JJEE Volume 10 | Number 2 | June 2024 | Pages 229-239

Jordan Journal of Electrical Engineering

ISSN (print): 2409-9600, ISSN (online): 2409-9619

Homepage: jjee.ttu.edu.jo

Real-Time Implementation of a Cognitive Radio Node for Video Transmission with Dynamic Channel Selection

Ogechukwu A. Kanu[1*], Nosa Bello[2]

[1, 2] Department of Electrical and Electronic Engineering, Faculty of Engineering, University of Benin, Edo, Nigeria
E-mail: kanu.ogechukwu@eng.uniben.edu

Received: Jul 05, 2023 *Revised: Sep 11, 2023* *Accepted: Sep 13, 2023*

***Abstract*—** This paper endeavors to exploit the extent to which the radio spectrum can be utilized by almagating and enhancing various methods proposed by previous researchers in the field. Cognitive radio research addresses the challenge of spectrum scarcity and underutilization through cognitive radio technology. The realm of cognitive radio research encompasses diverse aspects of its architecture, ranging from spectrum sensing to higher-level strategies. Notably, the navigation strategy employed by cognitive radios constitutes a pivotal layer in these implementations. Thus, this paper focuses on the real-time implementation of a cognitive radio node in a spectrum of licensed users. Leveraging the capabilities of software-defined radios, GNU Radio, GStreamer and Ubuntu OS computers, this investigation introduces a streamlined channel strategy. This strategy uses both short-term and long-term historical data, along with a buffering mechanism, to facilitate the real-time transmission of a high-definition video signal. The successful implementation of this system yields a promising enhancement of approximately 32% in radio resource utilization. Furthermore, the sensing time of the cognitive radio experiences a significant reduction. This research contributes to the advancement of cognitive radio technology and demonstrates the consolidated strategies' practical implications, underscoring their potential to overcome spectrum-related challenges.

***Keywords*—** Channel selection; Cognitive radio; Software-defined radios; Gstreamer; GNU Radio; Gaussian minimum shift keying; Video transmission.

1. INTRODUCTION

With the increasing development of more mobile computing applications like the Internet of Things (IoT), smart homes, smart cities, mobile health, Unmanned Aerial Vehicle Networks, and the increase in usage of smartphones around the world, there is a problem of shortage in the spectrum as the demand of spectrum and Quality of Service (QoS) has increased tremendously [1, 2]. This problem is even increased when considering video transmission due to increased bandwidth requirements [3].

Dynamic Spectrum Access (DSA) is a new management policy as opposed to the static radio spectrum management policy [4]. There are three models under the dynamic spectrum access: the exclusive use model, the open sharing, and the hierarchical access model [5]. Cognitive Radio is the first envision of the hierarchical model implementing overlay spectrum access. It was coined by Mitola in 1998 and then investigated by DARPA Next Generation (XG) [6]. The overlay spectrum access applies restrictions on the radios in the hierarchical model such as the transmission power (not severe) and opportunistic spectrum access. Such radios aim to target spatial and temporal spectrum white space, which should be identified and exploited locally and instantaneously in a non-intrusive manner. They are radios which can be

* Corresponding author DOI: https://doi.org/10.5455/jjee.204-1688576444

programmed and configured to automatically use a predetermined band as they have the feature of supporting different air interfaces [7]. Cognitive Radio functionalities are built upon the Software-defined Radios (SDRs) platform, it is a smart and intelligent radio that is aware of its environment, capable of self-reconfiguration and adaptation to any spectrum environment by learning about that environment [8].

One of the motivations of this work is to investigate the extent to which spectrum usage can be improved with real-time streaming given latencies and hardware constraints. This paper shows a practical demonstration of a cognitive radio which opportunistically accesses the white spaces while transmitting real-time video in a congested radio environment. It is greatly inspired by the works of [3]. They investigated the extent to which video source encoding and wireless transmission can be jointly optimized. Using open-source tools like GNU Radio and Gstreamer and a set of programmable radios or SDRs, they practically demonstrated HD video streaming and the improvement of SSIM (Structural Similarity Index Metric) and PSNR (Peak Signal to Noise Ratio) with adaptive encoding and DSA implementation. Thus, we investigated a third parameter, spectrum occupancy, in our work. Another practical demonstration of cognitive radio technique paramount to this work is that of [9]. With Labview and SDRs, they demonstrated the impact of predictive channel selection on the sensing time and throughput.

The studies evaluating the different techniques by which spectrum usage can be improved are discussed. The authors in [3] exploited the effects of adaptive encoding and dynamic spectrum access on the video quality of the agent (cognitive radio). In [9, 10], the authors proposed a method of channel selection that combines both long-term and short-term information to reduce sensing time, increase throughput and reduce collision with primary users of the spectrum. Furthermore, the authors in [11] explored the challenges of spectrum sensing and packet transmission in real-time video transmission. Some works also investigated the handoff strategies that optimize transmission by reducing the end-to-end transmission delay of real-time video packets [12, 13]. To the best of our knowledge, most of the research papers investigating cognitive radio techniques have different objectives like minimizing sensing time and interference or maximizing throughput but information regarding the radio resource usage was not included. Thus, this paper consolidates the closely related research in cognitive radio techniques and investigates the extent to which the radio resource can be greatly utilized. Through a system that implements a channel selection strategy that maximizes cognitive radio throughput by opportunistically accessing the white spaces in the spectrum and applying a buffer system to maintain real-time video streaming, we obtained a reasonable increase in the use of spectrum resources.

2. SYSTEM OUTLINE AND METHODOLOGY

The experiment setup consists of three Universal Software Radio Peripheral (USRP) B210 devices (SDRs). Two of the devices were used to implement the cognitive radio node and the remaining device functioned as the spectrum sensing module. The radio spectrum was active with signals transmitted with three HackRF Ones. They represented the primary users (PUs) or licensed users of the spectrum in the network. The experiment was carried out in the 900-band as it had the least activity in the location of the experiment and the testbed spectrum band used was a 10 MHz band from 900 to 910 MHz.

The overall block diagram of the testbed setup is shown in Fig. 1. The system design of the cognitive node starts at the transmitter. The live video is captured via the webcam which is then encoded using a Gstreamer pipeline prior to the wireless radio stack for communication developed with GNU Radio 3.7. At the wireless radio stack, we used GMSK (Gaussian Minimum Shift Keying) modulation scheme for wireless communication.

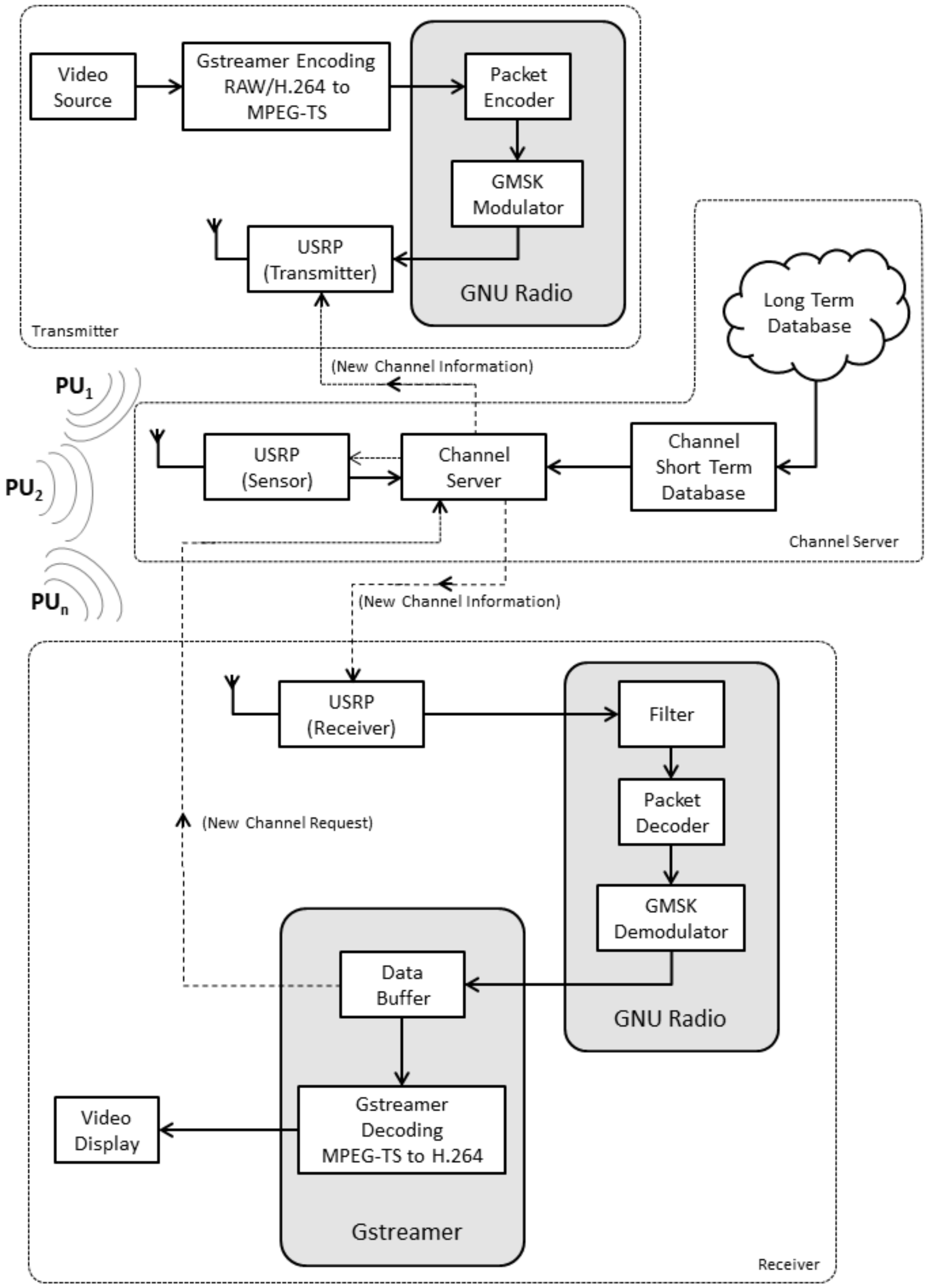

Fig. 1. Overall system design used for the experiment.

The radio packets in the GNU Radio (GR) radio stack are modulated onto a carrier frequency which is determined by the decision of the spectrum sensing module that senses the environment for free channels. The spectrum sensing module is part of the channel server part of the system architecture which handles all decision-making functions. It was implemented with the energy detection method at a predetermined threshold level of the radio testbed spectrum band. The information from the radio resource is stored in the databases (short-term and long-term). For the experiment, the PUs were programmed to use the spectrum with a considerable time of inactivity which can account for the sweep time of the spectrum sensor and channel mobility actions by the cognitive radio.The receiver block also communicates with the spectrum sensing module to know the frequency of communication with the transmitter. The receiver block receives the signal and demodulates it for further processing in the radio stack to generate the packets to be decoded by Gstreamer and displayed. In addition to the decision-making process of the channel server, we included a buffering system in the receiver block that contributes to channel switching when the channel in use is not generating enough throughput. Thus, while the channel server constantly checks for white holes in the spectrum and the return of the licensed users, the buffering system acts as a feedback system to measure the quality of the selected channel. The detailed methodologies used for the three blocks are discussed as follows:

2.1. Transmitter Block

The source for the video to be transmitted is a computer webcam. The captured video is then passed to the Gstreamer block to perform encoding. Encoding was done using an open-source Gstreamer multimedia framework [14]. The framework was used to convert our captured video from the webcam into the .ts (Transport Stream) format. This was achieved by using Gstreamer pipelines as shown in Fig. 2.

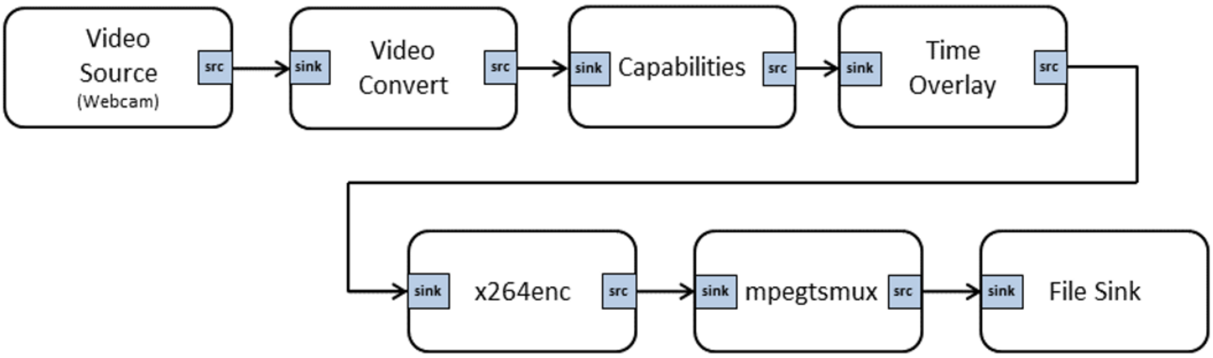

Fig. 2. Block diagram of a typical Gstreamer encoding pipeline.

The radio stack using GMSK was implemented with GNU Radio. There is a graphical interface that allows users to build systems with a drag-and-drop feature called the GNU Radio Companion [15]. The radio stack implemented a simple radio stack with the MAC (Medium Access Control) and PHY (Physical) layers. In the MAC layer, the packets were encoded with the Packet Encoder block in GNU Radio which packetizes the output stream from the file source into a form that can easily be modulated. The GMSK Modulator block performs modulation of the packetized data onto the carrier frequency sent over the air with a USRP B210 through the USRP Source block (an abstraction of the USRP Hardware Driver). GMSK was preferred because of its better spectral efficiency, ease of implementation, and reduced power consumption [16].

2.2. Spectrum Sensing Module

The energy detection method is a popular and simple spectrum sensing method. It detects the presence or absence of a signal in the received signal using a threshold value. The experiment was conducted with signals of different power levels. In the dynamic spectrum access model implemented, the primary users had a higher power level than the secondary users. As such the choice of the threshold level would detect the presence of the primary users and allow swift action. The sensor block was tasked with detecting the primary users and populating the database with the information of the spectrum. The results formed the long-term (LT) information. The spectrum sensing module was also implemented with GNU Radio. The method implemented a wideband spectrum sensing system that hops frequencies within the range to be sensed at an interval. The sensor sensed frequency range was 900 MHz to 910 MHz at a sample rate of 1.34 MHz (sweep interval). The system measured the channels' energy, noise, and time and stored the information in the LT database. This spectrum sensing is just a modification to the spectrum sensing code obtained from [17]. With the LT database, the most suitable channels were selected which helps reduce the sensing time for a cognitive radio node and they make up the short-term (ST) database. The decision of which channel to select and use is retrieved from the ST database. We can describe the operation of the LT and ST databases as follows:

a) Populate the LT database with a total of C channels.
b) From those C channels, select M channels and store them in the ST database.
c) Select one channel for use from M channels in the ST database.

The channel server, which can be called the engine of the cognitive radio houses the brain of the cognitive radio. This is where decisions like what channel to use are taken, and the communication with the transmitter and receiver for the next frequency to tune to is performed. This communication link between the engine and the radio stack was achieved using TCP protocol.

2.3. Primary Users

The presence of the primary users (PUs) was generated using three HackRF One SDRs. The setup used nine channels within a 10MHz band from 900 MHz to 910 MHz. Each of the channels had a different usage rate (duty cycles) in their channels which was predetermined. The PUs were FM signals with a higher power level in the spectrum. The choice of nine channels allowed us to demonstrate the impact of the two-fold database storing system on the sensing time.

2.4. Receiver Block

The receiver block implements filtering, synchronization, and demodulation of the received GMSK signal. This block was designed in GNU Radio 3.7 which again uses the abstraction of the USRP Hardware driver to interact with the USRP B210 device in receiving mode. The produced samples are filtered to remove noise and then synchronized using preambles encoded in the signal to remove frequency and timing offsets for the demodulator to correctly decode the signal. Finally, the GMSK demodulator demodulates and decodes the bytes and sends the packets to GStreamer for any order processing.

2.5. Data Buffer and Display

In GStreamer, the received .ts packets are decoded and displayed. In addition, the channel selection algorithm was also supplemented with a buffer. It was implemented using GStreamer in C programming. The addition of the buffer was to improve the robustness of the live stream. It acted as a feedback mechanism on the quality of the channel selected and its impact on the throughput of the stream. A channel request was made to the channel server whenever a limit is reached. Optimally, a limit of 85% of the maximum buffer size was used to issue a channel switch request to the channel server (cognitive engine) [18]. This secondary mechanism makes requests when the radio stack is receiving, which ensures that if the buffer drops below the limit, a request is sent supplementing the channel selection algorithm.

3. EXPERIMENTAL SETUP AND IMPLEMENTATION

The whole setup and experiment were done in one of the lecture theatres located at the Department of Electrical and Electronic Engineering of the University of Benin, Nigeria. A photograph of the setup is shown in Fig. 3.

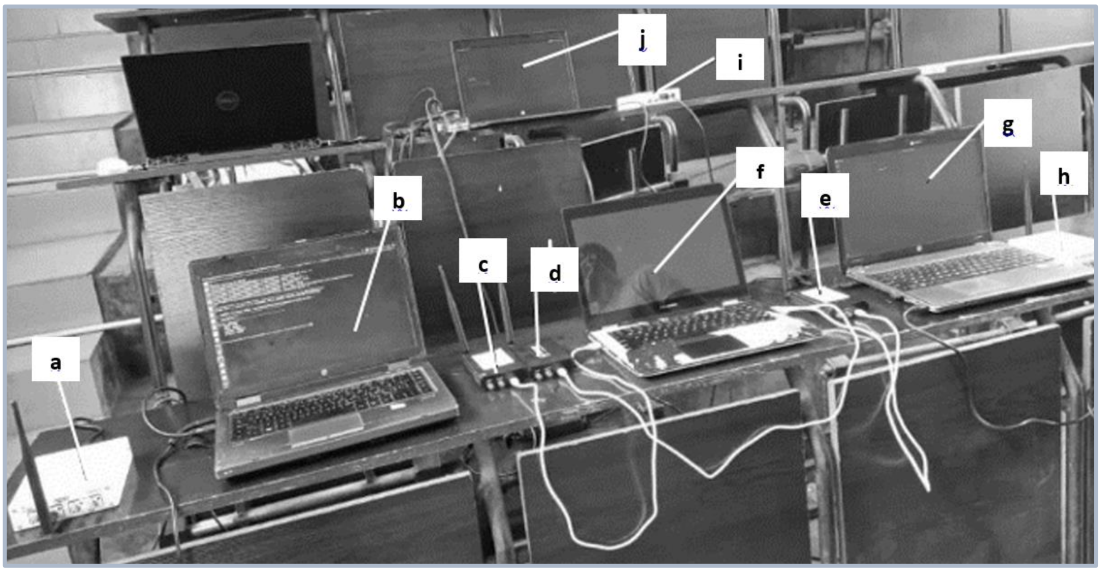

Fig. 3. The Experimental setup: a) the cognitive radio transmitter USRP B210; b) the computer running the transmitter block program; c, d, e) the HackRF Ones generating the Pus; f) the computer running the python program that generates the PUs' signals; g) the computer running the receiver block program; h) the cognitive radio receiver USRP B210; i) the spectrum sensing module USRP B210; j) the computer running the spectrum sensing and channel server program.

The setup comprises three USRP B210s, four computers, and three HackRF One SDRs. The HackRF one devices were used to generate the PUs of the model, while the three USRP B210 formed the complete cognitive radio node that comprises the transmitter, receiver, and spectrum sensing. Two computers were used to run the transmitter and receiver programs of the testbed while a third computer ran both the channel server and the spectrum sensing programs. All three computers were connected to a Wi-Fi network used to transmit the TCP protocol signals (channel assignment and channel request). Lastly, the fourth computer ran the program for generating the PU signals.

The steps taken to execute a demonstration included first running the channel server program, followed by initiating a communication link between the transmitter and receiver (TCP clients) at a fixed frequency and that involved creating a video source pipe using the $mkfifo$ command, then running the transmitter and receiver programs, and lastly the player command in GStreamer. Once the communication link was established, we introduced the PUs to the network to kick off the dynamic spectrum access mode.

In an ideal case, the steps would start up the real-time streaming and demonstrate dynamic channel access. However, in the experiment, running the programs to initiate a communication link between the transmitter and receiver ends was a little challenging majorly due to hardware constraints (mismatch in the processing speed of the computer and the SDRs). Thus, there were iterations made with the encoding rates, and sampling rates to manage the constraints. Also, the performance of the DSA mode was suboptimum at the earlier stages of the experiments and improved as we made more iterations in the experiment.

4. RESULTS AND DISCUSSION

Primarily, the channel selection algorithm depends on the information stored in the long-term database. The information stored during the experiment is given in Table 1. It is the information obtained from the spectrum for the nine PUs. Every entry in the database was the result of a complete sweep across the 10MHz band by the spectrum sensor measuring the power level. For 101 scans through the band, the Table shows the average power level, the number of occupied states in a channel and the resulting occupancy or usage rate (duty cycle).

The short-term (ST) database stores the channels with the least occupancy from the long-term database. The ST in the experiment was a list in the cognitive engine program. These channels were the ones that were sensed when a channel is needed to allow the cognitive radio to transceiver and when a request is made by the buffer system.

In the experiment, the two best channels were channel 1 and channel 2 at frequencies 900.56 MHz and 901.68 MHz respectively. As seen in Table 1, they have the least occupancy. These channels were consistently used by the algorithm to transmit and receive the live video shown in Figs. 5 and 6. This record is also seen in the spectrogram plot shown in Fig. 4 of the 10MHz band during one of the experiments that were taken. In this figure, the difference in the power levels between the PUs and the cognitive node in the network can be seen and the node majorly switches between channels 1 and 2. The analysis was done on two minutes of stored data to show the activity of the PUs and SU in the testbed during the experiment. The power levels are shown by the intensity of the colors.

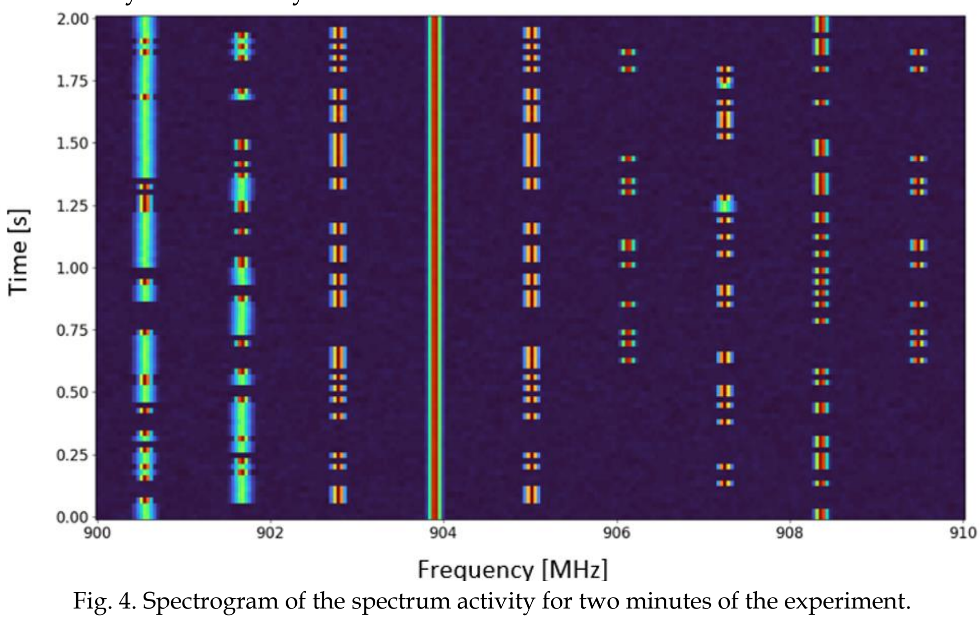

Fig. 4. Spectrogram of the spectrum activity for two minutes of the experiment.

The high-power signal is the PUs while the low power signal is the SU. Furthermore, the plot shows that the first two channels were the most used for continuous transmission by the SU. Fig. 5 shows the process of data processing from the webcam to the USRP B210 device. The cognitive radio starts transmitting once a centre frequency is received from the cognitive engine via TCP protocol. As seen from the figure, the two frequencies that were mostly selected by the channel selection algorithm were 900.56 MHz and 901.68 MHz. These were the two best frequencies that were stored in the short-term database located in the engine.

Table 1. Summary of information on the nine channels in the long-term database.

Centre frequency [Hz]	Power [dB]	Number of scans	Number of Occupied states	Occupancy
900562176	16.75576558	101	2	0.0198019802
901686528	21.87244458	101	1	0.009900990099
902810880	20.87074651	101	9	0.08910891089
903935232	51.43200968	101	101	1
905059584	26.04079116	101	9	0.08910891089
906183936	16.66641332	101	3	0.0297029703
907308288	17.25246411	101	5	0.0495049505
908432640	17.09035323	101	4	0.0396039604
909556992	18.31899325	101	3	0.0297029703

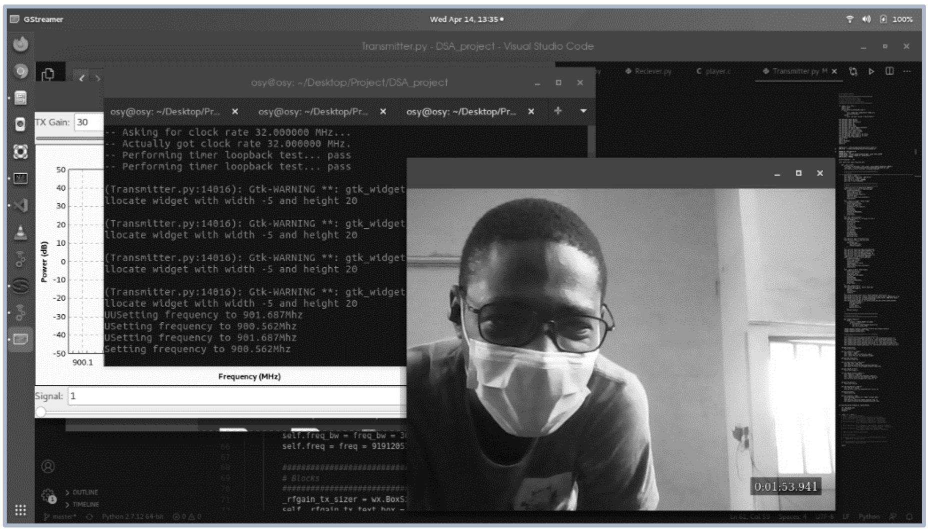

Fig. 5. The video capture and transmission running on the computer which runs the transmitter block program.

Fig. 6 shows the receiver block which also is automatically set to the same frequency as the transmitter received via TCP protocol, receives the video packets, and displays them using GStreamer.

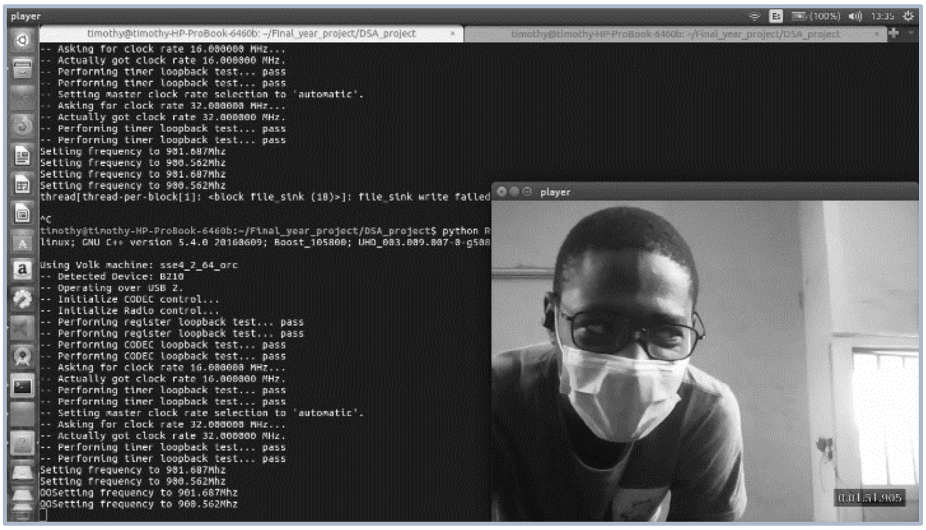

Fig. 6. The receiver block program running and displaying the received video packets with GStreamer.

The buffer system also supplements the channel selection algorithm to improve the quality of the video that is being streamed. The buffer size in GStreamer is 100 buffers, 2MB of data, or two seconds worth of data, whichever is reached first.

Implementing a buffer provides two advantages to the cognitive radio node; reducing interference with the primary users and identification of noisy channels. Though the experiment was carried out in a "free" band, there were no measures taken to ensure that the band was indeed free from extraneous signals. The interference of external signals with significantly low energy could coexist with the cognitive radio signal and therefore invisible to the sensing module. The buffer system helped detect such scenarios because its primary goal is to ensure the continuous reception of transmitted packets. Fig. 7 shows the standard output of the C executable file that implements the buffering and playback of received packets from GNU Radio. The buffer requests for a new channel from the cognitive engine and pauses the video when a threshold of 85% is reached.

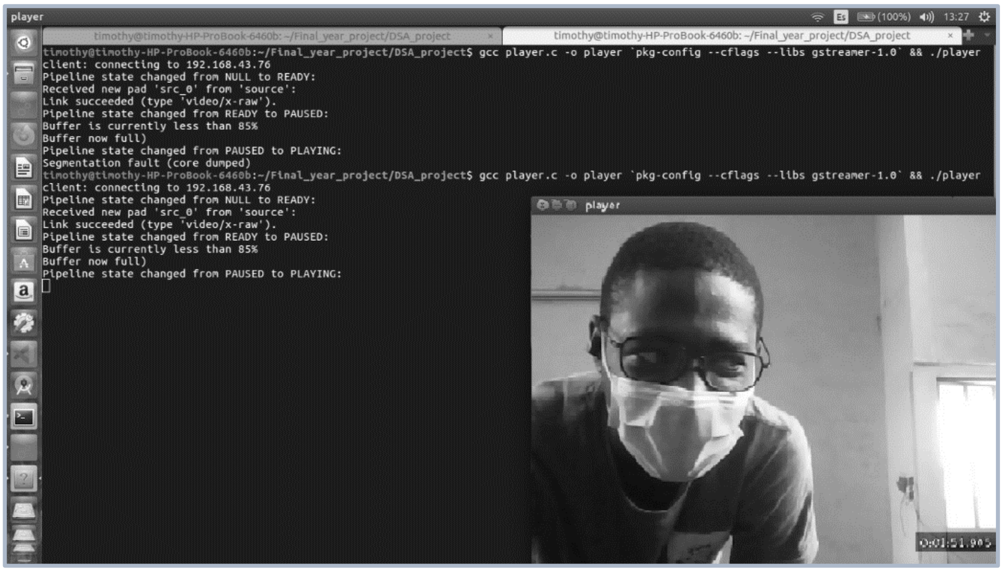

Fig. 7. Buffering in video streaming with the cognitive radio node.

A joint short-term and long-term database use has the combined benefit of reducing the sensing time by prioritizing the channels and allowing classification and prediction in the bands of interest [10, 19]. These benefits aid an improved operation of the cognitive radio by increasing the throughput and reducing the interference and sensing time. Though the goal of the research was to consolidate the works of literature and measure the impact on radio resource use, sensing time is a very crucial part of the operation of a cognitive radio and a cognitive radio is only as good as its sensing module. Thus, we also analyzed the impact of the design using a long-term and short-term database on the sensing time during the experiment. The long-term database was the accumulation of all the results from each of the experiments while the short-term database stored the prioritized channels for immediate use upon request but was sensed to ensure no interference to PUs. Instead of the cognitive engine always having to scan through all nine channels to get a free channel, it just queries the long-term database to get the two best channels having the least occupancy to use. These two channels are then stored in the cognitive engine and are sensed when a new channel is needed. This procedure, therefore, is characterized by a pronounced shift from sensing nine channels to sensing approximately two channels which underscores a substantial reduction of approximately 77.8% in sensing time assuming sensing time is constant for all channels.

In terms of radio resource usage, the use of the procedure records an increase of about 32.3% as shown in Fig. 8. The analysis was done on 2 minutes of the stored data from the experiment with the PUs transmitting alone in the spectrum which constitutes the non-DSA mode of operation and when the SU joins the network (the DSA mode). The occupancy was computed as the number of samples above a predetermined threshold and both modes of the experiment record a relatively low occupancy of approximately 6% and 7.97% for the non-DSA and DSA mode respectively. This is a result of white holes in the band shown in Fig. 4 between the active frequencies.

```
No DSA
Occupancy: 6.0274920886075956 %
DSA
Occupancy: 7.973090277777778 %


Occupancy gain: 32.27873484641327 %
```

Fig. 8. Radio resource usage gain comparing a testbed experiment with DSA and non-DSA mode of operation.

5. CONCLUSIONS

This investigation successfully implemented a cognitive radio node that uses a smart channel selection technique while operating in a shared radio environment. We recorded continuous streaming of a real-time video signal for the radio node in the testbed network of nine channels with different channel usage. With the design, the radio applied dynamic spectrum access using the long-term database of sensed data to prioritize the channels and the short-term database to reduce the sensing time. In addition to opportunistic channel access using the spectrum sensor to detect the return of the PU and issue a channel switch, the use of a buffer system as a feedback mechanism on the selected channel optimized the channel selection. In addition to opportunistic channel access using the spectrum sensor to detect the return of the PU and issue a channel switch, the use of a buffer system as a feedback mechanism on the selected channel also optimized channel selection by observing the received signal and requesting a channel switch when a limit is reached. The proposed design recorded an improvement in the use of the spectrum in the band of the testbed and a decrease in the sensing time.

Acknowledgement: The authors would like to thank Azuka Chiejina for his efforts towards the completion of the research.

REFERENCES

[1] A. Shamsoshoara, F. Afghah, A. Razi, S. Mousavi, J. Ashdown, K. Turk, “An autonomous spectrum management scheme for unmanned aerial vehicle networks in disaster relief operations,” *IEEE Access*, vol. 8, pp. 58064–58079, 2020, doi: 10.1109/ACCESS.2020.2982932.

[2] G. Liu, D. Jiang, “5G: Vision and requirements for mobile communication system towards year 2020,” *Chinese Journal of Engineering*, vol. 2016, 2016, doi: 10.1155/2016/5974586.

[3] D. Roy, T. Mukherjee, M. Chatterjee, E. Pasiliao, "Adaptive video encoding and dynamic channel access for real-time streaming over SDRs," 37th International Performance Computing and Communications Conference, 2018, doi: 10.1109/PCCC.2018.8710795.

[4] M. Admin, "Wireless innovation forum CBRS, SDR and spectrum sharing standards," *winnforum and the Marconi Society*, 2022. http://www.wirelessinnovation.org.

[5] M. Hassan, G. Karmakar, J. Kamruzzaman, B. Srinivasan, "Exclusive use spectrum access trading models in cognitive radio networks: a survey," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2192–2231, 2017, doi: 10.1109/COMST.2017.2725960.

[6] R. Hinman, "Application of cognitive radio technology to legacy military waveforms in a JTRS (joint tactical radio system) radio," IEEE Military Communications conference, 2006, doi: 10.1109/MILCOM.2006.302522.

[7] Q. Mahmoud, *Cognitive Networks Towards Self-Aware Networks*, Wiley & Sons Inc, 2007.

[8] J. Mitola, *Cognitive Radio Architecture*, in Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems, Dordrecht: Springer Netherlands, 2007.

[9] M. Höyhtyä, J. Korpi, M. Hiivala, "Predictive channel selection for over-the-air video transmission using software-defined radio platforms," Cognitive Radio Oriented Wireless Networks, 2016, doi: 10.1007/978-3-319-40352-6_47.

[10] M. Hoyhtya, J. Vartiainen, H. Sarvanko, A. Mammela, "Combination of short term and long term database for cognitive radio resource management," 3rd International Symposium on Applied Sciences in Biomedical and Communication Technologies, 2010, doi: 10.1109/ISABEL.2010.5702799.

[11] H. Thien, H. Van, I. Koo, "Implementation of spectrum sensing with video transmission for cognitive radio using USRP with GNU radio," *International Journal of Internet, Broadcasting and Communication*, vol. 10, no. 1, pp. 1–10, 2018, doi: 10.7236/IJIBC.2018.10.1.1.

[12] L. Fa, M. Yongkui, Z. Honglin, D. Kai, "Evolution handoff strategy for real-time video transmission over practical cognitive radio networks," *China Communications*, vol. 12, no. 2, pp. 141–154, 2015, doi: 10.1109/CC.2015.7084409.

[13] A. Amjad, T. Sikandar, I. Muddesar, F. Li, R. Imran, S. Muhammad, K. Ali, "Adaptive bitrate video transmission over cognitive radio networks using cross layer routing approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 3, pp. 935–945,2020, doi: 10.1109/TCCN.2020.2990673.

[14] *GStreamer*, 2022. https://gstreamer.freedesktop.org/documentation.

[15] *GNU Radio Wiki*, 2022. https://wiki.gnuradio.org.

[16] M. Tahir, H. Mohamad, N. Ramli, S. Jarot, "Experimental implementation of dynamic spectrum access for video transmission using USRP," International Conference on Computer and Communication Engineering, 2012, doi: 10.1109/ICCCE.2012.6271185.

[17] F. Zimmerle, "gnuradio/gr-uhd/examples/python/usrpspectrumsense.py at master zimmerle/gnuradio," 2022, https://github.com/zimmerle/gnuradio/blob/master/gr-uhd/examples/python/usrp_spectrum_sense.py.

[18] N. Bello, A. Muhammed, " Effect of buffer size variation on video quality transmission in a cognitive radio network," *Journal of Civil and Environmental Systems Engineering*, Vol. 19, No. 1, pp 34 -39, 2022.

[19] M. Höyhtyä, J. Korpi, M. Hiivala, "Predictive channel selection: Practical implementation and a social-aware vision for spectrum use," *EAI Endorsed Transactions on Cognitive Communications*, vol. 3, no. 10, p. 152186, 2017, doi: 10.4108/eai.23-2-2017.152186.