A Real Time Testbed for the Evaluation of Cognitive Radio MAC

Shen Gu, Pengchao Xu, Xinbing Wang, Xiaoying Gan, Hui Yu
Department of Electrical Engineering
Shanghai Jiao Tong University
Email: {gsgsgushen, xupengchao1986, xwang8, ganxiaoying, yuhui}@sjtu.edu.cn

Abstract—Cognitive Radio is a promising technology to solve the problem of increasing spectrum shortage in wireless communication. It allows secondary users(SUs) to opportunistically access idle spectrum of primary users(PUs) while ensuring collision perceived by PUs under a specific requirement. Testbed is essential to advance the development of cognitive radio networks. This paper presents a real time testbed for the evaluation of the cognitive radio MAC algorithm. The testbed supports the necessary components of cognitive radio including programmable RF transceiver, software-defined MAC layer and adaptive network layer to support multi-PU multi-SU cognitive radio networks. It is much easier to configure and control than the traditional FPGA-based testbed by using system-on-chip(SoC) programmable processors. We present the main features and the implementation details of the testbed in this paper. We also introduce an experiment using the testbed to evaluate the performance of an adaptive CR MAC algorithm proposed by us.

I. INTRODUCTION

The explosive growth in wireless services and the current static frequency allocation policy have resulted in the increasing shortage of useable spectrum, which will greatly hinder the development of emerging wireless services such as large-scale sensor network in the future. However, many experiments have showed that a large portion of allocated spectrum lies unused at any given time and location. For example, study from Federal Communication Commission(FCC) shows that temporal and geographical variations in the utilization of the assigned spectrum range from 15% to 85% [1]. So the feasible solution to mitigate the shortage of useable spectrum is to increase the spectrum usage effectively.

Cognitive radio has become one potential technology to solve the problem of spectrum shortage by allowing the SUs to opportunistically access the spectral white space of PUs and try the best to control the interference perceived by the primary users at the same time. Cognitive radio has attracted much research attention [2]. However, there are still a lot of works needed to do for the cognitive radio researches, especially for the testbed researches. On one hand, while a large number of CR algorithms have been proposed to improve the performance of CR networks, most of these algorithms have not been evaluated in a practical system. The feasibility and real performance of these algorithms remain unknown. On the other hand, the existing testbeds are not flexible enough for the evaluation of the complete cognitive radio protocol and do not support large scale cognitive radio networks.

Some of existing CR testbed researches are focusing on the algorithms of spectrum sensing and signal processing. In [3], the authors introduce a real time cognitive radio testbed for the evaluation and comparison of different spectrum sensing technologies based on a multi-FPGA emulation platform called BEE2. There are other testbeds based on the IEEE 802.11 hardware and targeting at upper layer protocol design. For example, the authors of [4] propose KNOWS, which has a reconfigurable transceiver based on a modified WiFi hardware. They study the bandwidth allocation problem and design a MAC protocol. However, because of limited hardware reconfiguration ability, the function of cognitive radio cannot be fully explored with these kinds of cognitive radio testbed. Some testbeds use powerful hardware to support large network scale. DARPA XG Radio Program [5], [6] is one well developed testbed. The authors implement large scale tests to demonstrate the spectrum sharing with no harmful interference. However, such a testbed is too expensive and the deployment is too time-consuming, so it is not suitable for most of researchers. There are some theoretical researches and testbeds [7] that propose to equip SUs with two transceivers and a dedicated control channel to obtain accurate spectrum state. However, these requirements on hardware may not be able to be set up in the practical situation.

In this work, we set up a real time cognitive radio testbed based on the SoC programmable processors that not only fully supports the cognitive radio function but also can support quick and economic deployment of large network scale. We realize the general functions of cognitive radio and establish a multi-PU multi-SU network to prove the feasibility of cognitive radio. The testbed supports high level program language to realize quick configuration of cognitive radio MAC algorithm. Through some changes of the configuration, the testbed can implement the algorithm of wireless sensor network such as [8] and achieve the combined application of cognitive radio and wireless sensor network. Both PU and SU in the network are controllable to achieve online adjustment. We apply energy sensing to supply straightforward and accurate radio environment information for SUs. Besides, we design an adaptive access control algorithm to fulfill the functions of spectrum analysis and spectrum decision of SUs. As an experiment, we use the testbed to study the performance of our proposed adaptive cognitive radio MAC algorithm and measure the cognitive functions of our testbed.
Our contributions are as follows:

- Establishment of a real time cognitive radio testbed that solves the key difficulties in the implementation and realizes the general functions of cognitive radio.
- Proposal of a new kind of cognitive radio testbed based on SoC programmable processors which decrease the complexity of configuration and the difficulty to set up large scale cognitive radio network.
- Development of an adaptive MAC strategy that enables multiple SUs to control the accumulated interference of all SUs under the PU requirement in a distributed way.

The rest part of the paper is organized as follows. We give the overview of the testbed in Section II. In Section III, we present the details of the implementation of the real-time cognitive radio testbed. The experiment and result are presented in Section IV. We conclude the paper in Section V.

II. OVERVIEW OF TESTBED

This section gives an overview of our testbed. A real time cognitive radio testbed should fulfill several basic requirements that includes the ability to support multiple radios, which can serve as PUs or SUs, respectively, the ability to obtain the radio environment information accurately and quickly to limit the interference to PUs and the ability to carry out online adjustment of operating parameters of SUs.

The testbed requirements stated above are well met by our proposed real time testbed. The testbed is based on the SoC programmable processors, CC1110 from TI. Fig. 1 shows a picture of the processor and its demonstration board.

CC1110 is a true low-power SoC processor designed for wireless applications. It has the state-of-the-art RF transceiver with the excellent performance and an industry-standard enhanced 8051 MCU. The MCU can be programmed in high level programming language, C++, which can support much quicker prototyping than the traditional testbeds based on FPGA. There are three 8-bit timer and one 16-bit timer in the processor that can be used to precisely control the busy/idle time of PU and sensing interval of SU. The RF transceiver is programmable with operating parameters like transmission power, frequency, data rate and so on. The RF frequency is fully programmable in the three ISM bands including 300.0-348.0 MHz, 391.0-464.0 MHz and 782.0-928.0 MHz. The small size and low cost make it feasible to set up a large scale network using these processors.

Through different configuration, the processors can represent SU and PU, respectively. Both PU and SU are controllable to adapt to different experiments. SU’s framework is illustrated as Fig. 2. Energy sensing based on Receive Signal Strength Indicator (RSSI) is used to realize accurate and quick spectrum sensing function. Here the channel selector considers whether it is necessary to hop to another PU channel for more chances of spectrum access and the access controller determines whether it is “safe” to transmit, and if yes, how long the transmission should be to avoid collision with returning PU. The transmission time threshold of SU is varied according to the estimation of the interference perceived by PU.

III. DESIGN AND IMPLEMENTATION OF TESTBED

In this section we describe the testbed implementation details through four key parts. The whole MAC protocol is implemented as C++ codes which run in CC1110’s 8051 core.

A. PU and SU Traffic Model Design

Our testbed is based on a multi-PU multi-SU cognitive radio networks. PUs and SUs are implemented through different configuration of SoC programmable processors. In our model PUs and SUs are not synchronized with each other. Each pair of PUs work at different channel and do not interfere with each other. PU’s activities follow an alternating IDLE-BUSY pattern. The idle time and busy time of PU follow general distribution such as exponential distribution. [9] shows that the inter-arrival time of voice user in CDMA network follows exponential distribution. The busy time of PU is composed of many slots. Each slot represent a transmission of a data packet. So if it collides with others, PU only retransmits the collided packet instead of all data. We use random number generator and timer to implement the IDLE-BUSY pattern. The idle time and busy time of PU follow general distribution such as exponential distribution. [9] shows that the inter-arrival time of voice user in CDMA network follows exponential distribution. The busy time of PU is composed of many slots. Each slot represent a transmission of a data packet. So if it collides with others, PU only retransmits the collided packet instead of all data. We use random number generator and timer to implement the IDLE-BUSY pattern of PU. Random number generator is used to generate proper countdown number for the 8-bit timer. Then the timer will generate an interrupt to inform 8051 core when its countdown process is over. In this way our testbed achieves the precise control of the transmission time and idle time of PU according to the specific distribution.
SU’s activities is also based on the slotted packet way. SU’s packet is shorter than PU’s packet. That means each collision will cause transmission failure of only one PU’s packet. SUs always have packets to transmit, which is consistent with the objective of maximizing spectrum usage. SUs need to choose one channel from the PUs’ according to a channel selection strategy before spectrum sensing. After choosing one channel, SU follows the listen-before-talk principle. The competition among multiple SUs targeting the same PU channel is solved by using CSMA/CA protocol. SU only initiates transmission when the PU is idle and there is no other SU accessing the channel before itself. The spectrum usage information obtained by the spectrum sensing part is analyzed by access controller of SU which decides whether and how long SU should access the channel. If it collides with PU, SU leaves the channel immediately and waits for the chance of transmission until the next idle time of PU.

B. Spectrum Sensing

Due to the fairly high SNR conditions, energy sensing can be used efficiently with very little complexity. So we use energy sensing in our testbed and pay more attention on the design and evaluation of upper layer strategy.

Energy sensing can be mathematically formulated as a binary hypothesis testing problem on a set of $N$ samples that either have just noise, or a mixture of signal and noise, respectively. In theory, the threshold $\gamma$ can be determined if closed-form expressions for the probability of false-alarm and success detection are obtained. In the real scenario, however, such expressions are not available because many factors other than signal and noise needed to be considered. So we calculate the value range of $\gamma$ with the theoretical expressions and tune the optional threshold through many experiments.

For our testbed, we use periodic sensing in the implementation. Our energy detection is implemented by using RSSI value. In each detection the sensor updates the RSSI value and compares it with the predefined threshold. The idle state is defined as the state when RSSI is lower than the threshold at the consecutive two times.

The oscillograph snapshot of Tx_on signal of PU and SU in Fig. 3 gives a real look of the performance of the spectrum sensing. The green line is the Tx_on signal of PU and the red one is that of SU. Here electric level 1 means transmission and 0 means idle. The PU’s transmission time and idle time are randomly chosen following uniform distribution, while the SU’s transmission time is controlled by an adaptive threshold to avoid collision with PU. It can be observed that the SU’s spectrum sensing part properly finds the beginning of PU’s idle time. The latency between the end of PU’s transmission and the beginning of SU’s transmission is due to the random defer interval to avoid the collision among SUs.

C. Channel Selection Strategy

There are multiple PUs in the cognitive radio network of our testbed and each PU works at different frequency as mentioned before. So SUs need to choose one PU channel before spectrum sensing and transmission. We assume all the PU channels are i.i.d., so they have same value for SUs. We also assume the number of PU are comparable to the number of SU. In this situation, the decisive fact of channel selection is the number of SUs sharing the same channel. For example, if there is only one SU accessing channel $C_1$, this SU can at mostly enjoy the whole idle time of the channel. However, if there are two SUs accessing $C_1$ and they can not transmit simultaneously because of interference, each one can only have half of chance to utilize the idle channel. So SU’s channel selection strategy is to find the channel that shared by least SUs to maximize their chance of accessing idle channel.

To find the least-shared channel, our strategy makes use of the result of random defer interval. After the random defer interval, only one SU finally wins the competition and starts transmission. All the losing SUs in the competition know the existence of another SU by sensing spectrum after the defer, then each of them hops to another channel randomly to try finding the still-no-SU channel. If there are no such channel, SU changes to search the channel already accessed by two SUs, then three SUs, until it finds the channel shared by least SUs. This strategy scatters the SU uniformly to the PU channels and helps maximize the capacity of the whole secondary networks.

D. Access Control Strategy

After spectrum sensor reports that the PU channel is idle, the upper layer will analyze the spectrum information and decide the access policy. Here, we firstly give the denotation used in the following discussion. We denote the mean of PU’s idle time and busy time as $l_I$ and $l_B$, respectively. The probability density function (PDF) of PU’s idle time and busy time are denoted as $f_I(t)$ and $f_B(t)$, and the cumulative distribution function (CDF) as $F_I(t)$ and $F_B(t)$. $\eta_p$ represents the mean of transmission packet number of PU and $\eta$ is the collision probability constraint predefined by PU. If we denote the transmission probability of the SU at $t$ as $d(t)$, then the optimal spectrum access policy for a single SU is as follow.

$$
\begin{align*}
    d^*(t) = \begin{cases} 
    1, & \text{if } g(t) < \gamma^*, O(t) = \text{Idle} \\
    p^*, & \text{if } g(t) = \gamma^*, O(t) = \text{Idle} \\
    0, & \text{otherwise}
    \end{cases}
\end{align*}
$$

Fig. 3. Tx_on Signal of PU and SU
where $O(t)$ is the sensing result at time instance $t$. The definition of $g(t)$ is as follows

$$g(t) = \frac{f_B(t)}{1 - F_B(t)} \quad (2)$$

Here $g(t)$ is the function representing the conditional probability density that the PU will return at time instance $t$ given that it has been idle until $t$. And $\gamma^*$ and $p^*$ can be obtained from the following expressions:

$$\gamma^* = \inf \{ \gamma : \int_{\tau : g(\tau) < \gamma} f_B(\tau) d\tau \leq n_p \eta \} \quad (3)$$

$$p^* = \frac{n_p \eta - \int_{\tau : g(\tau) > \gamma^*} f_B(\tau) d\tau}{\int_{\tau : g(\tau) = \gamma^*} f_B(\tau) d\tau} \quad (4)$$

The policy shows that we should not allow SU to access the channel when $g(t)$ is too large even if the spectrum sensing reports that the PU channel is idle. [10] has shown that the threshold-based policy mentioned above is optimal for the scenario of one SU.

For most of the distributions that have the monotonically increasing and continuous $g(t)$, there is a simpler time-threshold form as follows.

$$d^* = \begin{cases} 1, & \text{if } t \leq T^*, \ O(t) = \text{Idle} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where $T^*$ can be obtained from the following expression:

$$\int_0^{T^*} f_B(t) dt = n_p \eta \quad (6)$$

If the distribution of PU idle time is known by SUs at first or this information is estimated, then SUs can calculate the value of $T^*$ by (6). According to the access control strategy, SU controls its access behavior based on the time and stops its transmission when $T^*$ is reached.

However, when we study the scenario of multiple SUs or where the statistics of the PU traffic pattern change, the above access control strategy is poor at the protection of PU. When there are multiple SUs spread in the network, their interference to PU accumulate to an degree much larger than the requirement of PU. Moreover, PU’s traffic pattern may also change over time. Though there are schemes to estimate distribution function, they usually require a large number of time to achieve the accuracy. So we need an adaptive spectrum access algorithm for SU to tackle these scenarios.

Our testbed supports the scenarios with multiple SUs and changing PU’s traffic pattern. Here we assume that all SUs have knowledge about the average number of PU packets in a busy period, $n_p$, which is much easier to obtain than the distribution functions. And we assume that the mean of PU packets do not change when PU’s distribution changes. When PU’s traffic pattern or the number of SUs changes over time, the interference to PU is also dynamically changing. Because there is no control channel in our testbed, each SU only knows the number of collisions caused by it and is not able to obtain the total number of collisions by exchanging these collision numbers with other SUs. However, we may notice that the collisions directly cause a large number of transmission failures of PU. In our model, PU retransmits these failed packets immediately and keep retransmitting until receiving ACK from the receiver. The more collisions occur, the more retransmissions PU conducts. The retransmission causes the increase of the total busy time of PU. So SUs can use the number of retransmission packets or, more straightly, the changing length of PU’s busy time to learn the number of collisions perceived by PU totally.

In our model, we label $M$ SUs sharing one PU channel in the network with superscript $m = 1, 2, \ldots, M$. Each SU needs to record the length of PU busy time during the $k$th PU busy-idle period, denoted as $L_B^m(k)$. Because of period sensing, SUs can accurately detect the beginning and the ending of PU busy time. The interval between two times of sensing is constant, so the product of sensing interval and the consecutive times when sensing result is “Busy” can be used to estimate the length of PU busy time. The new estimation of the PU busy time is updated per $W$ times of PU busy-idle period as follows.

$$I_B^m(i) = \sum_{k=(i-1)W+1}^{i W} L_B^m(k) \quad (7)$$

The SU updates $T^m(i)$ per $W$ periods as follows.

$$T^m(i+1) = T^m(i) + \delta_i (T^m(0) - I_B^m(i)) \quad (8)$$

where $T^m(0)$ is the initial time threshold, and $\delta_i$ is the step-size. So when $I_B^m(i) > I_B$, i.e. if more collisions perceived by PU than its requirement, the SU reduces its transmission time; otherwise, the SU increases its transmission time. Here the value of $\delta_i$ and $W$ are determined according to the requirement of the convergence speed of the algorithm.

IV. EXPERIMENT DESIGN AND RESULTS

We establish an experiment to prove the cognitive function of our testbed and to evaluate the performance of the adaptive MAC algorithm. The test scenario of our experiment is shown as Fig. 4. It consists of two pairs of PUs and three pairs of SUs. The two pairs of PU transmit on two non-overlapping channels, Channel 1 and Channel 2, and do not interfere with each other. However, there are interference between SUs, so they try to choose different channel according to the channel selection strategy first. In our experiment, the final result of the channel selection are that SU1 chooses Channel 1 and both SU2 and SU3 choose Channel 2.

We use the testbed to evaluate the performance of the proposed adaptive access control strategy. The transmission rate is 250kBaud and the modulation method is GFSK. The packet length of PU, $K_p$, and the packet length of SU, $K_S$ are 4ms and 3ms, respectively. The random defer interval of SU is randomly chosen between 1ms to 16ms. Every $W = 50$ idle-busy periods SU updates its estimated mean of PU busy time and calculates the new time threshold. Initially, we set the busy time of PU following uniform distribution and the
idle time following exponential distribution. After 500 idle-busy periods, the PU’s idle time distribution changes from exponential distribution to uniform distribution. Finally, after 1500 idle-busy periods, SU3 leaves the network.

The performance of the proposed algorithm is shown in Fig. 5, including the collision probability, $p_c$, the time threshold, $T^*$ and the capacity of SU, $C_s$. The result shows that the initial time threshold $T^*$ is a little larger than the optimal one, so SU2 and SU3 distributively adjust $T^*$ of itself to control the collision probability $p_c$ back to the required level. When PU’s idle time distribution changes, SU2 and SU3 converge to the new time threshold quickly. The new time threshold is larger because the long idle time is more likely to occur with uniform distribution than with exponential distribution. After the leave of SU3, SU2 adaptively increases its access time and gains the whole opportunistic spectrum chances which is sharing between SU2 and SU3 before. The experiment shows that the testbed works quite well and proves the real performance of the proposed MAC algorithm.

V. CONCLUSION

In this paper we present a real time testbed for the evaluation of cognitive radio MAC algorithms. The testbed is based on programmable SoC processors, CC1110, which makes it easier and more economic to deploy and evaluate cognitive radio network than the traditional testbed based on FPGA. The testbed enables the full control and online adjustment of PU and SU. We introduce the implementation details of spectrum sensing of PHY layer, the channel selection strategy and the access control strategy of MAC layer. With such a testbed, we experiment the proposed adaptive MAC algorithm and prove the cognitive functions of our testbed.

VI. ACKNOWLEDGMENT

This work is supported by National Fundamental research grant (2010CB731803, 2009CB3020402), NSF China (No. 60702046, 60832005, 60972050, 60632040); China Ministry of Education (No. 20070248095); China Ministry of Education Fok Ying Tung Fund (No. 122002); Qualcomm Research Grant; China International Science and Technology Cooperation Program (No. 2008DFA11630); PUIJANG Talents (08PJ14067); Shanghai Innovation Key Project (08511500400); National Key Project of China (2009ZX03003-006-03, 2009ZX03002-003, 2009ZX03002-005); National High tech grant of China (2009AA01Z248, 2009AA11Z249, 2009AA011802);

REFERENCES