Integrating an Effective VoIP Service in a USRP/GNU Radio Testbed

Naceur Malouch

Université Pierre et Marie Curie - UPMC Sorbonne Universités
Laboratoire d'Informatique de Paris 6 LIP6/CNRS
Paris, FRANCE
Email: Naceur.Malouch@lip6.fr

***Abstract*—This work presents an implementation of spectrum handoff and selection in order to experiment a Voice over IP (VoIP) application over a cognitive radio testbed. VoIP connections use both WiFi bands and GSM bands to select available channels dynamically. We use the USRP/Gnu radio testbed to implement spectrum management functions and to run the experiments. We also experiment VoIP and data transfers together by implementing a satisfaction-based inter-service cohabitation strategy. Through subjective observations and objective measurements, we found that the quality of established phone calls can be maintained. Besides, VoIP can benefit from the flexibility of data transfers to perform handoffs more adequately. In the same time, data transfer is able to sustain its required average rate with minimum degradations.**

Keywords–Cognitive Radio Networks; VoIP; USRP; GNU Radio; Testbed.

I. INTRODUCTION

Providing classic services such as Voice over IP (VoIP) over Cognitive Radio Networks (CRNs) has become an interesting research topic during past few years. The first challenge is to guarantee uninterrupted services despite the dynamic nature of the spectrum. Unlike traditional Quality-of-service (QoS) mechanisms for wired networks which mainly depend on the traffic statistics, the quality of required services on CRN must be according to the spectrum properties as well. The second challenge is to evaluate the suitability of a new available spectrum for usage and selecting the best channel among multiple available channels. This challenge concerns channel selection and handoff algorithms.

In this paper, we focus on supporting a VoIP application [1] over a GNU Radio testbed [2] that uses Universal Software Radio Peripheral (USRP) devices [3]. Indeed, several works have studied theoretically and by simulations the feasibility of supporting such service [4]–[9]. Other works have evaluated the general performance using experimental testbeds without considering VoIP or real-time applications [10]–[12]. [12] studies latency between USRP devices and determines reasons of large latencies at the PHY layer. To the best of our knowledge, the most relevant work that has focused on experimenting VoIP over a cognitive radio testbed is [13]. However, the main objective of this work is to study the impact of spectrum sensing on the quality of VoIP. In our work, we rather study the impact of spectrum handoff and spectrum selection which is complementary to the previous work. The previous work has limited the tests to be only over 5 GHz frequency band. We use in this work both GSM and Wi-Fi bands using several channels. In particular, we show that VoIP works also when using low frequency bands. In addition, we use a real-world VoIP application [1] rather than a traffic generator as in [13]. The testbed also is different since [13] have performed their tests using the WARP platform [10]. Finally, in this work, we also investigate the cohabitation between VoIP and data transfer in order to share the spectrum while satisfying simultaneously both service requirements.

The lack of VoIP experiments over GNU Radio testbeds is mainly due to the difficulty of running a complete TCP/IP stack over existing testbeds. Besides, once simple frame transmissions are successfully set, usually no further investigations are done for upper layers.

During the experiments over the USRP/Gnu radio testbed, we evaluated the audio quality and measured the delay and the jitter for VoIP. The results of the tests show that it is possible to compensate the delay increased by the interruption of the transmission due to the presence of a Primary User (PU) on the channel, by performing spectrum handoff when it becomes a must. Here, finding adequate metrics are necessary to find when the handoff must be performed and which target channel to select for the handoff. These metrics are related on one hand to the service requirement, and on the other hand to the channel properties such as the availability ratio and the remaining availability period. We also experimented the efficiency of using prediction techniques to assess available periods.

The rest of the paper is organized as follows. In Section II, we introduce the testbed used in the experiments and we present related settings and constraints. In Section III, we show results of the experiments over the testbed for VoIP with and without the presence of data traffic. In Section IV, we conclude the paper and point out future directions.

II. TESTBED SETTINGS AND CONSTRAINTS

GNU Radio is an open source tool-kit that provides all functionalities to handle the radio interface and process radio signals at the software level. Gnuradio transceivers [2] are composed of many elements similar to hardware domain, like filters, demodulators, decoders, etc., called blocks.

The USRP device [3] is a radio hardware that GnuRadio can tie with. We have used USRP1 which is one of the numerous versions of USRP. The USRP1 platform can support two daughterboards The GnuRadio-USRP testbed that our experiments will be run on is shown in Figure 7.

ISBN: 978-1-61208-411-4

TABLE I. LIST OF FREQUENCIES AND THEIR AVAILABILITY UNAVAILABILITY PERIODS

Frequency (MHz)	Avail. range (sec)	Unavail. range (sec)	Avail. percentage
905	2-8	42-48	10%
915	9-15	25-31	30%
925	3-9	10-18	30%
2485	27-33	23-37	50%
2490	3-17	5-15	50%
2495	11-17	4-8	70%
2900	20-36	6-26	70%

The platform is composed of two USRP1 with two daughterboards each, RFX900 for GSM band, and RFX2400 for WiFi band and two Linux host machines, with GnuRadio installed on each one, connected via USB2.0 interface to the USRPs.

We have tested several supported modulation methods such as PSK, QPSK, GFSK, QAM, BPSK, and GMSK for both GSM and WiFi. Through the tests, gmsk gave the best results in terms of packets delivery and packet error rate.

GnuRadio and USRP do not allow straightforwardly to run simultaneous transmissions. However, this feature is necessary so that we can run more than one service simultaneously either on the same channels or on different ones. But USRP is designed to support at most two channels in one direction, one on each daughterboard.

In our work, the sensing module is the source of the information that we have to consider to decide how to manage spectrum mobility. The experiments are based on generating the sensing information using different random patterns corresponding to the availability and the unavailability of the channels. The benefit of this method is that we can compare results by using the same sets of availability patterns. Using realistic primary transmissions is not controllable and thus it is hard to compare selection and handoff algorithms fairly.

To generate the available/unavailable periods, we use the uniform distribution to control their durations and thus primary activity. For available/unavailable periods generation, we have set intervals, means and ranges of time to satisfy given percentages of availability for each channel. (Table I).

We implemented a realistic sensing-transmission cycle where the transmission operation will alternate with the sensing process periodically. We also implemented a GNU radio module in which we can plug any spectrum and handoff algorithm.

III. VOIP SPECTRUM SELECTION AND HANDOFF EXPERIMENTS

A. *Experiments with VoIP alone*

In order to generate different VoIP patterns, we establish real VoIP communications using mumble software while varying the duration of talkspurts and silence periods. Four types of experimentation are made during 10 minutes. The first type is with continuous communication without silence periods (Nosilence). The second one imitates a dynamic conversation. It is done with 10s for talkspurts and 10s for silences (10-10). The third represents also symmetric traffic but it is less dynamic with 60s for talkspurts and 60s for silences (60-60). The last one is with 20s for talkspurts and 120s for silences (20-120) to imitate a conversation where a speaker is listening more than speaking.

VoIP connection should select the channel with the greatest *predicted* remaining available period among the channels that have a residual bandwidth (capacity) larger than the codec rate. Besides, in order to evaluate the impact of using predicted values on the performance, we also perform experiments using the exact future remaining availability durations. This can be done since the channel patterns are generated in advance. Also, this can be useful for real systems where availability periods are known in advance through for instance regional databases. In case of a handoff, we select also the channel with the greatest predicted remaining availability period. We have used *Autoregressive Model Based Prediction (ARM)* [14] in our implementation.

First, Figures 1 and 2 show the variation of frequencies selected with spectrum management algorithms for VoIP. Since we assign more availability to WiFi channels (Table I), we notice that the algorithms tend to choose these channels more often. This is because the remaining available period of these channels is usually larger than the others. However, depending on the instantaneous availability of channels even low availability ratio channels can be used when necessary.

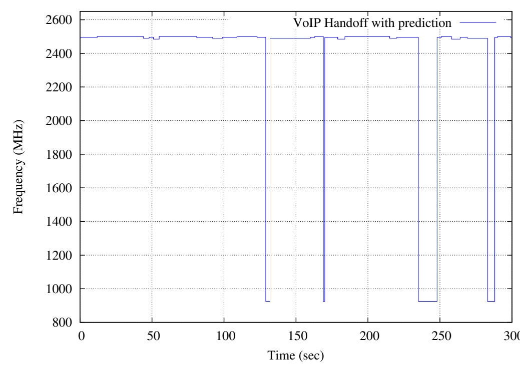

Figure 1. Channel changes during VoIP experiments

Then, we explore the number of handoffs. The results in Figure 3 are in accordance with the idea that the more

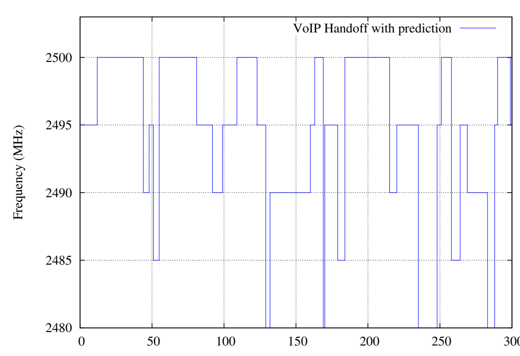

Figure 2. Zoom on channel changes over the WiFi band

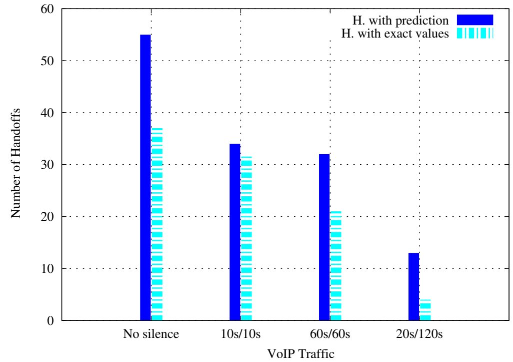

Figure 3. Number of handoffs function of VoIP traffic

important the traffic, the larger the number of performed handoffs. We notice that the algorithm with prediction gives different results than the one with exact values. This can be explained by the fact that the prediction mechanism gives sometimes inaccurate results that may lead to errors. We observe also that the prediction gives a result close to the exact values when the traffic is very dynamic because it is possible in this case to finish a talkspurt using the same channel before the arrival of a primary user. Knowing the exact value of the remaining available period is not really useful.

Notice that handoffs can occur during talkspurts. Especially, when there is no silence periods, the handoff does not impact the quality of the received audio except if the handoff is delayed. This happens when the signaling packet is lost, no channel is available immediately, or the transmission queue contains some packets. We conclude that it is better to send signaling packets using a different queue that has the scheduling priority. Handoffs however impacts a little the quality of the conversation interactivity. Globally, all tests being done, the quality can be evaluated by a subjective mean opinion score (MOS) of 3.8 since the phone calls pursue normally except few intermittent discomforts in terms of interactivity.

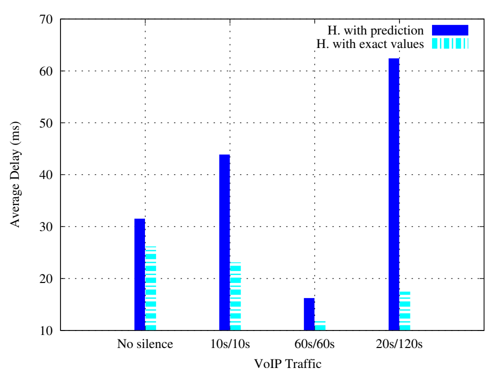

Figure 4. Average of VoIP packets delay function of VoIP traffic

From a delay point of view, we notice in Figure 4 that the algorithm with prediction has the same behavior as the one with exact values when the traffic is important (Nosilence and 60s-60s). The large delay variation noticed for some tests is caused by the fact that sometimes there are no available channels, in this case we stop the transmission (packets are buffered) until we get an available channel, and this impacts directly the delay.

The quality of the received audio confirm that if the spectrum handoff is performed immediately without any extra delay, then it does not affect packet transmissions. Indeed, we did not notice interruption of the communication. This is because the distance between the two USRP boxes is not long, hence audio packets can be delayed a little bit without a large impact. These observations confirm the idea that performing spectrum handoffs in advance when the conditions are optimal is an interesting approach to maintain a good quality of the VoIP communication. Of course, this can increase the number of handoffs and thus the energy consumption and the channel contention.

For the test 20s-120s we notice that the average of the delay exceeds 60 milliseconds. As we explained earlier, the channel unavailability causes this large delay. We draw the CDF of the packets delays shown in Figure 5. The figure zoomed on the interval [0-30]ms shows that despite that the average delay is large, most of the packets have delays less than 10ms.

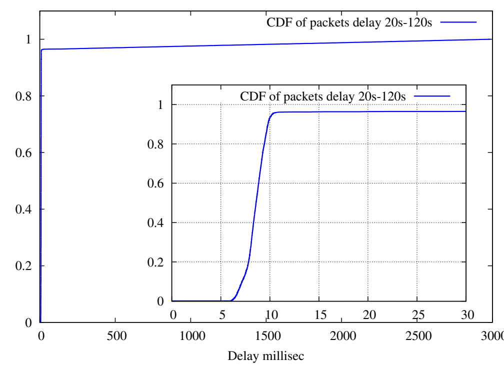

Figure 5. CDF of packets delay for 20s-120s talkspurt-silence periods and handoff with prediction

In Figure 6, when we use the exact values instead of the predicted ones for the remaining availability periods, we notice that the jitter is smaller especially when the traffic is more important because first packets of talkspurts require usually more transmission delays. Again, the algorithm with prediction has a closer result to exact values when the traffic is very dynamic and talkspurt durations are equal to silence durations (10s-10s).

B. VoIP and data transfer cohabitation experiments

In this part, each service (VoIP or Data) has to optimize its connection and guarantee at the same time the non disturbance of the other service. The Gnu Radio testbed imposes a strong constraint for the experiments: Two connections can not be established in parallel over two different channels of the same wireless card. Thus, we need to adapt the design to this constraint. The objective is to experiment the inter-service approach for cohabitation.

The inter-service approach is based on the fact that services have to consider each other. In other words, when performing spectrum selection and handoff, one service takes its decision

 ISBN: 978-1-61208-411-4

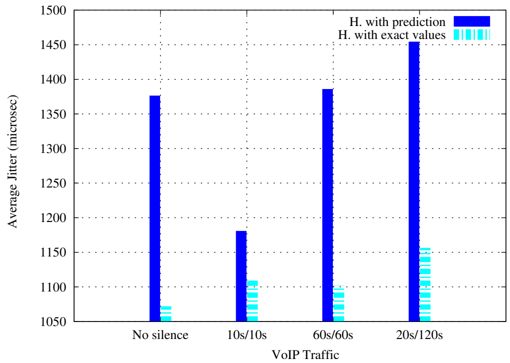

Figure 6. Average of the jitter function of VoIP traffic

not only based on its requirements but also based on demands of the other services, so that we satisfy both requirements as fairly as possible. The question here is how to assign channels to services, and how to know that this assignment is the best way to satisfy each service requirement. The basic idea is to find metrics that may describe better the state of channels for each service.

For VoIP, the goal is to find a channel that is available and allows the service to stay as long as possible on the same channel without handoff or perform the handoff in good conditions. To this aim, we can consider the remaining available period which describes only the channel instantly and does not give the global state of the channel. It is also necessary to measure the degree of satisfaction of the user. One possible metric is the mean number of handoffs that can be calculated for each channel in a given window time. Besides, this number provides somewhat a long term estimation of the channel status. However, it is difficult to relate this number to the quality perceived by the users, and also choosing the right measurement window size is not obvious.

Another possibility is to use in addition to the remaining availability period a threshold that represents a large acceptable value for the availability period. This threshold can correspond to the average VoIP duration or average duration of talkspurts. A channel that has this value of threshold for its remaining availability period is then considered as a best channel. In this case the satisfaction degree is equal to 1. In order to assess the satisfaction degree for a group of channels, we can use the maximum remaining availability period among all channels. More precisely, the satisfaction degree can be computed as follows

$$x_{VoIP_{DB_i}} = \frac{Max\ remaining\ available\ period\ on\ DB_i}{Threshold\ of\ remaining\ available\ period}, \tag{1}$$

where DB_i refers to the channels accessible through the GSM interface or the WiFi interface, $i \in \{GSM, WiFi\}$. This metric indicates whether it is worthy to stay on the current interface or it is better to move to the other interface. For instance, if at some time during the VoIP communication, the maximum remaining available period among all channels is low compared to the threshold, then in case of handoff, it is better to move to other interface to avoid more possible interruptions of the communication in the future.

For the data service, we have different parameters that we can take into account such as the available bandwidth during the availability period and the average achievable rate over the channel. The last parameter describes well the status of the channel but has again the drawback of choosing the right window size for past measurement. Alone, it is not sufficient to measure the satisfaction of the user. It should be compared to the rate demand. We use the following metric

$$x_{Data_{DB_i}} = \frac{Max\ (\frac{Available\ achievable\ rate}{Rate\ demand})\ on\ DB_i}{Tolerance} \tag{2}$$

Here, we also use the maximum value among all channels of a given wireless card. The $Tolerance$ parameter should be equal or close to 1. The smaller this parameter, the larger the tolerance to a rate decrease.

Since we have technical constraints, we have to select one channel on each wireless card and each channel is reserved for a service VoIP or data (Figure 7). Thus, few service configurations are possible and the goal is now to find what configuration to choose and when. In this case, we can calculate the fairness index for each configuration as follows:

$$\mathcal{FI}(x_1, x_2) = \frac{(\sum_{i=1}^{2} x_i)^2}{2 \cdot \sum_{i=1}^{2} {x_i}^2} \tag{3}$$

where $(x_1, x_2) = (x_{VoIP_{DB_1}}, x_{Data_{DB_2}})$ or $(x_1, x_2) = (x_{VoIP_{DB_2}}, x_{Data_{DB_1}})$. Then, we choose the configuration that has the best fairness index. In other words, we choose the configuration that tries to satisfy both services. Algorithms 1 and 2 provides the implemented spectrum handoff and selection procedures for each service.

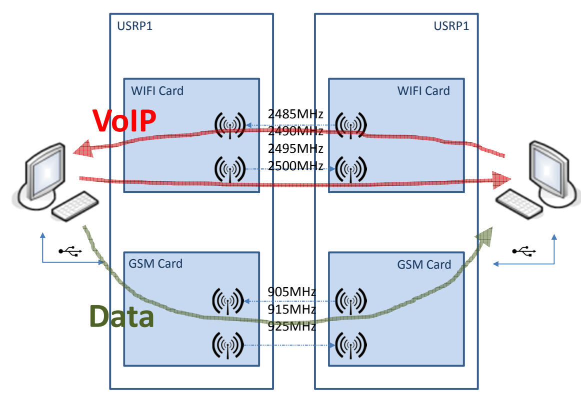

Figure 7. Example of daughterboards (wireless cards) configuration for VoIP and Data services cohabitation

Data Handoff and Selection with Cohabitation: When the data is transmitted on channel ch_c and a PU is detected, the algorithm tests if the available rate satisfies the required rate (line 3) to check if it is useful to wait. On the contrary case, we have to perform handoff immediately otherwise we wait for $t_1 = T_{OFF} + \sigma_{OFF}$ which is the sum of the average unavailability time of the current channel ch_c and the standard deviation. Then, we retest if the PU is still on the channel. This

Algorithm 1 Data Handoff and Selection with Cohabitation

Require: DB_c: The current daughterboard, Ch_c: The current channel, Ch_i: Channel i, AC: List of available channels on DB_c, OC: List of available channels on the other DB, T_{OFF}: The unavailability time mean of Ch_c, σ_{OFF}: The standard deviation of unavailability periods of Ch_c, Rav_i: Remaining availability time in channel Ch_i, Ar_i: Available rate of channel Ch_i, Rr: Required rate, W_{th}: Threshold weight, W_i: Weight on channel ch_i, W_{max}: Weight max on DB_c.
Ensure: Ch_{Data}: Selected channel for data, Ch_{VoIP}: Selected channel for VoIP.

if $P\ on\ Ch_c$ **then**
    $stop\ transmission$
    if $Ar_c >= Rr$ **then**
        $t_1 = T_{OFF}\ +\ \sigma_{OFF}$
        $wait\ t_1$
    end if
    if $(After\ t_1\ P\ still\ on\ Ch_c)\ Or\ (Ar_c < Rr)$ **then**
        for all $Ch_i\ in\ AC$ **do**
            $W_i = Ar_i/Rr$
        end for
        $W_{max} \leftarrow max_i\{W_i\}$
        if $W_{max} >= W_{th}$ **then**
            $Ch_{Data} \leftarrow argmax_{Ch_i \epsilon AC}\{Rav_i\}$
            return Ch_{Data}
        else
            $calculate\ fairness\ indexes\ :\ \mathcal{FI}_{Data\ on\ DB_c}\ and\ \mathcal{FI}_{Data\ not\ on\ DB_c}$
            if $\mathcal{FI}_{Data\ on\ DB_c} >= \mathcal{FI}_{Data\ not\ on\ DB_c}$ **then**
                $Ch_{Data} \leftarrow argmax_{Ch_i \epsilon AC}\{Rav_i\}$
                $Ch_{VoIP} \leftarrow argmax_{Ch_i \epsilon OC}\{Rav_i\}$
            else
                $Ch_{Data} \leftarrow argmax_{Ch_i \epsilon OC}\{Rav_i\}$
                $Ch_{VoIP} \leftarrow argmax_{Ch_i \epsilon AC}\{Rav_i\}$
            end if
            return Ch_{Data}, Ch_{VoIP}
        end if
    else
        $Go\ back\ to\ transmission$
    end if
end if

strategy aims at reducing the number of handoffs performed by data transfer so that it reduces also handoffs to channels required by VoIP. Indeed, if the current channel becomes available again, then there is no need for handoff and the rate of the data transfer can still be achieved because it depends on the *average* availability. Interrupting the transfer is tolerated and does not trigger necessarily handoffs. Now, if the PU is still on the channel after t_1, we have to perform handoff. In the two cases, the mechanism of selection of the new channel is the same. We look for channels on the same daughterboard, if there is one that verifies a satisfaction metric larger than the tolerance threshold, which means that we judge that the channel may satisfy the service requirement, we move to this channel. Otherwise, we calculate the fairness index so that if we have to switch daughterboards, we guarantee the best assignment of channels to satisfy requirements of both services.

VoIP Handoff and Selection with Cohabitation: For VoIP service, the presence of PU on the Ch_c triggers an immediate handoff. The new channel is first selected on the current daughterboard if the maximum remaining availability period is greater than the threshold (lines 5-6). If not, we calculate the fairness index (line 10) to find the best configuration. In one hand, we profit from the elasticity of the data service to handoff VoIP connections whenever required. On the other hand, we do not starve totally data transmission since we use the fairness index to provide always some resources to it. This means, we can accept some degradation in the VoIP quality in order to avoid stalling totally the data transmission.

Algorithm 2 VoIP Handoff and Selection with Cohabitation

Require: DB_c: The current daugtherboard, Ch_c: The current channel, Ch_i: Channel i, AC: List of available channels on DB_c, Rav_i: Remaining availability time for channel Ch_i, Rav_{max}: Max remaining available time on DB_c, Rav_{th}: Threshold of remaining available time.
Ensure: Ch_{Data}: Selected channel for data, Ch_{VoIP}: Selected channel for VoIP.

1: **if** $P\ on\ Ch_c$ **then**
2:     **for all** Ch_i in AC **do**
3:         $calculateRav_i$
4:     **end for**
5:     $Rav_{max} \leftarrow max_i\{Rav_i\}$
6:     **if** $Rav_{max} >= Rav_{th}$ **then**
7:         $Ch_{VoIP} \leftarrow argmax_{Ch_i \epsilon AC}\{Rav_i\}$
8:         **return** Ch_{VoIP}
9:     **else**
10:         $calculate\ fairness\ indexes\ :\ \mathcal{FI}_{Data\ on\ DB_c}\ and\ \mathcal{FI}_{VoIP\ not\ on\ DB_c}$
11:         **if** $\mathcal{FI}_{VoIP\ on\ DB_c} >= \mathcal{FI}_{VoIP\ not\ on\ DB_c}$ **then**
12:             $Ch_{VoIP} \leftarrow argmax_{Ch_i \epsilon AC}\{Rav_i\}$
13:             $Ch_{Data} \leftarrow argmax_{Ch_i \epsilon OC}\{Rav_i\}$
14:         **else**
15:             $Ch_{Data} \leftarrow argmax_{Ch_i \epsilon AC}\{Rav_i\}$
16:             $Ch_{VoIP} \leftarrow argmax_{Ch_i \epsilon OC}\{Rav_i\}$
17:         **end if**
18:         **return** Ch_{Data}, Ch_{VoIP}
19:     **end if**
20: **end if**

For the tests we choose to have medium traffic for VoIP (60s-60s talkspurt-silence), two rates for the rate demand of data traffic, 100kbps and 200kbps. We do not show results for higher data rates because the VoIP software client and server (mumble) need to exchange continuously signaling packets that permit the client to stay connected. In the tests, since the number of available channels is limited, we have to wait some time for frequencies to be available, so we stop transmissions. In this case, when the client does not receive server's packets, it considers that it is no longer connected which impacts our tests especially when we increase the rate demand. The results are summarized in Table II.

The first observation is that the effective rate measured for data transfer is slightly lower than the rate demand. This is the cost to pay to keep an acceptable quality for the VoIP conversation. Indeed, the audio was not distorted during the whole duration of the communication. Besides, a lower rate

 ISBN: 978-1-61208-411-4

TABLE II. PERFORMANCE RESULTS FOR VOIP AND DATA TRANSFER COHABITATION

Required data rate (kbps)	Number of VoIP handoffs	Number of data handoffs	Effective Data rate (kbps)
100	17	14	97.9
200	21	7	180

for data transfer implies usually a little longer delay before delivering the data (file), whereas for VoIP it is crucial to maintain interactivity and audio quality in all periods of the communication.

We observe also that the number of handoffs for data decreases when we increase the rate demand. This is because less channels can provide a better achievable rate for data and since VoIP has the priority when it needs handoffs, the data algorithm decides more often to stay on the same channel. This explains also the lower measured effective data rate compared to the case where the rate demand is lower. On the other hand, VoIP continues profiting from handoffs to choose the adequate wireless card and channel to pursue its communication. This confirms again that exploiting data flexibility is a practical approach to cohabit VoIP and data together.

IV. CONCLUSION

Cognitive Radio presents the perfect solution for many spectrum scarcity problems in several areas as long as classic services can be supported. Spectrum selection and handoff can achieve the quality required by these services when they experience degradations because of the appearance of primary users on the ongoing channel.

We have experimented real-world VoIP communications over the USRP/Gnu radio testbed in which we have implemented suitable spectrum selection and handoff algorithms. Our first observation is that even low availability ratio channels are useful to maintain VoIP calls through spectrum handoff without impacting substantially the quality. Moving from GSM band to Wi-Fi band and inversely can be done during the VoIP call. The spectrum handoff is performed to the channel with largest predicted remaining available period. We found that, the more the conversation is active (small talkspurts, small silence periods), the less the impact of prediction errors. However, when the talkspurts are large, it is better to perform the handoff in advance before the primary arrival to avoid abrupt interruptions.

We have also proposed a strategy and algorithms for cohabitation between two different services VoIP and data transfer. The results of experiments show that VoIP can benefit from the elasticity of data transfers in order to perform handoffs and choose adequate channels more easily. To do so, data transfer should avoid systematic handoff at primary arrival and do handoff to free the ongoing channel if required by a VoIP communication, while keeping an acceptable quality for itself in terms of achieved average rate. It is important to notice that a simple priority mechanism for VoIP as in traditional wired or wireless communications is not suitable to take spectrum decisions. Our mechanism is rather based on compromising satisfaction degrees of both VoIP and data transfer services.

More generally, the results of this work have demonstrated that acceptable quality for stringent services can be ensured in presence of primary users with dynamic activity. To go further, it is interesting to experiment the cohabitation with other types of services such as video streaming and remote desktop. In this case, the satisfaction degree and the spectrum handoff strategy of VoIP should not change. Also, the next step is the deployment of this testbed for everyday phone calls in a local network so that more statistics and more subjective opinion scores can be collected.

REFERENCES

[1] "Mumble: Audio and VoIP Application (LightSpeed Gaming LLC Company)," 2014, URL: http://www.mumble.com [accessed: 2015-05-12].

[2] "GNURadio: The Free and Open Software Radio EcoSystem," 2014, URL: http://gnuradio.org/redmine/projects/gnuradio/wiki [accessed: 2015-05-12].

[3] "ETTUS Research, USRP: Universal Software Radio Peripheral," 2014, URL: https://www.ettus.com/product [accessed: 2015-05-12].

[4] S. Lirio Castellanos-Lopez, F. Cruz-Perez, M. Rivero-Angeles, and G. Hernandez-Valdez, "Joint connection level and packet level analysis of cognitive radio networks with voip traffic," IEEE Journal on Selected Areas in Communications, vol. 32, no. 3, March 2014, pp. 601–614.

[5] S. Castellanos-Lopez, F. A. Cruz-Perez, M. E. Rivero-Angeles, and G. Hernandez-Valdez, "Performance comparison of voip cognitive radio networks under on/off and poisson primary arrivals," in IEEE 24th International Symposium on Personal Indoor and Mobile Radio Comm. (PIMRC), Sept 2013, pp. 3302–3307.

[6] Z. Wang, T. Jiang, L. Jiang, and X. He, "Voip capacity analysis in cognitive radio system with single/multiple channels," in 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Sept 2010, pp. 1–4.

[7] H. Lee and D.-H. Cho, "Voip capacity analysis in cognitive radio system," IEEE Communications Letters, vol. 13, no. 6, June 2009, pp. 393–395.

[8] T. Chakraborty, I. Misra, and S. Sanyal, "Selection of optimal transmission time in cognitive radio network for efficient voip performance," in 5th International Conference on Computers and Devices for Communication (CODEC), Dec 2012, pp. 1–4.

[9] H. Lee and D.-H. Cho, "Capacity improvement and analysis of voip service in a cognitive radio system," IEEE Transactions on Vehicular Technology, vol. 59, no. 4, May 2010, pp. 1646–1651.

[10] "Rice University WARP: Wireless Open-Access Research Platform," 2015, URL: http://warp.rice.edu/ [accessed: 2015-05-12].

[11] K. T. et al., "SORA: high-performance software radio using general-purpose multi-core processors," USENIX Symposium on Networked Systems Design and Implementation (NSDI), April 2009, pp. 75–90.

[12] N. Truong, Y.-J. Suh, and C. Yu, "Latency analysis in gnu radio/usrp-based software radio platforms," in IEEE Military Communications Conference, MILCOM 2013, Nov 2013, pp. 305–310.

[13] K. Tan, K. Kim, Y. Xin, S. Rangarajan, and P. Mohapatra, "Recog: A sensing-based cognitive radio system with real-time application support," Selected Areas in Communications, IEEE Journal on, vol. 31, no. 11, November 2013, pp. 2504–2516.

[14] Z. Wen, T. Luo, W. Xiang, S. Majhi, and Y. Ma, "Autoregressive spectrum hole prediction model for cognitive radio systems," in IEEE International Conference on Communications Workshops, May 2008, pp. 154–157.

ACKNOWLEDGMENT

Many thanks to Nada Abdelkader for all her works over the testbed and especially for solving parallel transmission issues. We also thank Jawad Seddar from Thales Communications and Martin Peres from LaBRI for their valuable remarks regarding the testbed settings. This work is partially supported by the French national research agency (ANR), LICORNE Project, grant ANR-10-VERS-005-03.