A real-time COFDM transmission system based on the GNU Radio – USRP N210 platform

Dien Nguyen Van
School of Electronics and Telecommunications
Hanoi University of Science and Technology
Hanoi, Vietnam
dien.nguyenvan.hut@gmail.com

Tuan Nguyen Anh
School of Electronics and Telecommunications
Hanoi University of Science and Technology
Hanoi, Vietnam
anhtuan245@gmail.com

Tu Tran Ngoc
School of Electronics and Telecommunications
Hanoi University of Science and Technology
Hanoi, Vietnam
tutn1990@gmail.com

Van Duc Nguyen
School of Electronics and Telecommunications
Hanoi University of Science and Technology
Hanoi, Vietnam
duc.nguyenvan1@hust.vn

Byeungwoo Jeon
School of Information and Communication Engineering
Sungkyunkwan University
Suwon, Korea
bjeon@skku.edu

Tien Hoa Nguyen
School of Electronics and Telecommunications
Hanoi University of Science and Technology
Hanoi,Vietnam
hoa.nguyentien@hust.vn

ABSTRACT
The term “Software Defined Radio” (SDR) has become familiar in research and development of wireless communication systems today. SDR allows users to adjust a system with its flexibility and re-configurability for any frequency band and different modulation of various physical parameters by using programmable hardware and software. In this paper, we implement an Orthogonal Frequency Division Multiplexing (OFDM) system combined with channel coding blocks based on Software-Defined Radio. The software design and implementation are proposed for a GNU Radio-based OFDM system in real-time wireless transmission. Different scenarios with and without channel coding have been implemented in order to evaluate the Packet Failure Rate (PFR) performance. Quadrature amplitude modulation QAM-16, convolutional encoder combined with block interleaver and Viterbi decoder are deployed for the implementation. It is shown that the introduction of channel coding blocks into GNU Radio-based OFDM system reduced the PFR of data signals transmitted over Universal Software Radio Peripheral (USRP) boards.

Categories and Subject Descriptors
C.2.1 **[Network architecture design]** Wireless communication

IMCOM (ICUIMC)'14, January 9–11, 2014, Siem Reap, Cambodia.

General Terms
Experimentation, Measurement and Performance.

Keywords
Channel coding, GNU Radio, USRP N210, Software-Defined Radio.

1. INTRODUCTION
The term “Software-Defined Radio” (SDR) [1] was first introduced by Joseph Mitola in 1992. Software-Defined Radio is a radio communication technology that is currently gaining its world-wide popularity. The revolutionizing idea behind the term SDR is the elimination of the conventional dependent on hardware, instead it takes advantage of software to manipulate radio signals. In general, the implementation of the radio signal will be processed in the digital domain instead of analog domain. It is possible for these programmable hardware modules to be utilized in different digital radio systems such as modulation, demodulation, channel coding in software. Given the re-configurability and flexibility of SDR, a typical communication system could be easily deployed and implemented, in which, the hardware part will be taken care by USRP boards, while the open-source software toolkit GNU Radio in a commodity computer or an embedded system will be accountable for handling the software part. SDR technology assists in providing a variety of communication systems and components.

OFDM is a frequency-division multiplexing (FDM) scheme used as a multi-carrier modulation method, which is commonly implemented in many emerging communication systems. The basic principle of OFDM is to split a high-rate data stream into a number of lower rate streams that are transmitted simultaneously over different subcarriers. These subcarriers are overlapping and orthogonal. OFDM provides several advantages over the traditional FDM. OFDM allows for greater spectral efficiency, reduced intersymbol interference (ISI), and resilience to multipath distortion. For the existing OFDM system built in GNU Radio toolkit, an enhancement of this system has been achieved by introducing channel coding blocks. It is universally acknowledged that channel coding plays a very important role in wireless communications systems. At the transmitter, it helps to encode the information into radio waves so that the receiver can recover the information reliably. Forward Error Correction (FEC) technique has been utilized for data transmission in this paper. Redundant data is added to the sent messages at the transmitter, which allows the receiver to detect and correct errors without asking the transmitter for retransmission.

In this paper, an improvement in the processes of transmitting and receiving signals between two USRP boards utilizing the open-source toolkit GNU Radio has been made by the introduction of convolutional encoder coupled with block interleaver and Viterbi decoder into the GNU Radio built-in OFDM system. By making some changes in configurable files of the GNU Radio-based OFDM system, it is possible for us to apply the channel coding theory in order to reduce errors during transmission. An improvement in the overall system performance by reducing PFR is expected to be made.

This paper is structured as follows. In section II, GNU Radio and USRP platform are introduced, section III presents the implementation of the COFDM system, section IV explains the technical setup and description of the system, section V concludes the paper followed by references.

2. GNU RADIO AND USRP PLATFORM

GNU Radio [2] is a free open-source software toolkit which provides signal processing parts to implement SDRs. It can be used in conjunction with a wide range of radio frequency hardware to quickly build actual implementations. The processing blocks are implemented in C++ and GNU Radio applications are written in Python scripting language to link the C++ signal processing blocks together. These C++ blocks can be linked to Python applications by using SWIG as an interface between two languages. In particular, GNU Radio provides a signal processing runtime environment which, combined with (low-cost) external RF hardware (USRP), allows users to deploy different SDR applications over wireless medium. The version of GNU Radio has been used in this paper is 3.6.3.

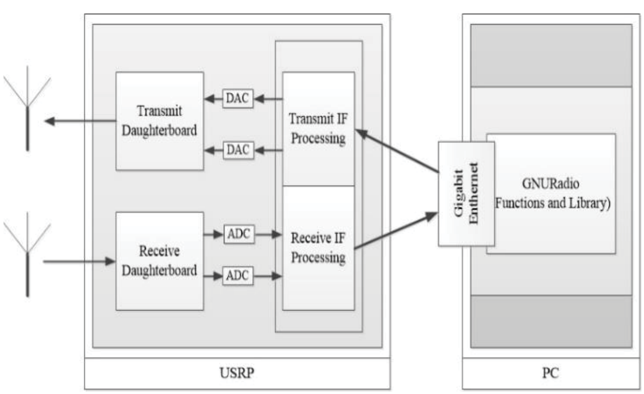

Figure 1. SDR Architecture

GNU Radio [2] is a free open-source software toolkit which provides signal processing parts to implement SDRs. It can be used in conjunction with a wide range of radio frequency hardware to quickly build actual implementations. The processing blocks are implemented in C++ and GNU Radio applications are written in Python scripting language to link the C++ signal processing blocks together. These C++ blocks can be linked to Python applications by using SWIG as an interface between two languages. In particular, GNU Radio provides a signal processing runtime environment which, combined with (low-cost) external RF hardware (USRP), allows users to deploy different SDR applications over wireless medium. The version of GNU Radio has been used in this paper is 3.6.3.

GNU Radio only deals with the software part of SDR, the hardware part for SDRs is developed on USRP. There are different USRP versions designed by Ettus Research [3] based on what a user needs. The remarkable advantage of using USRP is that it does not make any assumption about the modulation which will be implemented in it. However, it also leads to a major drawback is that all practical problems will have to be solved in software. A USRP only needs few parameters to work properly: a central frequency (which should be in the supported frequency range of its daughterboard), a sampling rate (which will implicitly define the usable bandwidth) and a gain. Other parameters may be available depending on daughterboards and our implementation purposes. For our experimentation, two USRP2 are used along with RFX2400 RF daughterboards [4], GNU Radio 3.6.3 and UHD (USRP Hardware Driver) 3.5.0 on a Linux Ubuntu 10.04 operating system. The USRP N210 includes a Xilinx Spartan-3A-DSP 3400 FPGA, 14 bits 100 Mega Samples Per Second (MSPS) dual ADC, 16 bits 400 MSPS dual DAC, and a GigaBit-Ethernet port. A modular design allows the USRP N210 to operate from DC to 6 GHz, while an expansion port allows multiple USRP N210 series devices to be synchronized. The RFX2400 is a high-performance, full duplex transceiver designed specifically for operation in the 2.4 GHz band. The daughterboard has a SAW filter in series with the TX/RX port to provide superior selectivity and spurious performance between 2.4 and 2.483 GHz.

Table 1. Overview of development system

Component	Type
Operating system	Ubuntu 10.04 LTS, 32 bit
GNU Radio	Version 3.6.3
USRP	Ettus Research N210
Daughterboard	RFX 2400
Antenna	VERT 2400
Carrier-frequency	2.45 GHz
Python version	2.6.3

3. COFDM

3.1 Convolution encoding and Viterbi decoding

Convolutional code [5] is a type of error-correcting code in which redundancy is added to original bit sequence. By adding additional bits, it makes bit error checking more successful and allow for more accurate transmission, therefore, increase the reliability of the communication. The purpose of a convolutional encoder is to take a stream of input bits and generate a matrix of encoded outputs. The convolution algorithm implemented in this paper is a convolutional encoder with rate 1/2, constrain length of K =3 and having generator $[7,5]_{8}$, which is suitable for transmitting data.

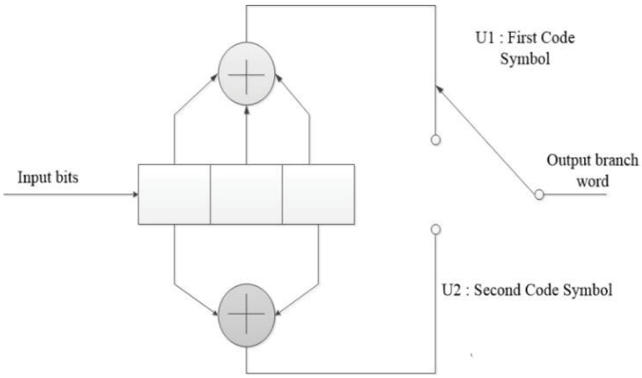

Figure 2. Convolution encoder

The relationship between the input and output bits and the state transition is illustrated in Table II.

Table 2. Convolution encoder

Current state	If input = 0	If input =1
	Next state (output)	Next state (output)
00	00(00)	10(11)
01	00(11)	10(00)
10	01(10)	11(01)
11	01(01)	11(10)

At the transmitter, the message is encoded bit by bit. After the last bit of the message passes, two flushing bits 0 are inserted to make sure the last transition state is 00. Its aim is to facilitate the decoding process of the receiver.

This paper describes the Viterbi algorithm [5] for the mentioned Convolutional encoder. First and foremost, the trellis diagram is constructed for the implementation of Viterbi diagram. Each state or output value is defined by 2 coded bits, hence the correct state could be found from 4 possible combinations. As shown in Table 1, any state can be reached from only 2 possible previous states, however, only one state is valid. At the receiver, assume that the total coded bits received are N, two coded bits are taken at a time for processing and computing *hamming distance, branch metric, path metric*.

After completing the trellis diagram, the next step is to find the *survivor path*, this can be done by following manner: the decoding process starts from the last time instance to the first time instance of the trellis diagram. At the last time instance, the state of the trellis diagram is always 00 (because of two flushing bits 00 inserted at the end of the transmitted message). Given the transition Table II and compare branch metrics from two previous states, we are able choose the next state during the decoding process at the receiver. This manner is continuously performed until the first state of trellis diagram is reached.

3.2 Block interleaver

In the modulation scheme QAM16, each I and Q symbol consists of 4 data bits. If the number of errors within a symbol exceeds the error-correcting capability of Viterbi decoder, then the process of recovering the original symbol is not feasible. There comes the need of constructing a block interleaver before the convolution encoder to avoid possible burst errors. The type of the block interleaver deployed in this paper is matrix interleaver.

According to [6], the decoding constraint length heavily depends on the characteristic of the code and can be derived from the code constraint length. For the code rate R=1/2, the satisfactory performance of BER is achieved if the decoding constrain length is approximately 5 times greater than the code constrain length which, in the used Viterbi decoder, is 3. Therefore, the matrix interleaver at the transmitter is designed with 4 rows and 4 columns, that is, the decoding constraint length at the receiver is 16, which is approximately 5 times greater than 3. The block interleaving is achieved by filling a matrix with the input symbols row by row, then sending the matrix contents to the output column by column. The deinterleaver takes the coded symbols into a matrix. The size of this matrix is equal to that of the interleaver matrix. However, the symbols are written column by column and then read out row by row, which can effectively preclude the possibility of misreading the received symbols at the receiver. The block diagrams of the OFDM transmitter and receiver, combined with channel coding blocks, are shown in Fig.3 and Fig.4. In comparison with the built-in GNU Radio-based OFDM system, the new channel coding blocks such as interleaver, convolutional encoder, deinterleaver and Viterbi decoder has been introduced in order to enhance the performance of the overall system.

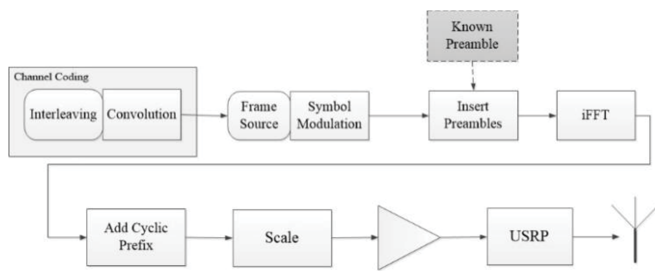

Figure 3. The transmitter of OFDM system using channel coding

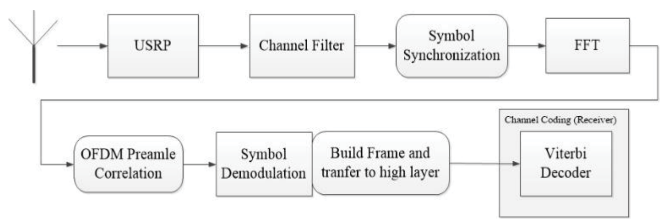

Figure 4. The receiver of OFDM system using channel coding

4. TECHNICAL DESCRIPTION AND IMPLEMENTATION

Two experiments have been conducted and the performance in terms of the average number of failed packets and PFR have been analyzed and reported in this paper. Two USRP N210 boards, two laptops were used to implement these experiments. The daughterboard used for these experiments is RFX 2400 whose frequency range is from 2.3GHz to 2.9GHz. Our experiments were conducted with narrowband B = 500 kHz, central frequency of 2.45 GHz. Experiments were conducted indoor, the distance between two USRP N210 boards is 1 meter.

Some GNU Radio configurable script files written in python have been used are *benchmark_tx.py*, *benchmark_rx.py* in *gnuradio/gr-digital/examples/ofdm* and *ofdm_utils_packets* in *usr/local/lib/python2.6/dist-packages/gnuradio/digital/*.

Figure 5. GNU Radio Architecture test bed

File *benchmark_tx.py* generates packets and frames in the format [8] as shown in Fig. 6. The size of the data field in the packet is 2 bytes. The software running in the PC generates the packets and frames and passed them through Gigabit Ethernet connection to USRP N210. At the receiver side, file *benchmark_rx.py* is accountable for establishing a proper configuration of the USRP in order to receive the transmitted packets.

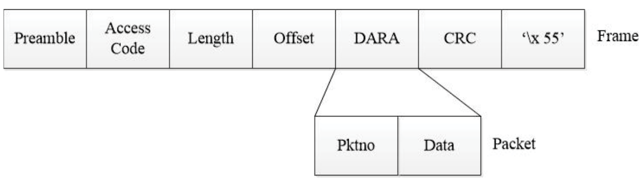

Figure 6. The relationship between the packet and the link frame

The bit rate of the system is calculated by the following equation:

$$R = r\frac{N_U M f_S}{N_{SC} + N_G} \quad (1)$$

Where r is the code rate of FEC, N_U is the number of subcarriers used for data transmission, M is the number of bits per subcarrier, f_s is the sampling rate (at the baseband frequency $f_s = B$), N_{SC} is the number of subcarriers, N_G is the length of cyclic prefix in samples.

In the modulation scheme QAM-16 with 1024 FFT bins, the bit rate that can be calculated by using equation (1) is 400 kbps.

The goal of the experiments is to evaluate and compare PFR performances before and after adding convolution encoding blocks to the OFDM system. In these experiments, PFR performance is measured in different modulation scenarios: QAM-16 with variation in the number of FFT bins. The length of the guard interval in these experiments is equal to one fourth of the FFT length.

5. EXPRIMENTAL RESULTS

The test environment in which experiments has been implemented was two pre-described USRP N210 boards with daughterboards RFX 2400. The experiments were set up at a central frequency of 2.45GHz, bandwidth of 500 kHz.

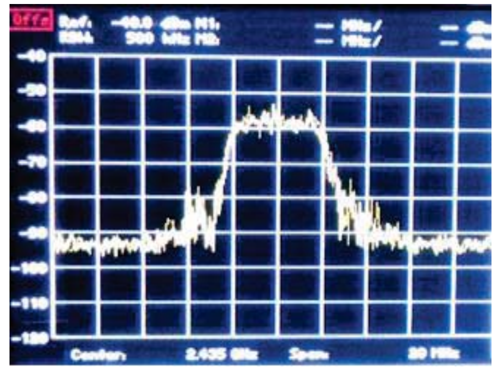

Figure 7. The spectrum of transmitted signal

Regarding the PFR performance of the proposed system, two experiments have been carried out with the variation in the number of FFT bins: 256,512 and 1024. Fig. 6 shows the analytical PFR performance of the system with and without channel coding.

The analytical PFR performance for OFDM system is calculated using the following equation:

$$PFR = \frac{Average\ number\ of\ failed\ packets}{Total\ number\ of\ received\ packets} \text{ X } 100\% \quad (2)$$

The first experiment is done using OFDM system without channel coding. The graph of the number of received packets versus FFT length is shown in Fig. 8. The graph shows that the number of received packets decreases dramatically as the FFT length decreases from 1024 to 256.

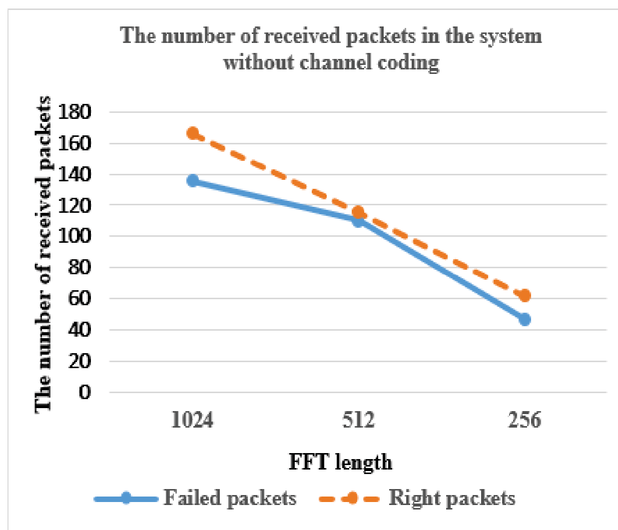

Figure 8. The number of received packets in the system without channel coding

The second experiment aims to illustrate the number of received packets in the OFDM system with channel coding. The graph of the number of received packets versus FFT length is shown in Fig. 9. It is clear that there are an increase in the number of right received packets and a decrease in the failed received packets in comparison with the previous experiment.

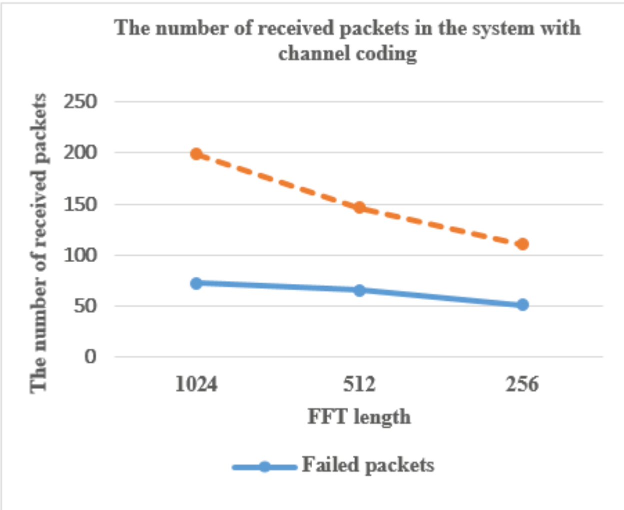

Figure 9. The number of received packets in the system with channel coding

As shown in Fig. 10, the improvement of PFR performance are compared among different modulation schemes with and without channel coding. The PFR improvement declined gradually with respect to the decrease in the number of FFT length. It is obvious that the modulation scheme with 1024 FFT bins has the best PFR performance. The PFR has been reduced by 40.5% in this scheme, while the PFR in the scheme with 256 bins has been lowered the least by 27.1%. This could be explained by the fact that the increase in the FFT length leads to the decrease in the transmission speed of each subcarrier. The synchronization method of the receiver is suitable for low transmission speed. Therefore, with a bigger FFT length, the receiver is capable of receiving and correcting more packets.

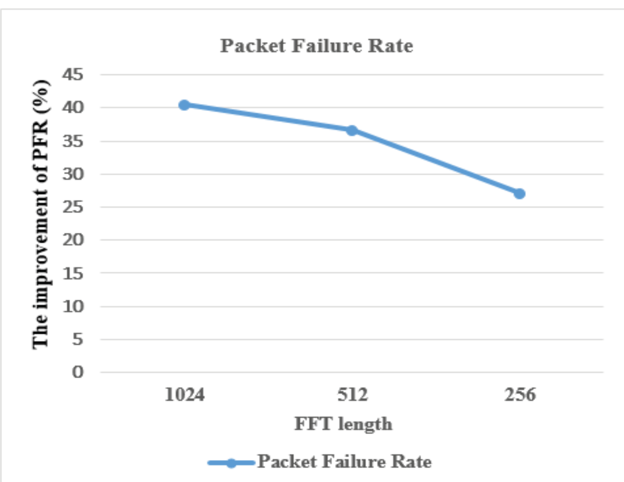

Figure 10. The improvement of PFR performance with channel coding

From the above conducted experiments, we can see that the number of received packets is relatively low. This could be explained by the fact that the number of received packets is relatively proportional to the DAC values of the USRP boards [10]. In this paper, the default configure of DAC value of USRP boards was used, so this could result in higher ratio of packet loss. As a consequence, the higher packet loss could potentially lead to the high packet failure ratio at the receiver.

6. CONCLUSION

In this paper, the implementation of a real-time OFDM system with channel coding using GNU Radio and USRP is described. The channel coding algorithm is implemented and tested in real-time transmission between two USRP boards for experimentation. The version of USRP boards were used in this paper is *Ettus USRP N210* and we did not make any change in the firmware of the Field-Programmable Gate Array (FPGA). Our analysis and measurement indicated the improvement of Packet Failed Ratio (PFR) by introducing channel coding to a built-in GNU Radio-based OFDM system. Also, the results has showed that an OFDM system with a bigger FFT length gives a better performance in terms of Packet Failure Rate (PFR) in comparison with other OFDM systems with smaller FFT length.

7. ACKNOWLEDGEMENT

This research is funded by the Vietnam Ministry of Science and Technology (MOST) under the grant number KC01.05/11-15.

8. REFERENCES

[1] J.Mitola, Software radios-survey, critical evaluation and future directions, *IEEE National Telesystems Conference*, April 1993, published in Aerospace and Electronic Systems Magazine, IEEE.

[2] Official GNU software radio website http://gnuradio.org/redmine/projects/gnuradio/wiki

[3] Ettus Research LLC website http://www.ettus.com

[4] Daughterboard RFX2400 website http://www.ettus.com/product/details/RFX2400.

[5] John G. Proakis and Masoud Salehi, *Digital Communications 5^{th} edition*, McGraw-Hill, 2007.

[6] Clark, G.C; Cain, J.B. *Error-Correction Coding for Digital Communication*, New York, Plenum Press, 1981

[7] Bastian Bloessl, Michele Segata, Christoph Sommer and Falko Dressler - An IEEE 802.11 a/g/p OFDM Receiver for GNU Radio, In *Proceedings of ACM SIGCOMM 2013, 2^{nd} ACM SIGCOMM Workshop of Software Radio Implementation Forum (SRIF 2013)*, Hong Kong, China, August, 2013.

[8] Alalelddin Mohammed – "*Studying Media Access and Control Protocols*", Master of Science Thesis.

[9] Van Duc Nguyen, Hans-Peter Kuchenbecker, Block interleaving for soft decision Viterbi decoding in OFDM systems, In *Proc. 54^{th} IEEE Semiannual Vehicular Technology Conference, IEEE VTC2001-Fall*, Atlantic City, USA, 7. -11. Oct. 2001, pp. 470-474.4.

[10] Arief Marwanto, Mohd Adib Sarijari, Norsheila Fisal,Sharifah Kamilah Syed Yusof, Rozeha A.Rashid Experimental Study of OFDM Implementation Utilizing GNU Radio and USRP - SDR, In *Proceedings of the 2009 IEEE 9th Malaysia International Conference on Communications 15 -17 December 2009,* Kuala Lumpur, Malaysia