Paris VI (UPMC) University and TELECOM ParisTech

Internship report
Master of digital communication systems

New Channel Coding for DASH7 Wireless Sensor Networks

Author:
Shahab SHARIAT BAGHERI

Responsbile of Internship:
Yordan TABAKOV
Supervisor(TELECOM ParisTech):
Ghaya REKAYA BEN OTHMAN
Supervisor (UPMC):
Jullien SARRAZIN

Master Science de l'ingénieur (SDI), Spécialité Systèmes Communicants, Systèmes de Télécommunication Numériques (STN)

September 2015

Learn from yesterday, live for today, hope for tomorrow. The important thing is not to stop Questioning.

Albert Einstein

PARIS VI (UPMC)

Abstract

Télécommunications
COMELEC Departement

Master Science de l'ingénieur (SDI), Spécialité Systèmes Communicants, Systèmes de Télécommunication Numériques (STN)

New Channel Coding for DASH7 Wireless Sensor Networks

by Shahab SHARIAT BAGHERI

The Dash7 Alliance Protocol is based on the ISO/IEC 18000-7 standard which is an international standard that describes a series of diverse RFID technologies. The main objective of this internship which has been held in a french Start up Company *Wizzilab* at Paris is to implement and enhance physical/data-link layer of DASH7 Specification. The proposed case in this internship are Bandwidth Channel Definition of protocol and the other is to have a new channel codes which would give a better performance in terms of error probability and power spectral efficiency simultaneously and finally design a Software Defined Radio (SDR) for Network using Octave/Matlab based programming for considering the theoretical issues.

Contents

List of Figures

List of Tables

To My Parents and *Wizzilab* . . .

Chapter 1

Introduction

1.1 Wireless Sensor Networks

A wireless sensor network (WSN) is spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, pressure, ... The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance. Today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

The WSN is built of ”nodes” from a few to several thousands, where each node is connected to one or sometimes several sensors. Each sensor network node has typically several parts: a radio transceiver, antenna, a micro-controller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

Figure 1.1: Parking Sensor Detector

1.2 Wizzilab

My internship has been held in a Start-up Company well called *Wizzilab* and i had a great pleasure to be under direction of Yordan Tabakov (CTO of Company) for this period of 6 months which Michael André is Co-founder. Actually Wizzilab has 5 engineers which they are located in Paris (18th Arr). Wizzilab is the only leader of DASH7 protocol for Wireless Sensor Network in France, they develop the solutions for Ultra-low-power and low latency Wireless Networks of DASH7 protocol, once an application is for detecting of Parking place which can be used for showing the empty places to drivers showed in Figure 1.1.

1.3 Scope and Objectives

By this way Wizzilab proposed an internship to have a Software/Simulator tool which permits us to model the physical layer of their protocol (DASH7) in terms of Power Spectral Density (99% Effective Bandwidth), Error Probability and generally Quality Of Service (QOS). A candidate who could be as an engineer to enhance their protocol (Physical Layer) and optimize it by theory methods and verifications. One can be adding a 'New Channel Coding', so they were searching a student who has a strong motivation in Research and Development to evaluate their objectives.

Wizzilab needed to watch precisely the Power Spectral Density of modulated signal and to define exactly the masks and bandwidth of its channels which has been announced in the recent version of DASH7 specification (DASH7 Alliance Wireless Sensor and Actuator Network Protocol Version 1.0).

IoT or *Internet of Things* is one of the most prestigious issues idea of technology in these years. its idea is to connect all objects or 'Things' to the Internet and finally together to have a absolute intelligent environment life. It means that for example you can leave your refrigerator and go to another country and from there you would have control of the temperature of it to increase or decrease it! it's really nice and amazing or everything that you can imagine. Actually DASH7 and properly wizzilab are the same idea as well as this way of thinking. Sensors of company are enable to connect to 3G Network and autonomously adapt their situation in function of changing system parameters.

In Chapter 2 we will explain some useful details of protocol and in the next Chapters 3 and 4 are our propositions to DASH7 and is all done projects with the first results part of Chapter 3 you will find the curves of power spectral densities of different options signalling, modulation, designing masks for channels and the second part of this chapter aims to design a Software-Defined Radio (SDR) who can be the mathematical model of system between Sensors (End-points) and gateway, we will design our simulation model and we will explain 'Time Syncronization', 'Carrier Aggregation','Adaptive Frequency Agility', techniques for data transmission in 8 adjustment channels. The chapter 4 will talk about new channel codings for DASH7 which can be used for DASH7 instead of previous channel codings and it is showed that by using these codes theoretically we gain approximately 2dB in SNR. Globally LDPC and Reed-Solomon codes are proposed instead of Convolutional and for using in DASH7 protocol. In this part you will find Block/Frame Error Rate (BLER) and Bit Error Rate (BER) curves with different algorithms of Encoding and Decoding (Hard/Soft) in different channel models.

1.4 Tools of Internship (Octave, Linux, Git, ...)

I had a great chance to be between some strong developers and engineers who know the practical issues and problems and who use alternatively the tools which are so important to study as an Researcher Engineer .

In this way i had to work with some tools like spectrum analyser, Linux environment, some useful software developer as Octave, C/C++, Git, bash Scripting code, LaTeX, Gvim Editor, Tricount Website and ... .

Chapter 2

DASH7

DASH7 or Developers Alliance for Standards Harmonization of ISO 18000-7 is an open source protocol RFID-standard for *Wireless Sensor Networks*, which operates in the 433 MHz unlicensed ISM band/SRD band. DASH7 provides multi-year battery life, range of up to 2 km, indoor location with 1 meter accuracy, low latency for connecting with moving things, a very small open source protocol stack, AES 128-bit shared key encryption support, and data transfer of up to 200 kbit/s. DASH7 is the name of the technology promoted by the non-profit consortium called DASH7 Alliance.

There are many choices for wireless sensing data devices today, ZigBee, RuBee, NFC, UWB, ... however the promise of DASH7 lies in its *low power*, *low cost*, *simple architecture* and a very efficient software code base dash7 technology is well positioned to become preferred solution in many applications the department of defense for example has been using 433MHz sensing devices for over 10 years and their new RFID contract requires all wireless identification devices to conform to the ISO 18000 DASH7 standard.

In addition to natinal defense deployments dash7 compliant products are being developed for private industries as well including energy, automotive, agriculture, civilian airspace, pharma and consumer electronics, Figure 2.1 just to name a few market potential dash7 is only limited by entrepreneurial spririt and adoption of an active standard that works for every one.

The purposes of the Alliance are:

- Manage the DASH7 Alliance specification (based on ISO 18000-7) in order to benefit overall market conditions and to foster interoperability of DASH7 based products and technologies.

Figure 2.1: Applications of Wireless Sensor Networks

- Provide an environment where its Members may meet to discuss and approve suggested revisions and enhancements that evolve the relevant specifications for protocol standards respecting the Alliance IPR policy.
- Provide a forum where users may meet with developers and providers of related products and services to identify requirements for interoperability and general usability of DASH7 specification.
- Promote the DASH7 Alliance specification and educate the business and consumer communities as to the value, benefits and applications for such interoperable consumer and commercial products.
- Protect the needs of users and increase competition among vendors by supporting the creation and implementation of uniform conformance test procedures and processes to foster the interoperability of products and services.
- Once applicable, make appropriate submission of the specifications to international standardization agencies, for ratification, approval and adoption of such specifications as an official national or international standard by such agencies or bodies.

2.1 Main Concepts of Specification

In vision of a Network, *Physical* and *Link* layers plays main role in my internship because Channel Coding does in this 2 layers one in 'data link' layer which just aims for detection packet errors and second is in physical layer in which we are interesting more in detecting and correcting errors. Physical layer also adds a part to packet namely 'syncronization' part which consists of *preamble + sync* code byte, this is because generally each receiver must be synchronized by its transmitter so receiver listens to medium just a little moment of received packet and by 'sync word' it can synchronize the clock of signal. There is also an interesting concept well namely 'Adaptive Frequency Agility' which includes native support for Adaptive Frequency Agility. Actually receiver is allowed to listen to multiple channels simultaneously, and a transmitter to communicate over any of these channels. In the request, a requester provides a list of channels it listens for the response. The responder can randomly select any channel of the list for its response.

Other important concept of physical layer of DASH7 is *Long Range RF compatible with International Regulations*, DASH7 specifies a bandwidth Physical Layer in the sub-GHz ISM bands compatible with all major International Regulations (FCC, ECC, China, etc). Generally there is 2 channels for communication in DASH7 'Normal-Rate' and 'High-Rate' which provide ultra long range capacity via the usage of 25 KHz ultra-narrow band channels. The 'Normal Rate' channel class provides the best compromise between range and low power and 'High-Rate' and new channel proposed 'Lo-Rate' which will be discussed more in Chapter 3.

There are also another different concepts and algorithms in upper layers of DASH7 which are not located in my objectives of internship so for more information please refer to DASH7 specification.

2.2 Packet and Data link Structure of Specification

Now we are interested more in considering physical layer 'Frame Structure' in terms of 'Control Network Packet' to show how we manage it in the network and after that details of 'Signal Processing'(signalling of packet, modulation , Coding, ...) which will be sent into channel are considered.

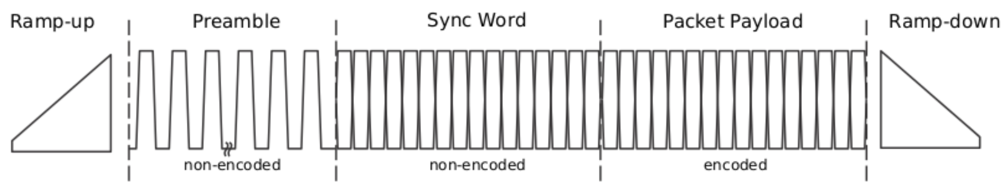


FIGURE 2.2: General structure for all frames

In Figure 2.2 shows different part of frame structure as you can see there is a part, 'Packet Payload' for physical layer in which data is coded by an Control Error Codes (CEC). The *'Sync Word'* part or 'Synchronization Word' is a known binary stream by

Part	Min	Typ	Max	Units
Preamble(Normal Class)	32	32	64	Symbols
Power Ramp-up	48	48	128	Symbols
Sync Word	16	16	16	Symbols

Table 2.1: Bit Order of Frame

Preamble + Sync	Length	Subnet	Control	TADR	Payload	CRC16
	$1Byte$	$1Byte$	$1Byte$	$0/2/8Bytes$	$0-251Bytes$	$2Bytes$

Table 2.2: Frame Structure

receiver for Synchronization. 'Preamble' part is a 101010... binary stream which makes a periodic (pseudo-sinusoid) signal just a way to estimate the delay between transmitter and receiver and so it's a work that is done in the last level of transmitter and first level of receiver and not need to be coded. Additionally 'Ramp-up' part and 'Ramp-down' periods may precede and follow the packet. These periods should be kept as short as possible, used for the purpose of meeting the channel stopband requirements.

In DASH7 *'Sync Word'* has 2 classes different Coding, i just chose and fixed '0xE6D0' as a 'sync word' which has 'CS0' Coding scheme. You can find the typical number of bits in DASH7 packet.

In Table 2.1 you can see the minimum, maximum and typical range of bits for different part of packet.

In Table 2.2 you can see more details of frame which are in addition to *sync part* consists of *Length*, which declares the Length of all packet (all of bits in the right of it), *Subnet*, allows configurable, data-based filtering of incoming frames. Each device contains an internal subnet value (the Device Subnet) that is compared with the value of the incoming frame, *CRC16*, is just simply a Cyclic Redundancy Check channel code added by data link layer to permit receiver to detect error in frame. *Control* and *TADR* are fields that is not considered in this internship.

In Table 2.2 the right part of syncronization will be coded by a channel coding code in physical layer which in DASH7 is previewed Convolutional(1/2).

So basically, there are 2 Error Control Codes which will be considered more in Chapter 3 in which we will propose 2 new channel coding codes who give a better performance (Bit Error Rate vs SNR) deisgning Algorithms for Practical aspects like memory usage and Mathematical order of calculation of each algorithm.

Chapter 3

Bandwidth Specification and Cognitive Radio

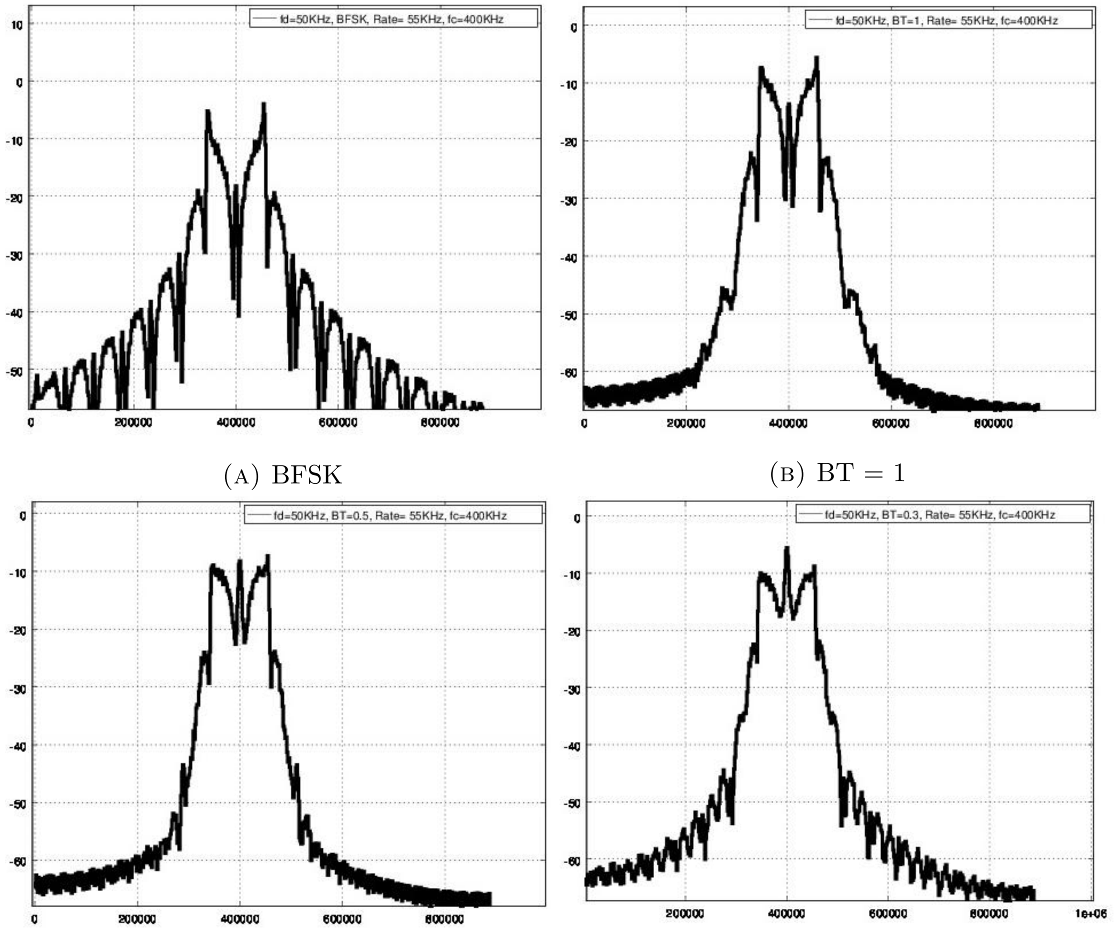
In this chapter we will discuss about 'Bandwidth Specification and Spectrum Utilization' in which we trace different curves of Power Spectral Density for different channels of DASH7 with designing a proper Mask, also 'Software Defined Radio for Sensors' that defines the real system between nodes. In each first part is going to talk about the objectives and results of work and then we will explain the details of project in terms of theory and practical aspects.

3.1 Spectrum Utilization and Channels Classes

3.1.1 Objectives and Resutls

DASH7 uses the 433 MHz ISM Band, occupying 433.05 to 434.79 MHz. The formal DASH7 spectrum extends from 433.056 to 434.784 MHz, organized into channels. Each channel is defined by an index specifying its center frequency and an index specifying its bandwidth. Any given device may, at any given time, permits communication on any combination of supported channels. Optimal practice, however, is to minimize the usage of overlapping channels within a single network.

Basically after my propositions in DASH7 we have 3 channels 'Hi-Rate', 'Normal-Rate', 'Lo-Rate' with rate and index modulation of (9.6 Kbps, 1), (55.555 Kbps, 1.8) and (166 Kbps, 0.5) respectively. we added and created 'Lo-Rate' channel and we reduced the rate of 'Hi-Rate' from 200 Kbps to 166.6 Kbps because the designed mask for Normal channel (main channel) couldn't adapt to Hi-Rate. In below you find typically curves of



(A) BFSK (B) BT = 1

(C) BT = 0.5 (D) BT = 0.3

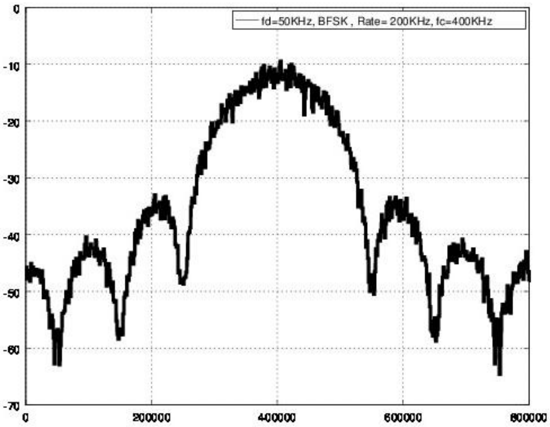
FIGURE 3.1: GFSK modulation, R = 55.55Kbps, fd = 50KHz

power spectral density of different channels by changing an important parameter (BT factor) which is actually a factor of pulse shaping of Gaussian filter.

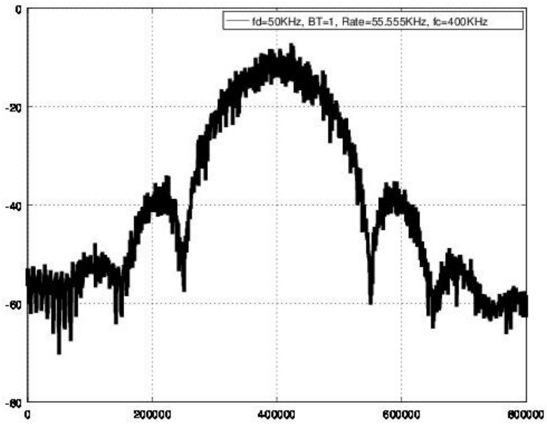
In Figures 3.1 you can easily compare the variation of BT parameter (1, 0.5 and 0.3) for Normal channel in which index of GFSK modulation is 1.8 in DASH7. If we don't apply a Gaussian pulse shaping we obtain BFSK modulation you can easily see by reducing BT the bandwidth is reduced.

Also In Figures 3.2 you see this effect in Hi-Rate channel which index of modulation is 0.5 and the rate is 200 Kbps.

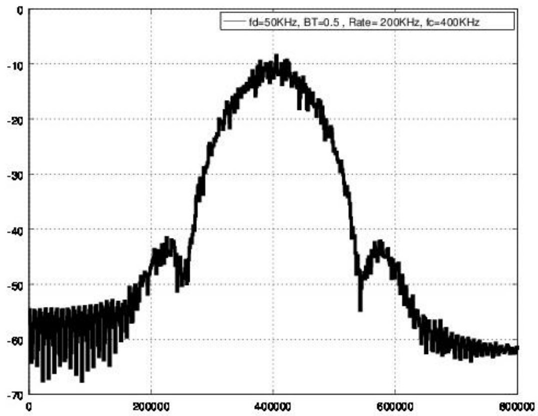
Finally In Figures 3.3 you see this effect in Lo-Rate channel which index of modulation is 1 and the rate is 9.8 Kbps. This channel can be used when the system detects a large noise on received signal because this channel has a smaller bandwidth the smaller noise will be in it.



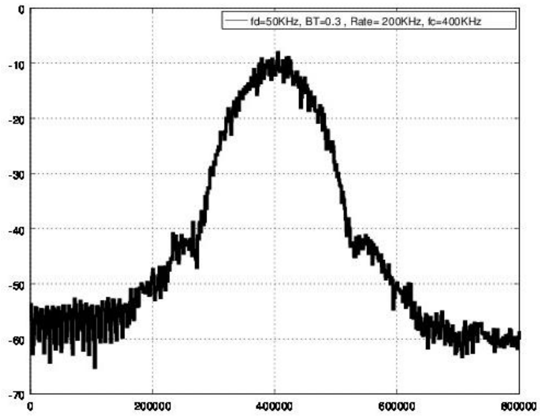
(A) BFSK



(B) BT = 1



(C) BT = 0.5



(D) BT = 0.3

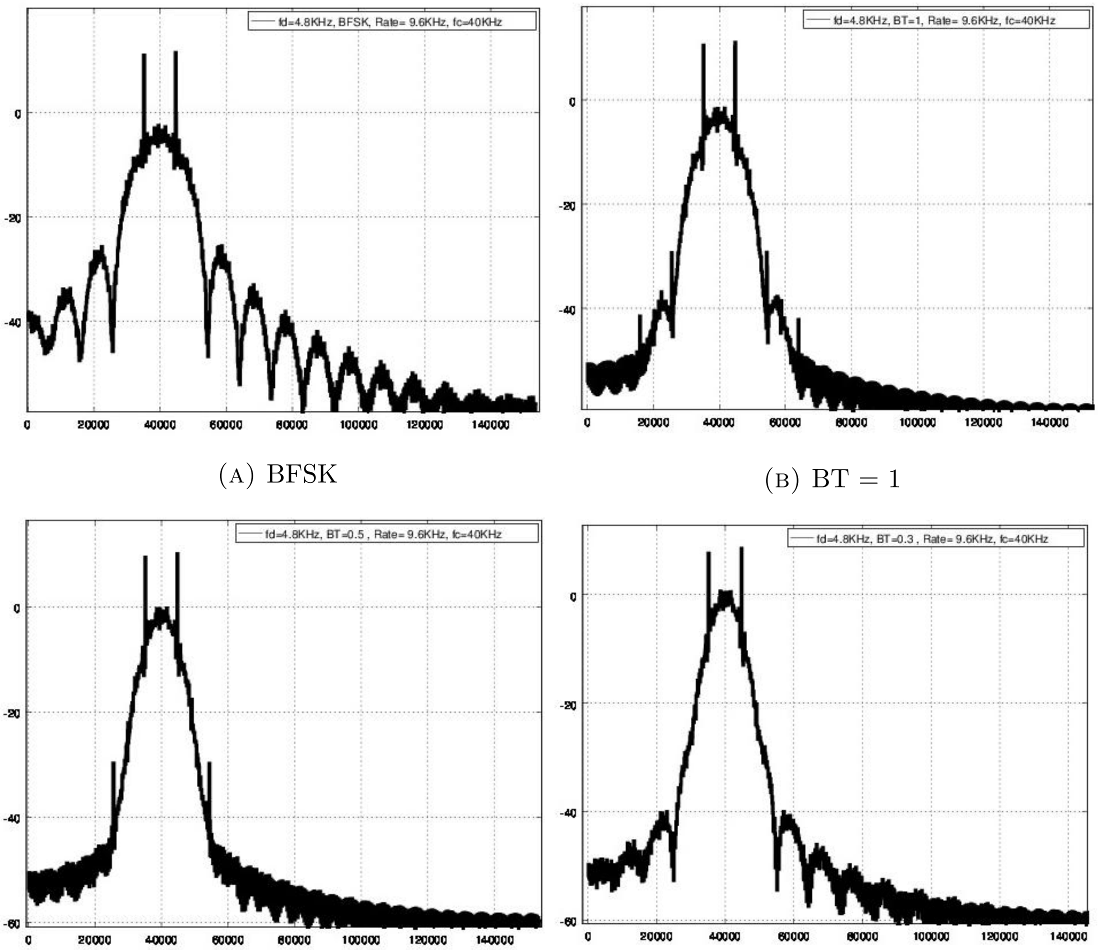
FIGURE 3.2: GFSK modulation, R = 200 Kbps, fd = 100 KHz

3.1.2 Power Spectral Desntiy of Channels and Masks Design

In this subsection we are going to simulate each channels and see the masks that Transmission Filtering Requirements which have choosen.

Actually in DASH7 Specification we have 3 choices for Gaussian filter in function of BT parameter ($BT = 1, 0.5, 0.3$). We saw that $BT = 1$ is a better filter for optimization of Normal channel power spectral efficiency and rate of channels and by fix this parameter, simulations and this subsection Figures shows that for merging the Mask of 'Normal channel' to 'Hi-rate' it should reduce the rate of 'Hi-rate' by setting a sixth of rate of Normal channel($R_{Low} = 1/6R_{normal}$) and rate of Normal should be a third rate of Hi-Rate($R_{normal} = 1/3R_{high}$). It means that the rate of low-rate and Hi-rate channel should be $9.26Kbps$ and $166Kbps$ respectively.

BT Parameter is the product of 'bandwidth of Gaussian filter' and 'symbol time duration' of modulation.



(A) BFSK

(B) BT = 1

(C) BT = 0.5

(D) BT = 0.3

FIGURE 3.3: GFSK modulation, R = 9.6 Kbps, fd = 4.8 KHz

- The transmission should be receivable by BFSK receivers.
- The filter should not decrease detectable SNR against BFSK more than 3 dB.
- If gaussian pulse shaping is used (GFSK), the bandwidth-time product of the gaussian filter shall be 0.3, 0.5 and 1.0.

Figures 3.4, 3.5, 3.6 are mask design of *Normal*, *Low* and *High* rate Channel. Also we can see in figure 3.8 by increasing factor of *BT* bandwidth of signal is increased which is a trade-off of a good Bit Error Rate.

Table 3.1 is the bandwidth (99% of whole power), in Hz, which are calculated by simulator Mathematically.

A summary of classes and modulation parameters in Table 3.2 .

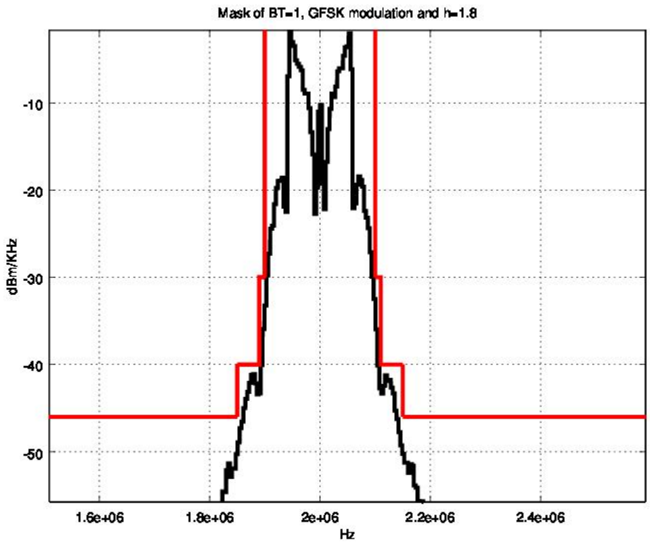


FIGURE 3.4: Mask of GFSK modulation,Normal, for BT = 1

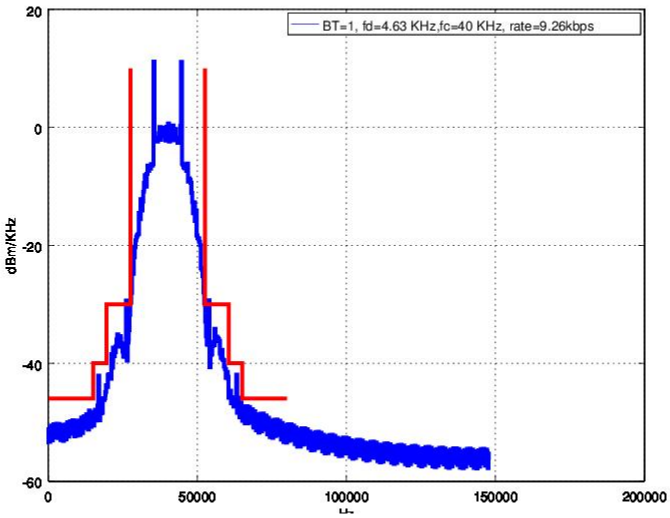


FIGURE 3.5: Mask of GFSK modulation, Lo-Rate, for BT = 1

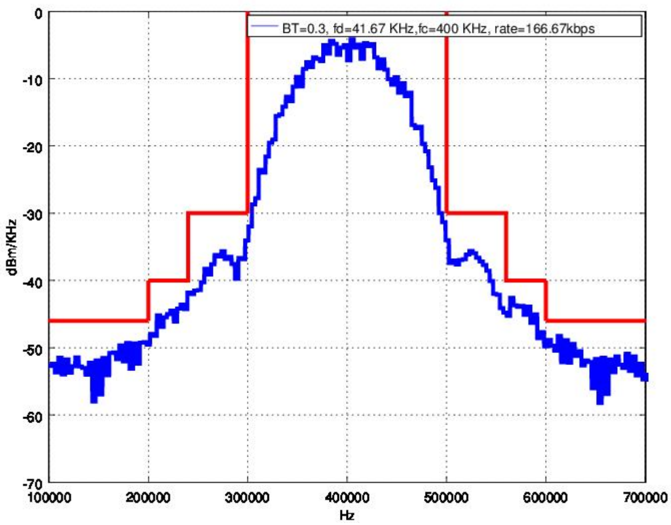


FIGURE 3.6: Mask of GFSK modulation,Hi-Rate channel, for BT = 1

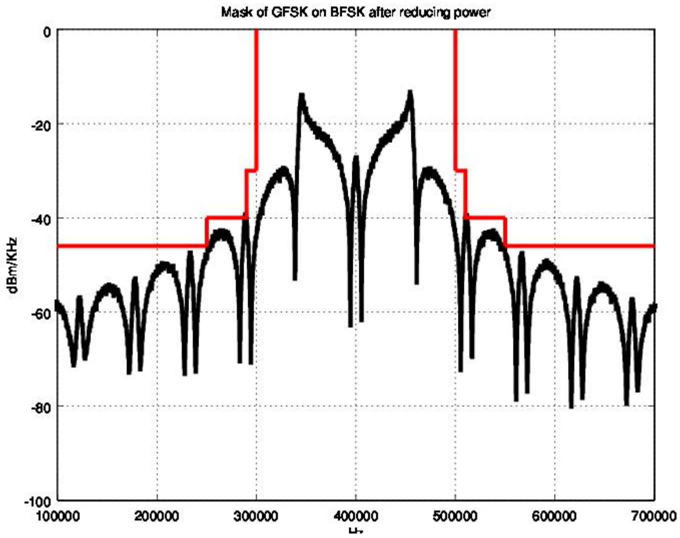


FIGURE 3.7: Mask of GFSK on BFSK power signal, $BT = 1$

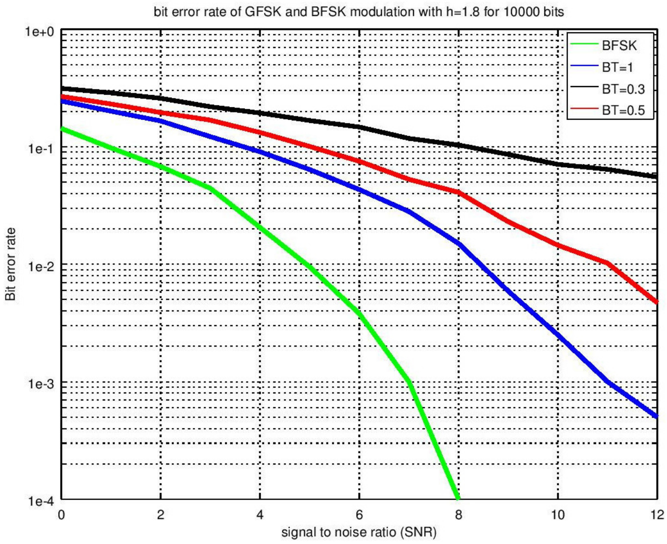


FIGURE 3.8: BFSK,GFSK $(BT = 1, BT = 0.5, BT = 0.3)$ Bit Error Rate

Channel	BW(KHz), $no - BT$	$BT = 1$	$BT = 0.5$	$BT = 0.3$
Normal	176.960	150.518	133.990	124.608
Hi-rate	234.200	217.780	198.852	179.834
Lo-rate	20.366	17.589	15.146	14.459

TABLE 3.1: bandwidth of 99% of whole power

Channel Class	Channel Spacing (MHz)	Modulation	Symbol Rate (kbps)	ndex modulation
Lo-Rate	0.025	2-(G)FSK	9.6	1
RFU	0.200	2-(G)FSK	55.555	1.8
Hi-Rate	0.200	2-(G)FSK	166.667	0.5

TABLE 3.2: Class Channels of DASH7

3.1.3 Mathematical Model of System

In this part we will consider more mathematical and Signal Processing view of modulation.

At first we want to send binary bits to receiver, by this way it must do a work well namely 'Pulse Shaping' which means that by applying a signal to binary stream we transform it to an analogue signal which is normally NRZ signalling in which we map '1' to '1' and '0' to '-1' and then we apply Oversampling meaning that we repeat our data for an arbitary, Os (our case consider:4,8 16) and then we apply our pulse shaping which can be any filter (Rectangular, Raised Cosine, Gaussian, ...) pulse filter. In GSM they use Gaussian Gaussian filter which give very good power spectral efficiency (1.5T puncturing of filter), because of this i also chose this pulse shaping (Figure 3.9).

$$d(t) = \sum_n I_n g(t - nT) \tag{3.1}$$

In 3.1 $g(t)$ is the filter of pulse shaping which is Gaussian, I_n is NRZ data which produce our analogue signal $d(t)$ ready for modulation, this process is named *pulse shaping*. By increasing BT factor Figure 3.9 $g(t)$ we have a narrower bandwidth filter.

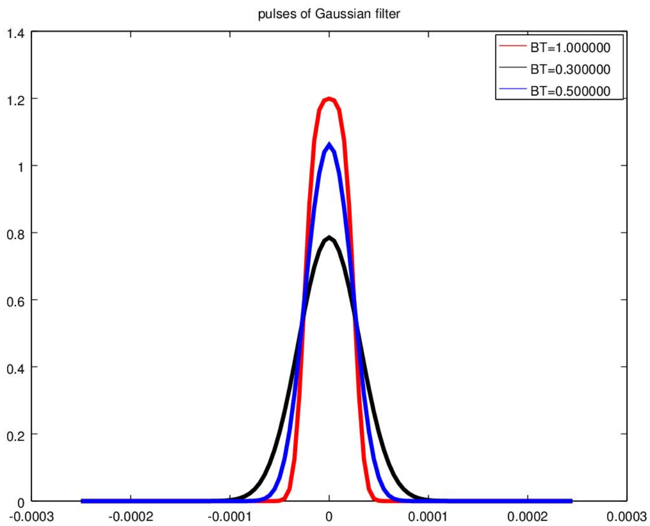


FIGURE 3.9: Gaussian pulse shaping for different BT

Now modulation is very simple to write, equation 3.4 from 3.2 which is produced by integration of $g(t)$ and $s(t)$ signal. To make a Gaussian Frequency Shift Keying (GFSKs) modulation we will have below equations[1].

$$\phi(t;\mathbf{I}) = 2\pi \sum_{k=-\infty}^{n} I_k h_k q(t-kT), nT \leq t \leq (n+1)T \tag{3.2}$$

$$q(t) = \int_0^t g(\tau)d\tau \tag{3.3}$$

$$s(t) = \sqrt{2E_b}cos(2\pi f_c + \phi(t;\mathbf{I})) \tag{3.4}$$

In equation 3.5, n is the nth symbol of input stream, T is symbol duration, $S(f)$ is the fourier transform of $s(t)$ and f_c is the carrier frequency and E_b is Bit energy. All Curves of Power Spectral Density in Section 3.8 comes from equation 3.5[2].

$$P(f) = \frac{|S(f|^2}{2T} \tag{3.5}$$

3.2 Software Defined Radio for sensors

Software-Defined Radio (SDR) is a radio communication system package where components that have been typically implemented in hardware (e.g. mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented by means of software on a personal computer or embedded system. Main objective of SDR's platforms is to have ability of Software control of communication parameters. One of the most prestigious block in Software Radio can be a Error Detection and Correction to detect and correct the occurred errors when we want to use different users in channels to eliminate error caused by interference, fading, noise or ... which is the main reason that we use these codes.

We are going to create packages of sensors (with SDR packets) for Digital Signal Processing aspect of physical layer of DASH7 protocol communication and its communication.

This package will be able to implement several software algorithms which are some introduced here to obtain it, the other name of this software radio is Cognitive Radio in which we have some intelligent algorithms for switching between channels in function of channel or user situations in terms of bit error rate, Interference,... which be considered in this chapter.

3.2.1 Design System and Simulation Models

In this Section we are interested in modelling and designing a communication system between nodes (sensors) and gateway. Figure 3.10 is the most simple globally block diagram of our system to implement, in Section 3.1 we talked about the mathematical model of pulse shaping and modulation blocks. Block of *AWGN Channel* just adds a random process into the input signal and can be changed if the channel would have fading effect. The signal received can be easily model by equation 3.6 in which $n(t)$ random process which has Normal distribution in practice and $s(t)$ comes from equation 3.4.

$$r(t) = s(t) + n(t) \tag{3.6}$$

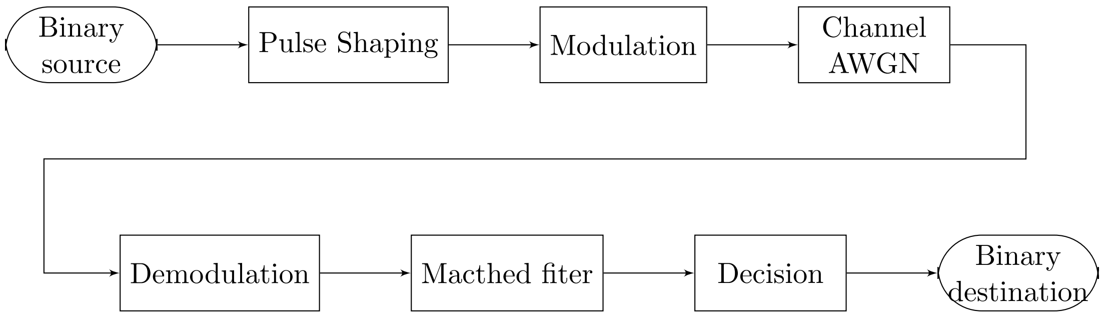


FIGURE 3.10: GFSK Modulation Pulse Shaping system

The equations of 3.1 is done in *Pulse Shaping* block and equations of 3.2, 3.3, 3.4 block are done in *Modulation*, actually the point is that the time duration of symbol (T), index modulation (h), carrier frequency (f_c) are the control parameters of Modulation block, so pulse shaping block is just a mapping from digital to analogue signal (like DAC) to give a analogue form for signal this work can be hold by different filters (Rectangular, Raised Cosine, Gaussian, ...) because of power efficiency benefits in using Gaussian filter, DASH7 uses Gaussian-FSK modulation.

Now it's time to talk about practical aspects of *Pulse Shaping* block, in practice we know that our signals can be just discrete datas so certainly always we have a frequency sampling on continuous signals and so the choice is very important once inside electronics capacity limits our data sampling and the other hand Nyquist theorem and anti-aliasing problems.

In DASH7 rate of Normal Channel is $55.55Kbps$ this means that the period of bit is $18\mu s$ and the bandwidth is 100KHz which will obtain $h = 1.8$. The main Question now is How many data is needed to introduce a Gaussian pulse for a bit($18\mu s$)? in other words, How should be chosen the value of sampling frequency of $s(t)$? The answer is very straightforward by Nyquist theorem - 2 times the highest frequency in $s(t)$ - this is correct but in practice actually most of time it's not enough and it should be applied an OverSampling.

Figure 3.11 is the simulation of figure System 3.10 which is Bit Error Rate curves for different OverSampling of data. It's clear that by increasing (OS) factor it will have a better performance as well as the curves fall downs. You also see after factor $OS = 4$ that the curves have been saturated asymptotically, so for a good trade off between complexity of calculation and performance we choose the smallest which is 4. Without including exceptions, in future and next sections simulations (BER, BLER) OS factor, in all of the and next sections are assumed 4.

Now we explain a little theorical and practical problems of the blocks: *demodulation*, *Matched filter* and *decision* in 3.10 respectively.

Actually in theory *demodulation* needs 2 signal I & Q for detecting the phase of signal by taking I & Q part of received signal. In practice implementation of *Arctan* is not really easy. Specially the microchips can calculate only the basic operator. So we have simplified the model to a beautiful loop (equation 3.8). We can write equation of 3.4 in form of an exponential equation 3.7 which is more practical for calculating of phase. The practical equation mathematics written in Octave for simulation is equation 3.8 in which $r(n)$ is the received signal (equation 3.6) and $\phi(n)$ is the phase signal and *arg* is the argument function which gives the phase of complex input number. This equation prevents to have a wrap change in phase signal(because of feedback) so it's simply the *demodulation* block[3], [4].

Block of *Matched filter* is just a filter to cancel the pulse shaping effect onto 1 & -1 data and now it's time to decision. There is an important issues for matched filter actually our Sensors the receiver doesn't know the kind of Pulse Shaping, by this way practically we shouldn't model it by a matched filter, instead we used because we had fixed the value of sample per samble equals to 4, so we put simply a downsampler with factor of 4[5].

The last block for designing is just a block of *decision* which comes from this idea that because of assumption equi-probability of symbols we give the maximum chance for received signal to be more nearer. So this block it's just a map the positive number to 1 and negative number to 0.

Figure 3.10 is mono-user (one channel) system without any complexity. Other block will be added in future parts.

$$s(t) = \sqrt{2E_b}e^{(j2\pi f_c + \phi(t;\mathbf{I}))} \tag{3.7}$$

$$\phi(n+1) = \phi(n) + arg(r^*(n-1) \times r(n)) \tag{3.8}$$

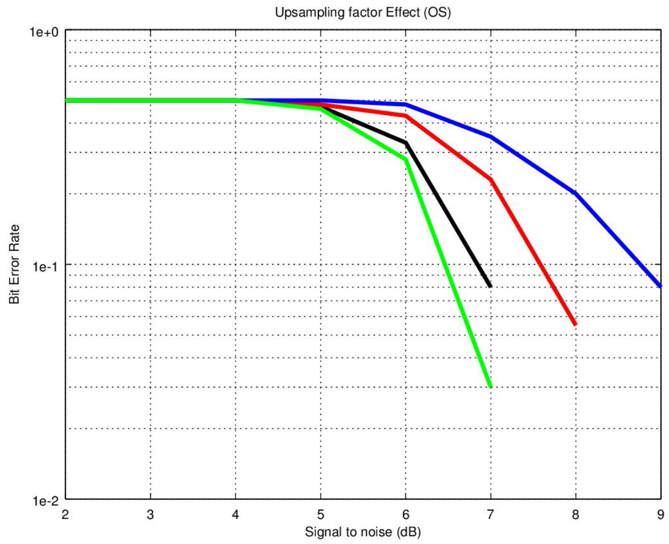


FIGURE 3.11: Bit Error Rate, Normal channel,upsampling factor(OS)

3.2.2 Developed Algorithms

In this subsection according to design the system of mono-carrier we are going to explain the algorithms for making this communication between Reception and Emission and the architecture of system which consists of *Sensors* or End-points for capturing data from environment, *Gateway* a gate for connect the sensors network to mobile network (3G). Mainly we explain *Time Synchronization*, *Carrier Aggregation Technique* and *Interleaving* algorithms.

3.2.2.1 Time Synchronization Algorithm

In Wireless Sensor Network (WSN), we have several nodes (sensors) which they want to communicate with server. By this way each time receiver has a different delay signal from different nodes and because of distances. In this way we are interested in estimation this delay, and after that receiver and Sensor will be synchronized (binary bits). It's the reason that in practice they preview a part of *preamble* and *syncword* in Figure 2.2

in this part we are going to explain more the reasons. The signal processing idea of estimation is very simple, actually in head of each packet which is going to send there is a part of synchronization in which *sync word* is a known binary stream by receiver and *preamble* is always a binary stream in this form : 101010... which produces a pseudo-sinusoidal, receiver listens to channel and it estimates the number of preamble symbol as soon as it detects the *sync word*, by this way it can always understand that which sensor is, and they can be synchronized after this moment.

Now mathematical algorithm of system will be studied, actually the idea is in this way that each time receiver stays to get the signal from channel then multiplies it by *sync word*, signal processing description would be to calculate of Cross-Correlation of 2 signals $r(t)$ and $sync(t)$ and then as soon as the value of this produced signal pass from a threshold (Calculable by theory) it declares the index of delay and preamble number in packet[6]. This algorithm does at the first step of receiver and can be done by analogue received signal, also it can be done just before of binary destination in the sense that we can do this algorithm in binary mode domain by measuring Hamming distance and clock of binary stream.

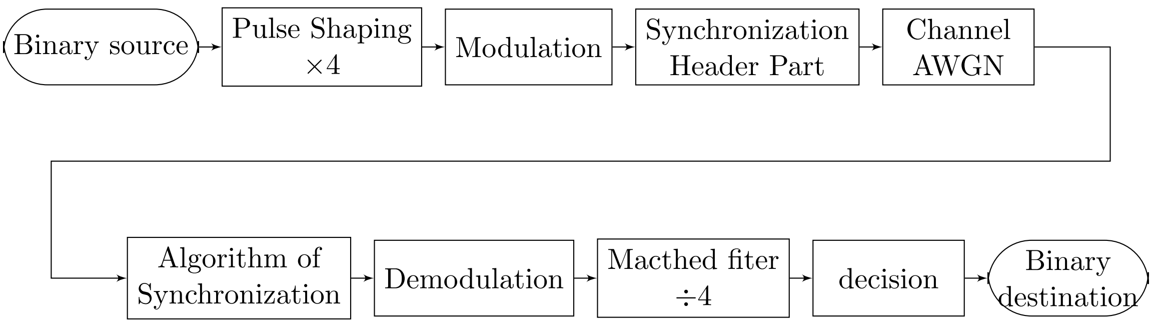


FIGURE 3.12: System of Mono-User with Synchronization Algorithm

In Figure 3.13 you see algorithm of synchronization doesn't effect on BER curves.

3.2.2.2 Carrier Aggregation Technique

As it has been indicated, each WSN communication needs several users by this way we are interested in using a beautiful technique called *Carrier Aggregation* which looks like Wave-length Division Multiplexing (WDM) technique in optical networks.

This technique uses a great idea of medium sharing, actually imagine 8 users want to communicate, these users are separated in frequency domain and now this technique tells, you can easily add the signal of these 8 users and then send it to receiver and in receiver like light which has several spectrum, each correspond user-destination can be

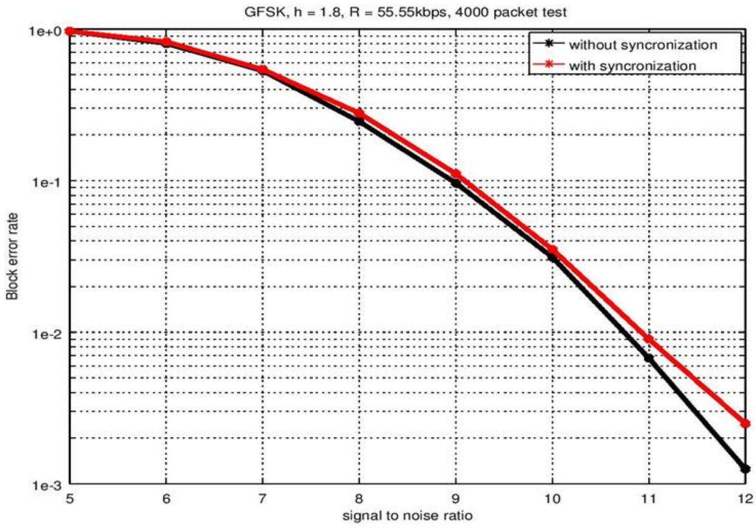


FIGURE 3.13: Algorithm of synchronization and Oversampling

recover signal just by applying a local frequency and then a low pass filter to have the exact signal in receiver.

This technique is done by me in *Wizzilab* and was compatible with DASH7 protocol. Figure 3.12 shows model to implement for each user of radio. As you see there is added 2 blocks *Up/DownSampling & Low pass filter* and *Frequency Carrier*, block of downsampling is for augmentation of immunity of signal against noise and interference between users and other practical advantages. The block of exponential is just to adjust the signal in different frequencies (channels), the point is in receiver we have a block of *low pass filter* which recover the main signal from others. Figure 3.12 is our system which has been tested and was accepted with technical problems. The algorithm of synchronization is independent of carrier aggregation technique and it can be after modulation and before demodulation.

Figure 3.14 indicates different curves of bit error rate against SNR by changing the coefficient of OverSampling (N) value we see that $N = 2$ is the best choice for this purpose. (complexity of calculation and performance) so the radio software will have 8 users with technique of Carrier Aggregation and each user has a specific number of preamble, now we have all th system.

The Next section we will concentrate on mono-user case, Figure 3.15 for developing this system.

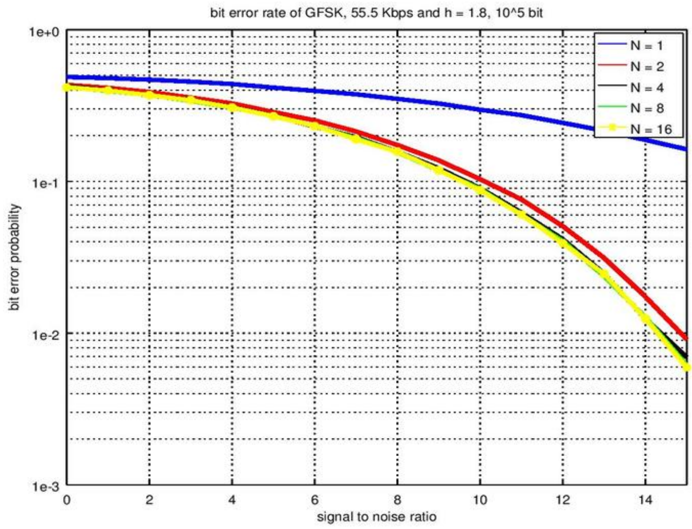


FIGURE 3.14: Bit Error Rate, Normal channel,upsampling factor(OS)

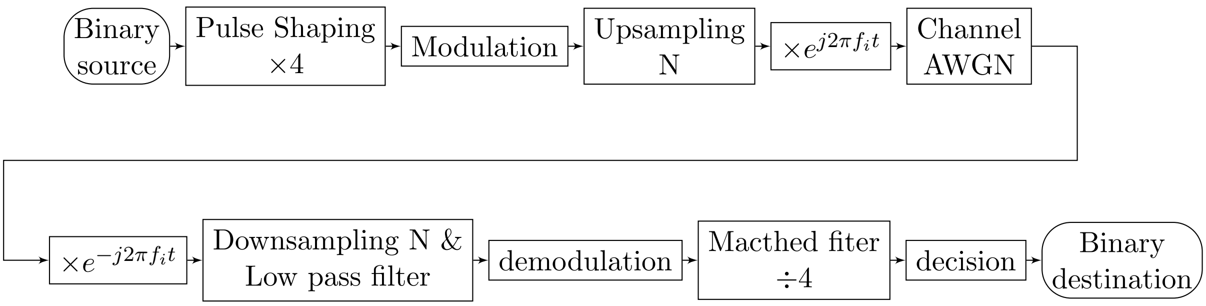


FIGURE 3.15: GFSK Modulation Pulse Shaping system

Symbol data	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Interleaved	30	31	22	23	14	15	6	7	28	29	20	21	12	13	4	5	26	27	18	19	10	11	2	3	24	25	16	17	8	9	0	1

TABLE 3.3: Bit Order of Frame

3.2.2.3 Adaptive Frequency Agility

Adaptive Frequency Agility (AFA) is a technique used by radio transmitters to avoid transmission in channels that are already occupied. The radio transmitter periodically monitors its local radio environment and notes channels that are occupied. Based on this information the transmitters select an operating frequency that is not yet used to avoid interference.

AFA can be very useful if a frequency band is shared among a large group of users or if the band has to be shared with another service which has a higher priority and therefore may not be be hindered.

This technique is generalized in DASH7 by this sense that for some parameters like Bit Error Rate or bandwidth of channel will be chosen and we have designed it for our systems in simulations in internship.

Symbol data	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Interleaved	0	8	16	24	1	9	17	25	2	10	18	26	3	11	19	27	4	12	20	28	5	13	21	29	6	14	22	30	7	15	23	31

Table 3.4: Bit Order of Frame

3.2.2.4 Interleaving

Interleaving is just a block exactly before signalling blocks, actually we can apply this block before all of binary datas which want to pass from channel. Idea of this block is because of a phenomena which is called *Burst Error* and mainly it happens in channels with frequency selective fading which are because of dependency between near symbols in receiver and if an error occurs in a symbol, it will destroy the near symbols.

By this way Interleaving just rewrite the binary stream in such a way to prevent the dependency errors. In DASH7 Specification Interleaving is done by Table 3.3.

This block is done as a box to order bits in that way the nature dependency which exits between near or neighbour symbol can be removed.

There are also some other methods of interleaving which are like Table 3.4 who are matrix interleaving and are useful specially for example in GSM system, but the interesting point is in DASH7 own interleaving is more efficient in terms of Bit Error Rate against GSM Interleaving method.

Chapter 4

New Channel Codings

4.1 Objectives and Scopes

In this Chapter we are interested in developing system with adding blocks called *Channel Coding*, it's clear that it must add it before all of signalling blocks of Figure 3.15 because the idea of channel coding is just adding some logically redundancy bits to permit to receiver at first to detect errors and then the ability of correction in packets. DASH7 protocol uses a channel codings and our scope is to propose a new channel codings to enhance the performance in terms of bit error rate of system.

In information theory, the noisy-channel coding theorem (Shannon's theorem), establishes for any given degree of noise contamination of a communication channel that it is possible to communicate digital information nearly error-free up to a computable theoretical maximum rate which can be calculated by 4.1 through the channel. The Shannon limit or Shannon *Capacity* of a communications channel is the theoretical maximum information transfer rate of the channel, for a particular noise level (SNR). In equation 4.1 B is the Bandwidth, C channel Capacity and SNR is the Signal to Noise Ratio power. Some people also are interested in $\frac{C}{B}$ which is called Spectral Efficiency (SE) of system. Each time that you appliy an Error Control Coding you will have a smaller SE which causes a smaller Bit Error Rate.

$$C = B \times log_2(1 + SNR) \tag{4.1}$$

Globally In Engineering One of the most important issues and parameters is the time that you want to put for a project or work, actually you can not never separate it from one of your minded parameters, it's the reason which for that in practice we are interested in Power and not just Energy, actually the power means the average energy that you can give to a project over time as job is very important, nor you can say easily i will stay 200 years to do my own project, while the technology is growing exponentially each day. So in this part heavily of calculation in function of processes done in code is very important also, more code is efficient more we have a short time to process calculation, for example in some simulations we had to wait 3 days to obtain the exact results which I put it to do during the weekends to use more efficient of time. while we were using a 4-core dell processor.

4.2 Payload Packet Coding

In DASH7 protocol has been previewed as we noted in Table 2.1 we have 2 parts to code, one is *Payload + CRC* in which Payload comes from upper layers, after encapsulation by data link layer 'CRC' will be added at the end of this packet the second is *header* which contains some information about given payload packet and it's ready to give this frame to physical layer to add syncornization part. In this subsection we will talk about coding of payload and the other will be applied on all this packet (without Syncronization) is Convolutional Channel Codes, in this subsection we will propose a new code *LDPC (Low Density Parity Check)* which are named *Capacity-Approaching Codes* because of their Bit Error Rates curves which are very near to Shannon limit against Convolutional codes and after that we will explain the different used algorithms.

In first part we introduce the curves of BER and BLER for different Coding in different channel, the next part is to introduce the *LDPC* & *Convolutional* channel codes and then we explain mathematical equations and ideas of Encoding and Decoding algorithms.

4.2.1 BLER and BER Curves in Different Channel Models of Different Codings

The BER vs SNR is the curve of Bit Error Rate against of signal to noise ratio, in digital modulation is $\frac{E_b}{N0}$ which definitely depends on Modulation and Signal Shaping. it's defined by the number of received error bits in packets, divided by the length of packet with an acceptable statistical of test. It means for watching bit error rate of 10^{-4}, it should have at least 10^4, actually theoricaly it's correct but in practice for a good immunity for example this error probability we test 10^5 or 4×10^4. BLER is just a measurements which is used more in higher layers (data link, ...) which means that the number of received errors packet.

Figure 4.1 is the bit error rate of coherent demodulation without coding of DASH7 (2-FSK, with index of modulation of 1.8 & 0.5) which by increasing when $h = 0.5$ is better because the discontinuity of signalling is lesser in this case. We can say that BLER is proper to BER, so from this time we only concentrate on BLER curves, because it is more important in terms of protocols.

Now figure 4.2 shows Block Error Rate of system 3.15, clearly the difference between system with coding convolutional and without coding. You can see also a better curve (green) which is better when we put *Zero Padding*. Actually when we put a known word in the end of coding packet to help the algorithm State Machine steps for Viterbi (Hard Decoding).

Figures 4.3 and 4.4 are the competitions between LDPC and Convolutional coding, Figure 4.4 shows BLERs figures where LDPC starts to catch Convolutional codes which for *Soft* decoding is about 4.2 dB and for *Hard* decoding is about 5.2. Hard and Soft decoding for LDPC means 'Bit flipping' & 'Log-Sum Product' and for Convolutional means 'Hard','Soft' decision of Viterbi algorithm. Details of algorithms will be discussed in future subsection. In all figures packet bits test was for 16 Bytes (128 bits).

In figure 4.3 you can see there are 2 models of channel AWGN which is famous case of just added randomly noise to the transmitted signal and the second is TU5 which is a model of channel used in GSM system for 'Typical Urban' and is implemented by Jakes algorithms in Octave and figure 4.4 is in TU5 model channel [8].

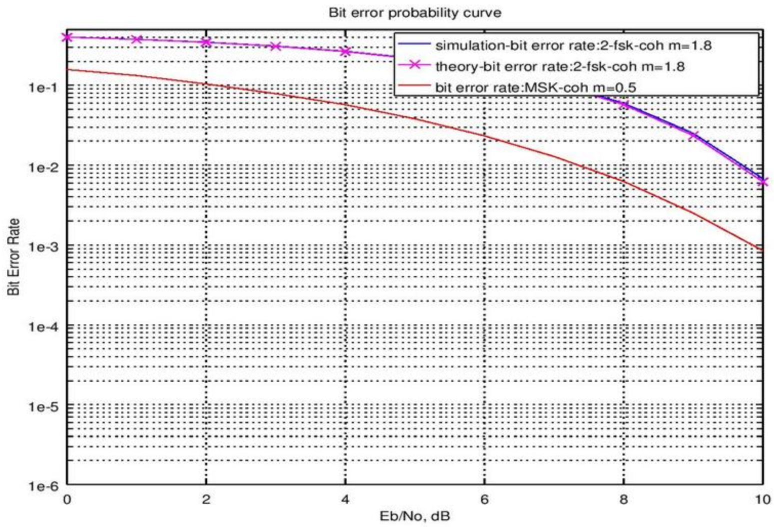


FIGURE 4.1: Bit Error Rate, 2-FSK modulation theory and simulation

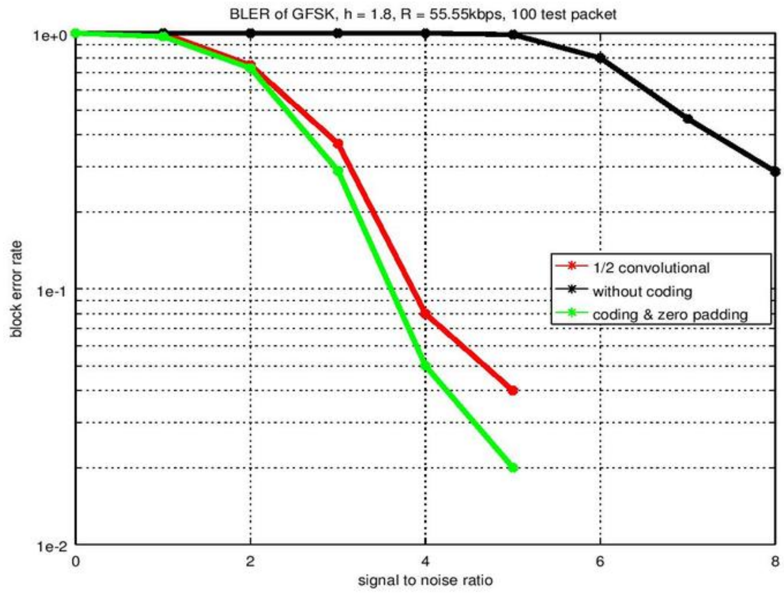


FIGURE 4.2: BLER, with & without Coding, Zero Padding

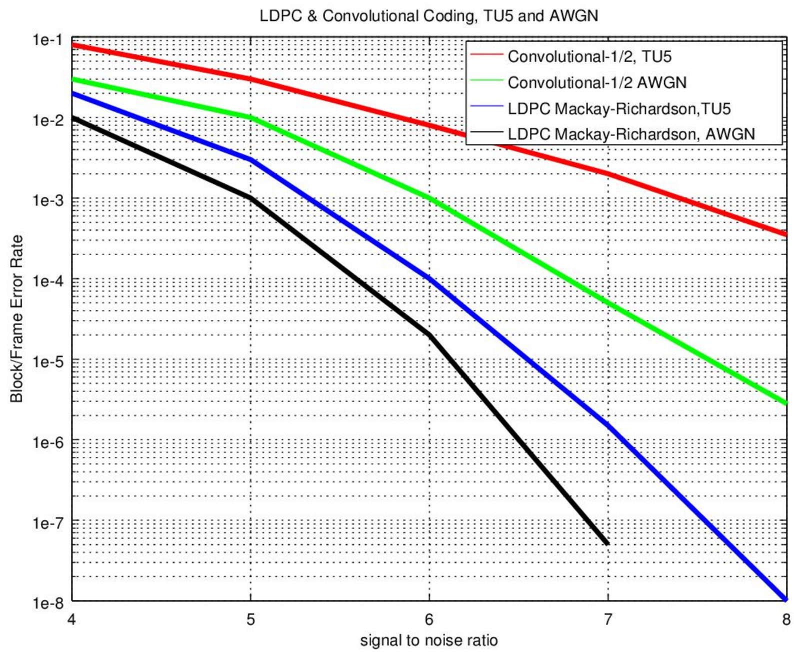


FIGURE 4.3: LDPC(256,128) and Convolutional in AWGN & TU5 channel model

4.2.2 LDPC and Convolutional Codes

In this subsection we are going to explain problems with theoretical and mathematical description. Globally Error Control Codes are divided by 2 parts 'Convolutional' and 'Block Linear' codes.

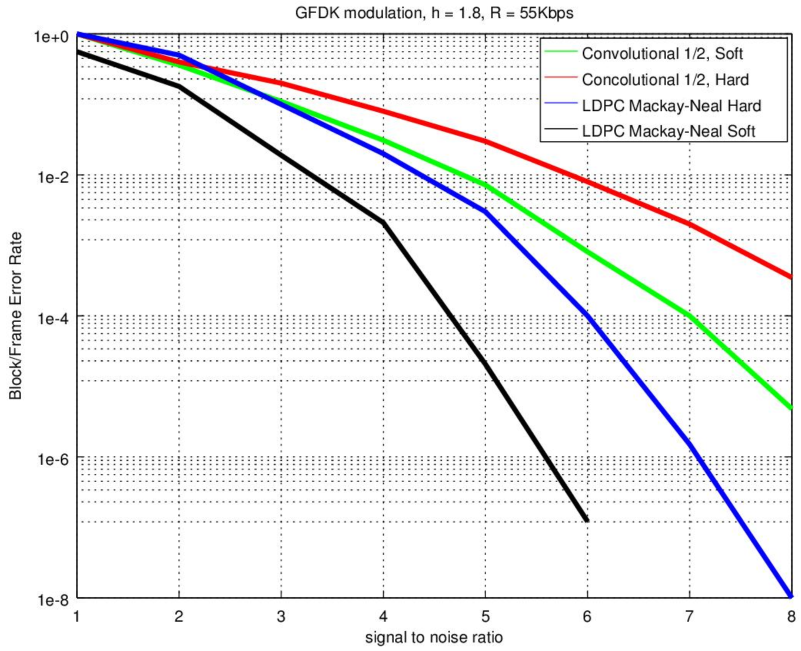


Figure 4.4: BLER of LDPC(256,128) vs Convolutional Coding with Hard and Soft decoding

Now we are interested in presenting our new channel coding, **LDPC** codes, which are very high performance and have strong results in Bit Error Rate curves.

Historically, these codes first developed by Gallager in 1963, and then were forgotten until his work was rediscovered in 1996, Turbo codes, another important class of capacity-approaching codes discovered in 1993, became the coding scheme of choice in the late 1990s, used for applications such as the Deep Space Network and satellite communications, but in the last few years, the advances in Low-Density Parity-Check codes have seen them surpass turbo codes in terms of error floor and performance in the higher code rate range, leaving turbo codes better suited for the lower code rates only.

In 2003, an irregular LDPC code beats six turbo codes to become the error correcting code in the new DVB-S2 standard for the satellite transmission of digital television. The DVB-S2 selection committee made decoder complexity estimates for the Turbo Code proposals using a much less efficient serial decoder architecture rather than a parallel decoder architecture. This forced the Turbo Code proposals to use frame sizes on the order of one half the frame size of the LDPC proposals.

In 2008, LDPC beat convolutional turbo codes as the forward error correction (FEC) system for the ITU-T standard. G.hn chose LDPC codes instead of turbo codes because of their lower decoding complexity because the proposed turbo codes exhibited a significant error floor at the desired range of operation. LDPC codes are also used for 10GBase-T Ethernet, which sends data at 10 Gigabits per second over twisted-pair cables. In 2009, LDPC codes are also part of the Wi-Fi, 802.11 standard as an optional part of 802.11n and 802.11ac, in the High Throughput PHY specification. These interests ,volunteers and reasons in LDPC codes caused to think about these codes for DASH7.

The main idea of *LDPC (Low Density Parity Check)* is to produce a parity check matrix which are typically Low Density, means that the number of '1's bit are very smaller than '0's (0.001). It's the reason that codes are very practical for registering datas we need only to register '1's. there are many methods to construct LDPC codes such Gallagar, Prototype, Mackay Neal, Finite Geometry, RS based, Irregular, Random, ...

In this internship we have tested the easiest to implement (gallagar, Regular, Mackay Neal) because of practical objectives of internship and we chose Mackay Neal because of Figure 4.5 which has gotten from 5, which is Bit Error Rate of 4 different codes, *Rate 1/2 64 state convolutional*, *Computer generated*, *Mackay Neal* and *EG-Gallagar* which can be concluded the best compromise between these codes, EG-Gallagar and Mackay Neal codes are the best performance in BPSK modulation in AWGN channel. Implementation of EG-Gallagar is difficult in practice. In contrast Mackay Neal codes are near to these codes and are more practical. So we chose Mackay Neal method to construct Parity Check matrix. LDPC codes are introduced using a sparse *bipartite* Graph specially in Designing and Decoding Techniques[9].

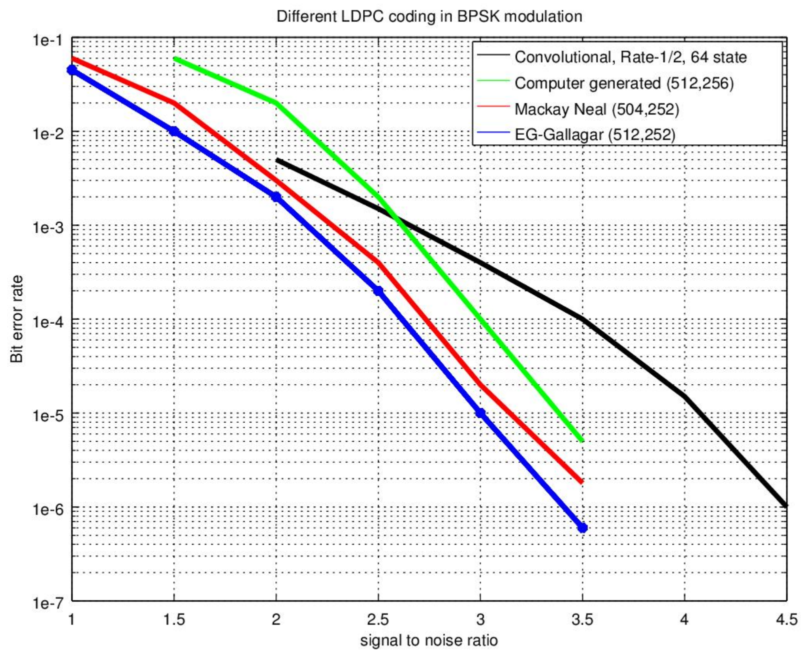


FIGURE 4.5: Rate-1/2 64 state Convolutional code, EG-Gallagar (512,256), Mackay Neal (504,252), Computer generated (512,252)

Convolutional codes are produced by shift registers which is actually a finite state machine and in function of states and input the output (binary coded) will be produced and it's very useful and common in practice. Convolutional code used in DASH7 has a rate of 1/2, 3 shift registers with $G(D) = [\frac{D^3+D+1}{D^3+D^2+D+1}, 1]$ matrix of generator which each D is just an agent of a shift register and 1 is just moment input bit, 2 arguments is because of rate which means that it transforms each bit to 2 bit. Mathematical description let's say if you want to code a given bit for first output of Coding it's just enough to add (in $GF(2)$) all of the register's bits and the input bit and for second output is just enough to add the 3th, second and own bit input to have the 2 bits coded of your suitable bit.

Linear Block Codes. The general idea of these codes is to create a linear function (matrix) of binary bits which are basically produced by algebric structure and algebra abstract concepts which own-self obtained by algebric polynomials. These codes have 2 matrix, *Generator*(G), *Parity Check*(H) matrices. Generator matrix is a $k \times n$ matrix which k is the number of message bits and n is the length of coded binary stream just a matrix that you can multiply your binary stream and the results is the coded binary stream. There are different methods for Decoding of 'Block linear codes' one can be syndrome decoding algorithm. Actually Parity Check matrix $((n-k) \times n)$permits us to understand if a binary stream is in codeword dictionary. We can check it by this method: Multiplying received bit stream by Parity Check matrix the result is called syndrome of stream. Syndrome is going to check, zero if there is no error in stream and if not by checking this stream in look-up table we can find the proper errors. In which look-up table is the correctable errors.

4.2.2.1 Algorithms of Encoding

Algorithm of Encoding of convolutional coding is by a 3 state memory shift register. The circuit is shown in Figure 4.6 which just transforms each bit to 2 bits depends on last moment state and input now, equations of matrix generator produce the coded message $G(D)$ form a feedback in practice to be stable so they are usually normalized.

Actually as we said encoding of linear blocks codes are done by a matrix generator called (G) which is just simply a multiplication, the other hand, basically constructing LDPC codes are done by Parity Check Matrix (H, which are low density), the problem is for encoding it must transform H to G, actually this procedure is not easy, specially when dimensions of parity check matrix is large. In my internship because we wanted to code payload and the minimum acceptable of payload was 16 Byte, we have chosen a parity

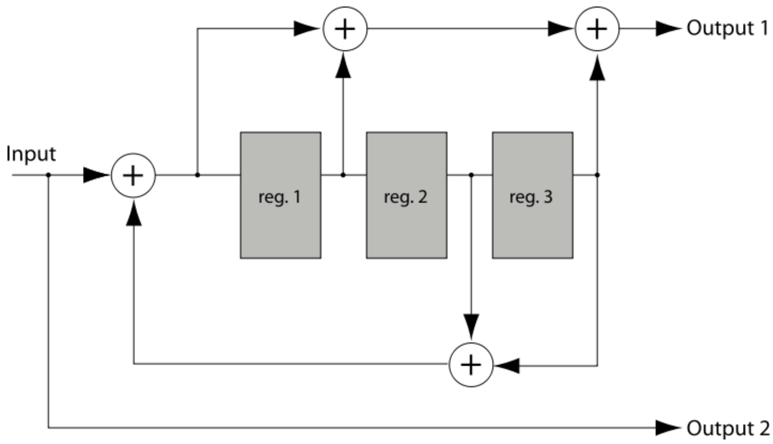


Figure 4.6: Logical Circuit of DASH7 Convolutional code

matrix of 128×256, which means that the rate of coding is $\frac{1}{2}$ and is constructed by *Mackay Neal* method and we have 128 redundant bit for 128 bit message.

Method of Mackay Neal LDPC is based on producing random columns of H matrix such that there will find no 2 columns or rows as vectors that have maximum in one place the same '1' common. This is because of preventing 4-cycles girth in proper graph The steps of algorithm is this: 1) Producing all zero matrix $M \times N$, 2) In all columns of H and some of rows there will be γ bit '1' 3) Become randomly the rows that consists any or just one bit of '1' to 2 bit '1'. 4) In each , $\rho = \frac{N_\gamma}{M}$, must be an integer number if ρ is not an integer, it can not be designed a regular code. 5) If the weight of *ith* row is greater than ρ randomly one of the bit '1' of this row is chose and it exchanges with another rows which have a smaller weight against ρ, if there is no row by this condition it repeats for the columns. 6) Preventing the 4-cycle path in proper graph.

There are also another methods which are just the subsection of this algorithm, which can be generated by producing 2 submatrices seperatley cunstructed by LDPC methods.

Girth of LDPC. In LDPC Parity Check matrices there is a parameter called *girth* of matrix, is just a parameter in LDPC codes which demonstrate the degree of ability of code some increasing in girth which progress the performance is basically to prevent 4-cycle in Tanner graph of code. One of the most important challenges in LDPC codes is *Cycle Decomposition*, 4-cycle (vertices of rectangle of bit '1') in H it can be heavily degraded the performance of code. Bipartite graph of code searches all of equations to be satisfied to be correct or not. Actually for preventing of these 4-cycles we transform each columns or rows of H by 2. This procedure is called *Cycle Decomposition* and we'll have an expanded code in which the cycle of parities check is lesser and it can increase the performance of code. This algorithm prevents these 4-cycles. We have used an *irregular* *matrix code with $\rho = 3$ & $\gamma = 6$ are the mean of rows, column's Hamming weight.* In practice Irregular LDPC codes are more useful because of their better performance.

Encoding LDPC. Compared with general linear block codes, LDPC encoding with *Lower Triangular Check Matrix* and *Approximate Lower Triangular Check Matrix* carry out encoding directly by parity check matrix H. There are two types of encoding by lower triangular check matrix structure. The first is to use the Gaussian elimination to convert the check matrix H into lower triangular matrix structure before encoding. The encoding complexity is $O(n^2)$, n is the column of check matrix. However, lower triangular check matrix produced in this method is not consistent with the sparse characteristics. The second is to directly use a given lower triangular sparse check matrix for encoding, which may result in loss of encoding performance.

Approximate lower triangular LDPC encoding was proposed by Richardson and Urbanke in 2001. The encoding is to disintegrate the check matrix H (as shown in Figure 4.7) into six (A, B, C, D, T, E) sparse sub-matrix before working out the redundant bit $p1$, $p2$ (Equations 4.2) according to the characteristics of the six sparse sub-matrix to complete the encoding. The encoding complexity is $O(n + g^2)$, g is the row of matrix E. Compared with the lower triangular encoding matrix, the complexity is lower and the encoding is consistent with the sparse characteristics; hence, the encoding performance is relatively higher. Coded message will be $c = [m, p1, p2]$.

FIGURE 4.7: Left : Lower Triangular Check Matrix , Right : Approximate Lower Triangular Check Matrix

$$\begin{cases} Am + Bp_1 + Tp_2 = 0 \\ (-ET^{-1}A + C)m + (-ET^{-1}B + D)p_1 = 0 \end{cases} \tag{4.2}$$

$$H = \begin{bmatrix} A & B & T \\ -ET^{-1}A + C & -ET^{-1}B + D & 0 \end{bmatrix} \tag{4.3}$$

Actually algorithm of Richardson-Urbanke has been designed for regular LDPC codes which have a Quasi-Cyclic, Block-Circulant which are more easy to encoding , i tested for irregular LDPC (Smaller Girth-Cycle, more Sparse Parity Check Matrix) and we gave a suitable performance. In Encoding We have registered 6 sub-matrices and it's done by a combination of these matrices to message without using a generator matrix (G)[7].

4.2.2.2 Algorithms of Decoding (Hard and Soft decision)

Decoding of Convolutional. Codes are done normally by a prestigious algorithm called *Viterbi*, which is used in any random Markov process, the idea is to find the shortest path that a binary stream could be coded. Shortest path is measured by a metric called *Euclidean Distance* which is in Galois binary Field ($GF(2)$) or can be in $\mathbb{R}$ numbers and will be produced Hard and Soft decision respectively. Imagine you have a n binary stream to decode, Viterbi algorithm starts from the first bit and calculate all possibilities of coded paths in each steps eliminates the paths that are larger value of metric, simply because of Maximum likelihood concept and at the end you will have your most probably path which have should be received.

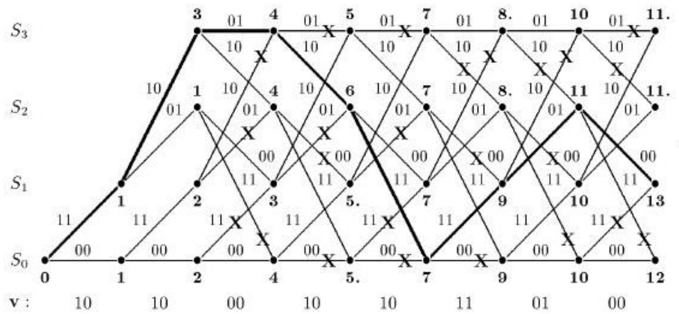


FIGURE 4.8: Viterbi decoder, $R = \frac{1}{2}, K = 3$, BSC channel

Normally Viterbi algorithm is done by *Trellis* graph (Figure 4.8), in which you can see the time effect and elimination of less probable paths on algorithm. If before of decision we apply received analogue signal to Viterbi somehow we will have an effectively non-binary Euclidean Metric result will be more correct which approximately gains 2dB in theory. In internship we didn't have access to inside of chips and the bases of chips get binary numbers and it can't get non-binary numbers.

Decoding of LDPC. Optimum (Maximum Likelihood) decoding of LDPC codes is in general not feasible for the reason of complexities. The algorithm of Viterbi is optimal while generally the algorithms of LDPC codes are suboptimal which are popular because of error probability performance. An LDPC coded can be decoded in various ways, namely: Majority-Logic (MLG), Bit Flipping (BF), Weighted BF (WBF), A Posteriori Probability (APP) and Iteratively Decoding based on Belief Propagation (IDBP) (commonly known as sum-product algorithm (SPA)). The first 2 types are hard decision, the last 2 are soft-decision decoding and the third one is in between. MLG decoding is the simplest one in decoding complexity. BF decoding requires a little more decoding complexity but gives better error performance than the MLG decoding. APP decoding and the SPA decoding provide musch better error performance but require much larger decoding complexity than the MLG and BF decodings. The weighted BF offers a good trade off between error performance and decoding complexity. SPA decoding gives the best error performance among the five types of decoding of LDPC codes and yet is partially implementable. APP decoding also provides the best error performance; however it is computationally intractable and hence will not be considered as a candidate of decoding algorithm[9]. These reasons make us to choose *Bit Flipping* (Figure 4.9) for hard decoding and *Log-Domain SPA* (an implementable version of SPA) proposed by [10] for soft decoding. We explain now these algorithms in more details

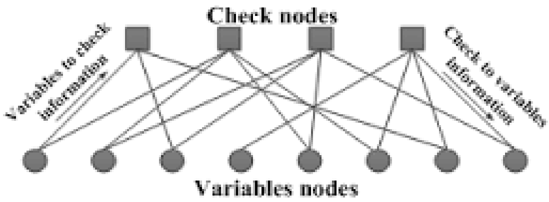


FIGURE 4.9: Bipartite Graph Demonstration for Decoding

Algorithm of *Bit Flipping* is done by this steps with Equations of 4.4 in which, z is the **Hard** decision , H parity check matrix and **s** syndrome of received signal. The steps of algorithm is below :

1) compute the parity check sums (syndrome bits) based on 4.4, if all the parity-check sums are zero, stop the decoding. 2) Find the number of failed parity check equations for each bit, denoted by $f_i, i = 0, 1, ..., n-1$. 3) Identify the set $\mathbb{S}$ of bits for which f_i is the largest. 4) Flip the bits in set $\mathbb{S}$. 5) Repeat steps 1 to 4 until all the parity check sums are zero (for this case, we stop the iteration in step 1) or a maximum number of iterations is reached.)

$$\begin{cases} \mathbf{s} = (s_1, s_2, ..., s_J) = z.\mathbf{H}^T \\ s_j = \mathbf{z}.\mathbf{h}_j = \sum_{l=0}^{n-1} z_l h_{j,l} \\ \mathbf{h}_j = (h_{j,0}, h_{j,1}, ..., h_{j,n-1}) \end{cases} \tag{4.4}$$

Soft decoding is done by *Log-Domain SPA* which is just a SPA using Log-Likelihood-Ratio (LLR- A posteriori probability) metric to measure the equations of normal-SPA and is done by these steps:

1) Initialization step: variable nodes are initialized with the belief of the corresponding variable, based solely on the received vector x. 2) Tentative decoding: variable node n computes, based on all the information it has available (i.e from the channel vector x

and messages from adjacent check nodes), the most likely value of the variable n, c_n. If the decoded nodes satisfies all checks ($\mathbf{H} \textbf{ . } \mathbf{c} = 0$), the decoding algorithm is halted. 3) Horizontal step: a message denoted is passed from variable node n to check node m. expressing the belief of the nth variable, given all the information from all connected check nodes, except check node m itself. 4) Vertical step: each check node m sends a message to adjacent variable node n, reflecting the belief of the nth given all the information from the channel and all variable nodes connected to check node m, except variable n itself. Go to step (2). Effectively in this procedure z of first equations 4.4 is the received signal before applying hard decision [9] [11].

4.2.3 Cyclic Redundancy Check Codes (CRC)

This family of Error Control Codes belongs to the family of *Detecting* error control codes actually there is a lot of kinds of this family like : CRC-8, CRC-16, CRC-32, CRC-64, Checksums algorithms and ... . Actually DASH7 uses CRC-16 ITT of this family and actually in data linke layer when a packet comes down, this layer is going to encapsulate this packet and make a frame by this way that it adds a 'header' at first of packet and a *CRC* part at the end of frame which contains 16 bits. In the receiver frame is checked and if the packet is not a good version it will send the No acknowledgement to sender to retransmit it again. Mathematical equation is given in 4.5:

$$g(x) = x^{16} + x^{12} + x^5 + 1 \tag{4.5}$$

As for note, Frame/Block Error Rate is a criteria measuring in data link layer of network, it means that if you want to measure your quality of service in this layer, theoretically because the bits have been framed so the smallest piece is the *frame*, by checking the error occurred inside frame you can define your criteria against energy which is Frame Error Rate. in Figure 4.10 you can find final coding Frame Error Rate of payload packet (*LDPC + CRC*).

4.3 Header Packet Coding

As it is mentioned in Table 2.1 from Chapter 2 we will apply a special channel coding on head of packet, so in parallel and we will propose 2 channel coding, generally channel codes are designed for detecting and correcting bits.

In DASH7 header consists of 28 bits which are some datas about the packet (128 bits payloads) which comes from upper layer as it has been well explained and in this way if

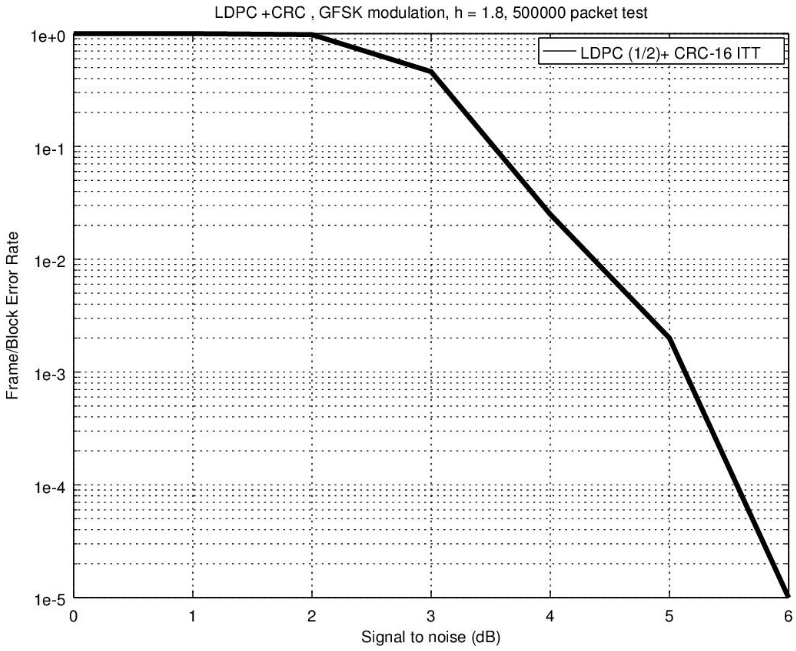


FIGURE 4.10: LDPC(256,128) + CRC16-ITT Frame Error Rate

we would design a code which could well correctly decode in receiver we will Header of packet and we will sure about header information.

4.3.1 BLER Curves of Reed-Solomon and LDPC

The point is in (BER vs SNR) figures the gradient of LDPC codes are fast and they are very near to their Shannon limit but also RS codes give good performance and fast slopes we can design a RS code which for a fixed arbitrary error probability because the gain of coding is a function of Bit/Block Error Probability. Figure 4.11 shows an RS code which satisfies this condition.

4.3.2 Reed-Solomon encoding and decoding algorithms

RS codes are one of the most important codes, extended of BCH codes in which the coefficient of generator polynomials can be in $GF(q)$ and in general form they have Equation 4.6 in which t is the number of correctable bit in code, α^i is the roots of extended binary Galois Field, $GF(2^m)$ which are made up by primitive polynomials. In this case $m = 4$ which codes 7 bit to 15 bit so the rate of code is $\frac{7}{15}$ and in desired region works we want. RS codes can achieve the maximum value of d_{min} (minimum Hamming distance in code words) which is $n - k + 1 = 2t - 1$ in which n is the length of number coded, k is the length of message, t is the number of ability of correction errors. This

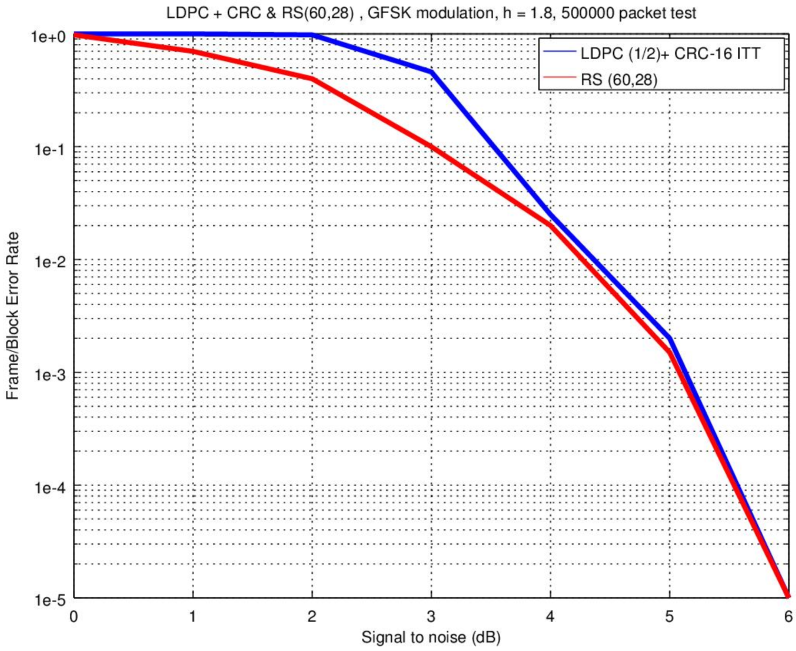


FIGURE 4.11: BLER of (LDPC(256,128) + CRC-16ITT) and RS(60,28)

code is very useful in the 'Burst' Errors which is the case of fading channels the other hand some applications like IEEE 802.16 Broadband Wireless communication they use a combination of this codes with convolutional codes with block of Interleaving and will give very good performances.

RS Coding We can use algebric appearances of these codes and multiplication message into generator polynomial, coded message will be obtained. Algebric mathematical method permit to write these codes in form of Equation 4.7

RS Decoding Main algorithms of RS decoding are *Error Trapping & Berlekamp-Massey*. Berlekamp-Massey uses some matrix calculation and so it will have heavily calculation of matrix. Error Trapping is a good algorithm specially in practice because it's very compatible with shift Registers and steps of algorithms are done by registers. The idea is division polynomials until the Hamming weight of residual would be equal to the number of correctable bits (t)[12].

An example can be RS(7,3;5) ($n = 7, k = 3, d_{min} = 5$) with generator polynomial of $g(x) = 1 + \alpha^4 x + \alpha^2 x^2 + \alpha^4 x^3 + x^4$ in $GF(2^3) = \{0, 1, \alpha, \alpha^2, \alpha^3 = \alpha + 1, \alpha^4 = \alpha^2 + \alpha, \alpha^5 = \alpha^2 + \alpha + 1, \alpha^6 = \alpha^2 + 1\}$ in basis polynomial of $p(x) = 1 + x + x^3$ which is seen $e(x) = x^3 + x^2$. Steps of error trapping is shown which is used in decoding algorithm ,below for this example:

$v(x)$	1	1	1	1	1	1	1
$e(x)$	0	0	1	1	0	0	0
$r(x)$	1	1	0	0	1	1	1
$g(x)$	1	α^4	α^2	1	0	1	α^5
	0	α^5	α^2	α^4	0	1	1
$\alpha^5 g(x)$		α^5	α^2	1	α^2	α^5	
	0	0	0	α^5	α^2	α^4	1
$\alpha^5 g(x)$	α^5			α^5	α^2	1	α^2
	α^5	0	0	0	0 α^5	α^6	
$\alpha^5 g(x)$	1	α^2	α^5	0	0	α^5	α^2
	α^4	α^2	α^5	0	0	0	1
$g(x)$	α^4	α^2	α^4	1	0	0	1
	0	0	1	1	0	0	0

- In each steps buffer shifts generator polynomial circulant (i) by α^i multiplication to get the desired result.

$$g(x) = \prod_{i=1}^{2t} (x - \alpha^i) \tag{4.6}$$

$$r(x) = m(x)g(x) + e(x) \tag{4.7}$$

Other algorithm, 'Berlekamp-Massey' is a useful computer implementable algorithm by follow: 1)Calculate of syndromes vectors $S_1, S_1, ..., S_{2t}$ by equation of 4.8.

2) Initializing algorithm by $k = 0$, $\Lambda^{(0)}(x) = 1, L = 0, T(x) = x$

3)$k = k + 1$,

4) If $\Lambda^k = 0$, jump to (8).

5) Update error place pointer polynomial by equation 4.10.

6)If $2L \geqslant k$, jump to (8).

7)$L = k - L$, calculate $T(x)$ from equation 4.11

8) $T(x) = xT(x)$.

9) if $k < 2t$, jump to (3) 10)roots of $\Lambda(x)$ polynomial are the placement of errors [12].

$$S_i = r(\alpha^i) \tag{4.8}$$

$$\Delta^k = S_k - \sum_{i=1}^{L} \Lambda_i^{k-1} S_{k-i} \tag{4.9}$$

$$\Lambda^k(x) = \Lambda^{k-1}(x) - \Lambda^k T(x) \tag{4.10}$$

$$T(x) = \frac{\Lambda^{k-1}(x)}{\Lambda^k} \tag{4.11}$$

This code can be used in parallel as coding of *Header* to have good performance and to be sure to have ability of correcting header because the information in header determines the length of upper given packets. and by applying LDPC codes on payload of frame we will have a good performance for Error Controlling of DASH7 protocol.

Chapter 5

Conclusions

Always in every Engineering systems there are trade off which Communication Systems also is not an exception of this history, each day we are going to develop our ideas about something maybe a great technological idea like IoT, SDN, ... Actually when you are an engineer that you understood when it should use a clever way to solve your problem. 'genius' is to apply a science in the real world is just a combination of art and science and it's beautiful.

To be a good engineer it should learn all of tools that are going to be useful and they define your power to progress your work without problems. Some days you don't get a good result, it's a good news because you got a great experience to not repeat it and it must always try to find the answers and suitable results. So it seems that you should have a project and you must define a goal for your project.

Network protocols are completely different by several reasons and visions so depends on your system your designing must be varied. In Wireless Sensor Networks we are limited in some constraints who are mainly definitive for our objects, control of objects from go away is now our ideas because the Thing that connect our real world to Computer, Internet or generally virtual worlds are *Sensors*

Channel coding is one of the most important in current days of communication because we have different systems in which the same codes don't work so effectively it's your systems who determines and declares the exact demands. LDPC codes are too useful in these days of telecommunication engineering there many place in which we are going to use it or it has been implemented like 5G, Wimax, Ethernet, ...

Appendix A

Recommendations and Next Steps

Wizzilab : Young group in Paris who has some passionate developers and engineers, they have beautiful new ideas for realization in Wireless Network Sensors application.

DASH7 : New version of DASH7 (version 1.0) with some recommended channel Bandwith and Specifications and New Channel Codings

Signal Processing Projects : Signal Processing with MATLAB/Octave is just a realization of mathematics in your real world, try to understand the concepts with 2 aspects in parallel, philosophy and applicable, this report has produced a good opportunity for Telecommunication students who want to have a beautiful and relative understanding of some applied mathematics tricks and its applications in Signal Processing and Digital Communication.

Bibliography

[1] Z. Yafeng, Cao Zhigang, M.Zhengxin *Modulation Index Estimation for CPFSK Signals and Its Application to Timing Synchronization*The state key Laboratory on Microwave and Digital Communications, China, 2004

[2] John G.Proakis, Masoud Salehi *Digital Communication.* (fifth Edition), 2008

[3] B. Yu, Student Member IEEE, L. Yang, Senior Member, IEEE, and Chia-Chin Chong, Senior Member, IEEE *Optimized Differential GFSK Demodulator*, The state key Laboratory on Microwave and Digital Communications, China, 2011

[4] R.Schiphorst, F. Hoeksema, K. Slump *Bluetooth demodulation algorithms and their performance.* University of Twente, Dpartment of electrical engineering, Laboratory of Signals and Systems, The Netherland, 2000.

[5] N.P Kontakos and J.K Pollard *Bluetooth RF Layer Performance Evaluation* University College London, 2001

[6] S. Nagaraj, S. Khan a,d C. Schlegel, Departement of Electrical Engineering University of Alberta, *On Preamble Detection in Packet-Based Wireless Networks*, 2006

[7] Thomas J. Richardson and Rüdiger L. Urbanke *Efficient Encoding of Low-Density Parity-Check Codes*, IEEE Transactions of Information Theory, 2001.

[8] Henrik Asplund, Kjell Larsson and Peter Ökvist *How typical is the "Typical Urban" channel model.* Ericsson Research

[9] Shu Lin, Daniel J.Costello, Jr. *Error Control Coding.* (Second Edition), 2004.

[10] Henk Wymeersch, Heidi Steendam and Marc Moeneclaey *Log-domain decoding of LDPC codes overGF(q).* DIGCOM research group, TELIN Dept., Ghent University, 2004.

[11] Y. Kou, Student member of IEEE, S. Lin, M.P.C. Frossorier, Seniors of IEEE *Low-Density Parity-Check Codes Based on Finite Geometries: A Rediscovery and New Results*, 2001

[12] Bo Zhou, Li Zhang, Jingyu Kang, Qin Huang, Ying Y. Tai and Shu Lin, Department of Electrical and Computer Engineering University of California, Davis *Non-Binary LDPC Codes vs. Reed-Solomon Codes*, The state key Laboratory on Microwave and Digital Communications, China, 2011