Adaptive Video Encoding and Dynamic Channel Access for Real-time Streaming over SDRs

Debashri Roy*, Tathagata Mukherjee†, Mainak Chatterjee*, Eduardo Pasiliao‡

* Computer Science
University of Central Florida
Orlando, FL 32826
{debashri, mainak}@cs.ucf.edu

† Computer Science
University of Alabama
Huntsville, AL 35899
{tm0130}@uah.edu

‡ Munitions Directorate
Air Force Research Laboratory
Eglin AFB, FL, 32542
{eduardo.pasiliao}@us.af.mil

Abstract*—In this paper we study and implement real-time adaptation schemes for video encoding and channel selection that work in tandem to facilitate HD video streaming for secondary users in a dynamic spectrum access network. Out-of-band feedbacks on instantaneous pathloss of the signal between the transmitter and the receiver, the received signal strength indicator (RSSI) at the receiver and the quality of the reconstructed video are used to continuously determine the most apt encoding parameters. At the same time, the radio transmitter continuously adjusts the channel parameters (i.e., center frequency and channel bandwidth) based on the transmission activities of the primary users who have prioritized rights on these channels. We consider the physical limitations of the encoder along with the channel statistics to determine *when* to change the encoder parameters and *when* to switch to a new channel. We propose a multi-level threshold based mechanism to find the optimal number of encoding bit rates. We also propose a threshold based algorithm to find the best available channel between the transmitter-receiver pair. We validate our theoretical propositions on an indoor testbed using software defined radios (SDRs) and the GNU Radio suite. Live video was captured, encoded using open source H.264 software libraries, streamed using GStreamer and transmitted over the 915 MHz ISM bands with omni-directional antennas. For the SDRs, we chose the universal software radio peripheral (USRP) B210s from Ettus Research and use them as the transmitter and the receiver. A third B210 was used to sense the energy levels on all the channels to detect the presence of primary transmissions. GNU Radio was used to build the signal processing pipeline, both for the transmitter and the receiver. We use PSNR and SSIM to measure the video quality and report experimental results that show that: (i) the video encoder and the USRP transmitter-receiver pair are able to adapt to the changing RF conditions, (ii) the adaptation schemes yield better video quality than non-adaptive schemes, and (iii) the USRPs can switch the channels fast enough allowing uninterrupted HD video streaming even when primary users preempt the secondary user's transmission.

I. Introduction

With the increasing demand for high definition (HD) video over wireless access networks, the wireless service providers are finding it difficult to handle the large volumes of the video traffic [1]. In recent years, the research community on video processing and delivery has contributed towards implementing sophisticated encoding techniques that ensure minimal storage and are resilient against transmission losses [2], [3]. Results have shown that knowledge about network status and coordination among users that share resources result in better encoding decisions and overall utilization of the bandwidth. In spite of these developments, guaranteeing QoS over wireless networks has always been a challenge due to the unpredictable channel characteristics [4]. The problem is even more aggravated for *real-time* video transmission because of the strict inter-packet delay and bandwidth requirements. Alongside, to alleviate the problem of bandwidth shortage in some of the spectrum bands, the concept of dynamic spectrum access (DSA) has been proposed and implemented [5], the objective of which is to efficiently manage the radio spectrum and still provide the quality of service (QoS) that the end users expect.

In a DSA network, secondary users (i.e., unlicensed users) typically equipped with SDRs continuously monitor the presence of primary users (i.e., licensed users) and opportunistically access the unused or under-utilized licensed bands of primaries [6]. SDRs define different hardware components through software implementations which can be programmed to continuously adapt to the changing RF environment by adjusting the waveforms, modulation schemes, filters, and other RF parameters. Technologies based on the DSA paradigm provide some relief to the bandwidth shortage problem; however it is yet to be seen what kind of data-intensive applications can the software defined radios handle. In particular, if they can efficiently handle HD video transmissions in a contested environment. Also, it is yet to be seen how well and how fast can the radios adapt to the changing network and channel conditions.

One of the motivations of this work is to investigate to what extent can video source coding and wireless transmissions be jointly optimized, given the network latencies and hardware constraints. Though there are some work that deal with joint source and channel coding [2] [7], they are mostly confined to the theoretical domain. Also, RF parameter adaptations for DSA systems do not consider the QoS vector of the application being supported [8], [6]. We not only provide a theoretical groundwork for joint video coding and channel selection for maintaining QoS, but also implement them on video encoders and USRPs (a software defined radio manufactured by Ettus Research) to demonstrate their practical feasibility.

In this paper, we propose and implement adaptive video encoding and channel adaptation schemes on an indoor testbed. Using an out-of-band real-time feedback from the receiver about the video quality, the encoder continuously adapts the

978-1-5386-6808-5/18/$31.00

encoding bitrate. Also, the transmission parameters, particularly the center frequency and the channel bandwidth, are changed based on the SINR at the receiver and the transmission activities of the primary users. The objective of the proposed adaptation techniques is to keep the video streaming uninterrupted by changing the source coding and channel parameters, based on QoS at the receiver. Our novelty lies in having the H.264 codec to continuously tune the parameters at any time granularity *and* for the radio transmitter to change the channel parameters simultaneously. Though extremely fast adaptation is desirable and have been proposed in theory [2], [9], in practical scenarios, we are limited by the hardware capabilities. Thus, instead of changing the encoding rates at infinitesimally small time scales, we use a multi-level threshold based mechanism for determining *when* to change the encoding rate and *what* that rate should be. The multi-level threshold is based on the probability density function (pdf) of the received power at the receiver. Moreover, we tune the transmitting frequency to the most desirable channel between the transmitter-receiver pair based on the sensed RSSI at the receiver. If the current channel is found to degrade the video quality, the transmitter and receiver switch to the next best channel.

Our theoretical propositions have been validated on an experimental testbed using USRP and GNU Radio. We used the 902 - 926 MHz ISM band to create 8 channels of 3 MHz each. Live video captured via a webcam was encoded using open source H.264 software libraries and streamed using GStreamer [10]. The video transport streams were transmitted using USRP B210 which used Gaussian Minimum Shift Keying (GMSK) modulation and omni-directional antennas. The encoding rate and the transmitting frequency were changed based on the reconstructed video quality measured using PSNR and SSIM. Results showed that the proposed encoding and channel adaptation techniques were able to substantially improve the quality of video displayed at the receiver. To the best of our knowledge, this paper is the first to propose and implement source encoding and channel adaptation schemes with a real-time feedback control mechanism in the DSA paradigm. Insights from the results reveal that leveraging off-the-shelf USRPs for real-time HD video transmission would ease the over-burdened radio resources.

The rest of the paper is organized as follows. Related works with respect to video transmission over wireless networks are discussed in Section II. System model and video assessment techniques used are presented in Section III. The proposed approach for video encoding and channel adaptation techniques are discussed in Section IV. Testbed implementation details are described in Section V. Experimental results are presented in Section VI. Finally, conclusions are drawn in the last section.

II. Related Work

This section summarizes the studies evaluating different source channel encoding techniques and video quality assessments in the DSA paradigm. Software defined radios have emerged as a promising solution for solving the spectrum scarcity problem and improving the spectral efficiency by leveraging the concept of spatial reuse of radio bands [11].

In [12], the authors prove that maximizing the number of served multimedia streams in a single-frequency network is NP-Complete and propose a heuristic algorithm with polynomial running time. Efficient utilization of the radio bands depends on how well the primary users (PU) and secondary users (SU) are able to coordinate their transmission activities. Many PU and SU activity models have been reported in [13]. In one such study, the primary user activity pattern has been assumed to follow a Poisson model [14]. However, the Poisson model has drawbacks of achieving lower throughput and high false-alarm rates. In [15], a parameterized primary user traffic model has been proposed by forming cluster of correlated primary users. In [4], the spectrum opportunities for secondary users based on primary users activity patterns was investigated. In [16], the authors modeled the primary user activity as a Semi-Markov ON/OFF process.

In [17], the authors proposed three types of channel selection schemes: best-fit channel selection (BFC), longest idle time channel selection (LITC), and p-selfish channel selection. The authors proved that non-selfish selection improves performance over selfish selection schemes. Both PU and SU activity models were considered to be 3-state Markov chain. A distributed channel selection scheme was proposed in [18] for efficient data dissemination in a multi-hop cognitive radio network, based on channel classifications. Authors in [19] tried to overcome the bandwidth requirement limitation for mobile videos by proposing a cross-layer design. An experimental deployment of a DSA system for video transmission using USRPs and GNU Radios was presented in [6]. It showed DSA based systems perform better than non-DSA ones. However, this work did not use any video quality metrics as we have done. A system level solution for a database-assisted DSA system using inexpensive software configurable RF chips was proposed in [8].

As far as video adaptation techniques over wireless access networks are concerned, there have been some breakthroughs. A video quality assessment technique, for different kinds of wireless networks in high mobility environments, was presented in [20]. In [2], the authors presented a popularity-aware joint source-channel coding optimization framework for video plus depth format. They proposed a framework to generate a rate embedded bitstream which would be optimally decodable at multiple data rates depending on the diversity of video perspectives. Another joint source-channel coding (JSCC) approach was proposed in [3] for 3-D stereo video transmission using video plus depth. The authors considered flat Rayleigh fading to model the noisy channel. In [9], the authors implemented variable-length codes with feedback to improve the delay-distortion trade-off and power constraints.

Most of the existing JSCC techniques focus on the various video coding standards; they do not necessarily consider the networking and radio parameters. Jointly optimizing source encoding parameters and RF parameters on a real wireless platform for real-time video streaming remains a challenge

because of (i) the limited bandwidth for a large number of users; (ii) non-reliable channels; (iii) hardware limitations to adapt changing channel conditions; and (iv) video quality assurance. Sustaining video quality along with high spectral efficiency for an over-burdened network is very desirable from both video and radio resources' perspective.

III. System Model and Assumptions

In this section, we present the wireless channel models, and the necessary background on video encoding and quality assessment techniques that are relevant for the rest of this work.

A. Channel Model

A good channel model must incorporate important features like pathloss, shadowing, and fading. We considered a wireless channel with multiple transmitters and receivers, with line of sight (LOS), primarily to mimic the testbed conditions discussed later. Next we discuss our assumptions for the wireless channel.

1) Pathloss Model: The received power depends on the distance between the transmitter and the receiver, denoted by d and is given by:

$$P_r(d) = P_t \times \left(\frac{\lambda}{4\pi d_0}\right)^2 \times \left[\frac{d_0}{d}\right]^{\gamma} \quad (1)$$

where P_t is transmitted power, γ is the pathloss exponent, λ is the wavelength in meters and d_0 is the reference distance for antenna far field which is usually 1 to 10 meters for indoor environments [11]. The corresponding pathloss in decibel is given by:

$$PL_{dB}(d) = 20\log\left(\frac{\lambda}{4\pi d_0}\right) + 10\gamma\log\left[\frac{d_0}{d}\right] \quad (2)$$

2) Shadowing and Fading Model: We assume that signals are affected by shadowing and fading that capture the multi-path effect for wireless channels. The received power envelop (P_s) due to shadowing is a random variable, the logarithm of which is a zero-mean Gaussian distribution [11] with density function given by:

$$f(ln(P_s)) = \frac{1}{\sqrt{2\pi \times \delta_s^2}} \exp\left(-\frac{(ln(P_s))^2}{2\delta_s^2}\right) \quad (3)$$

where δ_s^2 is the variance of P_s in dB^2.

We use Rician fading which is the most appropriate channel model for line of sight [21] scenarios. Here the received signal amplitude distribution is given by: $P(r) = \frac{r}{\delta_f^2}\exp\left[-\frac{r^2+s^2}{2\delta_f^2}\right] I_0\left(\frac{rs}{\delta_f^2}\right)$, where $s \geq 0$, and $r \geq 0$. I_0 is the zero order Bessel function of the first kind, δ_f^2 is the power contribution of scattered paths at the received signal, s is the peak amplitude of the dominant path, and r is the channel's faded amplitude [21].

For a Rician fading channel, the signal amplitude follows a Gaussian distribution. We assume that the power loss due to Rician fading, denoted by x, also follows a Gaussian distribution with density:

$$\psi(x) = \frac{1}{\sqrt{2\pi\sigma_s^2}} \exp -\frac{(x-\mu_d)^2}{2\sigma_s^2} \quad (4)$$

where μ_d is the mean powerloss at distance d, and σ_s is the variance.

B. Video Encoding

A continuous video stream typically has spatial (similarities with neighboring pixel blocks), temporal (similarities with pre-arrived frames), statistical, and perceptual redundancies. The video codecs implement efficient and pertinent data encoding by exploiting the different redundancies present in the source video. Most codecs cannot continuously tune the coding parameters based on network and channel conditions at any arbitrarily small time interval. This is because: (i) there is a finite delay between the moment the receiver sends a feedback and the moment the encoder acts upon it, and (ii) the hardware or software based video codecs have their fundamental limits on operational speeds.

If the channel conditions change during encoder adaptation, the codec adaptation is not effective. The situation is worse for fast fading channels. However, adaptation benefits can be achieved for slow fading channels. We consider both slow and fast fading.

C. Video Quality Assessment

Though there are several metrics to measure the video quality, we use two simple video quality assessments: Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [22]. PSNR is the most widely used video quality metric and SSIM is the most accurate to human perception [22]. PSNR relates to the quality of reconstructed video from lossy compression codecs. Higher PSNR generally refers to better video quality. PSNR values ranging from 30 to 50 represent good quality video with respect to wireless communication [23]. Measurement of SSIM relies on quantizing structural distortion rather than error. The motivation for this metric is that human vision responds positively to structural similarity than errors, therefore SSIM gives a better correlation to the human subjective impression. Good SSIM for wireless communication ranges between 0.8 - 1.0 [24].

IV. Proposed Adaptation Schemes

We propose two adaptations that work in tandem. The first one adapts the video encoding parameters in response to the channel conditions at the receiver such that the video streaming remains uninterrupted. This is achieved through a feedback control mechanism from the receiver to the encoder. As for finding the best encoding rate at any time, we use a multi-level threshold based mechanism where the thresholds are computed using the probability density function of the received power at the receiver. The second adaptation lies in continuously finding the best channel for the secondary transmitter-receiver pair. As in a DSA based system, when the currently used channel is found to be noisy or unusable due to return of the primary, both the transmitter and the receiver switch to another channel. In Section IV-A, we discuss the proposed encoder adaptation while in Section IV-B, we present the adaptive channel selection method.

A. Encoder Adaptation

We assume that the pdf $\psi(x)$ (recall that x is the power loss) has a mean of μ_d, and a standard deviation σ_s ($\sim$ [-2.6134 to 2.6134] dB [11]), for indoor LOS. It is intuitive to conclude that the encoding rate should be decreased (increased) as the channel conditions deteriorate (improve). However it is not immediately clear: (i) how frequently should the encoding rates be changed, and (ii) how do we quantitatively find the encoding rates. Our objective here is to adapt the video encoding parameters based on the channel conditions.

1) Frequency of "Encoding Rate" Changes: Theoretically the encoding rate should be changed as frequently as possible in response to the continuously changing channel condition. However, the hardware might not be able to handle such quick transitions. The rate at which the encoding bitrate can be changed is limited by: i) the hardware capabilities of the encoder, streamer, signal processor and SDR, and ii) the time lag for executing the feedback loop. Hence we seek to find the frequency at which the encoding rate must be changed such that it is the optimal response to the channel conditions within the capacity of the underlying hardware.

It must be noted that the rate of change of the encoding bitrate is dependent on how fast the powerloss is changing. The mean powerloss (μ_d) at a distance (d) for the distribution $\psi(x)$, from equation 4, is given by:

$$\mu_d = 20\log\left(\frac{\lambda}{4\pi d_0}\right) + 10\gamma\log\left[\frac{d_0}{d}\right] \tag{5}$$

Suppose ε_{MIN} and ε_{MAX} are the minimum and maximum encoding bit rates that a given encoder can support. Based on the distance between the transmitter and receiver, the encoder will use an encoding rate $[\varepsilon_{min}(d),\, \varepsilon_{max}(d)]$, where $\varepsilon_{min}(d)$ and $\varepsilon_{max}(d)$ are the minimum and the maximum encoding bitrates for distance d, respectively. Of course, $\varepsilon_{MIN} \leq [\varepsilon_{min}(d),\, \varepsilon_{max}(d)] \leq \varepsilon_{MAX}$. We formulate $\varepsilon_{min}(d)$ and $\varepsilon_{max}(d)$ as:

$$\varepsilon_{min}(d) = \begin{cases} \varepsilon_{MIN} & \text{for } d \leq 1 \\ \dfrac{\varepsilon_{MIN} \times \mu_{\psi(d=1)}}{\zeta(\mu_d)} & \text{for } d > 1 \end{cases} \tag{6}$$

$$\varepsilon_{max}(d) = \begin{cases} \varepsilon_{MAX} & \text{for } d \leq 1 \\ \dfrac{\varepsilon_{MAX} \times \mu_{\psi(d=1)}}{\zeta(\mu_d)} & \text{for } d > 1 \end{cases} \tag{7}$$

where $\mu_{\psi(d=1)} = \left[20\log\left(\frac{\lambda}{4\pi d_0}\right) + 10\gamma\log d_0\right]$ is the powerloss at a distance of 1 meter. The intuition behind equations 6 and 7 is that the encoding rates could be the best possible in the near field (i.e., $d \leq 1$). However, as d increases, the channel cannot sustain a higher bit rate and hence, the bit rate should be decreased. Though $\zeta(.)$ could be any increasing function of μ_d, for the sake of simplicity, we use a linear function for $\zeta(.)$ in our experiments (as discussed in Section V).

Once the range of the encoding rates are obtained (i.e., $[\varepsilon_{min}(d),\, \varepsilon_{max}(d)]$), we want to determine the number of discrete encoding bitrates in that range. In essence, we seek to find the quantized encoding rates that are optimized for the given $\psi(x)$.

The Optimization Problem: Suppose, we divide the encoding bit rate range into L partitions depending on $\psi(x)$. If L is too large, then the changes in encoding bit rate would be too frequent (contradicting overall hardware and timing lag constraints). If L is too small, then the changes in encoding rate would be less frequent, hence video quality would be less adaptive with respect to the channel conditions. Thus we need to optimize L so that we can ensure better video quality which adapts to the channel conditions and adheres to the hardware constraints. The discrete encoding rates should not be too close or too far from each other.

As $\psi(x)$ is Gaussian, most of the time the powerloss would be between $\mu_d \pm \sigma_s$. This phenomenon asserts the requirement of a *non-uniform* partitions for mapping the channel to the source coding. We propose to partition the density function such that each partition has the same probability mass. As $\int_{-\infty}^{\infty}\psi(x) = 1$, we seek to find L partitions with powerloss levels $x_1, x_2 \cdots, x_i, \cdots, x_L$ such that:

$$\int_{x_1=-\infty}^{x_2}\psi(x) = \int_{x_2}^{x_3}\psi(x) = \cdots = \int_{x_L}^{\infty}\psi(x) = \frac{1}{L} \tag{8}$$

Here, $x_1 = -\infty$ is possible as the powerloss is measured in dB, which can be negative. Also the last partition is open-ended, as powerloss can in theory be ∞.

Let us assume υ_{MIN} and υ_{MAX} are the minimum and maximum desired gap between two encoding rates, which are fixed for a given encoder. The encoding rate gap (υ_i) is calculated as:

$$\upsilon_i = \left(\frac{\varepsilon_{max}(d) - \varepsilon_{min}(d)}{L-2} \times \frac{x_i - \mu_d}{\sigma_s}\right),\ 2 \leq i \leq (L-1) \tag{9}$$

It is to be noted, that υ_{MIN} will prevent large L and υ_{MAX} will prevent small L. Thus, we get a range for the values of L that satisfies

$$\upsilon_{MIN} \leq \left(\frac{\varepsilon_{max}(d) - \varepsilon_{min}(d)}{L-2} \times \frac{x_i - \mu_d}{\sigma_s}\right) \leq \upsilon_{MAX} \tag{10}$$

Though any value of L within the range would suffice, we seek the minimum L so as to minimize the frequency of encoding rate changes that still adapts to the channel fluctuations. Thus, we can formulate the problem as the a constrained optimization problem as follows:

$$\begin{aligned} \underset{\forall i}{\text{minimize}} \quad & (\text{L}) \\ \text{subject to} \quad & Equation\ 8 \\ & Equation\ 10 \quad \text{for } 2 \leq i \leq (L-1) \end{aligned}$$

This objective function can be solved by non-linear integer programming (NIP) to get the optimal L for a given $\psi(x)$ and the hardware limitations. However, solving NIP is a NP Hard problem [25]. Since we have some intuition for the value of L (i.e., less than 20), we simply try all values of L starting from 2 for fast processing.

Note that even if we were to change the other encoding parameters, our formulations would still remain the same. In

that case ε_{MIN}, ε_{MAX}, υ_{MIN}, and υ_{MAX} will have to be updated based on the parameters selected.

2) Quantitatively Finding the Encoding Rates: Once L is known, we want to determine the exact values for the different encoding bitrates for encoder adaptations. As per equation 8, the density of the power loss is Gaussian. However, our optimization technique is applicable for any other distribution. While optimizing L, we get the values of $x_2, \cdots, x_i, \cdots, x_L$ using non-linear integer programming. To map the source coding parameters, we can get encoding bitrate ε_i for range between x_i and x_{i+1} as:

$$\varepsilon_i = \varepsilon_{i-1} + \upsilon_i \quad \text{for } 2 \leq i \leq (L-1) \tag{11}$$

where $\varepsilon_1 = \varepsilon_{min}(d)$, and $\varepsilon_L = \varepsilon_{max}(d)$.

B. Channel Adaptation

Channel selection by the secondary transmitter-receiver pair and subsequent transmissions will depend on the PU activities. First, let us discuss PU activity and then present how SUs can avail the channels, not used by PUs.

1) PU Activity Model: We consider a 2-state ON-OFF Markov model for the primaries. During the ON state, the secondaries must remain inactive. However, during the OFF state, the secondaries can choose to either remain idle or transmit.

For the PUs, let the transition rate from OFF state to the ON state be α and transition rate from the ON state to the OFF state be β. Then the probability of the PU being in the ON state is given by: $P_{ON} = \alpha/(\alpha+\beta)$. Similarly, the probability of PU being in the OFF state is, $P_{OFF} = \beta/(\alpha+\beta)$. Here the ON and OFF periods are assumed to be exponentially distributed with means $1/\beta$ and $1/\alpha$ respectively.

The cumulative distribution function (CDF) of the PU being in the ON state for time t_{on} is $F_{ON}(t) = P_{ON}(t \leq t_{on}) = 1 - exp(-\beta t_{on})$. Similarly, the CDF of the PU being in OFF state for time t_{off} is $F_{OFF}(t) = 1 - exp(-\alpha t_{off})$.

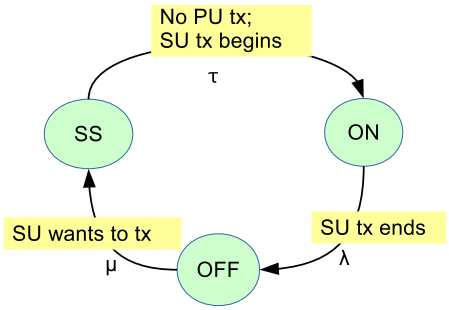


Fig. 1. 3-state Markov Model for SU Activity

2) SU Activity Model: Secondary users follow a 3 state Markov model [17] as shown in Fig. 1. The three states are: ON (SU transmitting), OFF (SU idle), and SS (Sensing State). If we assume the arrival and departure rates of different states according to Fig. 1, then the probability of SU being in the ON state is $P_{S_{ON}} = \tau/(\tau+\lambda)$; being in the OFF state is $P_{S_{OFF}} = \lambda/(\mu+\lambda)$; and being in the SS state is $P_{S_{SS}} = \mu/(\mu+\tau)$. The CDF of the SU being in the ON state for time $t_{S_{on}}$ is $F_{S_{ON}}(t) = 1 - exp(-\lambda t_{S_{on}})$. CDF of SU being in OFF state for time $t_{S_{off}}$ is $F_{S_{OFF}}(t) = 1 - exp(-\mu t_{S_{off}})$. Similarly, the CDF of the SU being in the SS state for time $t_{S_{ss}}$ is $F_{S_{SS}}(t) = 1 - exp(-\tau t_{S_{ss}})$, assuming exponential distribution as before.

Channel_Selection( )

Input : Set of PU Channels (PU), Energy Threshold for SU Channel (Th_{SU}), Threshold Increment (Th_{inc})
Output: Selected PU Channel ($PU_{Selected}$)

```
begin
    N = No_of_PUs(PU);
    for i ≤ N do
        E[i] = Energy (PU[i]);
        T[i] = PU_availability (PU[i]); /* From Sec.
         IV-B1*/
        StoreT[i] = T[i]; /*Stores the initial values of T_i*/
    end
    level 1: while length(T) > 0 do
        T[x] = maximum(T), ∀i;/* Maximum PU
         availability*/ if E[x] ≤ Th_SU then
            PU_Selected = x;
            break;
        end
        else
            T = T - T[x];
        end
    end
    if PU_Selected == NULL then
        Th_SU = Th_SU + Th_incr;
        for i ≤ N do
            T[i] = StoreT[i] ;
        end
        Goto level 1;
    end
    else
        return PU_Selected;
    end
end
```

Algorithm 1: Algorithm for Spectrum Sensing and Channel Selection for Secondary Users

3) Spectrum Sensing and Channel Selection: There are many spectrum sensing techniques like energy detection, matched filter, cyclostationary [14] etc. However, energy detection is most popular because of its simplicity and robustness. As channel allocation is a NP Hard problem [12], we present a heuristic threshold based energy detection mechanism for channel allocation to the SUs. For an SU, to use a channel for a longer period of time, two conditions are desired: (i) the energy level on that channel (or, any kind of interference) should be low and (ii) the PU should exhibit low activity i.e., longer OFF periods.

We present the channel selection scheme by the SU in Algorithm 1, where the SU selects a channel based on two factors: (i) channel is available, and (ii) the detected energy is lower than a threshold value. The first *for* loop stores all the available PU channels. The PU channel with the maximum availability is selected based on a threshold. Later, the threshold is incremented if no PU channel is available and

the process is repeated.

V. Indoor Testbed Implementations

We have implemented the proposed adaptation techniques on an indoor laboratory testbed. The overall block diagram is shown in Figure 2. Though the setup allows for multiple primary users and multiple secondary users, we implemented one secondary transmitter-receiver pair, using two USRPs. This could easily be extended for multiple SUs by considering the channel occupied if *any* of the PUs or other SUs transmit over that channel. Any smart channel allocation scheme can also be applied for multiple SUs requesting the same channel at same time, however, these issues are beyond the scope of this paper. Spectrum sensing was done using a third USRP. The SU scans a range of channels and maintains a list of channels that can be potentially used. First let us describe the different blocks used for the testbed.

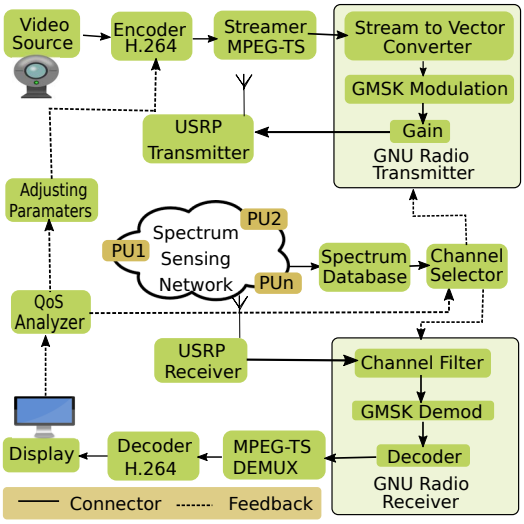


Fig. 2. Blocks for Video Transmission using USRPs and GNU Radio

Video Source: The video source was a Logitech web-cam with a resolution of 720p. Real-time video was captured at 25 fps rate and streamed to the encoder for encoding.

Encoder: We used the MPEG-4 Part 10, also known as H.264, to implement the Advanced Video Coding (AVC) standard. H.264 is the most widely used video codec due to its efficient implementation. We used *x264* open source software libraries for the implementation. We encoded the videos in 720p with 25 frames per second. Typically, once the encoding parameters are set before the beginning of a video transmission, the rates remain fixed for the entire duration of transmission. In our implementation, these parameters are continuously updated depending upon the feedback from the QoS Analyzer.

Streamer: The streamer uses UDP to stream the encoded video over the network. The open source GStreamer [10] framework has been used for the experiments.

GNU Radio: We used GNU Radio [26] for implementing the transmitter and receiver. The GNU Radio transmitter captures the video frames from the streamer and processes them for transmission over the RF channel, after signal processing. GMSK modulation was used for improved spectral efficiency and lower power consumption at the receiver [6].

USRPs: Universal Software Radio Peripherals (USRP) B210 with operational frequency range of 70 MHz to 6 GHz, was used for our implementations. All USRPs have free UHD driver support, open design schematic, and seamless integration features with the GNU Radio platform for signal processing. VERT900 antennas (from Ettus Research) were used to transmit and receive the processed signal. Both the transmitting and receiving USRPs receive feedback from the channel selector and set the RF parameters during runtime.

Mobile Transmitter and Lab Environment: The entire transmitter setup (web-cam, transmit antenna etc.) was mounted on a remotely controlled mobile robotic platform as shown in Fig. 3(a). Fig. 3(b) shows the lab environment. The indoor lab environment exhibited considerable multipath effects due to concrete walls. A Marvelmind indoor navigation system [27] working on 915 MHz ISM band was also present in the room with 6 stationary and 4 mobile beacon elements, making the RF environment congested. Along with all these, we had multiple PU transmitter-receiver pairs during the experiments, operating on the same ISM band. The receiver and the spectrum sensors were stationary as shown in Fig. 3(c).

Decoder and Display: At the receiver side, the H.264 decoder was used to decode each frame considering 16×16 unit size and 9 prediction modes. The H.264 decoder had integrated display of mplayer and GStreamer bin player.

Spectrum Sensing Network: One USRP was used for sensing 902 MHz to 926 MHz spectrum range with 1 second time interval for 1 MHz step size. For our experiments, we used a USRP B210 with 70 dB receiver channel gain, placed within 12 meters of both SUs. The sensed RSSI was fed to a primitive database management station. We modified the `usrp_spectrum_sense.py`, a file provided by the `uhd` driver installation.

Spectrum Database: We used a `mysql` database for storing the spectrum sensing data. Though such a database was not essential, nevertheless we implemented it for the sake of completeness for a larger distributed sensing framework. It took input from the QoS analyzer as a feedback, and operated on the data obtained from the sensing network.

Channel Selector: The channel selection tool was written in Python. This tool fetched data from the database and sent to the GNU Radio modules through TCP socket connections. We implemented the threshold based channel selection technique as per Algorithm 1 for use with this tool. The sensing network, mysql database, and channel selector worked in a live environment incurring about 1 second of overall delay.

QoS Analyzer: We used MSU video quality measurement tool [28] for analyzing video QoS. When the QoS is below a threshold at any time, a feedback is sent to: (i) the encoder to set bitrate to a lower rate as per equation 11, and (ii) the channel selector to inform the receiver and transmitter to hop to the next best available channel as per Algorithm 1. Spectrum

(a) Live Video Capture and Transmit (b) RF Environment (c) Video Receiver

Fig. 3. Experimental Scenario in Indoor Lab Environment

sensing, channel selection, and QoS analysis constituted the feedback control mechanism for the overall implementation.

Configuration Parameters: The configuration parameters are presented in Table I. All the experiments were conducted on ISM bands (902 MHz to 926 MHz) with minimum sustaining bandwidth of 3 MHz. Higher bandwidth did not show significant improvement of video quality. Usually antenna far field reference distance (d_0) is 1 to 10 meters for indoor environment. Pathloss exponent (λ) ranges between 1.6 and 1.8, inside buildings with Line Of Sight (LOS) [11]. During our experiments we used $\sigma_s^2 = 6.83$ for 910 MHz [11].

Parameters	Values
Experimental Scenario	Indoor
Pathloss Model	Simplified Pathloss
Frequency	902-926 MHz (ISM Band)
Bandwidth	3 MHz
Modulation Scheme	GMSK
Error Correction Scheme	None
Transmitter Channel Gain	80 dB
Receiver Channel Gain	70 dB
Antenna Gain	3 dBi
Antenna Far Field Distance (d_0)	10 meter
Pathloss Exponent (γ)	1.7 [11]
Powerloss Variance (σ_s^2)	6.83 dB^2 [11]
Powerloss Deviation (σ_s)	[-2.6134 to 2.6134] dB
Min Encoding Bitrate(ε_{MIN})	512 Kbps
Max Encoding Bitrate(ε_{MAX})	2048 Kbps
Encoding Frame Rate	25 fps
Video Resolution	720p
Streaming Encapsulation	MPEG-TS
Video QoS	PSNR and SSIM
Each Experiment Run	5 minutes

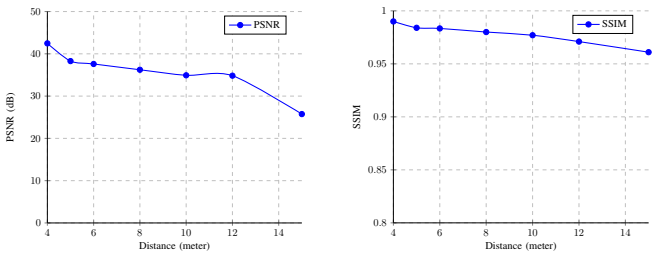
TABLE I
CONFIGURATION PARAMETERS FOR EXPERIMENTAL SETUP

VI. EXPERIMENTAL RESULTS

We present the experimental results for scenarios with: (i) no encoder adaptation and no channel adaptation, (ii) encoder adaptation but no channel adaptation, (iii) no encoder adaptation but channel adaptation, and (iv) both encoder and channel adaptations. These four scenarios are presented in sections VI-A to VI-D respectively. Due to the lack of existing study for the same attributes considered here, we could not present a comparative study of proposed method with any other implementation.

A. Video QoS without any Adaptation

In this set of experiments, we consider the channel to be quasi-stable, i.e., the channel condition does not change considerably. Here, we did not utilize the entire bandwidth by keeping encoding bitrate fixed at 512 Kbps. The frame rate was set to 25 fps. The SU transmitter was mounted on a remotely operated mobile robot for all experiments. We considered one secondary transmitter-receiver pair, where the receiver was stationary. We controlled the distance of the mobile transmitter such that it was at varying distances from the receiver. We observe from Fig. 4 that for the transmitter-receiver separation of 4 to 15 meters, the PSNR varied from 42 dB to 25 db. According to [23], PSNR between 30 dB to 50 dB, is considered to be good quality video. We achieved PSNR of 34 dB at 12 meters. We also achieved SSIM of 0.99 to 0.961 for distances of 4 to 15 meters.



(a) (b)

Fig. 4. Video Quality with Encoding Rate= 512 Kbps (a) PSNR; (b) SSIM

B. Video QoS With Encoder Adaptation Only

Here we consider a continuously changing channel. We simulated different types of channel diversity by number of changes in channel condition per minute. A *"change"* in the channel condition occurs when the powerloss $\psi(x)$ goes from one partition to another. We designed a software tool in C using GTK+ libraries for this purpose. The proposed channel adaptation technique was implemented to adapt to the varying channel conditions and thereby provide best possible video quality. The source coding bitrate was adapted based on the distance and channel conditions. We considered minimum (ε_{MIN}) and maximum (ε_{MAX}) supported bitrates as 512 Kbps and 2048 Kbps respectively. From the previous set of experiments, we inferred that we can get good video quality till 12 meters of separation between transmitter and receiver. So we conducted this set of experiment for three distances: 4 meters, 8 meters, and 12 meters. We ran the experiments for different number of channel adaptation rates (per minute). From Fig. 5, we see PSNR ranges between 42 dB to 30 dB for 0 to 60 changes per minute. The SSIM behaves like PSNR for all distances. From Fig. 5 and Fig. 6, we infer that:

- Video quality degrades as channel goes from slow to fast fading.
- Good video quality is achieved until 40 channel condition changes per minute for a distance of 8 meters.
- Good video quality is achieved until 30 channel condition changes per minute for a distance of 12 meters.

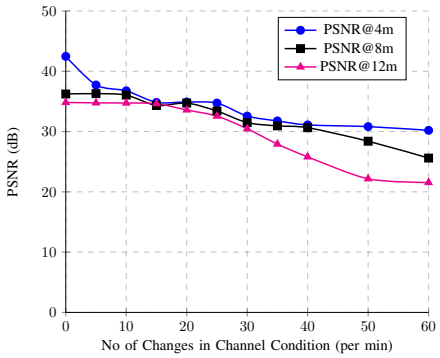


Fig. 5. PSNR for Continuously Changing Channel with Channel Adaptation

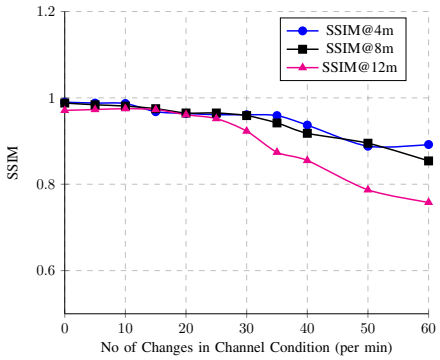


Fig. 6. SSIM for Continuously Changing Channel with Channel adaptation

C. Video QoS With Channel Adaptation Only

In this set of experiments, we implemented DSA considering multiple PUs, and one SU transmitter-receiver pair. We consider 8 PU transmitters, all of which use a bandwidth of 3 MHz. The first PU uses 902 - 905 MHz, the second PU uses 905 - 908 MHz, and so on. We started the SU transmission at 910 MHz, and after sometime the primary user of that frequency was activated based on the PU activity model. The SU transmitter pair continued to stream video by hopping to the next available frequency, which it obtained by consulting the spectrum database. The channel selector used the proposed threshold based energy detection algorithm to select most suitable channel for the SU to operate. We ran different sets of experiments considering that the SU needs to do frequency hopping (FH) for 1, 3, 5, 7 times per minute. We also implemented a non-DSA scenario, where the SU pair stops to operate as soon as the PU gets active on that particular frequency. Whenever the PU stopped its transmission, the sensor detects the PU absence and marks that channel as usable. Fig. 7 shows that PSNR ranges from 32 - 29 dB for channel hopping 1 to 7 times per minute at a distance of 4 meters. The non-DSA scenario provides worse video quality, which is below the acceptable range of recognizable video quality. In summary, from Figs. 7 and 8, we observe that:

- Channel adaptations provide better video quality.
- Video quality degrades as the frequency of channel hopping increases.
- Good video quality is achieved with upto 3-5 hops per minute in an indoor environment.

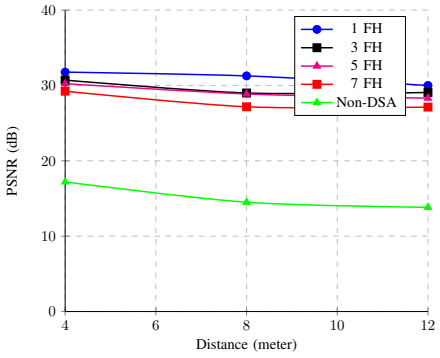


Fig. 7. PSNR for Fixed Channel Implementing Dynamic Spectrum Access

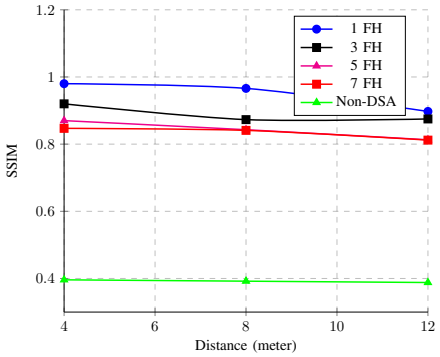


Fig. 8. SSIM for Fixed Channel Implementing Dynamic Spectrum Access

D. Video QoS With Both Encoder and Channel Adaptations

Here, we used the same set of experiments as we did for encoder adaptations (section VI-B) but with adapting the channel 5 times per minute. As expected, we observed improvements over previous results. In Figs. 9 and 10, the solid lines represent non-adaptive encoder ('nAdap') and the dashed lines represent adaptive channels ('Adap'). The behavior of PSNR and SSIM changes are like the previous set of experiments. We observe that the adaptive encoder provides a PSNR of 36.6 dB, whereas it was 32 dB for the non adaptive one. We achieved enhancement in video quality metrics in each case after implementing the proposed encoder adaptation technique. Finally, from Fig. 9 and 10, we infer that:

- Joint adaptations of the encoder and channel provide better video quality than the non-adaptive ones. The video stalls at the SU receiver during PU interruption in non-DSA setting, resulting a degraded overall video quality.
- The trend of how the video quality changes is the same as the previous set of experiments.

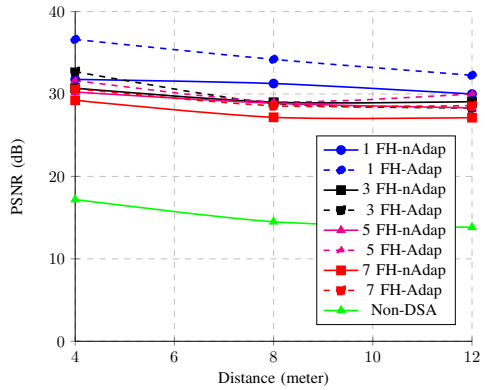


Fig. 9. PSNR for Changing Channel with Dynamic Channel Access

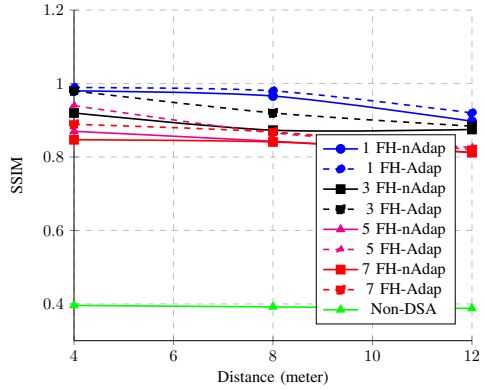


Fig. 10. SSIM for Changing Channel with Dynamic Channel Access

VII. Conclusions

This paper presents a testbed implementation of dynamic spectrum access for real-time video transmissions using SDRs for multiple PUs and a single SU receiver-transmitter pair. We propose a multi-level threshold based energy detection technique for spectrum sensing and selection based on SU and PU activity pattern. We have also presented a feedback-controlled channel adaptation technique targeted to enhance video quality for a continuously changing channel. We have concentrated on bitrate adaptation as source coding parameter, but the proposed method can be applied to any parameter using the same mechanism. We have shown that: (i) better video quality is achieved by implementing DSA as opposed to non-DSA based methods; (ii) the proposed channel adaptation technique provides good quality video for a certain number of channel condition changes. Finally, we have implemented the feedback-controlled channel adaptive mechanism for continuously changing channel using dynamic spectrum access. We observe that channel adaptation enhances the video quality for DSA implementation. Inclusion of live deployment of spectrum database and more automated physical deployment can be considered as a future direction of this work.

References

[1] C. VNI, "The Zettabyte Era-Trends and Analysis," http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html, 2016.

[2] J. Chakareski, V. Velisavljevic, and V. Stankovic, "View-popularity-driven joint source and channel coding of view and rate scalable multi-view video," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 3, pp. 474–486, 2015.

[3] A. Vosoughi, P. C. Cosman, and L. B. Milstein, "Joint source-channel coding and unequal error protection for video plus depth," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 31–34, 2015.

[4] C. N. Ververidis, J. Riihijarvi, and P. Mahonen, "Evaluation of quality of experience for video streaming over dynamic spectrum access systems," in *IEEE WoWMoM*, 2010, pp. 1–8.

[5] WinnForum, "Wireless Innovation Forum," http://www.wirelessinnovation.org/, 2017.

[6] M. Tahir, H. Mohamad, N. Ramli, and S. P. Jarot, "Experimental implementation of dynamic spectrum access for video transmission using USRP," in *IEEE ICCCE*, 2012, pp. 228–233.

[7] Z. He, J. Cai, and C. W. Chen, "Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 511–523, 2002.

[8] O. A. Al-Tameemi and M. Chatterjee, "A System Level Solution for DSA Systems: From Low-Cost Sensing to Spectrum Database," in *IEEE LCN*, 2016, pp. 104–111.

[9] V. Kostina, Y. Polyanskiy, and S. Verd, "Joint source-channel coding with feedback," *IEEE Trans. on Information Theory*, vol. 63, no. 6, pp. 3502–3515, 2017.

[10] Online, "gstreamer," https://gstreamer.freedesktop.org/, 2018.

[11] J. C. Liberti and T. S. Rappaport, "Statistics of shadowing in indoor radio channels at 900 and 1900 MHz," in *IEEE Military Communications Conference*, 1992, pp. 1066–1070.

[12] S. Almowuena and M. Hefeeda, "Mobile Video Streaming over Dynamic Single-Frequency Networks," *ACM Trans. Multimedia Comput. Commun. Appl.*, pp. 81:1–81:26, 2016.

[13] Y. Saleem and M. H. Rehmani, "Primary radio user activity models for cognitive radio networks: A survey," *Journal of Network and Computer Applications*, vol. 43, pp. 1–16, 2014.

[14] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor, "Spectrum sensing for cognitive radio: State-of-the-art and recent advances," *IEEE Signal Processing Magazine*, vol. 29, no. 3, pp. 101–116, 2012.

[15] B. Canberk, I. F. Akyildiz, and S. Oktug, "Primary user activity modeling using first-difference filter clustering and correlation in cognitive radio networks," *IEEE/ACM Trans. on Networking (TON)*, vol. 19, no. 1, pp. 170–183, 2011.

[16] J. Riihijärvi, J. Nasreddine, and P. Mähönen, "Impact of primary user activity patterns on spatial spectrum reuse opportunities," in *IEEE Wireless Conference*, 2010, pp. 962–968.

[17] S. Bayhan and F. Alagöz, "Distributed channel selection in crahns: A non-selfish scheme for mitigating spectrum fragmentation," *Elsevier Ad Hoc Networks*, vol. 10, no. 5, pp. 774–788, 2012.

[18] M. H. Rehmani, A. C. Viana, H. Khalife, and S. Fdida, "SURF: A distributed channel selection strategy for data dissemination in multi-hop cognitive radio networks," *Computer Communications*, vol. 36, no. 10, pp. 1172–1185, 2013.

[19] S. Jakubczak and D. Katabi, "A cross-layer design for scalable mobile video," in *ACM Mobile computing and networking*, 2011, pp. 289–300.

[20] D. Roy, M. Chatterjee, and E. Pasiliao, "Video quality assessment for inter-vehicular streaming with IEEE 802.11p, LTE, and LTE Direct networks over fading channels," *Computer Communications*, vol. 118, pp. 69–80, 2018.

[21] M. Abramowitz, *Handbook of Mathematical Functions, With Formulas, Graphs, and Mathematical Tables*, 1974.

[22] Z. W. Ligang, Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment using structural distortion measurement," in *IEEE Int. Conf. Image Proc*, 2002.

[23] N. Thomos, N. V. Boulgouris, and M. G. Strintzis, "Optimized transmission of JPEG2000 streams over wireless channels," *IEEE Transactions on image processing*, vol. 15, no. 1, pp. 54–67, 2006.

[24] A. C. Brooks, X. Zhao, and T. N. Pappas, "Structural similarity quality metrics in a coding context: Exploring the space of realistic distortions," *IEEE Trans. on image processing*, vol. 17, no. 8, pp. 1261–1273, 2008.

[25] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*, 1990.

[26] E. Blossom, "GNU radio: tools for exploring the radio frequency spectrum," *Linux journal*, vol. 2004, no. 122, p. 4, 2004.

[27] M. Robotics, "Marvelmind Indoor Navigation System Operating manual," https://marvelmind.com/pics/marvelmind_navigation_system_manual.pdf, 2017.

[28] Online, "MSU Video Quality Measurement Tool," http://compression.ru/video/quality_measure/video_measurement_tool.html, 2012.