Vol. 19, No. 1, 2022, ISSN 15965538, pp 34 -39

Journal of Civil and Environmental Systems Engineering

Department of Civil Engineering, University of Benin, Nigeria

Journal homepage: https://j-cese.com/

EFFECT OF BUFFER SIZE VARIATION ON VIDEO QUALITY TRANSMISSION IN A COGNITIVE RADIO NETWORK

Bello N. and Muhammed A. A

Department of Electrical/Electronic Engineering, Faculty of Engineering, University of Benin, Nigeria

nosabello@uniben.edu and abudu.muhammed@uniben.edu

Abstract

This paper shows how the quality of a video stream in a cognitive radio can be affected by the addition of a data buffer for the received video packets. Video encoding and decoding are done in the open-source software Gstreamer. The video buffer at the receiver is also implemented with Gstreamer. GNU Radio is then used for the physical (PHY) and medium access control (PHY) implementation of the cognitive radio. The setup also uses software-defined radios (SDRs) for the wireless communication between the transmitter and receiver. The performance of the video when played back on the receiver was evaluated using different threshold levels between 95% and 75% in decrements of 5% and compared to a baseline – a cognitive radio without a buffer. The achieved results show that in comparison to the baseline, at a threshold of 85%, there is an increase in the runtime – the amount of time the video can be played until an error occurs

Keywords

Transmission; Dynamic Spectrum Access; Smart Channel Selection; GNU Radio; Gstreamer; Software-defined Radio; Cognitive Radio

Article History

Received: March, 2022
Reviewed: March, 2022
Accepted: April, 2022
Published: May, 2022

1. Introduction

The idea of Dynamic spectrum Access (DSA) techniques has been proposed and demonstrated in various forms to address the issue of spectrum shortage by allowing for dynamic allocation and opportunistic use of underutilized spectrum channels. Cognitive radio is an important part of DSA, with cognitive radio, improved DSA techniques can be achieved and implemented, thus leading to improved spectrum awareness and management (Zhao and Sadler, 2007). The term cognitive radio was coined by Joseph Mitola in 1998 (Mitola, 2000). The cognitive radio is a smart radio system that is aware of the surrounding operational environment and can reconfigure itself and adapt to changes in its environment (Mitola, 2006.). The cognitive radio is based on Software-defined Radios (SDRs), which are radio systems that have functions that can be configured and implemented programmatically.

To achieve environmental awareness, cognitive radio has to be able to sense its surroundings or in other words, perform spectrum sensing. This allows the cognitive radio to sense its spectrum environment and search for vacant spectrum holes that can be taken advantage of by an unlicensed secondary user on the condition that the secondary user will vacate the spectrum space for the licensed primary user when they need to use that spectrum space. Therefore, a cognitive radio must have an efficient and effective algorithm to do all of this (Xu and Zhou, 2013). However, with the overall framework of DSA techniques that allow for the opportunistic use of the unutilized spectrum by unlicensed users, there is still the problem of the quality of service of the secondary users of the spectrum (Khan and Nakagawa, 2012).

In a paper by M. Tahir et al (2012), they implemented a technique for dynamic spectrum access for video transmission where they demonstrated a noticeable improvement in the quality of video at the receiver when tested with DSA enabled against the baseline, non-DSA case. Also, Roy, Debashri and Chatterjee, Mainak (2018) in their paper went further and proposed a method for adaptive video encoding at the transmitter to improve the quality of service at the receiver in dynamic spectrum access and implemented this with GNU Radio (GNU Radio, 2022) and Gstreamer (Gstreamer.freedesktop.org., 2022). They proposed adjusting the encoding bitrate, resolution and frame rate at the transmitter, akin to what some video streaming websites currently employ, to improve the receiver's service quality. Research has shown that spectrum resource allocation based on the buffer size of the PUs and SUs is mainly affected by the buffer setting of the PUs. So the actions of the SUs can be optimized considering the influence exerted by the buffer size setting of the PUs. Zhao, Y. and Yue, W. (2020) addressed the use of buffer setting and proposed a probability returning scheme to the interrupted SU packets using a Markov chain model. The proposed method was compared in terms of average delay of packets, bocking rate, throughput, average queue length and loss rate of the SU with the baseline, a non-buffer aided scheme for the SUs. Though the proposed method shows an two-fold increase in the capacity of the buffer improves the throughput of the SU, the Markov chain model was simulated with certain assumptions. Not many works on the use of buffers in the improvement of packet throughputs are available in the area of research regarding cognitive radios. So in this paper, we considered the method centred on packet buffering and monitor the effects on the quality of service for secondary users with a practical demonstration using video data. This method leverages the open-source software Gstreamer to modify the cognitive radio's receiver end, allowing the received video to be stored in a buffer. The amount of data available within the buffer is also considered when deciding whether to switch channels. This is an approach that is yet to be explored towards the improvement of the quality of service of a cognitive radio network.

This paper is structured as follows. Section 2 describes our experimental setup and demonstration environment, starting with an overview of the channel selection method used. In section 3, the results are presented and expanded upon and conclusions are drawn in section 4.

2. Experimental Setup

2.1 Overview of the Smart Channel Selection Method

A simplified overview of the smart channel selection method used in this experiment is presented in figure 1. On initialization, the Cognitive radio acting as the channel server sends a query to the long-term database which was prepared based on historical information of the demonstration environment, and receives a list of available channels, sorted in order of increasing occupancy, where a channel with a lower occupancy is more likely to be available at any given time. When the need to select a new channel arises, the channel server senses the list of channels gotten from the long-term database and selects the channel that is available then. It then communicates this data to the transmitter and receiver of the secondary users over a shared data link.

2.2 Addition of a Data Buffer

The addition of a data buffer to the receiver end allows for a channel switch request to be made in the unlikely event that a primary user is not detected in time. This is done by checking the amount of data available in the buffer and if the data is below a certain threshold, sending a request to the channel server for a new channel. This method works under the assumption that under normal conditions when a channel is free, the data within the buffer is unlikely to drop below a certain threshold.

2.3. Methods to Compare

To show the effects of adding a data buffer, the achieved runtimes are compared with and without the buffer as well as also comparing the false switching rates at various buffering thresholds.

2.4. Demonstration Environment

The Block diagram of the experimental setup used is shown in figure 2. A total of six Software Defined Radios were used; three USRP (Universal Software Defined Radio) B210 by Ettus Research (Files.ettus.com. 2022) and three HackRF (Hack Radio Frequency) One by Great Scott Gadgets. The experiment was carried out at the Faculty of Engineering at the University of Benin, in one of the available lecture theatres. Signal processing was done with GNU Radio flow graphs, and the video encoding, decoding and playback were done using Gstreamer.
The major components of the entire system are described in the following subsections.

2.4.1. The Primary Users

The primary users in this experiment consist of three HackRF One (Ossmann, M., 2022) configured with one operating on a single channel while the other two are free to hop between the available channel range in a regular pattern.

2.4.2. The Secondary User

The secondary user, in this case, consists of two USRP B210. One acts as the transmitter and the other acts as the receiver. The secondary user is free to move between the available channels only when it receives new channel information from the channel server.

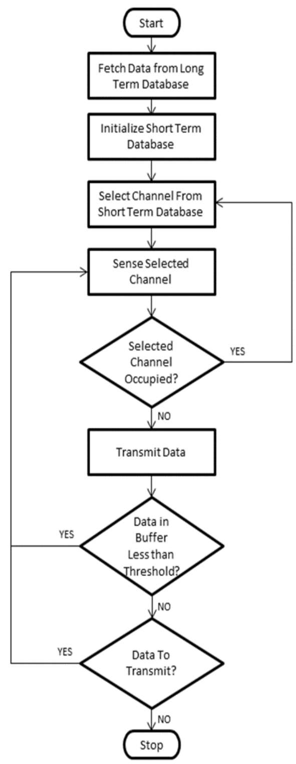

Figure 1: A flowchart overview of the channel selection method used.

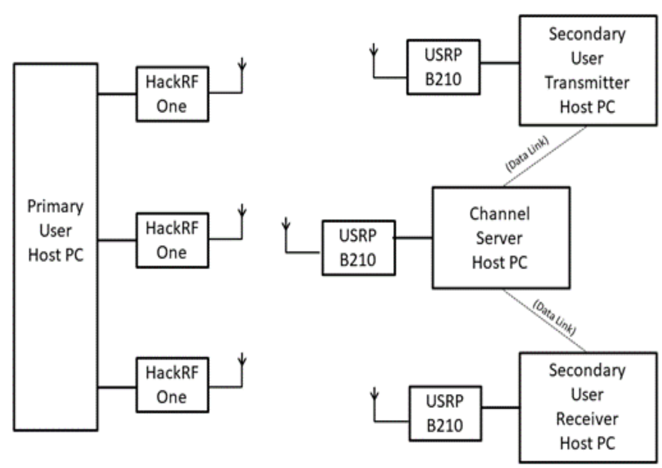

Figure 2: Block diagram showing the experiment setup

2.4.3. The Channel Server

The channel server consists of a single USRP B210. It senses the list of available channels and then chooses and sends an appropriate one over a data link to the secondary user only when interference from a primary user is detected.

2.4.4. The Video Source

A live feed from the webcam of the transmitter PC (Personal Computer) was used. The video was captured and encoded for transmission using Gstreamer and passed on to the transmitter through a Linux pipe.

2.4.5. The Video Playback

The data gotten at the receiver was passed on to Gstreamer for decoding, monitoring and playback. The buffering occurs at this stage.

3. Results and Discussion

The measurements taken were the runtimes in seconds and the number of false switches measured for each threshold level. To generate the data used, measurement was taken at each threshold level five consecutive times. Performance is evaluated in terms of the runtime of the experiment which in this case is taken as the duration for which the video can be played before an error or a crash occurs. Table 1 is a summary of the noteworthy results and shows the maximum and minimum runtimes rounded to the nearest second as well as the average runtime that was got as the arithmetic mean of five consecutive runs. It also shows the average number of false switches rounded to the nearest whole number and the false switch rate which is gotten by dividing the number of false switches by the average runtime in minutes. A false switch is considered to have occurred if a channel switch request is made when there is known to be no interference from the primary user. The buffer size in GStreamer is 100 buffers, 2MB of data, or two seconds worth of data, whichever is reached first.

For this experiment, the buffering threshold values above 95% were not considered because the behaviour of the setup became erratic with these higher threshold values. Also, it was observed that the threshold values below 75% did not produce any noticeable variation. Thus, the results of these values were neglected in the experiment.

Table 1. Summary of Experimental Results.

Buffer Threshold	Avg. Runtime (s)	Max. Runtime (s)	Min. Runtime (s)	No. of False Switches	False Switch Rate (false switches per min)
95%	105	123	98	6	3.43
90%	121	143	109	4	1.98
85%	398	421	378	3	0.45
80%	172	184	165	1	0.34
75%	163	175	154	0	0
No Buffer	182	214	132	0	0

Figure 3 below presents our findings in graphical form. It shows a drastic increase in both minimum and average runtimes for a buffer threshold level of 85% when compared to other threshold levels and the baseline with no buffer.

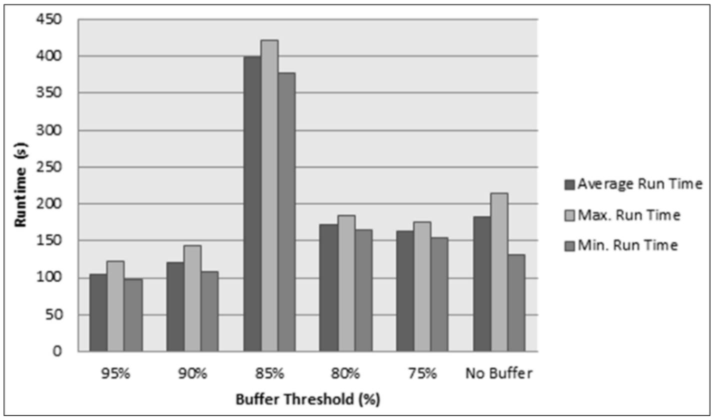

Figure 3: Runtimes for each buffer threshold used

Though the runtime is improved with the buffering setting at 85% threshold level, there are some reduncdancies with the design. The system supports channel switch request with the buffering setting and it was observed that there were false channel switch requests. Figure 4 shows the false switch rates for each threshold used. From the graph, it is seen that to avoid the problem of false switching, the buffer threshold of 75% or the baseline method should be used. It is also noted that the rate at which these false switches occur tends to decrease when a lower buffer threshold is used.

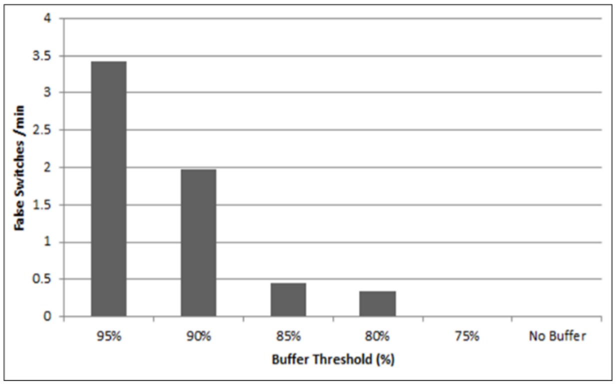

Figure 4: False switch rates for each threshold used.

From our findings, therefore, to increase the overall duration for which the video can be played, a threshold of 85% would be best, but this can, however, lead to the secondary user underutilizing the available spectrum due to the secondary user switching channels while it is still vacant and is thus inferior to the baseline in this regard.

4. Conclusion

The use of a buffer allows for longer more consistent playback in a cognitive radio network and offers some level of redundancy in interference detection. If the buffer is in use, a recommended threshold level is 85% as it gives the best performance during the duration of the video transmission where there are fewer false channel switches and high runtime duration of the cognitive radio. The false switches can be eliminated through a check system before acknowledging the request while maintaining the high runtime. This paper has shown the effects of this approach and compared it to one without a buffer. The video playback performance of the buffering approach is greatly improved albeit with the downside of introducing inaccuracies in channel switching leading to an underutilization of the available channels.

References

Ettus Research. 2022. *SDR Software – Ettus Research*. [online] Available at: https://www.ettus.com/sdr-software/ [Accessed 22 March 2022].

Files.ettus.com. 2022. *USRP Hardware Driver and USRP Manual: USRP B2x0 Series*. [online] Available at: https://files.ettus.com/manual/page_usrp_b200.html [Accessed 22 March 2022].

GNU Radio. 2022. *About GNU Radio · GNU Radio*. [online] Available at: https://www.gnuradio.org/about/ [Accessed 22 March 2022].

Gstreamer.freedesktop.org. 2022. *Application Development Manual*. [online] Available at: https://gstreamer.freedesktop.org/documentation/application-development/index.html [Accessed 22 March 2022].

Höyhtyä, M., Korpi, J. and Hiivala, M., 2016, May. Predictive channel selection for over-the-air video transmission using software-defined radio platforms. In International Conference on Cognitive Radio Oriented Wireless Networks (pp. 569-579). Springer, Cham.

Khan, F. and Nakagawa, K., 2012. Performance improvement in cognitive radio sensor networks. *the IEICE Japan*.

Mitola, J., 2000. Cognitive radio. Doctorate dissertation. Royal Institute of Technology (KTH).

Mitola, J., 2006. Cognitive radio architecture. *Cooperation in wireless networks: Principles and applications*, pp.243-311.

Ossmann, M., 2022. *HackRF One - Great Scott Gadgets*. [online] Greatscottgadgets.com. Available at: https://greatscottgadgets.com/hackrf/one/ [Accessed 22 March 2022].

Roy, Debashri & Chatterjee, Mainak. (2018). Adaptive Video Encoding and Dynamic Channel Access for Real-time Streaming over SDRs. 10.1109/PCCC.2018.8710795.

Tahir, M., Mohamad, H., Ramli, N., and Jarot, S. P. W. (2012). Experimental implementation of dynamic spectrum access for video transmission using USRP. 2012 International Conference on Computer and

Communication Engineering (ICCCE). doi:10.1109/iccce.2012.6271185.

Xu, H. and Zhou, Z., 2013, September. Cognitive radio decision engine using hybrid binary particle swarm optimization. In *2013 13th International Symposium on Communications and Information Technologies (ISCIT)* (pp. 143-147). IEEE.

Zhao, Q. and Sadler, B.M., 2007. A survey of dynamic spectrum access. *IEEE signal processing magazine*, *24*(3), pp.79-89.

Zhao, Y. and Yue, W., 2020. Performance analysis and optimization for cognitive radio networks with a finite primary user buffer and a probability returning scheme. *Journal of Industrial & Management Optimization*, *16*(3), p.1119.