

Initial Draft of SDR, SDN, and Cognitive Radio Literature

J.D. McCormack

I. RTL-SDR

The RTL-SDR is a recent tool discovered by the DIY and hacking community. Its original purpose is to be used as a digital TV tuner. However, it was discovered that this system could also be used to general SDR purposes. There is now a large community dedicated to using this tiny SDR to receive various different signals. Prior to the creation of the open source drivers for the RTL-SDR, the most popular devices for SDR came from USRP. The USRP devices are fantastic products, but cost at least \$1,000 and can cost quite a bit more with additional features. The RTL-SDR is based on the Realtek RTL2832U chip. This device can often be purchased for between \$20 and \$30 [1]. The range for the RTL-SDR is typically 64 MHz to 1700 MHz, however this varies depending on which tuner the manufacturer paired with it. The authors in [1] paired the RTL-SDR with a mixer in order to lower the range all the way to DC. For this, they used the NE6062AN chip.

Starting with Release 2013b, MATLAB/Simulink now have a support package that targets RTL-SDR devices. In Simulink, the package contains a single block called "RTL-SDR Receiver." This block allows the user to tune the center frequency, change the tuner gain, set the sampling rate, and alter the frequency correction factor. The block can then output the complex envelope (IQ) of the received signal in both floating point and integer formats[2]. Due to the open nature and low cost of the RTL-SDR, the authors in [3] propose using this as a tool set for teaching DSP and Communications principles to students.

When the cost of the system is far less than that of a textbook, it is easy to see why this could become a valuable learning tool for many students. UC Berkeley has already begun to use the RTL-SDR as one of the project assignments in their digital signal processing course. There have been efforts to use the RTL-SDR with the popular Raspberry Pi computing platform. However, at least with the B+ model there is not enough power available to process the signal. Instead, it has to be used as a TCP server that is then able to forward the data on to a more powerful computer[4]. Currently, it does not appear as though anyone has tested this with the Raspberry Pi 2 microcomputer. Other work has been done to estimate the cost savings of using a USRP in conjunction with several RTL-SDRs to replace existing DSP lab infrastructure [5].

The most unfortunate downside to using the RTL-SDR is that it is unable to transmit. However, researchers in [6] proposed a system in which a USRP SDR is used as a master device that broadcasts out to a series of slave nodes that can only listen for information. In their system they used a tool

called GStreamer to pass video data into GNURadio. This information was then broadcast to multiple computers in a room running an RTL-SDR with GNURadio. The nodes were all able to view the video stream in near real time.

It is interesting to note that an IEEE search for the RTL-SDR turned up 6 results, but a search on Funcube Dongle returned 0 results. These devices were in competition with one another for awhile, but the Funcube's prices continue to rise as the RTL-SDR's continue to drop. The Funcube has a larger range than the RTL-SDR but it seems the cost is still the key factor.

II. GNU RADIO AND MESH NETWORKS

In [7] and [8] the authors use GNU Radio as a way to verify the successful use of algorithms for mesh networking. However, they do not specify that they are using SDR's and it seems like they use GNU Radio for simulation. GNU Radio has also been used with the USRP to create a device capable of communicating with both Bluetooth and WiFi devices. However, this does not create a mesh network or attempt to bridge communication between the two protocols. However, a significant amount of information about communicating with each type of network is presented [9]. This also presents the concept of Police Nodes which monitor traffic in an attempt to block out improper use of the spectrum.

Research has been done in using the GNURadio toolset along with the USRP to test Mesh Network routing protocols. One test used varied data transmission rates to exploit opportunities in physically close proximity Nodes [10]. The GNU Radio toolset was also used to test using cognitive radio within a mesh network. USRPs were used as nodes trying to communicate on a "shared" frequency. A separate USRP was used to replicate a primary user, or one that had a license to operate in that spectrum. Whenever the primary user began to transmit in the spectrum, the other nodes would use reinforcement learning to move to an unoccupied channel automatically and continue transmitting [11]. A similar test bed is also presented in [12].

REFERENCES

- [1] M. Sruthi, M. Abirami, A. Manikoth, R. Gandhiraj, and K. Soman, "Low cost digital transceiver design for software defined radio using rtl-sdr," in *Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on*, March 2013, pp. 852-855.
- [2] A. Sergienko, "Software-defined radio in matlab simulink with rtl-sdr hardware," in *Computer Technologies in Physical and Engineering Applications (ICCTPEA), 2014 International Conference on*, June 2014, pp. 160-161.

- [3] B. Uengtrakul and D. Bunnjaveht, "A cost efficient software defined radio receiver for demonstrating concepts in communication and signal processing using python and rtl-sdr," in *Digital Information and Communication Technology and it's Applications (DICTAP), 2014 Fourth International Conference on*, May 2014, pp. 394–399.
- [4] R. Danyamol, T. Ajitha, and R. Gandhiraj, "Real-time communication system design using rtl-sdr and raspberry pi," in *Advanced Computing and Communication Systems (ICACCS), 2013 International Conference on*, Dec 2013, pp. 1–5.
- [5] M. Abirami, V. Hariharan, M. Sruthi, R. Gandhiraj, and K. Soman, "Exploiting gnu radio and usrp: An economical test bed for real time communication systems," in *Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on*, July 2013, pp. 1–6.
- [6] S. Nimmi, V. Saranya, G. Theertha Das, and R. Gandhiraj, "Real-time video streaming using gstreamer in gnu radio platform," in *Green Computing Communication and Electrical Engineering (ICGCCCE), 2014 International Conference on*, March 2014, pp. 1–6.
- [7] R. Alimi, L. Li, R. Ramjee, H. Viswanathan, and Y. Yang, "ipack: in-network packet mixing for high throughput wireless mesh networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008, pp. –.
- [8] L. Li, R. Alimi, R. Ramjee, H. Viswanathan, and Y. Yang, "munet: Harnessing multiuser capacity in wireless mesh networks," in *INFOCOM 2009, IEEE*, April 2009, pp. 2876–2880.
- [9] R. Miller, W. Xu, P. Kamat, and W. Trappe, "Service discovery and device identification in cognitive radio networks," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON '07. 4th Annual IEEE Communications Society Conference on*, June 2007, pp. 670–677.
- [10] C. Yu, T. Shen, K. Shin, J.-Y. Lee, and Y.-J. Suh, "Multihop transmission opportunity in wireless multihop networks," in *INFOCOM, 2010 Proceedings IEEE*, March 2010, pp. 1–9.
- [11] A. Syed, K. Yau, H. Mohamad, N. Ramli, and W. Hashim, "Channel selection in multi-hop cognitive radio network using reinforcement learning: An experimental study," in *Frontiers of Communications, Networks and Applications (ICFCNA 2014 - Malaysia), International Conference on*, Nov 2014, pp. 1–6.
- [12] P. Nagaraju, L. Ding, T. Melodia, S. Batalama, D. Pados, and J. Matyjas, "Implementation of a distributed joint routing and dynamic spectrum allocation algorithm on usrp2 radios," in *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, June 2010, pp. 1–2.