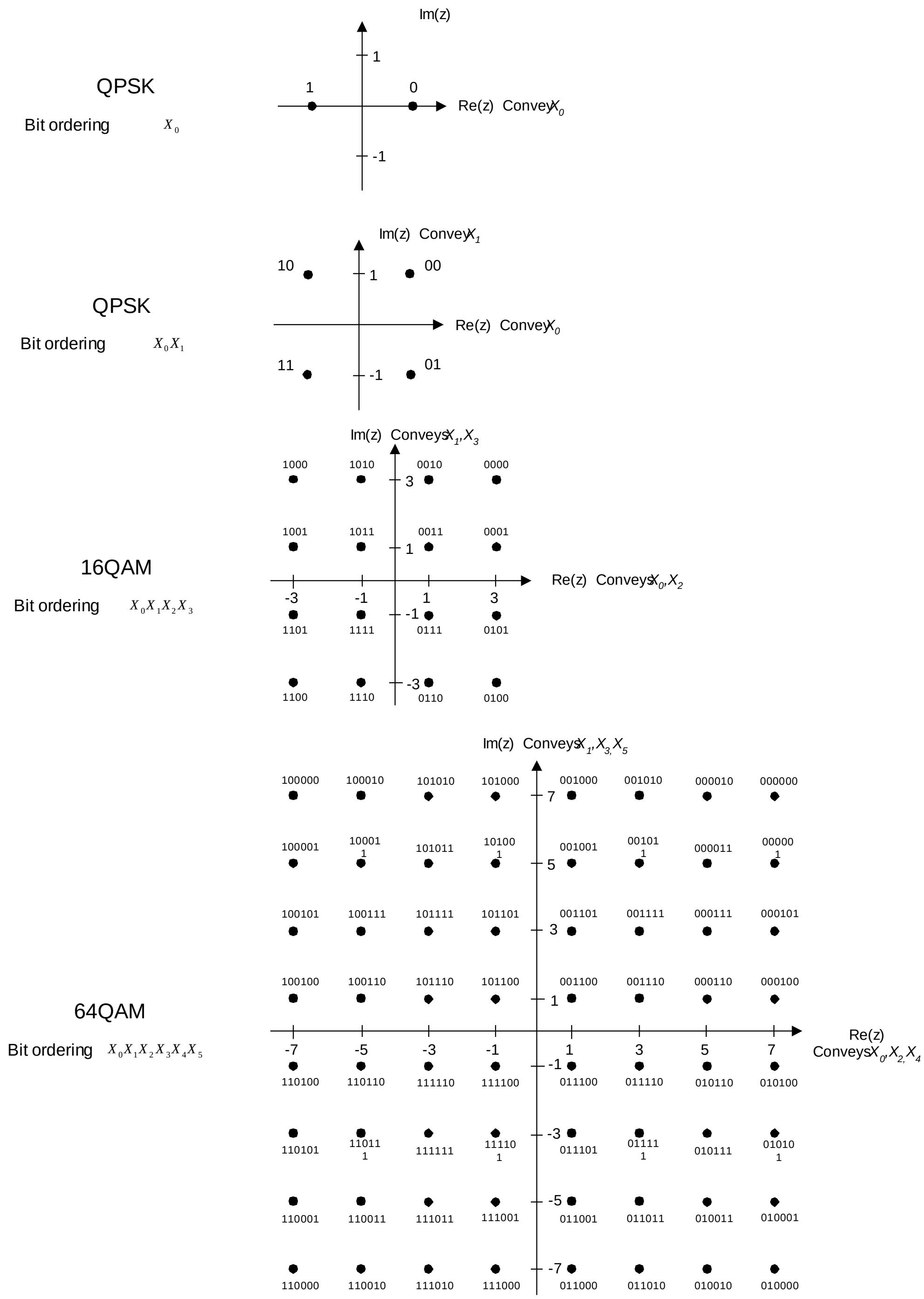



Figure 25: BPSK, QPSK, 16QAM and 64 QAM constellations

The complex number *z* shall be normalized by the value *c*, before mapping onto the carriers, by using the factor defined in the next table:

| Modulation scheme | Normalization Factor 3dB attenuation | Normalization Factor Reference 0dB | Normalization Factor 3dB Busting |
|---|---|---|---|
| BPSK | $c = z/2$ | $c = z$ | $c = 2 \cdot z$ |
| QPSK | $c = z/\sqrt{8}$ | $c = z/\sqrt{2}$ | $c = z\sqrt{2}$ |
| 16QAM | $c = z/\sqrt{40}$ | $c = z/\sqrt{10}$ | $c = z\sqrt{2}/\sqrt{5}$ |
| 64QAM | $c = z/\sqrt{164}$ | $c = z/\sqrt{42}$ | $c = z\sqrt{2}/\sqrt{21}$ |

The complex number $c$, resulting from the normalization process, shall be modulated onto the allocated data carriers. The data mapping shall be done by sequentially modulating these complex values onto the relevant carriers. The reference-normalizing factor is used for the US, and the DS defined for 0dB busting or attenuation. The normalizing factors used for attenuation and busting are for DS use only, this is defined in the DS parameters for a specific burst type and is used for Forward APC.

## 8.2 Pilot Modulation

Pilot carriers shall be inserted into each data burst in order to constitute the Symbol Structure (see clause 3.23.3) and they shall be modulated according to their carrier location within the OFDM symbol.
The Pseudo Random Binary Sequence (PRBS) generator depicted hereafter, shall be used to produce a sequence, $w_k$. The polynomial for the PRBS generator shall be $X^{11} + X^2 + 1$.

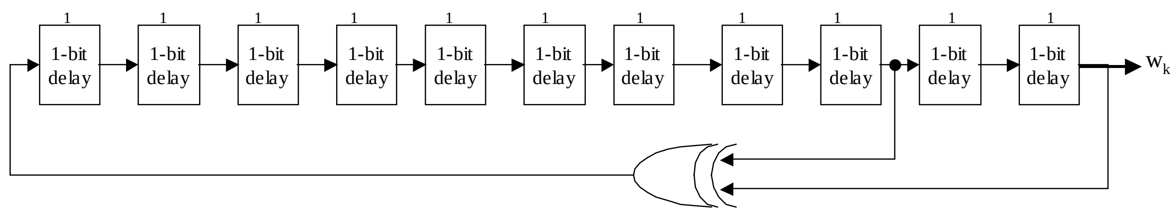



Figure 26: PRBS used for pilot modulation

The value of the pilot modulation, on carrier k, shall be derived from $w_k$.
When using data transmission on the DS the initialization vector of the PRBS will be: [11111111111]
When using data transmission on the US the initialization vector of the PRBS will be: [10101010101]

The PRBS shall be initialized so that its first output bit coincides with the first usable carrier. A new value shall be generated by the PRBS on every usable carrier. The DC carrier and the side-band carriers are not considered as usable carriers.
The pilots shall be sent with a boosting of 2.5 dB over the average energy of the data. The Pilot carriers shall be modulated according to the following formula:

$$\mathrm{Re}\{C_k\} = 4 / 3 \times 2\,(½ - w_k)$$
$$\mathrm{Im}\{C_k\} = 0$$

When Sub-Channels are used for pilots transmission only (preamble or midamble) the pilots shall not be boosted. The Pilot carriers shall be modulated according to the following formula:

$$Re\{C_k\} = 2\,(½ - w_k)$$

$$Im\{C_k\} = 0$$

## 8.3 Ranging Pilot Modulation

When using the ranging Sub-Channels the user shell modulate the pilots according to the following formula:

$$Re\{C_k\} = (½ - w_k) / 6$$

$$Im\{C_k\} = 0$$

$C_k$ is derived in clause 7.1.1.

# 9 System Throughput

We give an example for the system throughput on a 16MHz channel bandwidth using the 2k mode. The following table gives the Net data rates (in Mbit/s) for the system (assuming all Sub-Channels use the same modulation and coding rates):

| Modulation | Bits per sub-carrier | code rate | Net bit rate (Mbps) for different Guard intervals | | | |
|---|---|---|---|---|---|---|
| | | | 1/4 | 1/8 | 1/16 | 1/32 |
| BPSK | 1 | 1/2 | 6.85 | 7.62 | 8.07 | 8.31 |
| | 1 | 2/3 | 9.14 | 10.16 | 10.76 | 11.08 |
| | 1 | 3/4 | 10.29 | 11.43 | 12.10 | 12.47 |
| QPSK | 2 | 1/2 | 13.71 | 15.24 | 16.13 | 16.62 |
| | 2 | 2/3 | 18.28 | 20.32 | 21.51 | 22.16 |
| | 2 | 3/4 | 20.57 | 22.86 | 24.20 | 24.94 |
| 16-QAM | 4 | 1/2 | 27.42 | 30.48 | 32.27 | 33.27 |
| | 4 | 2/3 | 36.57 | 40.63 | 43.03 | 44.33 |
| | 4 | 3/4 | 41.14 | 45.71 | 48.40 | 49.87 |
| 64-QAM | 6 | 1/2 | 41.14 | 45.71 | 48.40 | 49.87 |
| | 6 | 2/3 | 54.86 | 60.95 | 64.54 | 66.49 |
| | 6 | 3/4 | 61.71 | 68.57 | 71.61 | 74.81 |

Table 7: System throughput using a 16MHz bandwidth and a 1k FFT

The allocated bandwidth for the upstream and the down stream can be different in order to satisfy different scenarios or demands.

In order to compute bit rates for other channel bandwidth a good approximation will be to use this table as reference and multiplying it by the factor of: $\frac{NewBandwidth(MHz)}{20(MHz)}$

Where the $NewBandwidth(MHz)$ parameter should be in MHz.

# 10 Co-Existence

## *10.1 Dynamic Frequency Selection (DFS)*

In order to gain the most from DFS we recommend that a base station shall use several Base Station elements each with it's own RF head (using all together several frequencies). The Base Stations can then scan several frequencies at once (or one at a time, for one head only), this will allow choosing the best and undisturbed frequency/frequencies around, and then choose the best frequency available. This technique when combined with the OFDMA capabilities can also use the dynamic transition of users between frequencies for load balance and excellent reuse factor.

## *10.2 Operation in presence of interference*

The interference in the MAN environment can be categorized into:

- Narrow band jamming
- Partial band jamming
- Pulse jamming
- Other operating system jamming and coexistence with IEEE802.11a, HiperLAN2.

### 10.2.1 Narrow band jamming

Narrow band jamming can be treated by:

- Using time shaping on the symbol and then equalization (the more FFT points used the better the shape is)
- Using jamming detection and then a smart ECC, which can erase bad symbols.

In any case when using large FFT sizes (especially in OFDMA), jammers at the base station are more effectively suppressed (due to the FFT filtering) and destroys less carriers (in percentage sense) then for small FFT size.

### 10.2.2 Partial band jamming

Detecting bad symbol can treat partial band jamming, which allow the usage of smart ECC, which can erase bad symbols. The OFDMA (2k mode) has a 15dB “processing gain” against wide band jammers or other 802.11a, HiperLAN2 interferers.

### 10.2.3 Pulse jamming

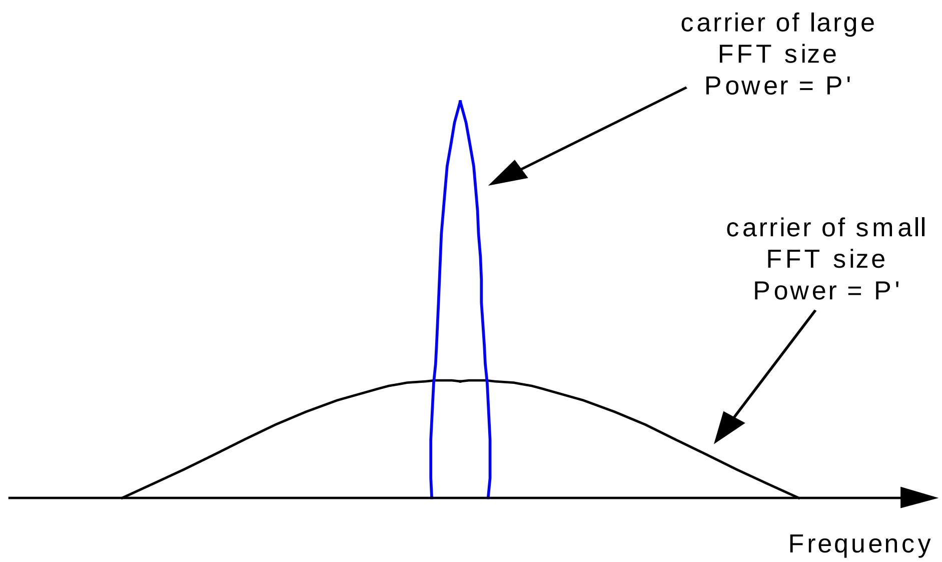
Short time interference can be sold by time interleaving the data. The usage of the Sub-Channel notion enables time interleaving of the Sub-Channel over time, the small packet length enables easy time interleaving and better statistical multiplexing.

### 10.2.4 Other operating system jamming and coexistence with IEEE802.11a, HiperLAN2

Best coexistence between the TG4 PHY and the IEEE802.1a, HiperLAN2 will occur when using large FFT sizes for TG4. When using large FFT size the carrier bandwidth is about 10KHz (compared to more then 300KHz for 64 FFT size). This bandwidth difference gains us a “processing gain” of 15dB in total.

Furthermore the FFT which is a filter help decreasing the all around interference by at least 13dB, when considering all the above in a scenario where both system are to work with the same power emittion the TG4 has a 28dB advantage in jamming rejection of the IEEE802.1a, HiperLAN2 signal. When using smaller FFT sizes the advantage decrease.

The following figure illustrates this scheme:




The figure illustrates the advantage when performing the large FFT over the small size FFT, where:

- for the large size FFT we get $\frac{S}{N} = OFDMA \big/ 802.11a - Jammer = 15dB$
- for the small size FFT we get $\frac{S}{N} = 802.11a \big/ OFDMA - Jammer = 0dB$

### 10.2.5 Other jamming rejection or Coexistence tools

There are more ways to have two systems work together without interfering one another:

- The usage of directive antennas
- The usage of adaptive array and null steering

Both are antenna-based techniques, which can remove or help prevent interferences.

# 11 Additional Possible Features

## *11.1 Adaptive Arrays*

Employing adaptive antenna arrays can increase the spectral efficiency linearly with the number of antenna elements. This is achieved by steering beams to multiple users simultaneously so as to realize an inter-cell frequency reuse of one and an in-cell reuse factor proportional to the number of antenna elements. An additional benefit is the gain in signal strength (increased SNR) realized by coherently combining multiple signals, and the ability to direct this gain to particular users. This is in contrast to sectored antenna approaches where most users are not in the direction of maximum antenna gain. Another benefit is the reduction in interference (increased signal to interference plus noise ratio, SINR) achieved by steering nulls in the direction of co-channel interferers.

The benefits of adaptive arrays can be realized for both the upstream and downstream signals using retro directive beam forming concepts in TDD systems, and to some extent in FDD systems using channel estimation concepts. These techniques do not require multiple antennas at the SS, although further benefits can be achieved by doing this.

Further benefits can be realized by combining adaptive antenna arrays with frequency spreading. These techniques are based on Stacked Carrier Spread Spectrum implementations.

The framing methods outlined in previous sections addressing adaptive arrays are designed to exploit these advantages.

Adaptive array could be designed to accommodate Narrow Band or Broad Band systems. Narrow band Systems could be achieved by defining the Sub-Channel carriers to be adjunct. The system inherently supports Broad Band channels, by using any other symbol structure (including the one were carriers of a sub-Channel are allocated adjunct).

When using Broad Band allocations in a Broad Band channel (up to 28MHz) there are several methods used to design adaptive arrays which are well known [23], this methods could comprise the use of matched receivers (amplitude and phase all over the band). Another method could comprise the use of non-matched receivers were processing could be done in the Base Band [ (by first sending internal testing signals and tuning the arrays in the Base Band, easily implemented for OFDM modulation, which is a frequency domain processing).

## *11.2 Transmit diversity Alamouti's Space-Time Coding*

Alamouti's scheme [25] is used on the downlink to provide (Space) transmit diversity of 2$^{nd}$ order.
*(This is an optionnal mode with little overhead and good benefits !)*
There are two transmit antennas on the BTS side and one reception antenna on the CPE side. This scheme requires Multiple Input Single Output -MISO- channel estimation. Decoding is very similar to maximum ratio combining.
Figure 27 shows Alamouti scheme insertion into the OFDM chain. Each Tx antenna has its own OFDM chain, but they have the same Local Oscillator for synchronization purposes.

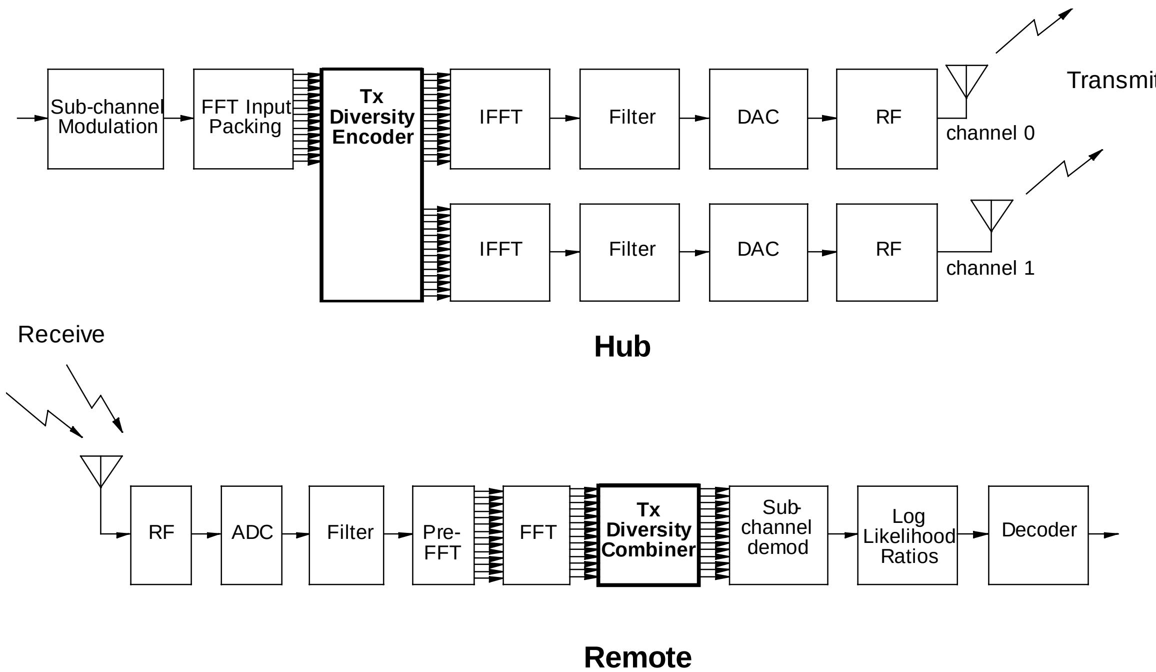



Figure 27: Illustration of the Alamouti STC

Both antennas transmit in the same time 2 different OFDM data symbols. Transmission is performed twice so as to decode and get 2nd order diversity. Time domain (Space-Time) or Frequency domain (Space-Frequency) can be used for this repetition.

MISO channel estimation and synchronization -
Both antennas transmit in the same time, and they share the same Local Oscillator. Thus, received signal has exactly the same auto-correlation properties as in the 1 Tx mode. Time and frequency coarse and fine estimation can so be performed in the same way as in the 1 Tx mode.
The scheme requires MISO channel estimation, which is allowed by splitting some preambles and pilots between the 2 Tx antennas.

## 11.3 STC for FFT sizes 64 and 256

Long preamble is transmitted once, either by one or both antennas. It is used for coarse synchronization.
Short preamble is transmitted once, antenna 0 using even frequencies, and antenna 1 odd frequencies. This allows fine synchronization and MISO channel estimation. Each channel (0 & 1) is interpolated with very little loss according to channel model.
Figure needed TBD.
Another option for short preamble is to transmit it twice alternatively from antenna 0 then antenna 1 . This yields to a preamble overhead, but with better fine synchronization.

Pilots tones are used to estimate phase noise. There are transmitted alternatively (on a symbol basis) from one antenna or the other. Since both antennas have the same LO, there is no penalty on phase noise estimation.

## 11.4 STC for FFT size 1k

Pilots tones are shared between the two antennas in time.
Again, synchronization, including phase noise estimation, is performed in the same way as with one Tx antenna.
The estimation of the two channels is unchanged, but interpolation is more used (in the time domain).

## 11.5 Alamouti STC Encoding

s* denotes complex conjugate of s.
(Scheme explanation) The basic scheme [25] transmits 2 complex symbols $s_0$ and $s_1$, using twice a MISO channel (two Tx, one Rx) with channel values $h_0$ (for antenna 0) and $h_1$ (for antenna 1).
first channel use: Antenna0 transmits $s_0$ , antenna1 transmits $s_1$.
Second channel use: Antenna0 transmits $-s_1^*$ , antenna1 transmits $s_0^*$ .
Receiver gets $r_0$ (first channel use) and $r_1$(second channel use) and computes $s_0$ and $s_1$ estimates:
$s_0 = h_0^* r_0 + h_1 r_1^*$.
$S_1 = h_1^* r_0 - h_0 r_1^*$.
These estimates benefit from 2nd order diversity as in the 1Tx-2Rx Maximum Ratio Combining scheme.

OFDM/OFDMA symbols are taken by pairs (equivalently, 2 Tx symbol duration is twice 1 Tx symbol duration, with twice more data in a symbol.)
In the transmission frame , variable location pilots are kept identical for two symbols, that means that the modulo L of the transmission same the same for the duration of two symbols.
Alamouti's scheme is applied independently on each carrier, in respect to pilot tones positions.

Next figure shows Alamouti's scheme for OFDMA. Note that for OFDM, the scheme is exactly the same except that a pilot symbol is inserted before the data symbols. Also note that since pilot positions do not change from even to odd symbols, and pilots modulation is real, conjugation (and inversion) can be applied to a whole symbol (possibly in the time domain)

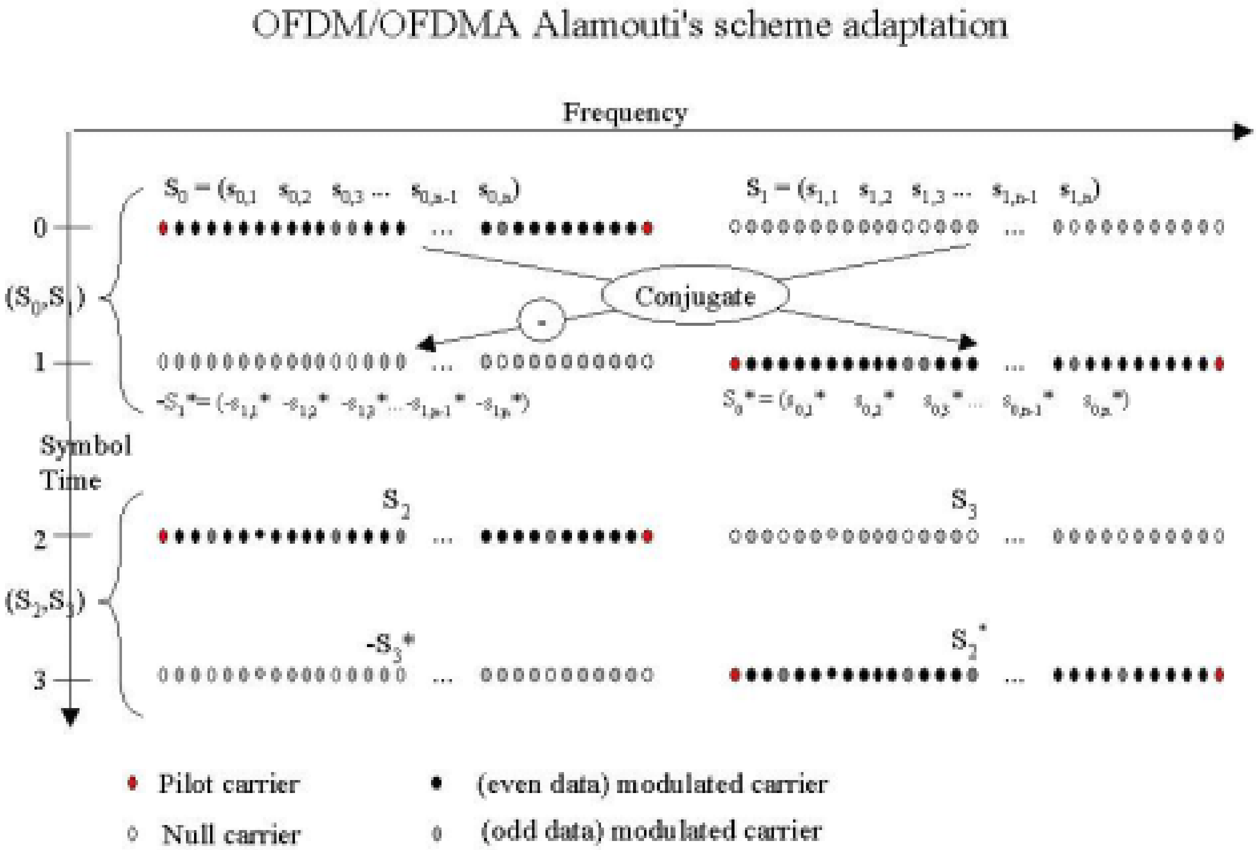


Figure 28: Using Alamouti’s Scheme with OFDM/OFDMA

### *11.6 Alamouti STC Decoding*

The receiver waits for 2 symbols, and combines them on a carrier basis according to the formula in section 11.5.

## 12 Intellectual Property

Intellectual Property owned by companies participating in the writing of this document, might be required to implement the proposed PHY specification. The authors are not aware of any conditions under which the companies would be unwilling to license Intellectual Property as outlined by the IEEE-SA Standards Board Bylaws, if the proposed specification will be adopted.

## 13 ANNEX A - Turbo Code Description

The Block Turbo Code is a Turbo decoded Product Code (TPC). The idea of this coding scheme is to use well-known product codes in a matrix form for two-dimensional coding, or in a cubical form for three dimensions.
The matrix form of the two-dimensional code is depicted in Figure 29. The $k_x$ information bits in the rows are encoded into $n_x$ bits, by using a binary block ($n_x$, $k_x$) code. The binary block codes employed are based on extended Hamming codes.
The redundancy of the code is $r_x = n_x - k_x$ and $d_x$ is the Hamming distance. After encoding the rows, the columns are encoded using another block code ($n_y$, $k_y$), where the check bits of the first code are also encoded. The

overall block size of such a product code is $n = n_x \times n_y$, the total number of information bits $k = k_x \times k_y$ and the code rate is $R = R_x \times R_y$, where $R_i = k_i/n_i$, i=x, y. The Hamming distance of the product code is $d = d_x \times d_y$.

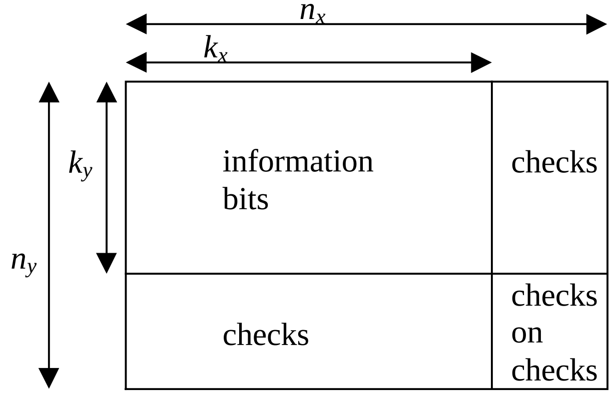


Figure 29: Two-dimensional product code matrix

## 13.1 Encoding of a Turbo Product Code

The encoder for TPCs has near zero latency, and is constructed of linear feedback shift registers (LFSRs), storage elements, and control logic. Encoding of a product code requires that each bit be encoded by 2 or 3 codes.

The constituent codes of TPCs are extended Hamming or parity only codes. Table 8A gives the generator polynomials of the Hamming codes used in TPCs. For extended Hamming codes, an overall even parity check bit is added at the end of each codeword.

| n | k | Generator Polynomial |
|---|---|---|
| 7 | 4 | $x^3 + x + 1$ |
| 15 | 11 | $x^4 + x + 1$ |
| 31 | 26 | $x^5 + x^2 + 1$ |
| 63 | 57 | $x^6 + x + 1$ |
| 127 | 120 | $x^7 + x^3 + 1$ |
| 255 | 247 | $x^8 + x + 1$ |

Table 8: Generators Polynomials of Hamming Codes

In order to encode the product code, each data bit is input both into a row encoder and a column encoder. Only one row encoder is necessary for the entire block, since data is input in row order. However, each column of the array is encoded with a separate encoder. Each column encoder is clocked for only one bit of the row, thus a more efficient method of column encoding is to store the column encoder states in a $k_x \times (n_y-k_y)$ storage memory. A single encoder can then be used for all columns of the array. With each bit input, the appropriate column encoder state is read from the memory, clocked, and written back to the memory.

The encoding process will be demonstrated with an example.

## 13.2 Example of a 2-Dimesional Product Code

Assume a two-dimensional (8,4) × (8,4) extended Hamming Product code is to be encoded. This block has 16 data bits, and 64 total encoded bits. Figure 30 shows the original 16 data bits denoted by $D_{yx}$. Of course the usual way is to have a serial stream of data of 16 bits and then label them as $D_{11}$, $D_{21}$, $D_{31}$, $D_{41}$, $D_{12}$,…, $D_{44}$.

| | | | |
|---|---|---|---|
| $D_{11}$ | $D_{21}$ | $D_{31}$ | $D_{41}$ |

| | | | |
|---|---|---|---|
| $D_{12}$ | $D_{22}$ | $D_{32}$ | $D_{42}$ |
| $D_{13}$ | $D_{23}$ | $D_{33}$ | $D_{43}$ |
| $D_{14}$ | $D_{24}$ | $D_{34}$ | $D_{44}$ |

Figure 30: Original Data for Encoding

The first four bits of the array are loaded into the row encoder in the order $D_{11}$, $D_{21}$, $D_{31}$, $D_{41}$. Each bit is also fed into a unique column encoder. Again, a single column encoder may be used, with the state of each column stored in a memory. After the fourth bit is input, the first row encoder error correction coding (ECC) bits are shifted out.

This process continues for all four rows of data. At this point, 32 bits have been output from the encoder, and the four column encoders are ready to shift out the column ECC bits. This data is also shifted out row-wise. This continues for the remaining 3 rows of the array. Figure 31 shows the final encoded block with the 48 generated ECC bits denoted by $E_{yx}$.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $D_{11}$ | $D_{21}$ | $D_{31}$ | $D_{41}$ | $E_{51}$ | $E_{61}$ | $E_{71}$ | $E_{81}$ |
| $D_{12}$ | $D_{22}$ | $D_{32}$ | $D_{42}$ | $E_{52}$ | $E_{62}$ | $E_{72}$ | $E_{82}$ |
| $D_{13}$ | $D_{23}$ | $D_{33}$ | $D_{43}$ | $E_{53}$ | $E_{63}$ | $E_{73}$ | $E_{83}$ |
| $D_{14}$ | $D_{24}$ | $D_{34}$ | $D_{44}$ | $E_{54}$ | $E_{64}$ | $E_{74}$ | $E_{84}$ |
| $E_{15}$ | $E_{25}$ | $E_{35}$ | $E_{45}$ | $E_{55}$ | $E_{65}$ | $E_{75}$ | $E_{85}$ |
| $E_{16}$ | $E_{26}$ | $E_{36}$ | $E_{46}$ | $E_{56}$ | $E_{66}$ | $E_{76}$ | $E_{86}$ |
| $E_{17}$ | $E_{27}$ | $E_{37}$ | $E_{47}$ | $E_{57}$ | $E_{67}$ | $E_{77}$ | $E_{87}$ |
| $E_{18}$ | $E_{28}$ | $E_{38}$ | $E_{48}$ | $E_{58}$ | $E_{68}$ | $E_{78}$ | $E_{88}$ |

Figure 31: Encoded Block

Transmission of the block over the channel may occur in a linear fashion, for example with all bits of the first row transmitted left to right followed by the second row, etc. This allows for the construction of a near zero latency encoder, since the data bits can be sent immediately over the channel, with the ECC bits inserted as necessary. For the (8,4)×(8,4) example, the output order for the 64 encoded bits would be

$D_{11}, D_{21}, D_{31}, D_{41}, E_{51}, E_{61}, E_{71}, E_{81}, D_{12}, D_{22}, \ldots, E_{88}$.

Alternatively, a block based interleaver may be inserted to further improve the performance of the system.

### 13.2.1 3-Dimensional TPC Encoding

For a three-dimensional TPC block, the element ordering for input/output for both encoding and decoding is usually in the order of rows, columns and then the z-axis. If we consider a serial stream of (i×j×k) data bits, labeled as:

$D_{1,1,1}, D_{2,1,1}, D_{3,1,1}, \ldots, D_{i,1,1}, D_{1,2,1}, D_{2,2,1}, \ldots, D_{,i,j,1}, D_{1,1,2}, \ldots, D_{i,j,k}$.

Note: this labeling is for convenience

Then the total size of the encoded block is ((i×j×k) + ECC bits), where there are p ECC bits for the x-axis, q ECC bits for the y-axis and r ECC bits for the z-axis, the bit order for input and output is:

$D_{1,1,1}, D_{2,1,1}, D_{3,1,1}, \ldots, D_{i,1,1}, \ldots, E_{p,1,1}, D_{1,2,1}, D_{2,2,1}, \ldots, E_{p,2,1}, \ldots, E_{p,q,1}, D_{1,1,2}, D_{2,1,2}, \ldots, E_{p,1,2}, \ldots, E_{p,q,2}, \ldots, E_{p,q,r}$

This is shown in Figure 32.

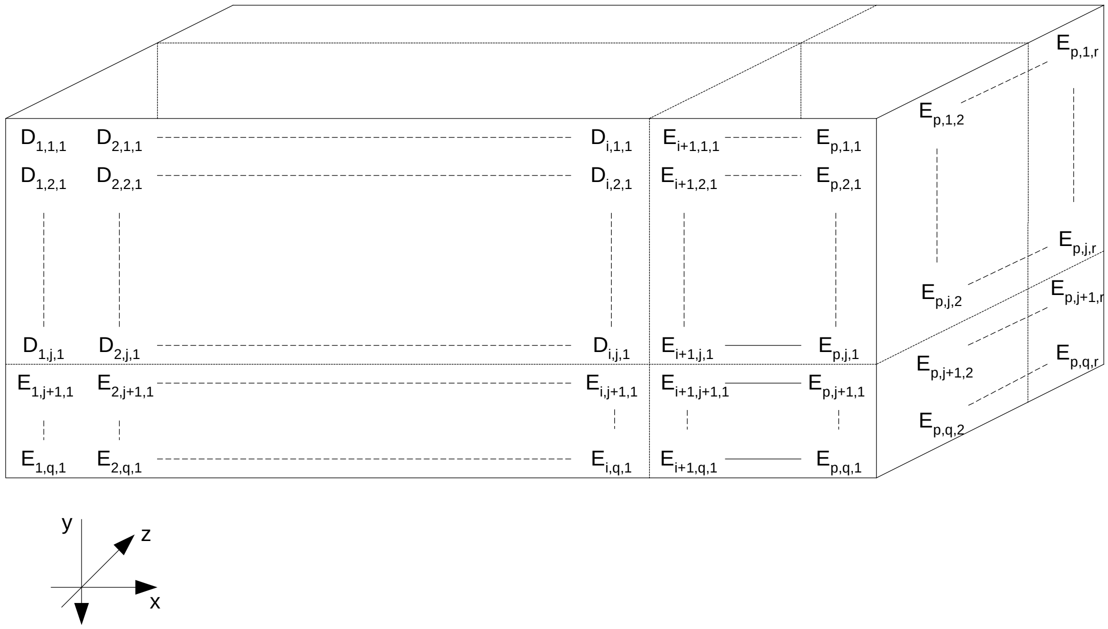



Figure 32: Structure of 3-Dimensional TPC

Notation:

- the codes defined for the rows (x-axis) are binary $(n_x,k_x)$ block codes
- the codes defined for the columns (y-axis) are binary $(n_y,k_y)$ block codes
- the codes defined for the z-dimension (z-axis) are binary $(n_z,k_z)$ block codes
- data bits are noted $D_{y,x,z}$ and parity bits are noted $E_{y,x,z}$

## 13.3 Shortened TPCs

To match packet sizes, a product code may be shortened by removing symbols from the array. In the two-dimensional case rows, columns or parts thereof can be removed until the appropriate size is reached. Unlike one-dimensional codes (such as Reed-Solomon codes), parity bits are removed as part of shortening process, helping to keep the code rate high.

There are two steps in the process of shortening of product codes. The first is to remove an entire row or column from a 2-dimensional code, or an entire X, Y, or Z plane from a 3-dimensional code. This is equivalent to shortening the constituent codes that make up the product code. This method enables a coarse granularity on shortening, and at the same time maintaining the highest code rate possible by removing both data and parity symbols. Further shortening is obtained by removing individual bits from the first row of a 2-dimensional code, or from the top plane of a 3-dimensional code.

## 13.4 Example of a Shortened 2-Dimensional TPC

For example, assume a 456-bit block size is required with a code rate of approximately 0.6. The base code chosen before shortening is the (32,26)×(32,26) code which has a data size of 676 bits. Shortening all rows by 5 bits and all columns by 4 bits results in a (27,21) × (28,22) code, with a data size of 462 bits. To get the exact block size, the first row of the product is shortened by an additional 6 bits. The final code is a (750,456) code, with a code rate of 0.608. Figure 33 shows the structure of the resultant block.

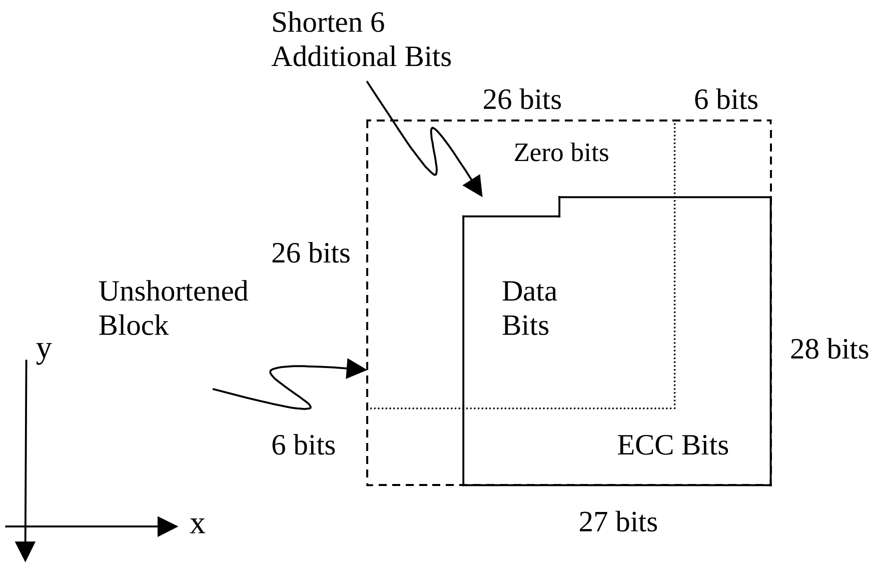



Figure 33: Structure of Shortened 2 D Block

Modifications to the encoder to support shortening are minimal. The shortening procedure is trivial, and yet an extremely powerful tool that enables construction of a very versatile code set.

## 13.5 Example of a Shortened 3-Dimensional TPC

Suppose a 0.4 - 0.45 rate code is required with a data block size of 1096 bits. The following shows one possible method to create this code.

Start with a (32,26)×(32,26)×(4,3) code. The optimum shortening for this code is to remove rows and columns, while leaving the already very short z-axis alone. Therefore, since a 1096 bit 3-Dimensional code is required, the desired vector data size can be found by taking the square root of 1096/3 and rounding up. This yields a row/column size of about 20. In fact, having a row size of 20, a column size of 19, and a z-column size of 3 gives the closest block size to 1096 bits.

The code size is now a (26,20)×(25,19)×(4,3) = (2600,1140). To get the exact data size, we further shorten the first plane of the code by 44 bits. This is accomplished by shortening 2 full rows from the first (xy)-plane, with each row removing 20 bits from the data block, and shortening another 4 bits from the next row. This results in a (2544,1096) code, with rate = 0.43. The following diagram shows the original code, along with the physical location of the shortened bits.

Figure 34 shows the original code along with the physical location of the shortened bits.

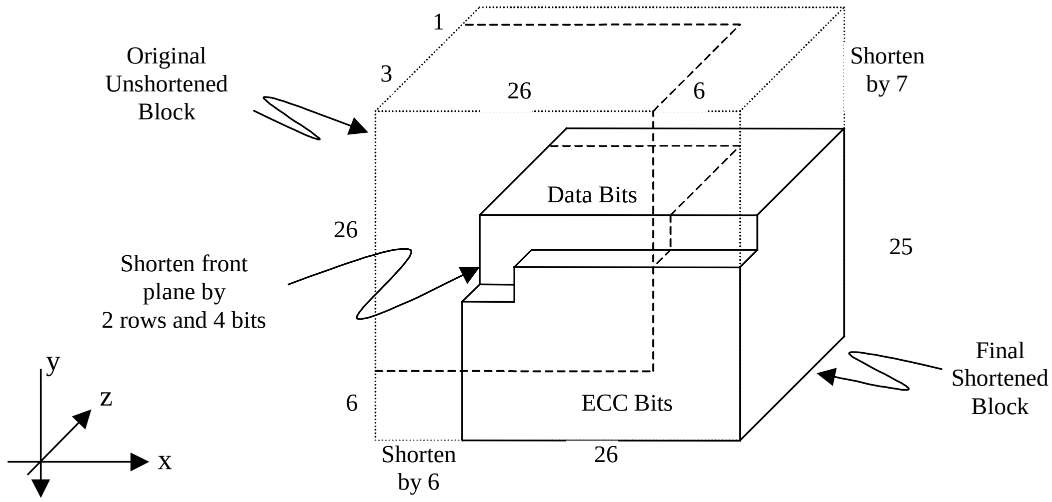



Figure 34: Structure of Shortened 3-D Block

## 13.6 Iterative Decoding

Huge performance advantages may be directly associated with the decoding mechanism for product codes. There are many different ways to decode product codes and each has its merits, however, the goal is maximum performance for a manageable level of complexity.

It is known that if it is possible to use unquantised information (so called soft information) from the demodulator to decode an error correcting code, then an additional gain of up to 2 dB over fully quantised (hard decision) information is achievable. It is therefore desirable to have soft information decision available to the TPC decoder.

Of course, we could in theory consider the decoding of this code a single linear code of size ($n_x \times n_y \times n_z, k_x \times k_y \times k_z$), using a soft decision decoder, but this will in general (apart from the smallest, and of course worst performing) be prohibitively complex.

It makes sense therefore, since these codes are constructed from (simple) constituent code that these soft decoders are used to decode the overall code. However until recently there have only been hard decision decoders for these constituent decoders. In recent years the computational power of devices has made it possible to consider (sub optimal) soft decision decoders for all linear codes. This is only half the solution as the main difficulty is with passing the information from one decoder to the next (i.e. when switching from decoding the rows to decoding the columns). For this, accuracy will need to be kept to a maximum, and so using soft input soft output (SISO) decoders will need to be considered. This is such that an estimate of the transmitted code word may be found and also an indication of the reliability. This new estimate may then be passed onto the next decoding cycle. Inevitably, there will be some degradation from optimal if we are to achieve our decoding using this method, but it does enable the complexity to be reduced to a level that can be implemented. Also, studies have shown that this degradation is very small, so this decoding system is very powerful.

What follows now is an explanation regarding the iterative nature of the decoding procedure. If we consider that, given 2-D TPC block, we define the first round of row and column decoding as a single iteration. We may then perform further iterations, if required. Thus, the main areas of investigation are that of the SISOs, and that of using some previously decoded information in subsequent decoding operations. These are both separate and yet connected areas of interest, as shall be explained.

With regards to the SISOs, there are many different methods including the following which have been described in detail in published academic papers:

1) Soft-Output Viterbi Algorithm (SOVA) [21]
2) The modified Chase algorithm [22]
3) The BCJR algorithm [25],

There have been many other papers explaining these algorithms both as independent algorithms for coding schemes and as part of turbo type decoding schemes. It must be noted that these are not the only algorithms that can achieve soft input soft output style decoding, but they are at present the most readily cited in academic literature.

Each block in a product code is decoded using the information from a previous block decoding. This is then repeated as many times as. In this way, each decoding iteration builds on the previous decoding performance.

Figure 7A illustrates the decoding of a 2-D TPC. Note here that prior to each decoding there needs to be a mathematical operation on all the data we have at that particular time, that is the current estimate of the decoded bits, the original estimate from the demodulator (this will not be used in the first decoding) and the channel information (where applicable).

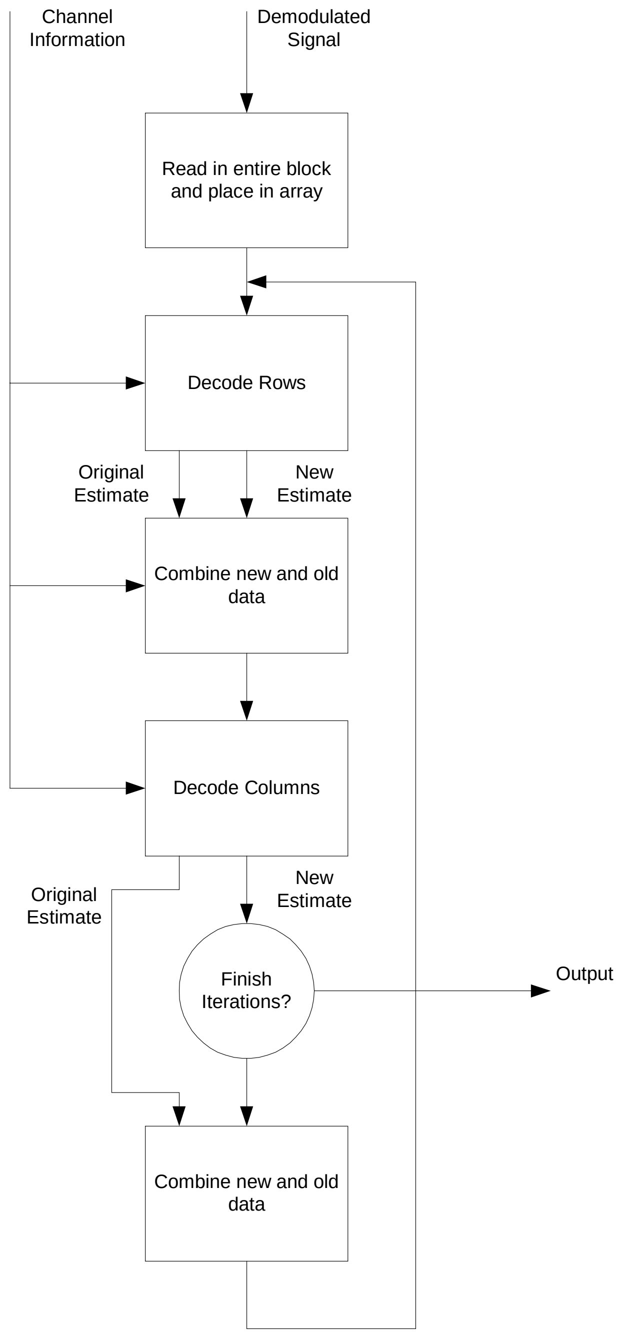



Figure 35: Procedure for decoding of 2-D TPC

It can easily be seen from Figure 35 that the iteration idea is applicable to one complete decoding of the rows and one complete decoding of the columns.
There is an obvious question as to how the iteration procedure is terminated. This is a question only answerable by the system provider and depends on performance and delay; more iterations imply better performance as the expense of a larger latency. Of course, over clocking the system in comparison can significantly reduce the

latency. When considering hardware, the problem of varying delays may be encountered, thus it may be advantageous to fix the number of iterations performed.

## 14 References

[1] H. Sari, Y. Levy, and G. Karam, "Orthogonal Frequency-Division Multiple Access for the Return Channel on CATV Networks," ICT '96 Conf. Rec., vol. 1, pp. 52-57, April 1996, Istanbul, Turkey.

[2] H. Sari and G. Karam, "Orthogonal Frequency-Division Multiple Access and its Application to CATV Networks," European Transactions on Telecommunications (ETT), vol. 9, no. 6, pp. 507-516, November - December 1998.

[3] Richard V. Nee and Ramjee Prasad, “OFDM for wireless multimedia communicatons”, Artech House Publishers,2000.

[4] 802163c-00_33 – Initial OFDM/OFDMA PHY proposal for the 802.16.3 BWA

[5] 802163c-00_34 – Initial OFDMA/OFDMA PHY proposal for the 802.16.3 BWA

[6] Document to the WirelessHUMAN™ study group - Proposal for an adaptive OFDM system for the WirelessHUMAN™ standard

[7] Savo Glisic and Branka Vucteic, “Spread Spectrum CDMA Systems for Wireless Communications”, Ch. 2-6, Artech House Inc., 1997.

[8] William C. Y. Lee, “Mobile Cellular Telecommunications Systems”, Ch. 6, McGraw-Hill Book Company, 1989.

[9] Cheong Yui Wong, Roger S. Cheng, Khaled Ben Letaief and Ross D. Murch, “Multiuser OFDM with Adaptive Subcarrier, Bit, and Power Allocation”, IEEE Journal on Selected Areas in Communications, Vol. 17, No. 10, pp. 1747-1757, October 1999.

[10] Ye Li and Nelson R. Sollenberger, “Adaptive Antenna Arrays for OFDM Systems with Co-Channel Interference”, IEEE Transaction On Communications, Vol. 47, No. 2, pp. 217-229, February 1999.

[11] Heidi Steendam and Marc Moeneclaey, “Analysis and Optimization of the Performance of OFDM on Frequency Selective Time-Selective Fading Channels”, IEEE Transaction On Communications, Vol. 47, No. 12, pp. 1811-1819, December 1999.

[12] Andres Vahlin and Nils Holte, “Optimal Finite Duration Pulses for OFDM”, IEEE Transaction On Communications, Vol. 44, No. 1, pp. 10-13, January 1996.

[13] ETSI EN 300 744 (DVB-T), 1999.

[14] Scott L. Miller and Robert j. O'Dea, "Peak Power and Bandwidth Efficient Linear Modulation", IEEE transactions on communications, Vol. 46, No. 12, pp. 1639-1648, December 1998.

[15] Kazuki Maeda and Kuniaki Utsumi, "Bit-Error of M-QAM Signal and its Analysis Model for Composite Distortions in AM/QAM Hybrid Transmission", IEEE transactions on communications, Vol. 47, No. 8, pp. 1173-1180, August 1999.

[16] EN 301 790, Digital Video Broadcasting (DVB) Interaction Channel for Satellite distribution system.

[17] A Viterbi Algorithm with Soft-Decision Outputs and its Applications, J. Hagenauer, P. Hocher, IEEE Globecom '89, Nov. 1989, pp. 1680-1685.

[18] David Chase, "A Class of Algorithms for De-coding Block Codes With Channel measurement. Information," IEEE Transactions of Information Theory, vol. 18, No 1, pp. 170-182, January 1972.

[19] RunCom Technologies, "Coding Performance of concatenated Reed-Solomon and Convolutional Coding, DVB-RCT Contribution", August 2000

[20] DVB-RCT standard draft, October 2000

[21] G.Drury, G.Markarian, K.Pickavance. "Coding and Modulation for Digital Television", KLUWER Academic Publishers, USA, 2001

[22] ETSI EN 301 021

[23] A. Cantoni and L.C. Godara, "Fast Algorithm for Time Domain Broad Band Adaptive Array Processing", IEEE Trans. Aerops. Electr. Syst.AES-18, 682, September 1982.