Proceedings of the 2009 IEEE 9th Malaysia International Conference on Communications
15 -17 December 2009 Kuala Lumpur Malaysia

Experimental Study of OFDM Implementation Utilizing GNU Radio and USRP - SDR

Arief Marwanto, Mohd Adib Sarijari, Norsheila Fisal, Sharifah Kamilah Syed Yusof, Rozeha A.Rashid
Telematic Research Group, Fakulti Kejuruteraan Elektrik
Universiti Teknologi Malaysia
Skudai, Johor, Malaysia.
efmarwanto@gmail.com
{adib_sairi|sheila|kamilah|rozeha}@fke.utm.my
ari

***Abstract-* A Software Defined Radio (SDR) is a radio which can tune to any frequency band, transmit and receive different modulations and different physical parameters across a large frequency spectrum by using a programmable hardware and powerful software. An SDR performs significant amounts of signal processing in a general purpose computer, or a reconfigurable piece of digital electronics or the combination of both. In this paper, we seek to explore the viability of using GNU Radio; an open source SDR implementation and the Universal Software Radio Peripheral (USRP); an SDR hardware platform, to transmit and receive the OFDM radio signal with QPSK and BPSK modulation. Quality of Service (QoS) in terms of Packet Received Ratio (PRR) on the data transmitted will then be investigated and analyzed.**

***Keywords*— Software Defined Radio, GNU Radio, USRP, OFDM, QoS and PRR.**

I. INTRODUCTION

The term "Software Defined Radio" (SDR) was introduced by Joseph Mitola from MITRE Corporation in 1991. His first paper on SDR [1] was published in 1992 at IEEE National Telesystems Conference. Though the concept was first proposed in 1991, software-defined radios have their origins in the defense sector since the late 1970's in both the U.S. and Europe. One of the first public software radio initiatives was a U.S. military project named SpeakEasy [2]. Examples of radio terminals that require support include hand-held, vehicular, airborne and dismounted radios, as well as base-stations; fixed and maritime.

Software Defined Radio (SDR) is where all the signal manipulations and processing works in radio communication are done in software instead of hardware. So, in SDR, signal will be processed in digital domain instead in analog domain as in the conventional radio. The digitization work will be done by a device called the Analog to Digital Converter (ADC). Fig.1 shows the concept of Software Defined Radio. This figure shows that the ADC process is taking place after the Front End (FE) circuit. FE is used to down convert the signal to the lower frequency called an Intermediate Frequency (IF); this is needed due to the limitation of the speed of current Commercial of The Shelf (COTS) ADC. The ADC will digitize signal and pass it to the baseband processor for further processes; demodulation, channel coding, source coding and etc. In conventional radio, all this processes are done in hardware.

In general, Software Defined Radio (SDR) is defined as a software based communication platform which characteristics can be reconfigured and modified to perform different functions at different times.

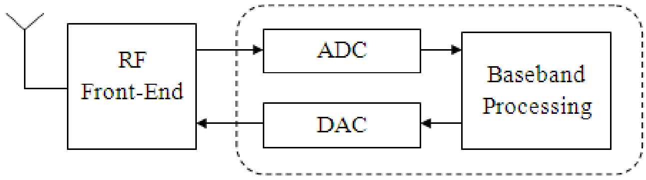


Fig. 1. Software Defined Radio Block Diagram

II. GNU RADIO AND USRP

Universal Software Radio Peripheral (USRP) is a flexible low-cost platform for SDRs developed by Matt Ettus [3]. USRP consists of two main boards; the daughter board and the mother board. The mother board consists of four 12-bit Analog to Digital Converter (ADC) with sampling rate up to 64MS/s, four 14-bit Digital to Analog Converter (DAC) with speed up to 128MS/s, two Digital up Converter (DUC) to up convert the baseband signal to 128MS/s before translating them to the selected output frequency, a programmable USB 2.0 controller for communication between USRP and GNU Radio and an FPGA for implementing four Digital Down Converter (DDC) and high rate signal processing. The daughterboard is acting as the RF front-end of the SDR. There are four slots on the motherboard which are used to connect the daughter boards with the mother board. Two of the four slots, labeled TXA and TXB, are meant for the TX daughterboard while another two, RXA and RXB, are for the RX daughterboard.

GNU Radio is an open source software toolkit which consists of signal processing blocks library and the glue to tie these blocks together for building and deploying SDRs [4]. The signal processing blocks are written in C++ while python is used as a scripting language to tie the blocks together to form the flow graph. Simplified Wrapper and Interface Generator (SWIG) is used as the interface compiler which allows the integration between C++ and Python language. Fig.2 shows the structure of GNU Radio and USRP SDR.

The USRP will digitize the inflow data from the air and passing it to the GNU Radio through the USB interface. GNU Radio will then further process the signal by demodulating and filtering until the signal is translated to a packet or a stream of data.

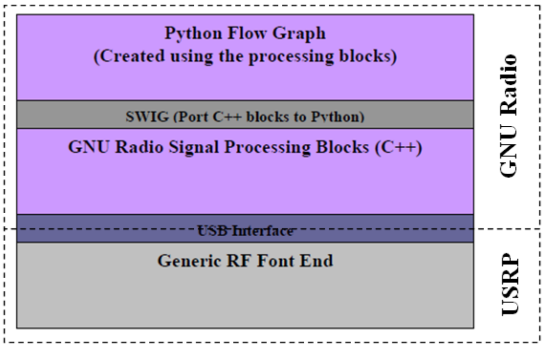
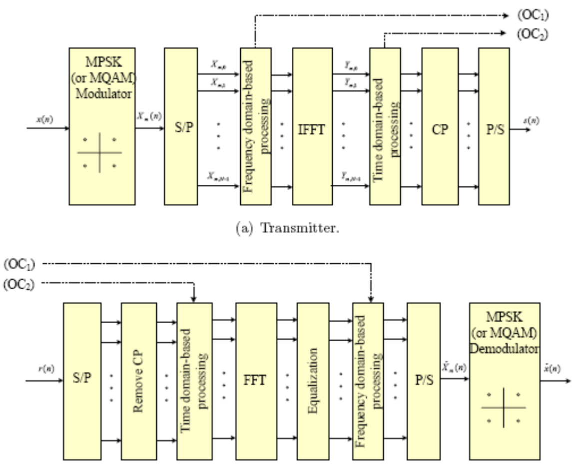


Fig. 2. GNU Radio Components

III. Orthogonal Frequency Division Multiplexing (OFDM)

OFDM is a frequency-division multiplexing (FDM) scheme utilized as a digital multi-carrier modulation method. A large number of closely-spaced orthogonal sub-carriers are used to carry data. The data is divided into several parallel data streams or channels, one for each sub-carrier. Each sub-carrier is modulated with a conventional modulation scheme (such as quadrature amplitude modulation or phase shift keying) at a low symbol rate, maintaining total data rates similar to conventional single-carrier modulation schemes in the same bandwidth.

OFDM offers several advantages over other transmission technology such as high spectral efficiency, robustness to fading channel, immunity to impulse interference, capability of handling very strong multi-path fading and frequency selective fading without having to provide powerful channel equalization.



(a) Transmitter.

(b) Receiver.

Fig. 3: An OFDM transceiver Block Diagram.

The basic principle of OFDM is to split a high-rate data stream from a data source into N lower rate streams. These streams are then individually modulated by using M-ary phase shift keying (PSK) or M-ary quadrature amplitude modulation (QAM). These streams are transmitted simultaneously over N orthogonal subcarriers using a serial-to-parallel (S/P) converter. These data over subcarriers are summed up to yield an OFDM symbol. Mathematically, if $X_{m,k}$, $k = 0, 1, \dots, N-1$, represents the complex input symbols of the k-th subcarrier at the m-th time instant, N is the number of subcarriers, and T the symbol duration, then one OFDM symbol starting at $t = t_s$ in complex baseband notation can be written as:

$$s_m(t) = \frac{1}{N}\sum_{k=0}^{N-1} X_{m,k}\exp\left(j2\pi\frac{k}{T}(t-t_s)\right) \qquad t_s \le t \le t_s T \tag{1}$$

The receiver performs the reverse operation of the transmitter that is mixing the RF signal to baseband for processing. Then, the signal is low pass filtered, converted to digital signal using an analog-to-digital (A/D) converter, and down sampled. The serial stream of sampled time signal is converted into parallel streams by the S/P converter and the cyclic prefix is discarded from the received composite signal, r_{mn}. Then, the DFT is used to transform the time domain data into frequency domain [5]:

$$\hat{X}_{m,k} = \sum_{n=0}^{N-1} r_{m,n}\exp\left(j2\pi\frac{kn}{N}\right) \qquad n = 0,1,\dots,N-1 \tag{2}$$

These parallel streams are then demodulated to yield digital data and are multiplexed together using the parallel-to-serial (P/S) converter to yield the serial bit stream, and delivered to the data sink.

IV. Experimental Setup

Fig.3 shows the experimental setup for the experimental study of OFDM SDR implementation utilizing GNU Radio and USRP. Four experiments have been conducted and the performance in terms of PRR (Packet Received Ratio) has been analyzed and reported in this paper. Two USRP, one Laptop and one personal computer (PC) were used to run these experiments. The PC with USRP A acts as the receiver while the Laptop and USRP B acts as the transmitter. Daughter boards used for these experiments are RFX2400 which can cover frequencies from 2.3GHz to 2.9GHz. For software part, the reconfigurable *benchmark_ofdm_tx.py* and *benchmark_ofdm_rx.py* in */gnuradio/gnuradio-example/python/ofdm/* are used. The distance of USRP A to USRP B is 660 mm. Experiments are conducted indoor. However, there is a neighboring access point to consider which can interfere with the USRP frequencies. Therefore, the USRP center frequency is set at 2.5GHz to avoid the interference and jamming with the said access point operating at 2.4 GHz. The aim of these experiments is to determine the relationship between PRR and Power Transmit (P_t) which utilizes OFDM environment using USRP. In these experiments, PRR is measured for two different modulations: BPSK and QPSK and varying FFT bins, 256 and 512.

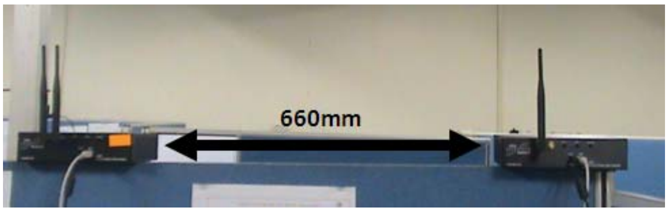


Fig. 3. GNU Radio Architecture test bed

In these experiments, we considered the relationship between DAC values and its power, P_{dac}, is shown in equation 1.

$$P_{dac} = \frac{DAC\ Value}{9830.1} \times 10mW \tag{3}$$

where 10mW is the maximum output power of the DAC chip used in the USRP. There is a Programmable Gain Amplifier (PGA) on the receiver side of the USRP which can give a gain up to 20dB. So the total power transmit is given as below;

$$P_t(dB) = P_{dac}(db) + P_{pga}(dB) \tag{4}$$

where P_{dac} used in this experiment is 10dB and P_{pga} is the power output from PGA. The PRR are plotted with the DAC values which are related to P_t.

V. EXPERIMENTAL RESULTS

The first experiment is done using OFDM with BPSK modulation and 512 FFT bin. The graph of the DAC value (which is proportionate to P_t) versus PRR is shown in Fig. 6. The graph shows that the PRR values increase drastically when the DAC values change from 0 to 1000. After the 1000 mark, the PRR stabilizes at 0.98 to 1.00. Therefore, the transmitter has to set its DAC value at least to 1000, so that 98% or more of the packets transmitted will be successfully received. However, none of the data can be received if the DAC value is set at less than 180 due to the low transmitted power.

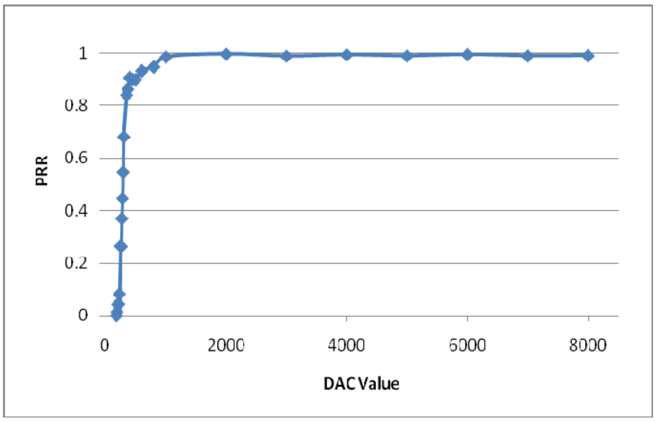


Fig.6 PRR for OFDM with BPSK modulation with 512 FFT bin

Second experiment aims to evaluate the performance in terms of PRR for OFDM with BPSK modulation and 256 FFT bin. Graph of DAC values versus PRR for this experiment is shown in Fig. 7. According to this graph, we need to set the DAC value at more than 2000 to achieve PRR more than 0.98.This graph also shows that the receiver cannot receive a single correct data if the DAC value is less than 240 due to similar reason mentioned before.

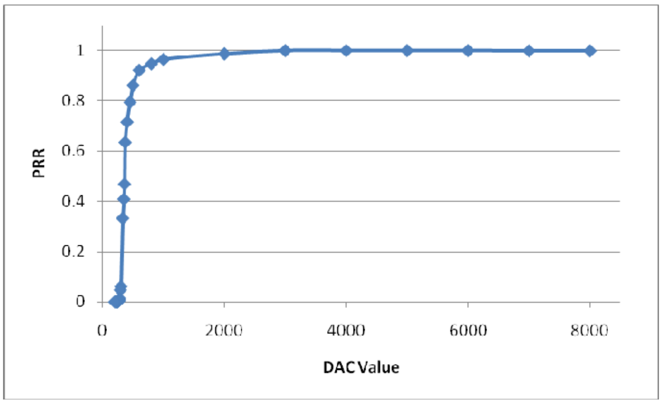


Fig.7 PRR for OFDM with BPSK modulation with 256 FFT bin

The third experiment is done by using the OFDM with QPSK modulation and 512 FFT bin. Figure 8 shows the PRR measurement versus the DAC values for this experiment. This graph shows that 0.98 PRR value can be achieved if the DAC value used is higher than 2000 and the packet is totally lost if the DAC value is less than 270.

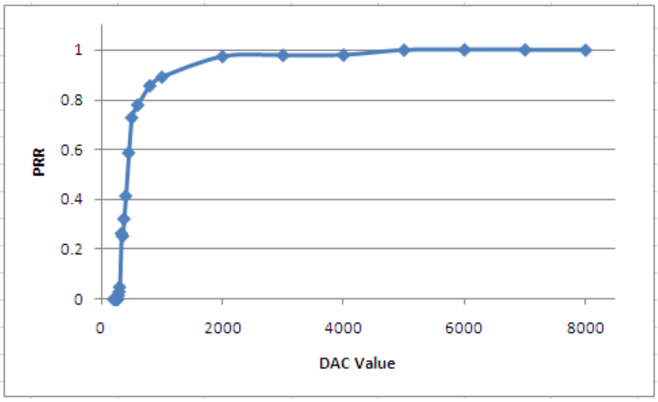


Fig.8 PRR for OFDM with QPSK modulation with 512 FFT bin

The last experiment is for evaluating the performance in terms of PRR for OFDM with QPSK modulation and 256 FFT bin. Graph of PRR versus DAC values for this experiment is plotted as shown in Fig. 9. 98% or more of received data can be successfully achieved with DAC value higher than 3000. The data is completely lost if the DAC value is less than 450.

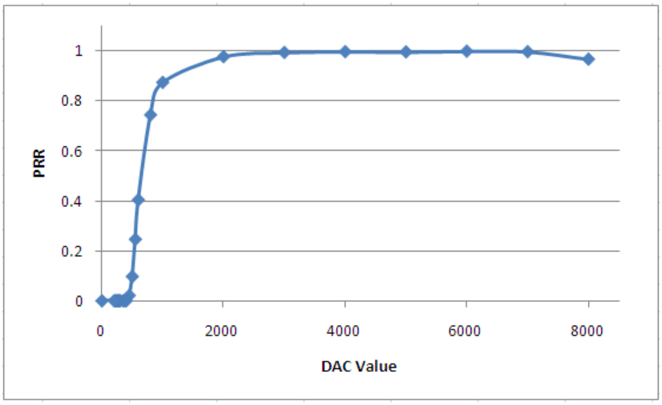


Fig.9 PRR for OFDM with QPSK modulation with 256 FFT bin

VI. CONCLUSION

The results show that OFDM with BPSK modulation gives a better performance of communication compared to the OFDM with QPSK in terms of PRR. This is in line with the theory as the error distance of BPSK is larger than QPSK due to its optimum signaling format, referred as antipodal signaling [21]. Similarly, the performance of PRR against the FFT length is in accordance with the theory [22]. The greater the length, the less the error and hence, a better PRR will be observed.

In conclusion, the radio parameters such as modulation, FFT bin and power used should be carefully chosen to achieve optimum performance of communication.

REFERENCES

[1] J.Mitola, *Software radios-survey, critical evaluation and future directions*, IEEE National Telesystems Conference, pages 13/15-13/23, 19-20 May 1992.
[2] Minh Nguyen, *Software Radio (R) Evolution and Its Application to Aeronautical Mobile Communications*, May 19-22, 2003.
[3] Matt Ettus, Universal software radio peripheral. http://www.ettus.com.
[4] Naveen Manicka, *GNU Radio Testbed*, Thesis for Master of Science in Computer Science, University of Delaware, Spring 2007.
[5] S.B. Weinstein and P.M>Ebert, "Data Tansmission by Frequency Division Multiplexing Using Discrete Fourier Transform", IEEE Trans. Commun. Technol, Vol 19 PP. 628-634, Okt 1971.
[6] Peter G. Cook ,Wayne Bonser, *Architectural Overview of the SPEAKeasy System*, IEEE Journal On Selected Areas In Communications, Vol. 17, No. 4, April 1999.
[7] Jeffrey Hugh Reed, *Software Radio: A Modern Approach to Radio Engineering*, Prentice Hall PTR, 2002. ISBN0130811580, 9780130811585.
[8] Raymond J. Lackey and Donald W. Upmal., *Speakeasy: The Military Software Radio*, IEEE IEEE Communicdtions blagctzine May 1995.
[9] *Software Communications Architecture Specification (Version 2.2.2)*, Joint Program Executive Office (JPEO) Joint Tactical Radio System (JTRS), 15 May 2006.
[10] GNU Software Radio project, http://www.gnu.org/software/gnuradio/.
[11] Thomas Sundquist, "Waveform Development using Software Defined Radio", Master Thesis, Department of Science and Technology,Linkopings Universitet, Sweden, 25 April 2005.
[12] David A. Scaperoth, *Configurable SDR Operation for Cognitive Radio Applications using GNU Radio and the Universal Software Radio Peripheral*, Thesis for Master of Science in Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 4 May 2007.
[13] Thomas Schmid, Tad Dreier, Mani B Srivastava, *Software Radio Implementation of Short-range Wireless Standards for Sensor Networking,* SenSys 2006 , November 2006.
[14] Thomas Schmid, Oussama Sekkat, Mani B Srivastava, *An Experimental Study of Network Performance Impact of Increased Latency in Software Defined Radios*, WiNTECH`07 , September 2007.
[15] Zhi Yan, Zhangchao Ma, Hanwen Cao, Gang Li, and Wenbo Wang, *Spectrum Sensing, Access and Coexistence Testbed for Cognitive Radio Using USRP*, 4th IEEE International Conference on Circuits and Systems for Communications, 2008. ICCSC 2008, 26-28 May 2008.
[16] Ahmad Ali Tabassam, Stefan Heiss, *Bluetooth Clock Recovery and Hop Sequence Synchronization Using Software Defined Radios*, Region 5 Conference, 2008 IEEE 17-20 April 2008.
[17] D.Cabric.S.M., Mishra.R.W., Brodersen, *Implementation Issues in Spectrum Sensing*, In Asilomar Conference on Signal, Systems and Computers, November 2004.
[18] Amalia Roca, *Implementation of a WiMAX simulator in Simulink*, Thesis for Diploma In Institiut Of Nachrichtentechnik Und Hochfrequenztechnik, Torreblanca Castellón, Spain, February 2007.
[19] Dawei Shen, *Tutorial 1: GNU Radio Installation Guide – Step by Step*, May 19, 2005.
[20] GNU Radio Trac, http://gnuradio.org/trac/wiki
[21] Wayne Tomasi, "Electronic Communications Systems Fundementals Trough Advanced", Printice Hall, ISBN0120306428, Fourth Edition 2001.
[22] U.Wasenm¨ uller, T. Brack, and N. When, "*Analysis of communications and implementation performance of FFT based carrier synchronization of BPSK/QPSK bursts*", Adv. Radio Sci., 6, 95–100, 2008.