2016 Joint International Conference on Artificial Intelligence and Computer Engineering (AICE 2016) and
International Conference on Network and Communication Security (NCS 2016)
ISBN: 978-1-60595-362-5

Establishing IEEE802.11p Test Platform Based on GNU Radio and USRP

Zheng RUAN[a], Rong-Jie GU[b*], Dong-Li WANG[c]

The Third Research Institute of Ministry of Public Security, Shanghai, China

[a]ruanzh2009@163.com, [b]rongjiegu@vip.163.com, [c]wdl2609@163.com

*Corresponding author

Keywords: ITS, GNU Radio, USRP, 802.11p.

Abstract. With the development of Intelligent Transport System (ITS), IEEE802.11p has just been approved as an international standard. However there are still challenging problems to be solved, for example high Doppler frequency shift and high speed hand-off. This paper proposed a method to establish a test platform based on GNU Radio and USRP1.1 to facilitate us further study in this field. The test platform established by us has been verified in two scenarios: low level TCP/IP communication and chat software IPTUX. The results show the test platform established in this paper works well and the average time consumed is 25.15ms for software processing reasons.

Introduction

Intelligent Transport System (ITS) is an important field in the future for its efficiently deal with traffic jam alarm, collision avoidance and commerce transactions in vehicles. Wireless access in the vehicle environment (WAVE)[1] is a key technology to resolve communication in vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) applications for ITS. IEEE 802.11p[2] is an approved amendment to the IEEE802.11 standard. It defines enhancements to IEEE802.11[3] required to support ITS applications. This includes data exchanges between high-speed vehicles and between the vehicles and the roadside infrastructure in the licensed ITS band of 5.9 GHz.

IEEE802.11p physical layer is based on orthogonal frequency-division multiplexing (OFDM) which is a multi-carrier modulation scheme. Multiple modulation schemes with OFDM are employed to support bit rate from 3Mbps to 27Mbps. However in the high mobility environment, many new problems emerge such as high speed hand-off, larger Doppler frequency shift, fast selective frequency fading, inter-symbol-interference (ISI) problem and so on. Although IEEE802.11p has modified some areas in the base of IEEE802.11, for example adding WAVE BSS (Basic Service Set), Timing Advertisement in MAC layer and reducing communication speed to 6Mbps in PHY layer, however these modifications may not fulfill the requirement of the high mobility.

This paper is organized as follows: Section 2 introduces the system architecture needed by the testbed, related work is also presented as a indispensable part. The logical ideal of system implementation in GNU Radio and principle of hardware are presented in Section 3. In section 4, test work in two scenarios according to IEEE 802.11p is conducted as well as performance analysis. The last section concludes the whole paper and further work directions.

Background and Related Work

The testbed is established on GNU Radio and universal software radio peripheral (USRP). In the following section, background and the related work is introduced in this field.

GNU Radio

GNU Radio[4] is a software-defined radio (SDR) platform. Nowadays, GNU Radio has been a most important SDR widely applied in amateur radio, academic institutions and business organizations for researching and developing the wireless communication problem and other various purposes, for it is open source software running on most general purpose processors and operating systems like Linux, Mac OSX and Windows XP.

The software architecture of GNU Radio consists of three layers including the application development layer, signal processing layer and scheduler controlling layer.

GNU Radio utilizes scripting language Python[5]. The connection of signal flow graph is created from source to destination. Signal processing units called as blocks such as filters, amplifiers, modulators, demodulators and more than other frequently used 100 blocks for constructing wireless communication system. These blocks are written by C++ language as it is quite efficient when program run. New algorithms can be developed by C++ too. There exists another powerful tool called SWIG (Simplified Wrapper and Interface Generator)[6] in Python. It works as glue between C++ classes and Python language, generates the wrapper for C++ modules and corresponding Python code (*.py) and library (*.so).

Universal Software Radio Peripheral (USRP)

The USRP is intended to be a comparatively inexpensive hardware device facilitating the building of a software radio. It is designed to be flexible, allowing developers to make their own daughter boards for specific needs with regard to different frequency bands and connectors. The hardware platform established utilizes USRP1.1[7]. Each USRP motherboard supports four daughter-boards act as receiving signal to be sampled by high-speed ADCs or send signal produced by DACs, two for transmitter and two for receiver. The daughter-board is RF part, several selectable daughter-boards work in different frequency band.

Related Work

The authors introduce the WAVE standardization history, MAC amendment details and PHY amendment details based IEEE802.11 for vehicular communications in[8]. In[9] the authors evaluated the maximum communication distance for 802.11p transceivers in a highway scenario, and conclude that 90% of the successful communications were conducted at a distance of less than 750m. Burguillo-Rial et al.[10] present a testbed demonstrates rapid algorithm described for 802.11p technology in a multi-agent simulator and Yi et al.[11] showed that the infrastructure data collection mode of 802.11p standard does not perform well under the current static schemes. The work provides an evaluation of throughput, collision probability and average delay based on the NS-2 simulator for vehicular networks specifies in[12]. Differing from the previous work, our paper focuses on establishing the testbed in GNU Radio, in order to verify the developed algorithm for high speed in detail, test in two different scenarios is conducted and analyzed, multiple factors can significantly affect the performance of 802.11p.

Establish Testbed

Testbed is established as the block diagram in Figure 1. The application software in PC is responsible for producing data packets to send and for processing the received data packets also.

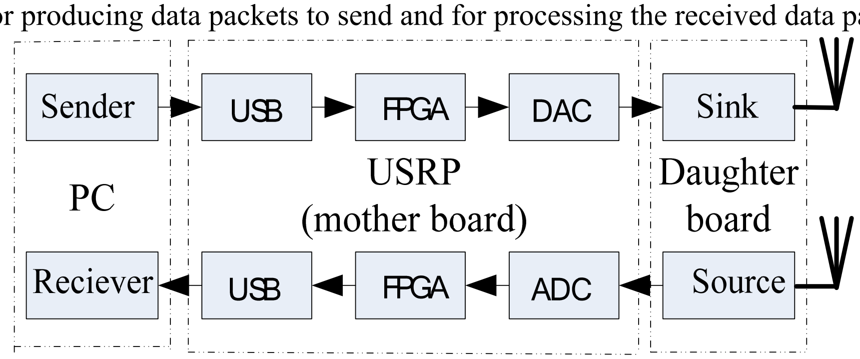

Figure 1. Framework of Test Platform.

Baseband signal processing algorithm or new communication algorithm can modified in the system. Mother board of USRP1.1 is responsible for frequency up-convert or down-convert and the daughter board is for RF processing.

Signal Processing in the Sender

Each block accomplishes corresponding signal processing algorithm required for 802.11p.

The first block sets transmit parameters such as Modulation and Coding Scheme(MCS), send and receive gain, transmission interface block calls Preamble format and Payload format blocks to complete baseband processing of preamble code, which form integrated OFDM symbols, then the signal will be processed by mode such as scrambler, interweave, these data will be further mapped to different constellation points, finally OFDM symbols inserted pilots be IFFT transformed in order to be received by hardware interface, the rest work is that how USRP1.1 transmits data.

Signal Processing in the Receiver

The receiver firstly completes the decoding of frame header then decodes the payload data, finally transmits the data to MAC layer. The first decoding function of blocks need set parameters such as the demodulation mode and the number of antenna, then Header decode, field decode, frame detection are called to complete decode of the frame header. Channel estimation estimates the channel by using long training sequence to produce the important channel parameters such as SNR. Correct phase is generated according to fine phase estimation in pilot estimation, the last decoded data are sent to MAC layer interface.

Hardware

USRP1.1's motherboard includes an Altera Cyclone EP1C12 FPGA, four 12-bit 64M sample/sec ADCs, four 14-bit 128M sample/sec DACs and a high-speed Cypress USB2.0 controller [13] with the rate of 480Mbps. The FPGA implements digital down converters (DDC), CIC interpolators and the interface with DACs and ADCs. Among them, the DDC is implemented with 4 stages cascaded integrator-comb (CIC) filters, each DDC has two inputs I and Q, each of four ADCs can be routed to either I or Q input of any DDC. DDC down converts the signal from IF (Intermediate Frequency) band to base band then decimates the signal so that the data rate can be adapted by the USB2.0 and be suit for the computer's computing capability. The complex input signal (IF) is multiplied by the constant frequency exponential signal then decimated by DCC with a factor N in the range of 1 to 256. The digital up converter (DUC) is implemented in AD9862 and FPGA. The daughter-board in the test is the transceiver RFX2400[14] which frequency band is 2.3-2.9GHz. This is different from the frequency band range of 5.850 to 5.925 GHz specified in IEEE802.11p. However it does not influence the baseband signal processing algorithm and is low cost.

Tests and Results

Real testbed is illustrated in Figure 2. Physical layer of The basic IEEE802.11p implemented in GNU radio while MAC layer in Click Router[9] which is also an open source software. USRP1.1 communicates with the PC host (Pentium Dual-core CPU @2.6GHz) by USB interface. Two detailed scenarios are following.

Figure 2. Photo of Testbed.

Scenario 1: Ping IP addresses

To detect the connectivity and analyze the speed of communication network respond, in Figure 3, Ping is used for testing communication quality between hosts. Local computer address is 192.168.255.13 while the other test side is 192.168.255.14.

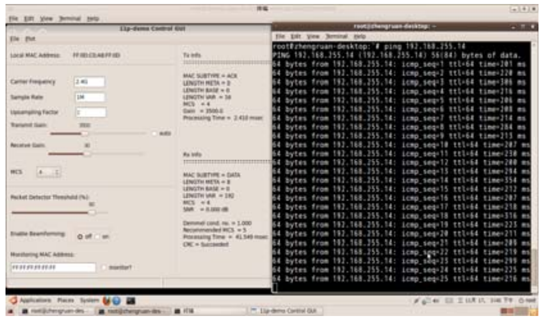

Figure 3. IEEE802.11p-demo GUI and Ping Statistics for Scenario 1.

From the control GUI (Graphical User Interface) provided on the left of Figure 3, parameters like carrier frequency, sample rate, transmit and receive gain can be adjusted to test communication in all cases. In terminal, ping packet can be processed in 200ms~300ms although software processing speed is low. There's no data packet loss, sometimes it occurs CRC (cyclic check code) error because noise interfere is existent anywhere. Modifying MCS parameter can affect communication speed. RF signal envelop has been captured by Agilent E4406A during communication in this case, as Figure 4 illustrates.

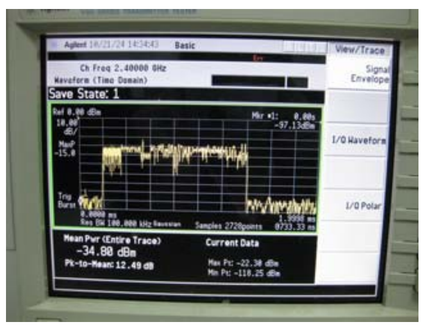

Figure 4. RF Signal Envelope in Agilent E4406A.

Scenario 2: Chat in iptux tool

Chat software "iptux" is widely existed in Ubuntu OS. Figure 5 shows the results under text applications "iptux", it's worth noting that the upper applications data are at last transfer through USRP in wireless way rather than ether cable. On the other hand this production significantly facilitates the release of traffic emergency under the rapid environment, which provides drivers critical time to avoid collision in the case of emergency, and also improves current urban crowded transportation.

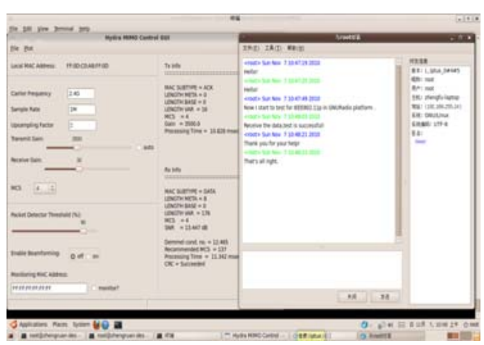

Figure 5. Chat in Iptux.

Conclusion

From the tests above, rapid algorithm development for 802.11p is demonstrated from theory to hardware implementation. With the help of GNU Radio and USRP, wireless communication in vehicular environment has been realized. The average delay time from transmission to arrival is 25.15 ms once. The next work is how to cut down this time which mostly is consumed on signal processing in PHY layer. In order to make 802.11p system perfect, the communication system needs to optimize by improving code quality for decreasing delay time. On this base, further work will be transplanting all the optimized code into specific application hardware system prototype such as FPGA or Power PC, this will considerably reduce delay time resulted from software itself, and largely meet the intelligent traffic communication requirements for real-time.

References

[1]A.U. Roberto, A.M. Gulielmo, Wave: a tutorial, IEEE Commun. Mag. 47(2009) 126–133.

[2]IEEE P802.11p/D5.0, Nov. 2008, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Amendment 7: Wireless Access in Vehicular Environments (WAVE).

[3]IEEE Std. 802.11-2007, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std. 802.11, 2007.

[4]Ke-Yu.Chen, Zhi-Feng.Chen,GNURadio, http://www.wu.ece.ufl.edu/projects/softwareRadio/documents/Project%20Report_James%20Chen.pdf.

[5]E. Blossom, Expolring GNU Radio, Free Software Foundation, 2007. http://www.gnu.org/software/gnuradio/doc/exploring-gnuradio.html.

[6]M. Adib Sarijari, A. Marwanto, N. Fisal, S. Kamilah Syed Yusof, R. A. Rashid, M. Haikal Satria, Energy Detection Sensing based on GNU Radio and USRP: An Analysis Study, IEEE 9th Malaysia International Conferenceon Communications. (2009) 15-17.

[7]M. Ettus, Ettus Research LLC, http://ettus.com/.

[8]D. Jiang, L. Delgrossi, IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments, Vehicular Technology Conference, 2008. 2008(2036-2040).

[9]L. Stibor, Z. Yunpeng, H.-J. Reumerman, Evaluation of Communication Distance of Broadcast Messages in a Vehicular Ad-Hoc Network Using IEEE 802.11p, Wireless Communications and Networking Conference, 2007. 2007(254-257).

[10]J.C. Burguillo-Rial, E. Costa-Montenegro, F. Gil-Castineira, P. Rodriguez-Hernandez, Performance analysis of IEEE 802.11p in urban environments using a multi-agent model, Personal, Indoor and Mobile Radio Communications. 2008(1-6).

[11]W. Yi, A. Akram, K. Bhaskar, P. Konstantions, IEEE 802.11p performance evaluation and protocol enhancement, Proc.2008 IEEE Int. Conf. Vehicular Electronics and Safety. 2008(22–24).

[12]T. Murray, M. Cojocari, F. Huirong, Measuring the performance of IEEE 802.11p using ns-2 simulator for vehicular networks, Electro/Information Technology, 2008. 2008(498-503).

[13]The USRP Board, http://www.snowymtn.ca/gnuradio/ gnuradiodoc-4.pdf.

[14]transceiver daughterboards for the USRP software radio system. http://www.ettus.com/downloads/ettus_ds_transceiver_dbrds_v6c.pdf.