A Framework for Cognitive Radio Wireless Sensor Networks

Luiz H. A. Correia*‡, Erasmo E. Oliveira*§, Daniel F. Macedo§,
Pedro M. Moura*, Antonio A.F. Loureiro§, Jorge Sá Silva‡
*Federal University of Lavras, Brazil
§Federal University of Minas Gerais, Brazil
‡Coimbra University, Portugal
Emails: lcorreia@dcc.ufla.br, erasmo@dgti.ufla.br, damacedo@dcc.ufmg.br,
pemmoura@comp.ufla.br, loureiro@dcc.ufmg.br, sasilva@dei.uc.pt

***Abstract*—The growth of mobile computing has increased the demand for wireless communication, causing a higher demand for the wireless medium as well as spectrum pollution. Smart radios, also called cognitive radios, monitor the network to identify the best available channel, in order to avoid interference. This paper proposes a framework for the development and testing of protocols for wireless sensor networks that employ cognitive radios (CRSN). We also developed two spectrum decision protocols for CRSN, which provide distributed mechanisms to select the best wireless channel based on the application's QoS requirements. Simulations of low, medium and high noise scenarios have shown that the protocols improve the delivery rate by up to 69%, while keeping the delay and energy consumption unaltered.**

I. Introduction

The development of mobile and ubiquitous devices increased the demand for communication and consequently the competition for frequency spectrum. Most of these devices use the ISM (*Industrial, Scientific and Medical*) frequency band, which does not require a license for operation. However, this band is used in applications such as cordless phones, remote control, microwave ovens and audio and video systems. The diversity of applications, along with the numerous communication standards employing the ISM frequency, has increased the level of interference and even caused the unavailability of spectrum in certain regions. In some locations the 2.4 GHz frequency band reaches an occupancy of up to 90% [1, 2].

Current wireless sensor networks (WSN) employ fixed ISM frequencies, which are also used by other wireless standards such as Wi-Fi, Bluetooth and IEEE 802.15.4. Akan et al [3] proposed a new paradigm for sensor networks, called Cognitive Radio Sensor Networks (CRSN). CRSN improve the communication by employing the dynamic spectrum access techniques found in cognitive radio networks. Typically, CRSN are more challenging than cognitive radio networks [4]. First, nodes have energy and processing constraints, requiring simpler algorithms. Second, due to the small size and cost, the nodes must use simple antennas and radios, which allow only the use of the ISM frequencies.

The literature proposes different cognitive frameworks for ad hoc networks and wireless sensor networks, however the approaches are either based on multi-radio or multi-channel. The C-MAC protocol proposes a multi-radio and multi-channel solution, but it employs a centralized decision in the allocation of discrete time slots [5].

This paper proposes distributed a framework for testing and developing MAC protocols for CRSN. This framework has been developed in the Castalia network simulator [6]. The framework is based on distributed algorithms and simulates nodes equipped with several multi-channel radios. Further, we developed two MAC protocols that sense the conditions of the channel and also take into account the application requirements when choosing the best channel. Simulations show that the use of CRSN technology increases the delivery rate by up to 69% compared with conventional WSNs, at the cost of a moderate increase in latency and energy consumption.

This paper is organized as follows. Section II presents the related works. The proposed framework is described in Section III. The operation of the CRSN framework is described in Section IV. Section V presents the simulation results. Finally, Section VI presents the conclusions and future works.

II. Related Works

In 2003 the FCC (Federal Communications Commission) analyzed the usage of the electromagnetic spectrum, showing opportunities for improved spectrum usage [7]. They proposed *cognitive radios*, which employ cognition techniques to identify unused frequencies, adapting the spectrum allocation in real-time in order to increase its overall usage.

Akyildiz et al. proposed a framework for spectrum management in cognitive radios [4]. This framework is based on a cross-layer model in which the MAC layer reconfigures the radio based on the application requirements as well as the state of the network. However, this framework does not address the problems of coexistence and interference.

Cognitive Radio Sensor Networks (CRSN) are wireless sensor networks that employ cognition to opportunistically use the idle ISM spectrum [3]. CRSN usually limit their frequencies to ISM because their transceivers must be very simple, in order to reduce energy and processing usage. A broadband radio, although highly desirable, would require a more expensive, energy-consuming hardware, demanding more processing and energy resources to scan and analyze more frequencies. In [3], the authors present a node architecture and a cognitive framework for CRSN. Despite this, no simulations or experiments were conducted to evaluate the reduction of interference or the coexistence among networks.

Zhou et al [8] demonstrate experimentally how electronic devices operating at 2.4 GHz can cause interference and even hinder the operation of WSNs based on the 802.15.4 standard. To avoid this, the authors proposed a multi-channel approach for today's WSNs, which have multi-frequency radios. They also proposed a middleware between the physical and MAC layers in order to support multiple channels. Nevertheless, this model was not validated.

Yuming et al proposed a spectrum decision algorithm for CRSN [9] that employs statistical information of the presence of primary users in the monitored region as input. The decision algorithm is based on AHP (Analytic Hierarchy Process) [10]. The method considers the probability of arrival of primary users, the application requirements as well as the characteristics of the wireless network (channel capacity, delay, packet loss and jitter). They also proposed a mechanism, based on the entropy concept, which automatically determines the weights of the input parameters in the decision method. Although the simulations present satisfactory results, the radio model and the simulation scenarios are quite simplistic. They also do not address issues such as antenna diversity or how the method would be implemented in MAC and PHY protocols.

Recently, [11] proposed an architecture for multi-radio sensor nodes to increase the reliability of communication. The IRIS platform and the IEEE 802.15.4 standard have been modified to operate with two radios with frequencies of 900 MHz and 2.4 GHz. Experiments demonstrated the feasibility of the proposed solution, showing improvements in the delivery rate and link stability with a moderate increase in energy consumption. Nevertheless, the approach was limited to a single channel implementation, where the radios were used independently, and as such spectrum sensing and spectrum decision were not implemented. The proposed framework, on the other hand, supports multiple multi-channel radios.

III. CRSN Framework

The proposed framework provides cognitive features to reduce the amount of interference and improve the coexistence in the ISM frequencies. It was implemented on the Castalia WSN simulator [6], since it implements a very realistic physical layer. The simulator models existing WSN radios (CC1000 and CC2420). It also provides RSSI (Received Signal Strength Indicator), SINR (Signal Interference Noise Rate) and LQI (Link Quality Indicator) readings. It also allows the variation of the modulation and the transmission power. The propagation model is based on empirical data, in which the attenuation varies in time. The simulator also implements classic WSN MAC protocols (S-MAC, TMAC and IEEE 802.15.4), and supports node mobility.

Castalia, however, lacks traffic models and test applications, there are no routing protocols implemented and the radios are single channel. Thus, we improved the simulator to support multi-radio and multi-channel operation, and modified the noise model based on empirical parameters (data from [12]). In addition, we added mechanisms to support cognitive radios, such a spectrum monitoring and decision, and extended the TMAC protocol for multi-channel and multi-radio operation.

Our CRSN framework is based on the models proposed by [3, 4], as shown in Figure 1. The gray rectangles represent the new or improved modules, which are described in the following subsections.

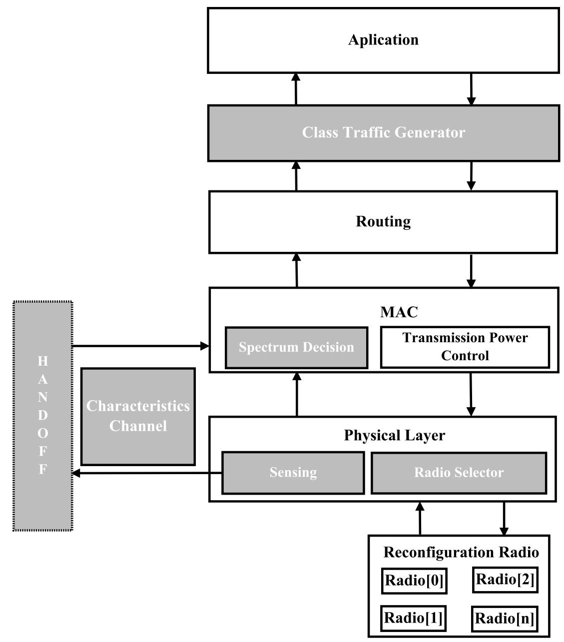

Fig. 1. Framework for CRSN (adapted from [6]).

Physical Layer: We extended the CC1000 and CC2420 radio models for multi-channel operation. For example, CC2420 supports 16 channels spaced by 5MHz. Our extensions also allow the detection of nearby networks, as well as to evaluate the occupation of the channels. We also added the support for multiple radios. As shown in Figure 1, nodes can employ radios having different communication standards. We also added a radio controller module to the physical layer in order to control the operation of each radio. This controller periodically enables each radio to perform spectrum sensing.

Medium Access Control: Spectrum decision algorithms include multi-layer functions such as environment sensing in the physical layer, medium access control for signaling, as well as taking into account the application's requirements. The most suitable layer to control those operations is the MAC layer.

Thus, we added a spectrum decision module to the MAC layer. The module employs distributed algorithms, and does not require a common control channel. To meet those requirements, we extended the TMAC [13] protocol for operation in CRSNs. This protocol supports event-driven applications that have a low data rate, and perform latency insensitive streaming or periodic data, which is the case of most WNSs. Moreover, it is contention-based, employs medium reservation and uses schedulers to identify when neighbor nodes will be ready to receive data. We also added the support for the spectrum decision protocols described in Section IV.

Traffic Model: We implemented a module that generates various types of network traffic. It supports three distributions for the inter-arrival rate of packets: uniform, constant and exponential. We employ a well-known code for generation of statistic distributions [14].

IV. Operation of the CRSN Framework

The proposed CRSN framework supports the classic spectrum sensing, sharing, decision and mobility functionalities [3]. In order to support multi-channel and multi-radio operation, it was necessary to add a channel synchronization mechanism so that sender and receiver would employ the same channel. The execution flow of this mechanism is shown in Figure 2. This procedure is performed at the initialization of the nodes, and is re-executed periodically. First, a node must find its direct neighbors and the two hop neighbors, which are stored in neighbor lists. Each node then performs spectrum sensing and chooses the best channel based on one of the two spectrum decision algorithms, CogTMAC or AHPTMAC. After choosing the best channel, the neighbors must be notified of the selected channel. Finally, the framework provides a channel synchronization mechanism, which is executed before each packet transmission.

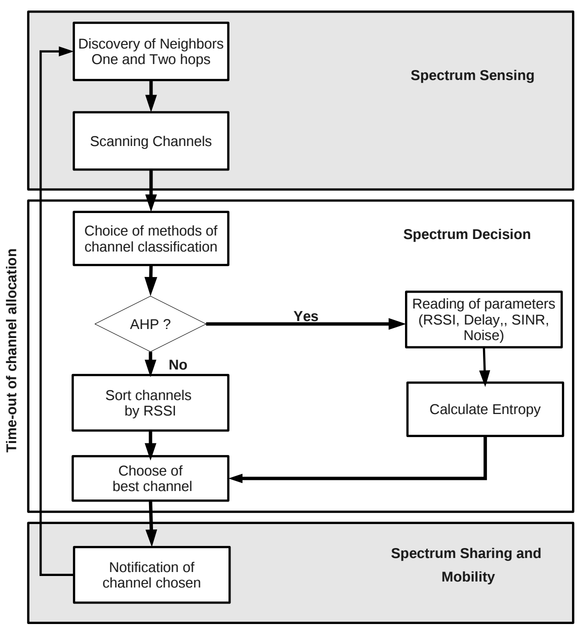

Fig. 2. Operation of the CRSN Framework.

A. Spectrum Sensing

Nodes periodically discover the immediate neighbors and the two hop neighbors. This protocol is based on the classic *echo/probe* protocol [15]. Each node broadcasts a FN (*Find Neighbors*) packet to its one hop neighbors, which reply with a NF (*Find Neighbor*) packet. When the node receives the reply NF, it adds the ID of the sending node to its list of neighbors (LVN_i).

The interference model used in this framework considers that two interfering nodes cannot transmit at the same time. In this framework two nodes will interfere with each other if they are up two hops distant from one another [16]. This model is appropriate since the radios can only communicate with an immediate neighbor at a time. One hop neighbors do not interfere with one another, since we assume that their transmissions are handled by the channel reservation mechanisms (RTS/CTS) of TMAC.

The identification of two hop neighbors occurs periodically. Each node (N_i) build a list of its two hop neighbors ($LV2HN_i$) based on the list of one hop neighbors (LVN_i) of its neighbors. This is shown in Algorithm 1.

Procedure List2HopNeighbors(N_i, LVN_i);
/* LVN_i: One hop neighbors of i */
/* N_i: Whose neighbors are N_j */
$LV2HN_i \leftarrow \emptyset$;
for $N_j \in LVN_i$ **do**
  $LV2HN_i \leftarrow LV2HN_i \cup (LVN_j - LVN_i) - N_i$;
return $LV2HN_i$

Algorithm 1: List of two hop neighbors.

After one hop and two hop neighbor are discovered, each node performs spectrum sensing. All channels of the radios are inspected in order to collect the input parameters for the spectrum decision algorithm.

B. Spectrum Decision Methods

The decision methods were implemented and adapted in the TMAC protocol. The two decision methods available in the framework are described below.

1) CogTMAC - Channel Selection Based on RSSI: The CogTMAC method chooses the best available channel according to the local noise observed by the node. The local noise is estimated using Received Signal Strength Indicator (RSSI) readings. CogTMAC utilizes the Clear Channel Assessment (CCA) function of CC2420 to assess if the channel is free for transmission (line 3 of Algorithm 2). The selected channel is the free channel that has the lowest RSSI value, and therefore offers the least amount of interference (lines 3-5). Once the best channel is chosen, it is marked as busy (line 6) and a notification is sent to all neighbors (lines 7 and 8). The notification method is detailed in section IV-C.

2) AHPTMAC - Multi-Criteria Channel Selection: A decision model based on only one channel parameter may not be sufficient to choose the best available channel or frequency. AHP is a multi-criteria selection algorithm that has been successfully used in previous works for spectrum decision [9]. In this paper, we adapted this method to the TMAC protocol, classifying the channels from parameters based on PHY readings as well as the requirements of the applications. To

```
Procedure channelChoiceRSSI(ListChannel, LVN_i);
for C_i ∈ ListChannel do
    if C_i is free then
        if C_i.RSSI <= bestChannel.RSSI then
            bestChannel ← C_i;
    Updates the list of assigned channels;
    /* Notify one hop neighbors */
    NotifyLV1Hop();
    /* Notify two hop neighbors */
    NotifyLV2Hop();
```

Algorithm 2: Channel selection based on RSSI.

do so, a weight must be defined for each parameter in order to identify the best channel. Some methods manually specify these weights according to user-defined criteria. However, most users have no prior knowledge of how much each parameter influences the choice of the best channel.

AHPTMAC automatically calculates those weights using the concept of information entropy, as in [9]. The calculation of the weights uses a Parameters Matrix $M_{C \times P}$, in which each line C represents a channel, and the rows P represent the channel parameters (Noise, RSSI, SINR, Delay).

The first step, described in Algorithm 3, is to calculate an entropy constant $C_{entropy}$ (line 2), given as a function of the “disorder”, or variance, of the collected readings for each parameter. Then, the entropy is calculated for each parameter of the matrix $M_{C \times P}$. Next, the entropy is calculated for each parameter P_k (lines 4-7).

```
1  Procedure AHPWeights(Channel, M_{C×P});
2  C_entropy ← (−1 ∗ (1/log(number_lines C))) ;
3  for i ← 1 to P do
4      for j ← 1 to C do
           /* Sum of parameter values of i */
5          F_ji ← B_ji/Sum_Column_i;
6          H_i ← F_ji ∗ log(F_ji);
7          H_i ← C_entropy ∗ sum(H_i);
       /* Calculation of weights W */
8      for i ← 1 to k do
9          W_i ← (1 − H_i)/(k − sum(H));
10     Choose max(W_i)
11 return (Entropy, Weight, Highest_Weight)
```

Algorithm 3: AHP method with dynamic weights.

Finally, AHPTMAC calculates the vector of weights w_i as a function of the entropy vector H_i and the number of analyzed parameters k. Then, the weight of each parameter is calculated (lines 8–9). The algorithm concludes by choosing the maximum weight (line 10). The algorithm returns the entropy, weight and the highest weight parameter (line 11) to the decision method to choose the best available channel. These weights are used as input to AHP, shown in Algorithm 4. They are normalized (lines 2-3), and the score is calculated for each channel (lines 4-5). The channel having the highest score is selected (line 6).

```
1  Function AHPTMAC(TableValue, ListofChannels);
2  weights ← AHPWeights(TableValue);
3  normalizeMetrics(ListofChannels);
4  for Channel_i ∈ ListofChannels do
5      Channel.QualityIndex ←
       weights.RSSI ∗ Channel_i.RSSI +
       weights.SNIR ∗ Channel_i.SNIR +
       weights.NoiseFloor ∗ Channel_i.NoiseFloor +
       weights.delay ∗ Channel.delay
6  bestChannel ← arg max_i(Channel_i.QualityIndex)
   return (bestChannel)
```

Algorithm 4: Choosing the best channel using AHP.

C. Notification of the Chosen Channel

A node, when choosing the best channel, notifies the one and two hop neighbors of its selection. The notification message is sent in broadcast, but to avoid repeated notifications, each node identifies the intersections between the lists of neighbors. This procedure is described in Algorithm 5.

```
1  Procedure Resend Broadcast(LVN_i, LV2HN_i);
2  toVisit ← LV2HN_i;
3  N ← sort(LVN_i, ↓);
4  for N_j ∈ N do
5      if toVisit ∩ LVN_j ≠ ∅ then
6          toVisit ← toVisit − LVN_j;
7          send(N_j);
```

Algorithm 5: Procedure for resending broadcast.

The algorithm is based on flooding. The sender forwards the message to its neighbors having the highest degree of connectivity, in order to reduce the amount of packet forwards. Nodes with decreasing amount of neighbors are added to this set at each time (lines 4-7). The process halts when all two hop nodes have been visited.

D. Spectrum Mobility

The dynamic channel selection can interrupt the communication between nodes when both nodes are listening/transmitting at different channels. We solved this problem by inserting a synchronization mechanism in TMAC. Thus, before sending a RTS frame, the sender switches to the channel of the receiver ($CH_j \rightarrow CH_k$). Then, the node sends the RTS indicating the channel that will be used for data transmission (CH_j). The receiver, upon receiving the RTS, changes the channel ($CH_k \rightarrow CH_j$) and sends a CTS frame. When receiving the CTS, the sender is already in its original channel (CH_j), and thus transmits the DATA frame. After acknowledging the transmission, the receiver returns to its source channel (CH_k).

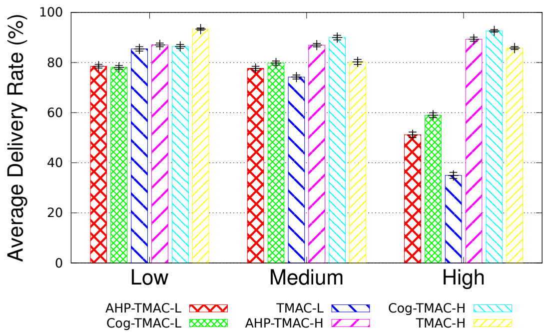

Fig. 3. Average Delivery Rate.

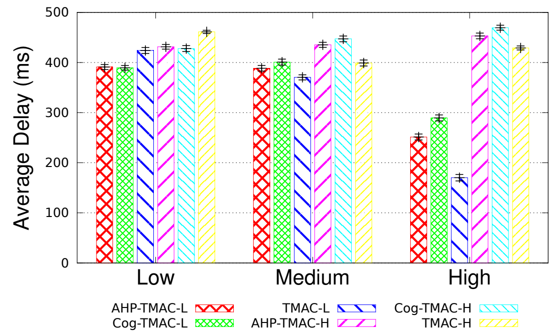

Fig. 4. Average Delay.

V. Evaluation

The framework was implemented in the Castalia Simulator version 3.0. The simulations were configured as follows. We simulated 18 nodes, placed on a grid (6x3), sending data periodically in the interval of 2 seconds (due to space constraints, we only show the results for uniform traffic). In each simulations we randomly chose four senders, so that the sender N_i randomly chooses one of its one hop neighbors to transmit data. Thus, there will be four simultaneous streams on the network, simulating the interference from multiple networks. We simulated 500s of network operation, so that 600 transmissions were generated.

Despite our implementation being multi-radio and multi-channel, we simulated only one multi-channel radio, since simulating n radios with m channels is similar to simulating one radio with $n \times m$ channels. The channel sensing frequency was set to 30s. The bandwidth used was 250 Kbps (same as CC2420 radio) and coverage are of 50m.

The objective of this evaluation was verify if cognitive protocols can reduce the interferences and improve the delivery rate in the network when compared with the original TMAC protocol. In all, 33 simulations were performed for each scenario. The evaluated metrics were average delivery rate, end to end delay, energy consumption and handoff (channel switching). We varied the channel noise into two different levels: Low ($-100dBm$), Medium ($-95dBm$) and High ($-92dBm$). We also simulated two different node densities by varying the size of the simulated area: Low (75x30m) and High (50x20m). The graphs represent the mean values, plotted with a confidence interval of 99%. To improve the readability, we mark on each curve the name of the protocol as well as the density of nodes ($-L$ for low, $-H$ for high).

A. Results

Figure 3 shows the average delivery rate. In the low density scenarios we observe that TMAC has a higher delivery rate than AHPTMAC and CogTMAC when the local noise is low, about 8% and 9% higher, respectively. These values are more pronounced when the local noise is high. CogTMAC presented an increase of 68.7% in relation to TMAC and 15.3% when compared to AHPTMAC.

The behavior of the delivery rate is similar for high and low density scenarios. When the local noise is low, the protocols have a cognitive delivery rate lower than that of TMAC. The poor performance of the cognitive protocols when compared to TMAC in low noise scenarios occurs due to periodic channel changes. During the channel switching, the cognitive protocols are subject to packet losses. In the high noise scenarios, though, it is observed that the cognitive protocols present better results, since the choice of the best channel improves the robustness of the communication.

Figure 4 presents the end-to-end delay. TMAC presented the highest delay for low noise, about 8.5% higher than the cognitive protocols. This is justified by the delivery rate of TMAC. In the scenario of low density with high noise, CogTMAC presented the worst performance, with a delay 15% and 70% higher than that of AHPTMAC and TMAC, respectively. It is important to note that, for this scenario, the delivery rate is reduced and the number of channel exchanges made by CogTMAC is superior to AHPTMAC's. However, for low noise and high density, the difference in the delay is reduced to approximately 3% and 9% when compared to AHPTMAC and TMAC.

The transmission's energy consumption is shown in Figure 5. The protocols had a very similar energy consumption. The greatest variation occurred for the high density and high noise scenario, in which TMAC consumed about 1% and 1.4% less energy than AHPTMAC CogTMAC, respectively. The highest consumption of the cognitive protocols is due to the overhead of channel selection and mobility. However, this difference is not statistically significant, considering a maximum error margin of ± 0.06 *mJoule* for a confidence interval of 99%.

Figure 6 shows the average number of handoffs for the cognitive protocols (TMAC is not presented since it uses a fixed channel). CogTMAC presented the highest number of handoffs. For the low density scenarios, the highest number of handoffs occurred for the high noise scenario, where CogTMAC presented 80% more handoffs than APHTMAC. Despite having a higher number of handoffs, the amount of handoffs for CogTMAC does not increase significantly when the noise level increases. The difference among the protocols is less pronounced in the high density scenarios. The highest number of handoffs occurred for the low noise scenario, where CogTMAC performed 14% more handoffs than APHTMAC. For high densities, however, we observed a reduction on the number of handoffs when the noise level increased.

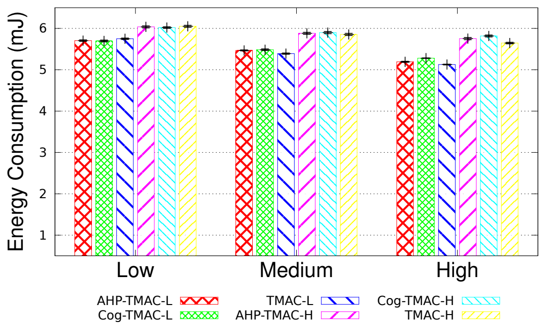

Fig. 5. Energy Consumption.

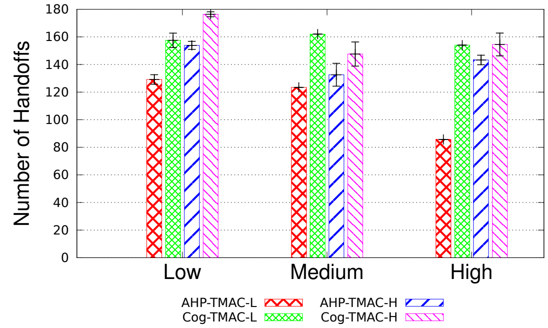

Fig. 6. Number of Handoffs.

VI. Conclusions and Future Work

This paper proposed a framework for the simulation of cognitive radio sensor networks (CRSN). We have implemented the main features of CRSN (spectrum sensing, decision, sharing and mobility), and two spectrum decision algorithms over TMAC: CogTMAC, which employs RSSI readings, and AHPTMAC, which takes decisions based on the RSSI, transmission latency, noise levels and SINR.

Results showed that the cognitive protocols obtain the highest delivery rate improvements in scenarios where the noise is more intense. CogTMAC obtained the highest delivery rate in low density scenarios, however at the cost of a moderate increase in the latency and the delivery rates. AHPTMAC consumed slightly less energy and presented a better delay than CogTMAC for low noise scenarios, since it tends to perform less channel handovers.

As future work we will implement other cognitive MAC protocols for CRSNs, investigate new techniques for spectrum decision such as evolutionary algorithms, and evaluate the protocols in multi-hop scenarios.

Acknowledgements

This work is supported by CAPES in the post-doctoral BEX 6668/10-0. The authors acknowledge the financial support of agencies CAPES, CNPq and FAPEMIG.

References

[1] Mark A. McHenry, Peter A. Tenhula, Dan McCloskey, Dennis A. Roberson, and Cynthia S. Hood. Chicago spectrum occupancy measurements & analysis and a long-term studies proposal. In *Workshop on Technology and Policy for Accessing Spectrum (TAPAS)*, 2006.

[2] Jeffrey H. Reed and Charles W. Bostian. Understanding the issues in software defined cognitive radio. Tutorial at IEEE DySPAN'05, November 2005.

[3] Ozgur B. Akan, Osman B. Karli, and Ozgur Ergul. Cognitive radio sensor networks. *Network Magazine of Global Internetworking.*, 23:34–40, July 2009.

[4] Ian F. Akyildiz, Won-Yeol Lee, Mehmet C. Vuran, and Shantidev Mohanty. A survey on spectrum management in cognitive radio networks. *IEEE Communications Magazine*, 46(4):40–48, April 2008.

[5] Carlos Cordeiro and Kiran Challapali. C-MAC: A Cognitive MAC Protocol for Multi-Channel Wireless Networks. *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 147–157, 2007.

[6] Athanassios Boulis. Castalia: A simulator for Wireless Sensor Networks and Body Area Networks. User's Manual., 2011. http://castalia.npc.nicta.com.au/pdfs/.

[7] FCC. ET Docket No 03-322 notice of proposed rule making and order, 2003. http://www.cs.ucdavis.edu/~liu/289I/Material/FCC-03-322A1.pdf.

[8] Gang Zhou, John A. Stankovic, and Sang H. Son. Crowded spectrum in wireless sensor networks. In *Workshop on Embedded Networked Sensors*, 2006.

[9] Yuming Ge, Yi Sun, Shan Lu, and E. Dutkiewicz. Adsd: An automatic distributed spectrum decision method in cognitive radio networks. In *International Conference on Future Information Networks (ICFIN)*, 2009.

[10] T.L. Saaty. *Fundamentals of the Analytic Hierarchy Process*. RWS Publications, 2000.

[11] B. Kusy, C. Richter, Wen Hu, M. Afanasyev, R. Jurdak, M. Brunig, D. Abbott, Cong Huynh, and D. Ostry. Radio diversity for reliable communication in WSNs. In *International Conference on Information Processing in Sensor Networks(IPSN)*, pages 270–281, 2011.

[12] Luiz H. A. Correia, Daniel F. Macedo, Aldri L. dos Santos, Antonio A. F. Loureiro, and José Marcos S. Nogueira. Transmission power control techniques for wireless sensor networks. *Computer Networks*, 51:4765–4779, December 2007.

[13] Tijs van Dam and Koen Langendoen. An Adaptive Energy-efficient MAC protocol for wireless sensor networks. In *ACM Int. Conf. on Embedded Networked Sensor Systems (SenSys)*, pages 171–180, 2003.

[14] Richard Saucier. Computer generation of statistical distributions. http://ftp.arl.mil/random/random.pdf, 2000.

[15] Gregory R. Andrews. Paradigms for process interaction in distributed programs. *ACM Computing Surveys*, 23:49–90, March 1991.

[16] Jitendra Padhye, Sharad Agarwal, Venkata N. Padmanabhan, Lili Qiu, Ananth Rao, and Brian Zill. Estimation of link interference in static multi-hop wireless networks. In *Internet Measurement Conference (IMC)*, 2005.