SPECTRUM SENSING ON LTE FEMTOCELLS FOR GSM SPECTRUM RE-FARMING USING XILINX FPGAs

Jörg Lotze (CTVR, Trinity College Dublin, Ireland. jlotze@tcd.ie);
Suhaib A. Fahmy (CTVR, Trinity College Dublin, Ireland. suhaib.fahmy@tcd.ie);
Barış Özgül (CTVR, Trinity College Dublin, Ireland. ozgulb@tcd.ie);
Juanjo Noguera (Xilinx Research Labs, Ireland. juanjo.noguera@xilinx.com);
Linda E. Doyle (CTVR, Trinity College Dublin, Ireland. linda.doyle@tcd.ie).

ABSTRACT

Femtocells are a promising solution to provide high coverage and high data rates inside consumer's homes while cutting operator costs significantly. Next generation Long Term Evolution (LTE) femtocells are likely to be deployed in GSM spectrum, increasing frequency utilisation and allowing a smooth transition to LTE. This paper proposes to use a spectrum sensing technique specialised for LTE signals to avoid interference between neighbour femtocells without operator intervention. Simulation results of the detection characteristics are given. A demonstrator has been implemented on a Xilinx ML507 Virtex 5 prototyping board, using our flexible FPGA-based cognitive radio framework, which allows to use runtime reconfiguration of the FPGA to switch between sensing and normal operation. It demonstrates the sensing algorithm on a real platform.

1. INTRODUCTION

Femtocells are consumer-deployed small base stations that use the Internet as backhaul to provide fast and exclusive cellular services to consumers' homes. They offer great potential to increase system capacity and coverage, and at the same time minimise operational costs [1]. Autonomous deployment is essential since operators have no control over where exactly a femtocell will be installed. It is required that femtocells avoid interference with macrocells and neighbouring femtocells, through careful power control and frequency allocation.

It is likely that next generation Long Term Evolution (LTE) systems will be deployed in vacant GSM spectrum to maximise frequency reuse [2]. This is a perfect opportunity for femtocells, enabling a smooth transition to LTE. Figure 1 illustrates a deployment scenario, where the frequency channels allocated to GSM macrocells B and C comprise the spectrum available for the operation of femtocells inside the area covered by GSM macrocell A.

Unlike UMTS, where all base stations use the same frequency bands and can be distinguished via code, LTE requires exclusive frequency access. Therefore femtocells within close proximity need to coordinate the use of spectrum to avoid interference. However, due to autonomous setup, femtocells do not know about the properties of other nearby femtocells in advance and cannot coordinate with them.

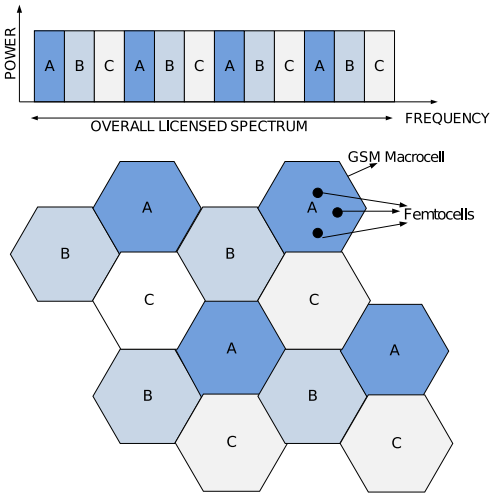

Fig. 1: GSM frequency re-use and femtocell deployment.

An attractive solution to this problem is to avoid interference by carefully controlling transmission power so as to only just cover the user's home. Yet, this method cannot guarantee interference-free operation since the femtocell must also provide complete coverage in the user's home. If the user places the femtocell too close to an outside wall or window, it may not be able to give full coverage while avoiding leakage to a neighbour at the same time. Thus, an LTE femtocell needs to detect if the frequency band it intends to use is already occupied by another nearby femtocell before starting to operate. This can be achieved by acting like a mobile handset and trying to decode the neighbour's control channel, but the signal can be too weak for a reliable detection, while still likely to cause interference.

A promising solution to this problem is spectrum sensing. It allows a femtocell to detect the presence of neighbouring femtocells without the need to decode their signals. It is even possible to detect weak signals with this technique, to guarantee interference-free operation.

Sensing is required during initial system startup, and very rarely during operation since the femtocells are not expected to change their operating frequency. Hence it is beneficial to use the same hardware resources for both sensing and normal operation modes, and reconfigure to one mode or the other, as needed. Xilinx FPGAs allow the reconfiguration of parts of the device while other parts continue to operate. This capability can be leveraged to allow switching between sensing and normal operation without wasting hardware resources and power, and thus using a smaller device.

In this paper we describe a sensing technique suitable for LTE femtocells as well as an implementation and demonstration of it on a run-time reconfigurable FPGA-based cognitive radio platform. The sensing algorithm, along with detection characteristics, is described in Section 2, and the hardware implementation is discussed in Section 3. Section 4 draws the conclusions.

2. SENSING LTE SIGNALS

The sensing algorithm in this paper is based on estimating time-averaged power spectral density (PSD), performing moving average filtering and decision thresholding, followed by an appropriate peak detection technique. The proposed algorithm is tailored for sensing LTE signals and also takes into account femtocell network frequency planning, as adopted by the mobile operator, to improve detection performance and reduce false alarms. In order to retrieve the frequency planning information, the femtocell basestation can decode the macrocell basestation's identity over the air after locking onto the GSM broadcast control channel (BCCH) carrier available in the GSM downlink. It can then use the decoded information to interrogate an operator-specific database through its Internet backhaul and obtain the frequencies and transmission bandwidth allowed for use by LTE femtocells. This is why the algorithm discussed here is only designed to sense LTE signal activity in these predefined frequencies; this improves robustness and computational efficiency significantly. In the remainder of this section, after giving brief information on LTE signals, the proposed sensing algorithm is described in detail and simulation results are presented to illustrate detection characteristics.

2.1. LTE Physical Layer Properties

The LTE [3] downlink and uplink transmission schemes are based on orthogonal frequency division multiple access (OFDMA) and single carrier frequency division multiple access (SC-FDMA), respectively. LTE supports several channel bandwidth modes ranging from 1.4 MHz to 20 MHz. Scalable bandwidth is a property of LTE that makes it attractive for deployment in existing GSM uplink and/or downlink bands which can reach 35 MHz and 75 MHz for GSM-900 and GSM-1800, respectively [4]. The basic LTE scheduling unit in both downlink and uplink is called a *resource block* (RB) and consists of 12 subcarriers with a spacing of 15 kHz (corresponding to 180 kHz overall) in the frequency domain and 6 or 7 consecutive OFDM symbols (SC-FDMA symbols for the uplink) in the time domain. The number of available RBs in the frequency domain varies depending on the channel bandwidth, which increases from 6 to 100 when the bandwidth changes from 1.4 MHz to 20 MHz, respectively. In the time domain, each RB spans a *slot*, with a duration equivalent to 6 or 7 symbols (0.5 ms). Two slots correspond to a *subframe* and 10 subframes typically form a *frame* (10 ms). LTE supports both time division duplexing (TDD) and frequency division duplexing (FDD). For TDD, a subframe within a frame can be allocated to downlink or uplink transmissions. In the case of FDD, because the downlink and uplink transmissions are separated in the frequency domain, there is no allocation of subframes in time.

2.2. Sensing Algorithm

As explained in the beginning of Section 2, the spectrum available to the LTE femtocell is divided into channels. Since a femtocell basestation serves only a few home-based users within a very short transmission range, the channel bandwidth is likely to be the smallest possible LTE bandwidth, 1.4 MHz. In the simulations and implementation we assume this bandwidth, although the same algorithm can be applied to higher LTE bandwidths. Using a small bandwidth is also preferable, since it allows the allocation of more frequency channels in the unoccupied GSM spectrum and, therefore, increases the number of neighbouring LTE femtocells which can coexist without mutual interference. In order to avoid such interference, a femtocell basestation needs to detect the signal activity in the frequency channels used by the neighbouring femtocells.

The sensing algorithm in this paper is based on signal detection using a time-averaged PSD estimate. The key feature that is making this sensing method reliable is the LTE-specific control and synchronisation signalling which typically occupies 72 subcarriers (all available subcarriers in 1.4 MHz mode, see Section 2.1) around the DC carrier. For instance, regardless of TDD or FDD mode, corresponding subcarriers of some OFDM symbols in particular downlink subframes are always allocated, even with no user activity. These carry data for certain LTE physical channels, such as data for the physical broadcast channel or for primary and secondary synchronisation signalling [3]. If we also consider the fact that LTE or GSM signals are the only sources of activity in the refarmed GSM spectrum, we see that it is possible to obtain a distinctive spectral shape in the presence of an LTE signal after sufficient time-averaging. It is also important to note that a long time average is reasonable since sensing takes place during initial system startup and very rarely during runtime.

The sensing algorithm for LTE femtocells has to discover whether there is an LTE signal present in a particular channel. A flow-chart of the complete sensing algorithm is shown

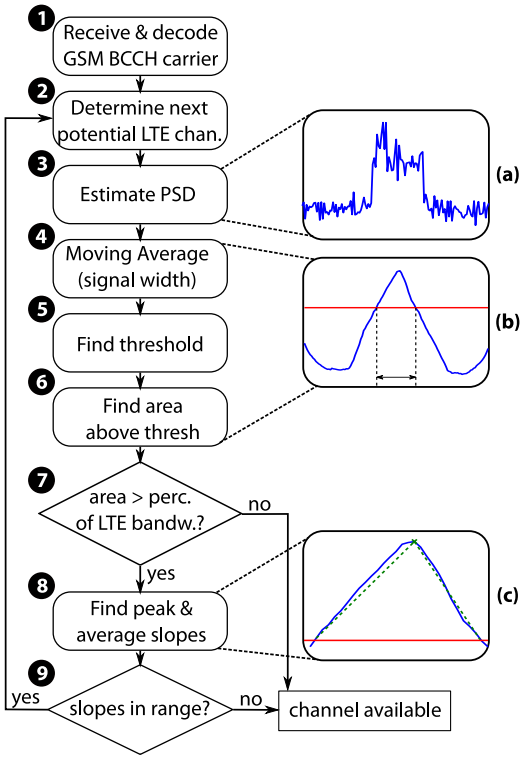

Fig. 2: LTE signal detection algorithm.

in Fig. 2. Details of all steps in the figure are given in the following.

In steps 1 and 2 the femtocell retrieves information about the channels and bandwidth available to it from the GSM BCCH carrier and the operator core network, as explained in the beginning of Section 2. To estimate the PSD (Step 3), an FFT of the incoming signal is computed, with a bandwidth wide enough to fully include the LTE signal with some additional margin. The PSD estimate is obtained by computing the square magnitudes of the instantaneous FFT, and averaging over a given time period. This period can be relatively long, since accuracy is more important than speed during setup. The PSD is then normalised to its mean power, to ensure power-independent characteristics for the remainder of the algorithm. An example of such an averaged PSD is shown in plot (a) in Fig. 2.

Applying a rectangular moving average window (Step 4) smoothes the PSD and effectively correlates the PSD with the expected LTE footprint (which is also approximately rectangular). This results in a triangular-shaped spectrum if an LTE signal is present, as shown in plot (b) in Fig. 2.

Next, we apply the thresholding technique proposed in [5] (Step 5), which is based on the mean μ and standard deviation σ of the spectrum. The threshold δ is determined by

$$\delta = \mu + c \cdot \sigma, \tag{1}$$

where c is a positive constant. Typical values for c are around 1. The threshold ($c = 1$) is illustrated by the horizontal line in plot (b) of Fig. 2.

In Step 6 we determine the width of largest consecutive area above the threshold δ (dotted lines in plot (b) of Fig. 2). If the LTE signal is present, it is expected that the width of the area above the threshold is around 50–70 % of the full LTE signal bandwidth (with $c = 1$). Therefore, we choose a percentage p below which we assume that no LTE signal is present in the current channel (Step 7), and thus, the channel is available.

If the width of the area above the threshold is above p %, we further check the slopes of the triangle. With an LTE signal present and in the absence of noise, the slopes of the triangle in the windowed spectrum are known. We use these slopes as another criterion for signal detection. We detect the peak of the area above the threshold and compute the average slopes left and right of the peak (slopes are illustrated by the dashed lines in plot (c) of Fig. 2). If these slopes are within range of the known slope, an LTE signal has been detected. Otherwise the signal is not an LTE signal, and the channel is therefore available to the femtocell.

This search continues for all available channels in the current GSM macrocell until a free one has been found. In the unlikely event that no channel is available, the femtocell chooses the one with the lowest detected signal power, and moderates its own power level, perhaps sacrificing full coverage of the user's home. Other strategies might be possible in this case, but this is beyond the scope of this paper.

Note that the computational complexity of this algorithm is only slightly higher than for pure energy detection, with the only additional operations being those necessary for steps 6 to 9. Furthermore, these steps only need to be performed once, after PSD estimation is completed. The PSD estimation requires repeated FFT calculations and averaging and hence dominates the computational complexity. This makes the algorithm suitable for implementation on low-cost embedded devices, as required for femtocells.

2.3. Detection Characteristics

The proposed sensing algorithm has the following parameters: number of FFT bins F, PSD averaging time t_{avg}, threshold factor c, and required percentage of samples above the threshold p. To determine the best parameter set and the algorithm's detection characteristics, we simulated the scenario in MATLAB.

The sensing algorithm is required to find the LTE signals even without active user traffic, i.e., using only the synchronisation and control LTE symbols (see Section 2.2). Thus, we generate an LTE downlink signal without user data, with a bandwidth of 1.4 MHz. As explained earlier, the algorithm presented here is not limited to this bandwidth. We upsample the signal to 2 MHz and add a small frequency offset, to

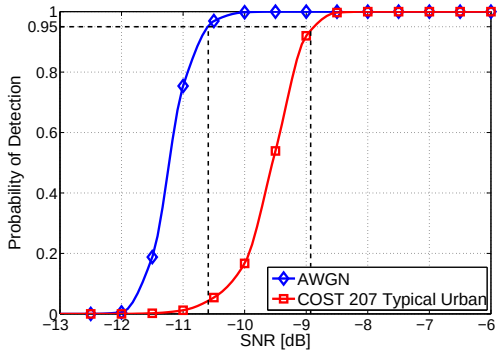

Fig. 3: Sensing Detection Characteristics for Rayleigh-fading and white noise channels. ($F = 128$, $c = 1$, $p = 60\,\%$, $t_{\text{avg}} = 0.1$ s). SNRs for 95 % Probability of Detection are -10.6 dB and -8.9 dB.

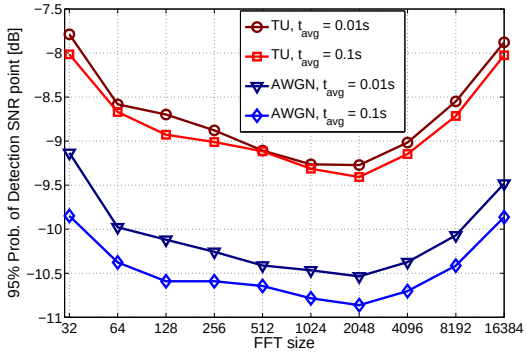

Fig. 4: 95 % Probability of detection SNR points vs. FFT size. Different averaging times are shown, as well as the white noise (AWGN) and COST 207 Typical Urban (TU) channels.

destroy the subcarrier orthogonality and simulate local oscillator offsets at the receiver. The signal is then sent through a wireless Rayleigh-fading channel, with parameters according to the COST 207 Typical Urban scenario [6], and an Additive White Gaussian Noise (AWGN) channel. The sensing algorithm is applied to the received signal.

Note that the COST 207 Typical Urban channel model can be considered a worst-case scenario, since femtocells are intended to be used indoors in private homes. We use it as a benchmark here, to demonstrate the capabilities of the proposed sensing algorithm.

From plot (b) in Fig. 2 it can be seen that the parameter c, which determines the level of the threshold, and the parameter p, which is the percentage of the LTE bandwidth required above the threshold, depend on each other. Through a large number of simulations we found that the best results for $c = 1$ are obtained with $p = 60\,\%$. We used these values for all other simulations.

The simulation to determine the detection characteristics was executed 2,000 times, for a range of SNR values, to obtain good averages for the probability of detection and probability of false alarm. We also simulated the dependency of the results on different FFT sizes F. With averaging times 0.01 s and 0.1 s, equivalent to 1 and 10 LTE frames, respectively, we found no false alarms for all simulated SNR values and FFT sizes. This means the probability of false alarm is less than $5 \cdot 10^{-4}$. The probability of detection versus SNR for $F = 128$ is shown in Fig. 3.

Given a fixed averaging time, as the number of FFT bins F increases, giving higher frequency resolution, less instantaneous FFT windows can be averaged. A higher resolution also reduces spectral leakage, which increases the quality of the estimated PSD. To examine the dependency on F, Fig. 4 shows the SNR point for 95 % probability of detection versus the number of frequency bins F.

It can be seen that for $F \leq 2048$, more resolution in frequency gives better detection results. However, the improvement decreases as F increases, with only fractions of a dB at high F values. If F goes beyond 2,048, the detection performance worsens again, which can be explained by the lower number of FFT windows for averaging. It is expected that the optimum point, 2,048 in this case, depends on the averaging time and the experienced channel conditions. However, the difference between $F = 64$ and $F = 2048$ is only about half a dB, while the computational complexity of the FFT increases with $O(F \log F)$ [7]. Additionally, as F increases, more computations are required for computing the square magnitudes and average, in order to obtain a PSD etimate. This is a trade-off to be carefully considered in practical implementations.

It can be seen that the proposed sensing algorithm is able to detect LTE signals of neighbouring femtocells reliably, even at SNR values around -9 dB.

In the following section we discuss the implementation of this technique on a flexible FPGA-based platform.

3. FPGA-BASED PROTOTYPE

We have implemented a demo that illustrates the use of our LTE signal detector in a femtocell-like scenario. We use a pre-recorded LTE signal for the interference, and a prior implementation of coded narrowband video transmission for the femtocell transmission, though in a real system this would clearly be done using LTE. The hardware implementation of this system is built upon our FPGA-based Adaptive Systems framework described in [8,9], which we have previously used to implement a number of other applications, including adaptive coding in a video transmission [9] and frequency rendezvous using spectrum sensing [10].

3.1. IRIS Framework

The IRIS framework allows radio designers with no hardware experience to leverage the performance and flexibility advantages of FPGAs as a target platform for cognitive radios. This

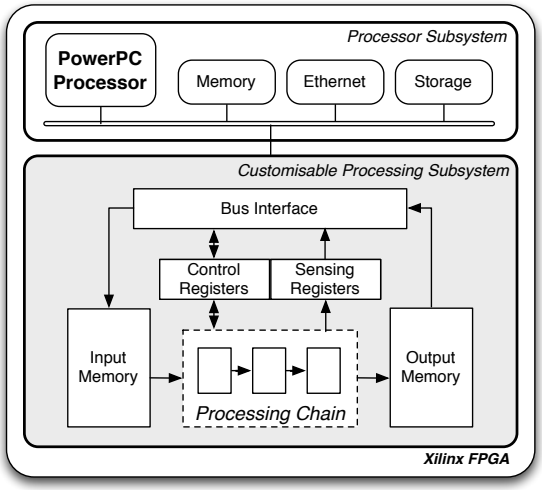

Fig. 5: The virtual architecture. The Processor Subsystem contains basic hardware for running Linux, the Customisable Processing Subsystem is used for radio configurations, which can be reconfigured at run-time.

is achieved through three aspects of the framework: a *Virtual Architecture* that abstracts away the physical implementation platform and runs Linux, and atop it our Software Radio platform, IRIS (shown in Fig. 5); a *Runtime System* that executes the cognitive part of the radio, managing reconfiguration seamlessly from the perspective of the radio designer; and *Compile-time tools* that partition and prepare the necessary FPGA configurations for use by the Runtime System.

The framework separates the design of the processing plane from the control plane. The processing plane is described using an XML file that connects components selected from a pre-existent component library. The control is implemented in a piece of custom software called the *Decision Engine*. This is written in C++ and accesses the high-level description of processing chains and the parameters of components within. Since the Runtime system manages access to the chains and components at an abstracted level, it can combine software and hardware components together and does not care about the type of component it is accessing. Compile-time tools then take care of partitioning and generating the necessary bitstreams, in a process that is hidden from the designer. This allows for rapid radio design, taking full advantage of the speed and flexibility of FPGA implementation without the need to be versed in FPGA design.

We have implemented this demo on a Xilinx ML507 development board [11], which hosts a Xilinx Virtex-5 XC5VFX70 FPGA. The FPGA is partitioned into two regions; the static processor subsystem, that contains the hard PowerPC core along with some basic bus/interface logic, and the customisable processing subsystem, which can be configured to undertake processing tasks. On the processor subsystem we run a standard Linux Kernel, and the IRIS Runtime Engine, which has been extended to deal with hardware components. We are able to combine custom hardware modules written in VHDL, or other hardware design languages, to build processing chains that are configured in the customisable processing sybsystem by the Runtime System through the runtime partial reconfiguration capability of Xilinx FPGAs.

3.2. Sensing in Hardware

The computational complexity of the sensing algorithm is concentrated in the estimation of the PSD. An average of successive FFT windows has to be computed in real time, for a large number of windows. Therefore, it is this part of the algorithm that is implemented in the FPGA logic. The detection part of the algorithm (from Step 4 onwards in Fig. 2) only needs to run once the PSD has been estimated, and does not need to be executed in real-time. Since this part would not benefit from a custom hardware implementation, it is implemented as an IRIS software component, to be executed on the PowerPC. This also allows us to reuse the hardware PSD estimator for other types of signal analysis.

In order to implement the LTE signal detector in hardware, a new VHDL module was designed. The basic outline of the energy detector contains two units; the FFT and an averaging memory. The FFT gives an instantaneous snapshot of the current spectrum of the signal for a single input window. To implement the FFT, we use the Xilinx FFT core, which provides some useful features including a run-time adjustable FFT size. Computing the instantenous PSD from the complex FFT output simply consists of taking the squared magnitude. To accommodate the averaging, we store successive windows of PSD results in a memory. For the first window in each averaging period, we store the result directly, then for successive windows in the same period, we simply accumulate the results for each output bin. So in the final window of an averaging window of size K, we have an accumulation of K power density results. To output the average, we only need to divide by K. Since we only allow numbers of averaging windows that are powers of 2, this division maps to a right-shift of the result by the corresponding number of bits. The averaged result is then used by the software component.

The detection component is implemented as a standard IRIS component in C++. It performs steps 4 to 9 of Fig. 2, as described in Section 2.2. If an LTE signal is detected in the current channel, the Decision Engine is informed, which tunes the radio frontend to the next channel and sensing executes once again.

3.3. Demo Setup

The demonstration consists of three radio nodes as illustrated in Fig. 6. The first broadcasts the pre-recorded LTE-like signal and acts as the interfering femtocell (1 in Fig. 6). The second node is the transmitter, which searches the spectrum

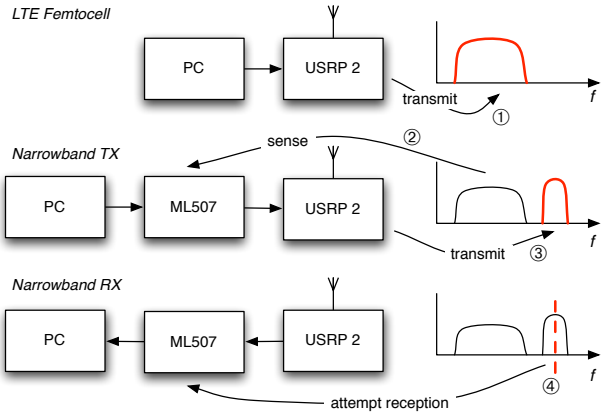

Fig. 6: The three nodes in this demonstrator.

to find spare channels (2 in Fig. 6), then begins to transmit the streaming video using coded DQPSK modulation (3 in Fig. 6). The final node is the receiver, which attempts to receive on each of the available channels, locks onto the transmission, and decodes the video (4 in Fig. 6).

The broadcast node is fixed, and is used to simulate the presence of a neighbouring femtocell's LTE signal, as would be the expected case in the environs of a femtocell. The transmitter node starts in sensing mode, and uses the algorithm described in Section 2.2 to locate a channel that does not interfere with the LTE signal. Once it has found a free channel, it switches into transmission mode and begins transmitting the streamed video, using framed, DQPSK transmission, coded using a convolutional code with constraint length 9 and code rate 1/2, streamed over UDP from the VLC Player application. The FPGA reconfiguration time is in the order of milliseconds, though since this happens very rarely, it is not a limitation.

The receiver attempts to receive the transmission in each channel successively. Once a transmission has been found, the receiver demodulates the signal and displays the video. The signal has been correctly identified if the fixed frame access code (inserted at the transmitter) is received correctly. If at any point, the transmission fails, the receiver begins to try and receive on the next channel, and so on. Switching to different channels is performed by the Decision Engine, which is informed by the reception chain when the signal is lost.

The transmission and reception nodes are each implemented on an ML507 FPGA development board, while the broadcast node is simulated using a PC, each of which is connected to a USRP 2 radio frontend [12]. Switching modes at the transmitter, between sensing and transmission, entails replacing the whole processing chain in the FPGA processing region. This is managed transparently, with the radio designer implementing the Decision Engine for each node as he would for software. The compile-time tools prepare the necessary FPGA configurations, and the Runtime System takes care of managing the reconfiguration. The system has been implemented and tested using real over-the-air transmission, and performs as expected.

4. CONCLUSION

We have developed a spectrum sensing technique, based on PSD estimation, that has been tailored for LTE femtocells in a GSM spectrum re-farming scenario. Simulation results show that the proposed technique can reliably detect LTE signals with SNRs as low as -9 dB with a very low probability of false alarm. We have shown the algorithm to perform as expected in a real implementation, using our run-time reconfigurable FPGA-based cognitive radio framework. The framework facilitates the easy design and prototyping of such a system on an FPGA, leveraging the reconfiguration capabilities to switch between sensing and normal operation, and thus allowing a smaller device to be used, saving cost and power.

REFERENCES

[1] V. Chandrasekhar and J. G. Andrews, "Femtocell networks: A survey," *IEEE Commun. Mag.*, vol. 46, pp. 59–67, Sep. 2008.

[2] "Spectrum Analysis for Future LTE Deployments," Motorola, Inc., white paper, 2007.

[3] Motorola, Inc. (2007) *The Drivers to LTE*. Solution Paper. [Online]. Available: http://business.motorola.com/experiencelte/pdf/TheDriverstoLTESolutionPaper.pdf

[4] 3GPP TS 45.005, v8.4.0, "3rd Generation Partnership Project; technical specification group GSM/EDGE radio access network; radio transmission and reception," Apr. 2009.

[5] T. J. O'Shea, T. C. Clancy, and H. J. Ebeid, "Practical signal detection and classification in GNU Radio," in *SDR Forum Technical Conference (SDR)*, Denver, Colorado, USA, Nov. 2007.

[6] M. Failli (chairman) and COST 207 Management Committee, *Digital Land Mobile Radio Communications : Final Report*. Luxembourg: Commission of the European Communities, 1989.

[7] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing (2nd Edition)*. Prentice Hall, 1999.

[8] J. Lotze, S. Fahmy, J. Noguera, L. Doyle, and R. Esser, "An FPGA-based cognitive radio framework," in *Irish Signals and Systems Conference (ISSC)*, Galway, Ireland, 2008, pp. 138–143.

[9] S. Fahmy, J. Lotze, J. Noguera, L. Doyle, and R. Esser, "Generic software framework for adaptive applications on FPGAs," in *IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Napa, CA, USA, Apr. 2009.

[10] J. Lotze, B. Özgül, S. A. Fahmy, J. Noguera, L. Doyle, and R. Esser, "Spectrum sensing to achieve frequency rendezvous using Xilinx FPGAs," in *IEEE Symposia on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, Chicago, IL, USA, Oct. 2008, demonstration paper.

[11] *ML505/ML506/ML507 Evaluation Platform – User Guide*, Xilinx Inc., Nov. 2008. [Online]. Available: http://www.xilinx.com/support/documentation/boards_and_kits/ug347.pdf

[12] *Universal Software Radio Peripheral – The Foundation for Complete Software Radio Systems*, Ettus Research LLC, Mountain View, California, USA, Nov. 2006. [Online]. Available: http://www.ettus.com/downloads/usrp_v4.pdf