

Research Article

A Self-Learning MAC Protocol for Energy Harvesting and Spectrum Access in Cognitive Radio Sensor Networks

Mohammed Hawa,¹ Khalid A. Darabkh,² Raed Al-Zubi,¹ and Ghazi Al-Sukkar¹

¹Electrical Engineering Department, The University of Jordan, Amman 11942, Jordan

²Computer Engineering Department, The University of Jordan, Amman 11942, Jordan

Correspondence should be addressed to Mohammed Hawa; hawa@ju.edu.jo

Received 2 September 2015; Accepted 31 January 2016

Academic Editor: Hana Vaisocherova

Copyright © 2016 Mohammed Hawa et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The fusion of Wireless Sensor Networks (WSNs) and Cognitive Radio Networks (CRNs) into Cognitive Radio Sensor Networks (CRSNs) is quite an attractive proposal, because it allows a distributed set of low-powered sensor nodes to opportunistically access spectrum bands that are underutilized by their licensed owners (called primary users (PUs)). In addition, when the PUs are actively transmitting in their own bands, sensor nodes can switch to energy harvesting mode to obtain their energy needs (for free), to achieve almost perpetual life. In this work, we present a novel and fully distributed MAC protocol, called *S-LEARN*, that allows sensor nodes in a CRSN to entwine their RF energy harvesting and data transmission activities, while intelligently addressing the issue of disproportionate difference between the high power necessary for the node to transmit data packets and the small amount of power it can harvest wirelessly from the environment. The presented MAC protocol can improve both the network throughput and total harvested energy, while being robust to changes in the network configuration. Moreover, *S-LEARN* can keep the cost of the system low, and it avoids the pitfalls from which centralized systems suffer.

1. Introduction

Wireless Sensor Networks (WSNs) are made up of a big number of distributed sensor nodes, fitted with various sensors and typically spread over a wide geographical area [1]. Such WSNs have found a wide range of applications nowadays, including periodic monitoring, security, surveillance, and health monitoring and control [2]. Energy consumption is a major issue in the design of WSN protocols, since each WSN node is equipped with limited power supply, due to cost considerations. Such limited power supply dramatically reduces the life time of a WSN system, especially when WSN nodes keep operating in the active mode.

To prolong the life of WSN nodes, and to avoid the need for continuous replacement or recharging of the batteries in such nodes, many researchers focused on developing more energy efficient MAC and routing protocols for WSNs. Such protocols utilized different innovative techniques and structures that minimize the energy consumption per transmitted packet [3–10].

However, no matter how energy efficient a WSN node is in transmitting data, a finite power supply is bound to dry up eventually, especially when high throughput data transmission is required, which makes the energy problem in WSN systems one of significant importance. This is why research in wireless energy harvesting techniques has surged recently [11–14].

In wireless energy harvesting, the WSN nodes are equipped with hardware to convert the ambient radio frequency (RF) signals from nearby sources into electricity to recharge the node's battery. WSNs represent an ideal application for energy harvesting since it means the WSN nodes can rely on a very small rechargeable battery, so long as they take the appropriate steps to harvest energy to maintain a sustainable power supply. Since stray RF power is widely available nowadays with the almost ubiquitous use of TV and radio broadcasting towers, cellular base stations, and Wi-Fi access points around the globe, finding a source from which to harvest power is easier than ever [11–14]. Dedicated RF sources can also be used in special cases when needed [15].

Many WSN nodes rely on license-free spectrum bands to transmit their data. This can cause heavy interference due to other networks operating in the same frequency spectrum. Hence, a promising advancement to WSNs is the proposition of combining WSNs with Cognitive Radio Networks (CRNs). This avoids the aforementioned spectrum access limitation and also provides the extra benefit of energy harvesting to WSN nodes [16–18].

CRNs were envisaged to solve the problem of spectrum inefficiency, which results because large portions of the licensed spectrum are actually underutilized at specific times and specific geographical areas [19]. In cognitive radio, secondary users (SUs) are allowed to opportunistically access the spectrum bands underutilized by the spectrum owners [20, 21], but on the condition that this is done in such a way to avoid harmful interference to the primary users (PUs), which are the original license holders. The combination of WSNs and CRNs into Cognitive Radio Sensor Networks (CRSNs) is quite an attractive proposal, because when the spectrum bands of a PU are not being utilized, CRSN nodes can transmit their data over such high-bandwidth bands. Conversely, when the PU starts utilizing such spectrum, CRSN nodes can switch to energy harvesting mode to obtain their energy needs (for free) from the PU itself [17, 18].

In this paper, we introduce a novel MAC protocol for sensor nodes in a CRSN allowing such nodes to schedule RF energy harvesting from the PUs in the system, while at the same time intelligently interweaving their data transmissions inside the spectrum holes of such PUs.

As will be explained shortly, the amount of energy a CRSN node is able to harvest from a PU is quite small compared to the energy required by the same CRSN node to transmit its data packets. This is the status quo in state-of-the-art hardware and is expected to last for quite some time unless an unexpected breakthrough occurs in hardware. This means that a single CRSN node will have to spend much more time harvesting energy than transmitting data. Our proposed MAC protocol turns this limitation into an opportunity to improve the overall throughput of the CRSN network, to adapt to variations in the network, and to minimize the packet transmission delay at each sensor node. This is done by utilizing a progressive self-learning process at each CRSN node to schedule energy harvesting and data transmission. To the best of our knowledge, no other researchers considered the discrepancy between the amount of energy harvested by sensor nodes and that required to transmit data, nor did they try to utilize this problem to the benefit of the system. Our MAC protocol is named *S-LEARN* (Self-Learning Energy harvesting and spectrum Access in cognitive Radio sensor Networks) and is based on a fully distributed architecture, which helps in reducing the cost of administration and maintenance. It also avoids the problem of single point-of-failure from which centralized systems suffer.

This rest of this work is organized as follows: In Section 2 we provide the necessary background for CRSN networks and energy harvesting. In Section 3 we describe our proposed MAC protocol for sharing the spectrum while harvesting energy in CRSN systems. Simulation details and the performance evaluation results illustrating the benefits of our

proposed algorithm are explained in Sections 4 and 5, respectively. More discussion of the tradeoffs achieved by varying the control parameters of our suggested method is given in Section 6. Finally, Section 7 concludes our discussion.

2. Background and Related Work

There has been tremendous research regarding the hardware circuit design for energy harvesting devices. Most of the implementations are based on the CMOS technology, though other technologies have also been used on occasion. In a harvesting device, generally the efficiency of energy conversion from the AC input radio signal to the output DC voltage is modest unless the received RF power level is high enough (above -15 dBm). For example, the work in [22] can achieve about 70% efficiency at 0 dBm received RF power on the 2.45 GHz spectrum, and the design in [23] can achieve 75% efficiency at -10 dBm received power at 900 MHz. However, the efficiency drops to 40% when the power drops to -17 dBm for the state-of-the-art work in [24] at 868 MHz. This means that the amount of harvested energy is typically small, even with the best possible hardware, especially due to the large propagation loss in wireless environments. For example, the authors in [24] managed to harvest $2 \mu\text{W}$ at a distance of 27 meters away from a 1.78 W transmitter running at 868 MHz. Another study managed to obtain a harvested energy of $5.5 \mu\text{W}$ at 15 meters from a 4 W power supply running at 902–928 MHz [25]. In addition, the authors in [26] managed to harvest $60 \mu\text{W}$ at 4.1 km from a high-powered 960 kW TV tower broadcasting in the 674–680 MHz frequency range.

It is fair to expect that more advanced technologies for wireless energy harvesting will be available in the near future due to better multiband electronics and better design of highly efficient beam-forming antenna arrays [27–29]. However, comparing the levels of power that current state-of-the-art harvesters can achieve (which is in microwatts), with the power required for transmitting data packets (which is in milliwatts), we see a big discrepancy. For example, a typical WSN node transmitter running at 2.4 GHz might send at a power level of -4.4 dBm, at which time its electronics will consume an average power of 4.5 mW [30]. If the transmit power is increased to 0 dBm, the transceiver of a sensor node consumes approximately 87 mW [30]. Another state-of-the-art ultra low power 915 MHz transmitter sending at -3 dBm requires a total power of 1.78 mW [31].

This effectively means that sensor nodes relying on RF energy harvesting to achieve perpetual lifetime are limited in their data transmission capabilities because they need to spend a great amount of time harvesting enough energy to be able to transmit a single packet. This is a big problem for WSNs that rely on energy harvesting, but one that we turn into an opportunity as we design our MAC protocol seamlessly around this issue, without sacrificing robustness and simplicity and without abandoning the distributed nature of WSN networks.

It is noteworthy that many of the scheduling techniques and MAC protocols proposed for WSNs that employ energy harvesting have utilized a centralized approach. For example, the authors in [32] proposed a scheduling protocol that

requires sensor nodes to harvest energy from downlink broadcast signals before they use this harvested energy to transmit their independent uplink data packets based on a Time Division Multiple Access (TDMA) framework coordinated by the access point. The access point jointly optimizes the time allocations for the downlink energy harvesting and uplink data transfer, which results in maximizing the overall system throughput.

A similar TDMA-based system was studied in [33] but with the base station equipped with two antennas. The optimization problem maximizes the total throughput of the system subject to time constraints or minimizes the total harvesting time and data transmission time of the system subject to data rate constraints.

The MAC protocol featured in [34] uses a contention-based approach based on CSMA/CA for a star-topology sensor network. In this topology, a single controller node collects data from and emits wireless energy to other sensor nodes. The presented protocol uses an energy adaptive technique and out-of-band RF power harvesting to manage the sensor node's duty cycle based on the node remaining energy level and to control the node's backoff time based on the node energy harvesting rate.

The authors in [35] considered a Multiple-Input and Multiple-Output (MIMO) system with imperfect channel state information and proposed a protocol for frame-based transmission that splits each frame into three phases. In the first phase, the centralized base station estimates the downlink channels by utilizing pilots. The base station then broadcasts radio energy to all nodes under its control for them to perform harvesting in the second phase. In the third and final phase, nodes transmit their data packets to the base station. The optimization problem is executed at the base station to optimize the time and energy allocation with the aim to maximize the minimum rate among all nodes.

The work in [36] used a multiantenna base station in an OFDMA broadband setting to not only communicate but also transfer RF energy to the wireless nodes. The base station optimization algorithm maximizes the system throughput under the constraint of power consumption by the base station and the wireless nodes. The authors in [37] allowed the base station to optimize energy efficiency (i.e., bit/Joule) in the downlink of a multireceiver OFDMA system. The optimization problem is based on the transmit power allocation from the base station and receiver operation to harvest energy.

Though these are valid optimization schemes, they rely on a centralized and powerful controller. This is not a perfect fit for WSNs since it suffers from intractable computational complexity when the system size grows. In addition, it exhibits single point-of-failure issues.

Decentralized approaches that are based on game theory have also been suggested to reduce the complexity issue when working out the optimal solution. For example, the authors in [38] proposed a bidding strategy to achieve a Nash equilibrium between the nodes when competing to harvest RF energy. The work in [39] extended the bidding to both data transmission and energy resources. In [40] a repeated coalition formation game was considered, where RF-powered

wireless nodes cooperate in packet transmission to improve the long-term payoff for the system. Such techniques suffer from overhead and the need for out-of-band mechanisms to perform bidding. Additionally, the power-limited and energy-constrained WSN nodes are not typically capable of running such sophisticated algorithms.

Other game-theoretic approaches used for optimizing energy harvesting in sensor networks include queuing theoretic transmission policies [41] and modified back-pressure-based algorithms with energy queues [42].

In [15], the authors proposed a distributed MAC protocol based on CSMA/CA to optimize in-band RF energy harvesting from multiple wireless energy sources in a CRSN. The goal was to minimize the impact of interference between the multiple wireless sources and to maximize energy transfer, but the authors did not consider the issue of either minimizing the interference between the secondary nodes and the primary users or minimizing the collisions between the secondary nodes themselves when they find an opportunity to transmit data.

A partially observable Markov decision process is used at the secondary nodes in [43, 44] to select between opportunistic spectrum access and energy harvesting to maximize the system throughput in CRSNs. This system incurs extremely high computational complexity when it comes to low-powered secondary sensor nodes, especially when the state space (related to energy queue size and data queue size) is large.

In light of our prior discussion, our proposed MAC protocol is the first distributed protocol for CRSNs that intelligently addresses the disproportionate difference between data transmission power and harvested power and uses that to improve both the network throughput and total harvested energy.

3. S-LEARN MAC Protocol

3.1. Network Architecture. We consider in this work a CRSN in which sensor nodes are scattered around a particular geographical area. Such nodes collect data from their sensors and report to dedicated data sinks as shown in Figure 1. The nodes opportunistically utilize the spectrum of PUs in the area when such PUs are inactive. When the PUs are active, sensors do not send data in the spectrum band to avoid harmful interference with the licensed owners. Rather, they use such opportunity to harvest RF energy from the radiated power of the PUs.

The S-LEARN MAC protocol that we propose in this work assumes that sensor nodes rely on harvested wireless energy to operate in such CRSN. To bring the cost of the hardware further down, we limit the sensor node to accessing only one frequency band at any point in time, though it can switch frequency to other available spectrum bands in the future, if it so chooses. Hence, sensor nodes cannot carry out energy harvesting and opportunistic data transmission at the same time, which reduces cost because it allows the sensor node to reuse the same antenna for both the transmitter and harvester circuitry, as shown in Figure 2.

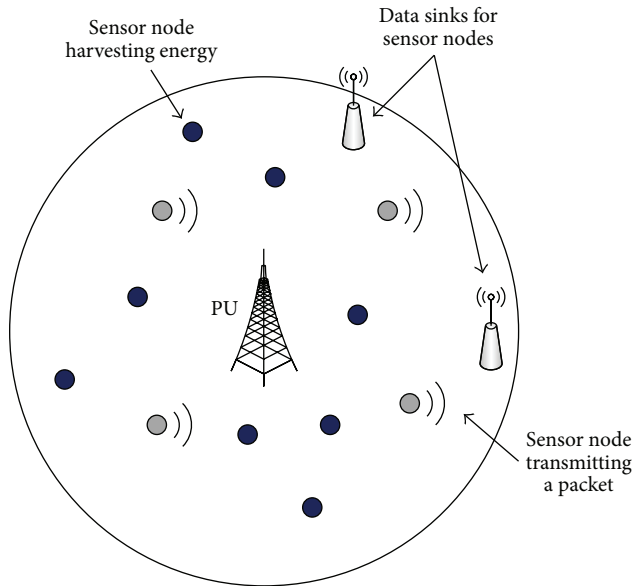


FIGURE 1: Cognitive sensor nodes utilize the spectrum of PUs to either send data to sink nodes or harvest energy from PUs.

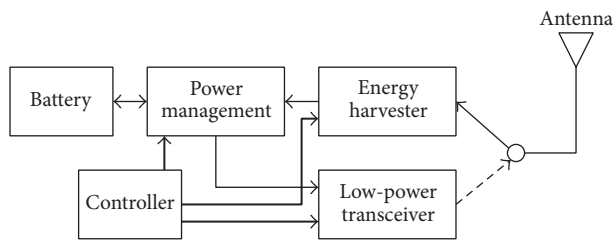


FIGURE 2: Hardware for the sensor node. Notice that both the transmitter and energy harvester use the same antenna.

We assume the time axis to be slotted into epochs, each of duration τ seconds. Each sensor node needs to make a decision every time epoch on which spectrum band it wants to access, and after a small period of sensing that band it can decide if a PU is active or not, so it can tell whether it should utilize that spectrum band for transmitting a packet or for harvesting energy.

The nodes that employ the S-LEARN protocol operate autonomously and independently in a completely distributed fashion, without the need to communicate their intentions with each other or with a centralized controller. Particularly, they adaptively learn about the behavior of the PUs and other sensor nodes in the network by utilizing their own spectrum sensing information. This keeps the system cost extremely low and makes for a very robust network.

Due to the limitations of hardware technology, we also assume that the *energy consumed* by the sensor node as it transmits a packet in one time epoch, which we denote by E_T , is much higher than the energy that the sensor node can wirelessly harvest during a different time epoch of the same period. This latter one is denoted by E_h . In other words $E_T \gg E_h$. Hence, the sensor node should spend more of its time attempting to harvest energy rather than transmitting so it can maintain a constant and healthy battery level.

As will be illustrated momentarily, S-LEARN requires sensor nodes to take turns in sending information in empty spectrum opportunities, while other sensor nodes busily work on harvesting energy from an available PU source. This will minimize possible collisions between the competing sensor nodes and maximize the total system throughput.

3.2. Algorithm Description. Collisions represent a major challenge for distributed systems, and in power-limited scenarios (such as sensor networks) it is doubly problematic, because not only does it limit the throughput of the available spectrum, but it also wastes valuable energy (equal to E_T) without resulting in successful packet transmission. This causes the node to spend excessive time in harvesting more energy to reattempt sending the same data, which unduly increases the delay the packets experience.

A trivial solution to this problem comes to mind, which is to use a TDMA protocol run by a centralized controller. In this case, the central controller will poll each sensor node in a round-robin fashion to allow it to transmit data. However, as explained previously, solving this issue using a centralized controller is not desirable in the context of WSNs. This is because the central coordinator will add extra unjustified cost to the system, as it requires high computational power, large memory, and extra transceiver power budget to allow it to communicate with every single sensor in the network. In addition, the central controller represents a single point-of-failure for the system and results in scalability issues when the number of sensor nodes increases. It also has to employ a special protocol to register and access newcomer sensor nodes and to detect and release defective energy-starved nodes from its polling schedule.

Central control becomes even more difficult to operate in cognitive radio environments because spectrum is not guaranteed to be available, which means a dedicated control channel might not persist over time. Finally, the central controller can easily become a security vulnerability for Denial-of-Service (DoS) and jamming attacks. Distributed systems do not suffer from all these problems, as they provide a lower cost solution that is scalable and does not have a single point-of-failure.

Another simple approach to coordinate among sensor nodes is for each sensor node to broadcast a schedule of its *intended transmissions* to all other sensor nodes in a beacon at the beginning of each time epoch. However, this approach is also undesirable because it incurs extreme overheads in using the channel capacity to send such control information instead of actual data, thereby reducing the channel throughput. More important, in cognitive radio scenarios (where there are more than one spectrum band available) it means each sensor node has to have multiple receivers tuned to the different available spectrum channels to collect the control information from other nodes in the system, which is extremely expensive in terms of hardware costs and power requirements for the nodes. The approach also suffers from the famous *hidden station problem*, where sensor nodes might not hear each other's beacons but still affect each other's transmission at the receiver side.

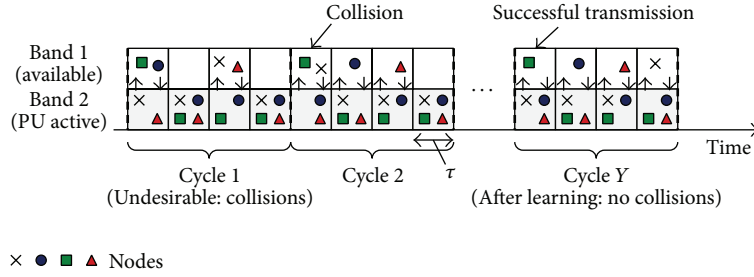


FIGURE 3: Four cognitive sensor nodes are executing the S-LEARN MAC protocol to coordinate access to two spectrum bands in a CRSN.

Therefore, autonomous and independent distributed operation of sensor nodes is preferable so long as there is an intelligent mechanism to reduce possible collisions. Our S-LEARN algorithm is such a technique, in which sensor nodes use their observations of the behavior of other nodes in the network to progressively learn about the environment and hence coordinate their transmissions.

In our MAC protocol, the long time it takes for the sensor nodes to harvest energy is utilized to our advantage to create proper schedules at the different nodes to specify in which bands and which time slots they are going to transmit their data to avoid colliding with other nodes. Meanwhile, other nodes use that time (when a specific node is transmitting) to harvest RF energy from the PUs in the system. The idea is illustrated by means of a simple example in Figure 3. In this scenario, four sensor nodes are trying to coordinate access to two spectrum bands in a CRSN. For simplicity, we assume that the first band is always empty, while the second band is occupied by an always-active PU.

From Figure 3, notice how the majority of sensor nodes spend most of their time in the second band to harvest RF energy, while leaving this band only once during a cycle to attempt transmitting data. Observe that, due to the distributed nature of the system, initially the nodes might collide with each other. However, from these collision incidents, the sensor nodes start to learn and change their transmission behavior until they settle on a configuration that has no collisions (see cycle Y in Figure 3).

To progressively reach that desired configuration, each node executing the S-LEARN MAC protocol maintains a tentative schedule of its intended data transmission and harvesting operations. The schedule is maintained and updated by the sensor node by factoring in the results of its own spectrum sensing operations. The schedule is only used locally and is never sent to other nodes in the network, which avoids extra data overhead that affects throughput.

In addition, the schedule is limited to a length of K time epochs (slots) to reduce memory requirements at the node, where K is an integer number called the cycle length. We require that the schedule at all sensor nodes be set to the same K value by design, but we do not require system-wide synchronization. Therefore, the schedules of different nodes do not have to be perfectly aligned (as illustrated in Figure 4).

In Figure 4, the cycle length is set to $K = 4$ time epochs (time epochs are referred to as time slots in the context of

scheduling since schedules repeat over time). The cycle length is selected in our MAC protocol to allow the sensor nodes to send a maximum of one data packet during one cycle. The remaining slots in the cycle must be used by the sensor node to harvest enough energy for one transmission. Hence, we must maintain the following condition on the cycle length

$$(K - 1) E_{\#} \geq E_{\Gamma} \quad (1)$$

or

$$K \geq \frac{E_{\Gamma} + E_{\#}}{E_{\#}}. \quad (2)$$

Typically, however, we need K to be slightly larger than this value because the sensor node is not guaranteed to find an available PU from which to harvest energy during all the $(K - 1)$ time slots, which means, depending on the activity behavior of the PUs in the system, K might need to be larger than that specified by (2) to ensure that the sensor nodes harvest E_{Γ} energy to be able to transmit one packet in that cycle.

Remember that we assumed (in our simple example in Figures 3 and 4) that the PU in band 1 is switched off, which means the band is always empty and available for data transmission, while the PU in band 2 is always on, which makes this band a perfect candidate for energy harvesting. If the sensor nodes somehow knew about this fact, they would schedule their transmissions in band 1 and their energy harvesting activities in band 2. This is the first problem our MAC protocol attempts to solve, which is to identify the bands that are best candidates for energy harvesting. The nodes in S-LEARN identify this information autonomously by observing the behavior of the PU activation/deactivation within a cycle using two simple counters that they maintain, $\#_0^{k,m}$ and $\#_1^{k,m}$ (to be explained shortly).

The next challenge the sensor nodes encounter is to select one proper time slot within their schedule to transmit their packet, such that they avoid selecting the same time slot as other nodes in the network, hence resulting in a collision. Figure 3 shows how the sensor nodes in S-LEARN start with random selections thus sometimes successfully transmitting and sometimes colliding with other nodes. However, due to two more counters in each node, called $\Gamma_0^{k,m}$ and $\Gamma_1^{k,m}$, the nodes progressively and independently learn

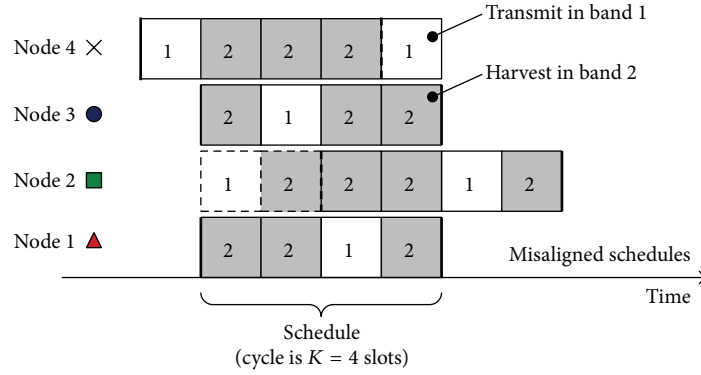


FIGURE 4: Tentative schedules maintained by the four sensor nodes during cycle Y in the example of Figure 3.

about the behavior of other nodes and start selecting the time slot/spectrum band combinations that are different from others. Figure 4 illustrates one desirable set of schedules at the different nodes in our example. We can see from the figure that node 1 decided to send its packet in band 1 during slot 3 of the cycle. Notice also that the schedule of each node does not have to be synchronized with other nodes for this technique to work, because the schedules are repetitive and of the same length K . For example, node 2 decided to transmit its packet in band 1 in time slot 3 (relative to its own schedule), but this is slot 1 relative to node 1's schedule, and hence no collision occurs between nodes 1 and 2.

3.3. S-LEARN Counters. To achieve the above described scheduling policy, each sensor node running our proposed MAC protocol must maintain a total of *four counters* for each time slot $k \in [1, K]$ in the K -long schedule and for each available spectrum band $m \in [1, M]$ in the CRSN system. Those counters are as follows.

- (i) $\mathcal{H}_1^{k,m}$: which represents the number of *successful* energy harvesting incidents the sensor node observed whenever it attempted to harvest energy from band m during time slot k of its own schedule.
- (ii) $\mathcal{H}_0^{k,m}$: which is similar to $\mathcal{H}_1^{k,m}$ but counts the number of *failed* energy harvesting incidents the sensor node observed, because no active PU was found in that band at that time slot.
- (iii) $\Gamma_1^{k,m}$: which counts the number of *successful* packet transmissions (for which an ACK was received) that the sensor node observed whenever it attempted to transmit through band m during time slot k of its own schedule.
- (iv) $\Gamma_0^{k,m}$: which is similar to $\Gamma_1^{k,m}$ but counts the number of *collisions* the sensor node observed whenever it attempted to transmit through that band and time slot.

Each sensor node builds the tentative schedule for the next cycle with the help of these counters, as will be described shortly, and then starts abiding by the schedule for the whole duration of the cycle. Notice that the sensor node can only

access one band at any given time slot due to its limited hardware capabilities (see Figure 2). When accessing a band at a certain time slot, the node senses that band at the beginning of the slot, at which time four possibilities could arise: (i) the sensor node attempts (based on its schedule) to harvest energy from band m at time slot k , and when it senses the band it finds that a PU is already sending RF energy in that band. In this case, the sensor node is lucky and harvests $E_{\mathcal{H}}$ energy from that band for the duration of the time slot and then increments its $\mathcal{H}_1^{k,m}$ counter by 1. (ii) Another possibility is for the node to attempt to harvest energy, but when it senses band m at the beginning of slot k , it discovers that the band is empty. This happens when the PU is inactive during that time slot. In this case, the sensor node increments its harvesting *failure* counter $\mathcal{H}_0^{k,m}$ by 1 and refrains from sending information to avoid colliding with other nodes that might have scheduled transmission in that slot/band pair. (iii) The third possibility is when the sensor node attempts to access the spectrum for data transmission (assuming there are packets awaiting transmission in its queue), and it is fortunate enough that sensing band m at the beginning of slot k finds it empty. In that case, the node sends its data packet and waits for an ACK from the data sink. If the ACK arrives at the end of the time slot then the $\Gamma_1^{k,m}$ counter is incremented by 1; otherwise a collision must have happened and rather the collision counter $\Gamma_0^{k,m}$ is incremented by 1. (iv) The final possibility is for the sensor node to access the band for data transmission but finds it occupied by an active PU, at which case the node takes this opportunity to harvest energy anyway, increments its harvesting success counter $\mathcal{H}_1^{k,m}$ by 1, and delays the transmission of its packet until the next cycle.

It cannot be missed that the four counters are initialized to zero and then continuously updated in the aforementioned manner as the S-LEARN protocol is executed by the sensor nodes. However, at the end of each cycle (and *after* building the tentative schedule for the next cycle), *each* of the four counters is *multiplied* individually by a factor $A < 1.0$ called the *aging factor*, which is common to all the counters. This aging process helps the sensor nodes to slowly forget old information that might not be relevant any more as the network structure changes over time.

It is worth mentioning that the counters are cumulative and are not reset for as long as the sensor node is running. However, multiplying by the aging factor, which is smaller than unity, will reduce the value in these counters to almost zero as time passes on.

3.4. Building the Schedule. At the beginning of each scheduling cycle, the sensor node builds its own local schedule by inspecting the values of its four counters. The schedule involves two objectives: (a) reducing the chance of colliding with other sensor nodes when the PU is inactive and (b) harvesting RF energy when the PU is active.

Since the sensor node does not have prior knowledge of the spectrum occupancy of the PU in the next cycle, it uses the harvesting success and failure counters to estimate, in a probabilistic way, the chance of finding an active PU in a particular band. Hence, the sensor node calculates at the beginning of the cycle the *harvesting scores* χ_m for bands $m \in [1, M]$, where

$$\chi_m = \max \left[a_h \frac{1 + \sum_{k=1}^K \mathcal{H}_1^{k,m}}{1 + \sum_{m=1}^M \sum_{k=1}^K \mathcal{H}_1^{k,m}} + (1 - a_h) \frac{1 + \sum_{k=1}^K \mathcal{H}_1^{k,m}}{1 + \sum_{k=1}^K \mathcal{H}_1^{k,m} + \sum_{k=1}^K \mathcal{H}_0^{k,m}}, p_m \right]. \quad (3)$$

The first term inside the $\max[\]$ operator is an estimate of the probability of finding an active PU in band m when compared to other bands in the system, while the second term estimates the chance of successfully harvesting energy (as opposed to failing) from band m when joining that band. Notice that the first term only considers harvesting success but not failures, while the second term considers both success and failure of harvesting, albeit in one band only for the latter. This combination allows the sensor node to quickly gravitate toward bands with more PU activity, but at the same time not completely abandon other bands with less PU activity. This is useful since the PU activity can change over time. Finally, p_m in (3) is a small probability, which we set to $p_m = 0.01$ to prevent sensor nodes from setting $\chi_m = 0$, thus ensuring that sensor nodes will visit band m for harvesting from time to time to see if new PUs (that are suitable for harvesting) have been recently activated in that band.

The parameter a_h is a constant that can be used to control the dominant term in the above equation. The tradeoffs resulting from this (and other parameters in S-LEARN) will be described later in Section 6.

For each time slot in the cycle (except for the one chosen for data transmission), a coin is flipped to choose one of the M bands for energy harvesting, using uniform distribution with the probability assigned to picking any band $m \in [1, M]$ being

$$\hat{\mu}_m = \frac{\chi_m}{\sum_{m=1}^M \chi_m}. \quad (4)$$

Once harvesting scheduling is performed, the second task of the scheduler is to find a proper slot/band pair in which to attempt the data packet transmission within that cycle. This

is done by calculating a *transmit score* $\gamma^{k,m}$ for each slot k and band m pair in the cycle utilizing three of our earlier counters as follows:

$$\gamma^{k,m} = \max \left[\Gamma_1^{k,m} - a_c \Gamma_0^{k,m} - a_p \mathcal{H}_1^{k,m}, 0 \right]. \quad (5)$$

The transmit scores of all slot/band pairs are calculated and compared to find the maximum score. The transmit scores equal to the maximum possible $\gamma^{k,m}$ are then collected and one of them is chosen at random with uniform probability. That selected slot/band is no longer used for harvesting, but for data transmission instead. On the other hand, if the sensor node does not have any packets queued for transmission by the start of the cycle, it skips the step of choosing a slot/band pair for transmission and just uses the earlier algorithm to schedule energy harvesting in all time slots.

Equation (5) is easy to understand. The slot/band pairs that witnessed more successful packet transmissions and fewer collisions in the past receive a higher transmit score $\gamma^{k,m}$, and hence they tend to be picked up by the schedule for data transmission. Clearly, if a slot/band pair is heavily contested with other nodes, $\Gamma_0^{k,m}$ counter will detect this and the sensor node will penalize it with lower $\gamma^{k,m}$ score to avoid transmitting a packet in that location in the future. In addition, the last term in the equation prevents sensor nodes from attempting to transmit in bands and/or slots where PUs were most active in the past, as this means missing the opportunity to transmit a packet in the current cycle.

Again, the parameters a_c and a_p are control parameters that can fine-tune the S-LEARN protocol to fit the user needs. The tradeoffs resulting from such parameters are explained later.

4. Simulation Framework

We test the performance of the S-LEARN MAC protocol using simulations. We will first describe the simulated network setups and then discuss the performance measures used to evaluate the performance of our technique. Finally, we present the simulation results in the next section.

4.1. Simulation Parameters. We investigate three different setups of cognitive radio sensor networks. Such setups will illustrate different capabilities of our proposed MAC protocol. In all such scenarios, we will assume a geographical area with $N = 900$ cognitive sensor nodes contending for access to $M = 5$ underutilized spectrum bands. Each spectrum band is licensed to a PU that is active only part of the time. For all setups used, we employ the most common PU activity model used in literature, which is the one shown in Figure 5 [45]. This model assumes slotted PU environments, where the activity of the PU follows a two-state Discrete-Time Markov Chain (DTMC). The transition probability of the m th PU going from the inactive state to the active state at the beginning of a time epoch is given by β_m , while the reverse probability is α_m . Solving this DTMC with the indicated transition probabilities, we notice that the probability of the PU to be active at any given time epoch

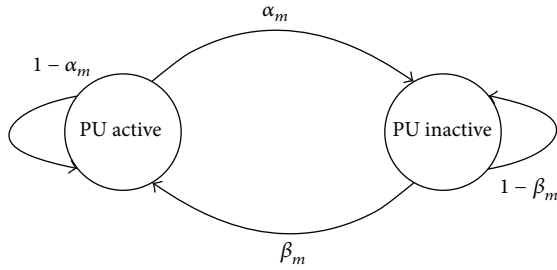


FIGURE 5: Activity state diagram for the PU in band m .

is $\mu_m = \beta_m / (\alpha_m + \beta_m)$. We assume that the sensor nodes do not know those transition probabilities; rather they will try to figure them out from the sensing information they obtain about the spectrum bands.

Packets are assumed to be fixed in size and each time epoch is enough to perform sensing, send one packet, and then receive the corresponding ACK. Packets are generated (arrive) at a sensor node according to a random Bernoulli process, and in each of the network setups (explained below), we vary the *mean packet arrival rate* λ_n at each sensor node $n \in [1, N]$, thus increasing the load ρ on the system. The main simulation parameters are summarized in Table 1. The simulation run is executed for a total of $T = 200,000$ time epochs, which is more than sufficient for the system to reach steady-state.

To provide a reference for comparison, and to show how progressive learning in S-LEARN can benefit the CRSN system, we also simulate two more MAC protocols that fit with our system model and hardware capabilities: the first is a *purely random harvest and transmit* technique, and the other is a *modified-CSMA* protocol [46].

A sensor node in the first technique (the purely random algorithm) picks, every time epoch, a band purely randomly with equal probability out of the M permissible bands and then senses the band for the first part of the time epoch. If the band turns out to be occupied by an active PU, the sensor node starts harvesting energy from that PU for the rest of the time epoch. On the other hand, if the node senses a spectrum band that is unoccupied by a PU, the sensor node attempts to transmit its data packet in that band immediately and then waits for an ACK in the same time epoch. Whether the packet went through successfully or not, the sensor node *does not* attempt to send a new packet (or retransmit a failed packet) until it has managed to harvest enough energy from PUs equal to the transmission energy E_T , even if it finds another empty transmission opportunity. Rather, the node keeps harvesting energy until its total harvested energy is equal to the transmission energy, and only then the node repeats its operations one more time.

Collisions are possible between nodes in the random harvest and transmit technique, but randomness in transmission helps in reducing such collisions. The randomness occurs because of three factors: the first factor is the randomness in picking the spectrum band, and the second factor is the time it takes a sensor node to collect E_T energy, which is random as it varies depending on the random PU activity in each picked

TABLE 1: Main simulation parameters.

Parameter	Value
Cognitive sensor nodes N	900
Spectrum bands M	5
Simulated time epochs T	200,000
PU active probability μ_m	{0.1, 0.3, 0.5, 0.7, 0.9}
Harvested energy per slot E_h	$10 \mu\text{W} \times \tau$
Transmission energy per packet E_T	$2.25 \text{ mW} \times \tau$
Cycle length K	256
Harvesting parameter a_h	0.6
Minimum probability p_m	0.01
Transmit score parameter a_c	0.5
Transmit score parameter a_p	0.2
Aging factor A	0.8
Battery maximum stored energy per sensor node B	$10E_T$
Initial battery energy	$0 \text{ mW} \times \tau$ (empty)
Mean packet arrival rate λ_n (packet per epoch)	0.78×10^{-3} to 3.52×10^{-3}
$[W_{\min}, W_{\max}]$ for modified-CSMA	[8, 10]

channel. The third contribution to randomness is obviously due to finding the empty slot, which occurs at different times for different nodes depending on the picked spectrum band and the inactivity of the PU at that instant (which is in itself is random). Hence collisions, though large, will not result in a catastrophic collapse of the system, especially that it takes much more time to charge the node (by wireless energy harvesting) than to drain it (by transmitting a data packet).

The second algorithm we implement is the slotted CSMA protocol modified for energy harvesting from the description in [46]. Again, the sensor node picks every time epoch a band purely randomly with equal probability out of the M bands and senses that band. If the band is occupied by an active PU, the sensor harvests energy from that PU. Conversely, if the node senses a spectrum band that is unoccupied by a PU, and the node has enough energy in its battery, it attempts to transmit its data packet in that band assuming that the CSMA backoff counter reached zero. Otherwise the backoff counter is decremented and the sensor node tires again in the next empty transmission opportunity.

The backoff counter is part of the well-known binary exponential backoff technique (used in the CSMA protocol in IEEE 802.11 [47] and IEEE 802.15.4 [48]). It aims to reduce possible collisions when the load on the system is high. This technique requires each sensor node to defer its transmission by a total of R empty transmission opportunities, where R is a random integer drawn from a uniform distribution over the interval $[0, 2^w - 1]$, which is known as the contention window. The integer value of w is dependent on the number of consecutive collisions i the node suffered in sending the packet. The value of w increases as i increases but is

maintained between the minimum and maximum values of W_{\min} and W_{\max} , respectively, as follows:

$$w = \min(W_{\min} + i, W_{\max}). \quad (6)$$

The values we use for W_{\min} and W_{\max} are shown in Table 1. We will see that reducing collisions using this CSMA backoff technique is beneficial in CRSNs, since it saves wasted energy due to repeated retransmissions, which in turn also saves on the overhead of harvesting wireless energy to make up for such waste.

4.2. Network Setups. We will test our MAC protocol under three different network setups to evaluate different aspects of the progressive learning technique. These network setups are as follows.

Setup 1: Stable Cognitive Radio Sensor Network. In this first test, a total of $N = 900$ sensor nodes operate for the whole duration of the simulation within $M = 5$ spectrum bands occupied by five different PUs. The conditions of the network are stable, as the different PUs are activated and deactivated randomly, each within its own band, according to the probabilities $\mu_m = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Each sensor node has a queue to which packets arrive, with an average arrival rate of λ_n packets per time epoch (the arrival rate is identical for all nodes, i.e., $\lambda_n = \lambda, \forall 1 \leq n \leq N$). We vary the packet arrival rate λ to observe how our MAC protocol handles increasing the load on the system.

Setup 2: New Sensor Nodes Joining the System. Compared to the first network setup, this is a similar experiment except that at the start of the simulation only 750 sensor nodes are active in the 5 spectrum bands. In the middle of the simulation, another 150 are added to the system for a total of 900 sensor nodes. This allows us to investigate the robustness of our technique as it allows the old and new nodes to progressively learn about the new topology of the network and adapt to this change. We maintain the same PU behavior as in the first experiment.

Setup 3: Sudden Change in PU Behavior. As the PUs activate and deactivate, the sensor nodes in our S-LEARN MAC protocol start learning the behavior of such PUs to better utilize their spectrum bands. The final test looks at the possibility of the PUs changing their behavior to which the nodes are now habituated. We observe how the sensor nodes can easily detect the change in the PU behavior and readjust their conduct to suit the new actions of the PUs. We design this network setup to be a variant of the first configuration, where the PUs are still activated and deactivated randomly. However, only during the first half of the simulation, the activation probabilities for the PUs are given by $\mu_m = \{0.1, 0.3, 0.5, 0.7, 0.9\}$. In the middle of the simulation, the most active PU becomes the least active and the least active PU becomes the most active. In other words, the activation probabilities for the PUs in the second half of the simulation suddenly change to $\mu_m = \{0.9, 0.3, 0.5, 0.7, 0.1\}$. Notice that the sensor nodes have to quickly and skillfully learn about this new situation because they rely on finding the most active PU

for maximizing their harvested energy, and this most active PU changes at time epoch $t = T/2$.

4.3. Performance Measures. The following performance measures are used to evaluate the performance of our proposed MAC protocol and compare it to both the purely random harvest and transmit technique and the modified-CSMA algorithm.

Successfully Transmitted Packets (Throughput). We count the number of packets transmitted successfully from the N sensor nodes per time epoch (i.e., with an ACK received), and we denote this by $s(t)$. Notice that this number is expected to be smaller than M , because there are only M spectrum bands available for the N sensor nodes to transmit their packets at any time epoch. Considering also that during a sizable chunk of the time PUs will be active, thus preventing sensor nodes from using such bands, the number will be even smaller. In addition, some of the transmission opportunities will be missed and some will also be occupied by collisions, further reducing the $s(t)$ value. In the results, we sometimes evaluate the average number of successfully transmitted packets per epoch over all time epochs, which is given by $S = \sum_{t=1}^T s(t)/T$.

Colliding Packets. We also count the number of packets transmitted by the N sensor nodes that have collided per time epoch (i.e., did not receive an ACK), and we denote it by $c(t)$. Notice that the actual number of collision incidents per time epoch is smaller than $c(t)$ because two or more colliding packets are required to result in one single collision. However, the value of $c(t)$ is more relevant to our discussion here because it shows both the missed spectrum opportunity and wasted energy by sensor nodes. Since energy is hard to come by, a packet that suffers a collision represents not only a waste of spectrum but also a waste of energy that needs to be harvested for excruciatingly long time before being used to transmit a successful packet again. The average number of colliding packets per epoch over all time is $C = \sum_{t=1}^T c(t)/T$.

Successful Harvesting Events. We denote by $h(t)$ the count of sensor nodes that each managed to successfully harvest $E_{\#}$ energy from an active PU during a time epoch t . The average of $h(t)$ over all time epochs is $H = \sum_{t=1}^T h(t)/T$. A desirable value of $h(t)$ is to be as close as possible to N , because that means sensor nodes are mostly sitting in a PU-filled band to harvest energy. However, since PUs are not always active, and since nodes can be unlucky in guessing where to find an active PU, the number will drop below N .

Node Energy. Throughout the simulation, we also track the energy level stored in the battery of each sensor node, which we denote by $e_n(t)$. The average energy for all sensor nodes per epoch is $e(t) = \sum_{n=1}^N e_n(t)/N$, and the average energy for all sensor nodes over all time epochs is $E = \sum_{t=1}^T e(t)/T$. Higher values of $e_n(t)$ are desirable in a CRSN because it allows nodes to adapt to changes in the network configuration, such as the change of PU activity without quickly running out of energy to transmit data. Nevertheless, to make the simulation more realistic, we limit the capacity of

the battery to a maximum value of $B = 10E_T$. This represents a very small sized and quite inexpensive battery that can only transmit 10 packets before needing a recharge.

Queue Size at Sensor Nodes. The queue size at each sensor node is also recorded at every time epoch as the simulation progresses and is denoted by $q_n(t)$. The queue size is an important parameter because it is directly proportional to both the memory needed by the sensor node (which adds to system cost) and also the delay the packets experience as they are traversing the sensor node. The average queue size over all sensor nodes per time epoch is evaluated as $q(t) = \sum_{n=1}^N q_n(t)/N$, and the average queue size for all sensor nodes over all time is denoted by $Q = \sum_{t=1}^T q(t)/T$. Smaller values of Q are, of course, desirable to reduce system cost and to provide better performance to end users. We do not limit the queue size in our simulations, and packets are never discarded.

Learning Time. The robustness of our MAC protocol to changes in the network configuration when new sensor nodes join the system is measured by the learning time, denoted by T_l . The learning time measures the time interval needed for these newcomers (and incumbent nodes) to get acclimated with the new system configuration, thus reaching a new steady-state that is different from the old steady-state. We define this interval to be equal to the number of consecutive time epochs required for the number of successfully transmitted packet $s(t)$, $t > T/2$, to reach the new average value of successful packets $S_{\text{new}} = \sum_{t=T/2}^T s(t) \times 2/T$ after the system change happens at $t = T/2$.

Adjustment Time. The adjustment time, represented by T_a , is used when the activity of any of the PUs in the system changes. It measures the time it takes nodes in our MAC protocol to figure out the new PU activity behavior and adjust their harvesting/transmission schedules to suit that new configuration. We set this interval to be equal to the number of consecutive time epochs required for the number of successful harvesting events $h(t)$, $t > T/2$, to reach the new average value of successful harvesting $H_{\text{new}} = \sum_{t=T/2}^T h(t) \times 2/T$ after the PU behavior changes at $t = T/2$. Small leaning and adjustment times are desirable features of our S-LEARN technique, and we show in the results that we can control how small these values can get as part of a tradeoff between overall system throughput and quick learning of the system configuration.

5. Results and Discussion

5.1. Stable Network. We use the first network setup to observe how our intelligent S-LEARN algorithm behaves compared to the random harvest and transmit technique and the modified slotted CSMA technique. Figures 6–10 show the results for the stable CRSN presented earlier. The figures shown are for the case where the average packet arrival rate at each sensor node is $\lambda = 1.95 \times 10^{-3}$ packets/epoch (or 0.5 packets per scheduling cycle). The figures are drawn with the aid of a

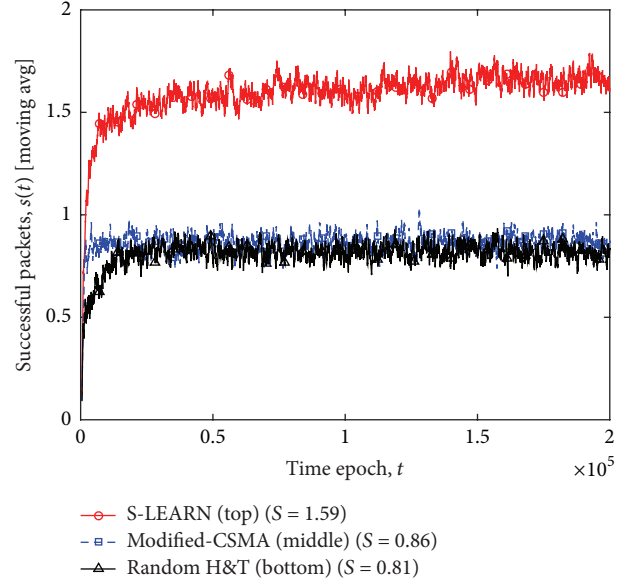


FIGURE 6: Number of successfully transmitted packets by sensor nodes at each time epoch in the first network setup when $\lambda = 1.95 \times 10^{-3}$ packets/epoch.

500-point moving average to allow for more solid lines and better clarity.

Figure 6 shows the number of successfully transmitted packets $s(t)$ by sensor nodes at each time epoch. The figure clearly shows the contribution of the proposed MAC algorithm which provides performance almost double that of the random and modified-CSMA techniques. This is achieved by the sensor nodes progressively and intelligently learning about the behavior of the PUs in the system and each other and dynamically adjusting their internal schedules to minimize collisions amongst themselves (see Figure 7) and also increase the harvested energy (see Figure 8). Both effects contribute to the increase in throughput since collision reduction increases the available spectrum for successful data transmission and reduces the wasted energy at the node. Saving such energy, and being able to find where the active PUs are located thus harvesting even more energy, means that each sensor node always has a healthy amount of energy stored in its battery (see Figure 9) available to send any packet when an opportunity is presented, without the need to delay such packet until energy is harvested at a very slow pace. This, of course, reduces the queue size (and packet delay) at each node, as evident by Figure 10. Of course, being able to transmit the packet with minimum number of collisions, and hence minimum number of retransmissions, also contributes to the lower packet delay and the smaller queue size seen in Figure 10.

The behavior of S-LEARN under various loads on the system is similar. We show a comparison between the number of successful packets for each of the three algorithms versus load in Figure 11. The superiority of the S-LEARN MAC protocol compared to others techniques is remarkable. Of course, as the load on the system increases beyond a certain point, the amount of energy needed by the sensor nodes

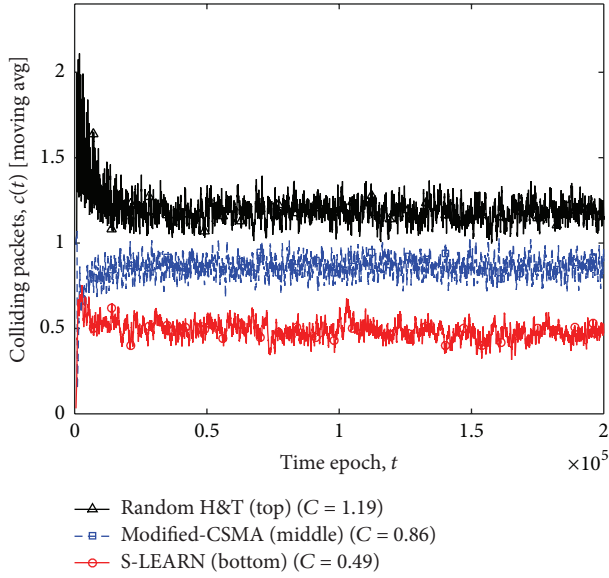


FIGURE 7: Number of colliding packets from sensor nodes at each time epoch in the first network setup when $\lambda = 1.95 \times 10^{-3}$ packets/epoch.

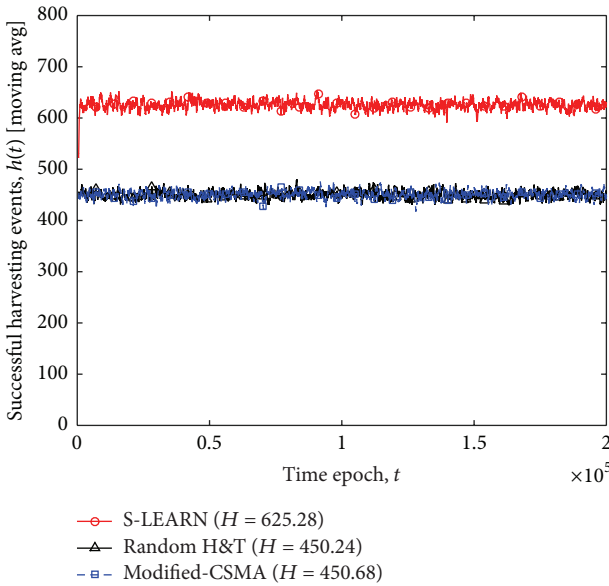


FIGURE 8: Number of sensor nodes successfully harvesting energy at each time epoch in the first network setup when $\lambda = 1.95 \times 10^{-3}$ packets/epoch.

becomes quite large that the nodes can no longer harvest enough energy to meet such demand. In this case the throughput (number of successful packets) of all algorithms (including ours) levels off and the amount of stored energy in the node battery starts diminishing (see Figure 12). Clearly, though, our MAC protocol provides much higher throughput even at higher loads compared to the random harvest and transmit and the modified-CSMA protocols. In addition, S-LEARN retains energy in the battery to a better extent compared to these two techniques, of which the random

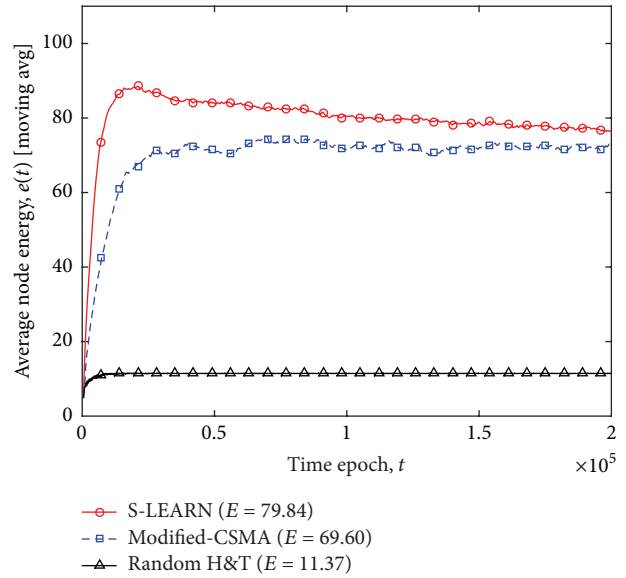


FIGURE 9: Energy stored in each node averaged over all nodes at each time epoch in the first network setup when $\lambda = 1.95 \times 10^{-3}$ packets/epoch.

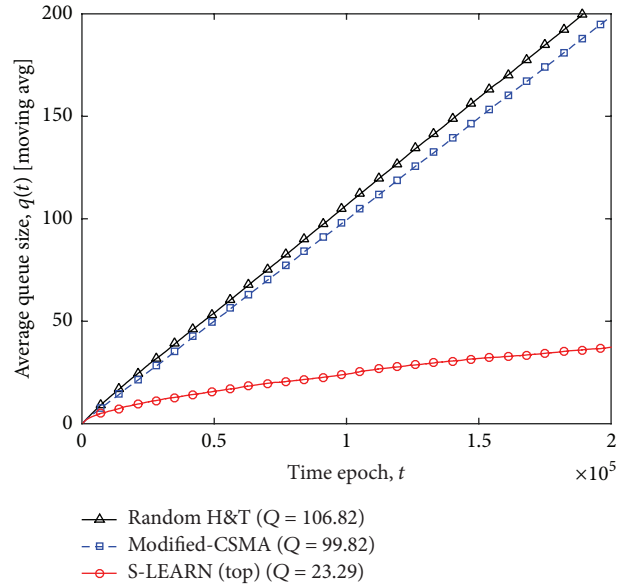


FIGURE 10: Queue length at each node averaged over all nodes during each time epoch in the first network setup when $\lambda = 1.95 \times 10^{-3}$ packets/epoch.

harvest and transmit technique is the worst. That is why its packets start accumulating heavily in the queue waiting for a long time, as evident by Figure 13.

5.2. Newcomer Nodes. In the second scenario, we investigate what happens when the CRSN is supplemented by extra sensor nodes while in operation. We want to see if the new nodes can be seamlessly integrated into the system or if they do cause undesirable instability. Hence, we run the network using only 750 sensor nodes, but at time $t = T/2$, another 150

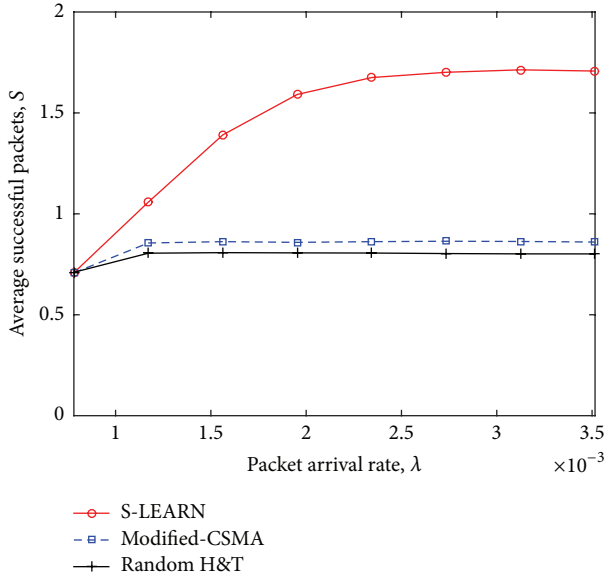


FIGURE 11: Number of successfully transmitted packets by sensor nodes versus arrival rate (system load) in the first network setup.

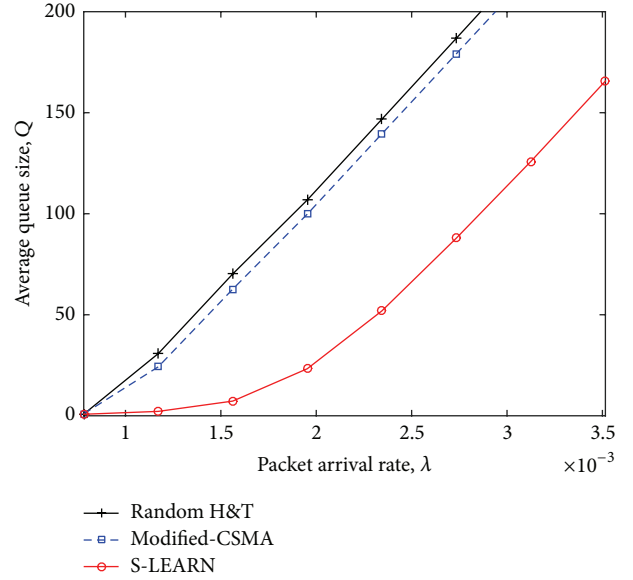


FIGURE 13: Queue length at each node averaged over all nodes versus arrival rate (system load) in the first network setup.

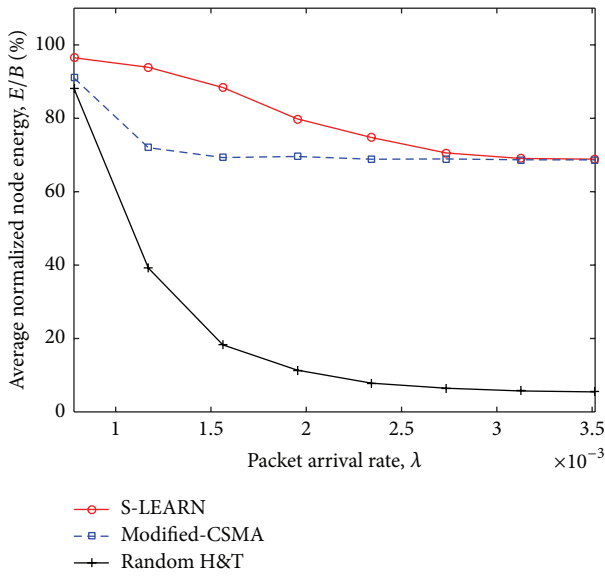


FIGURE 12: Energy stored in each node averaged over all nodes versus arrival rate (system load) in the first network setup.

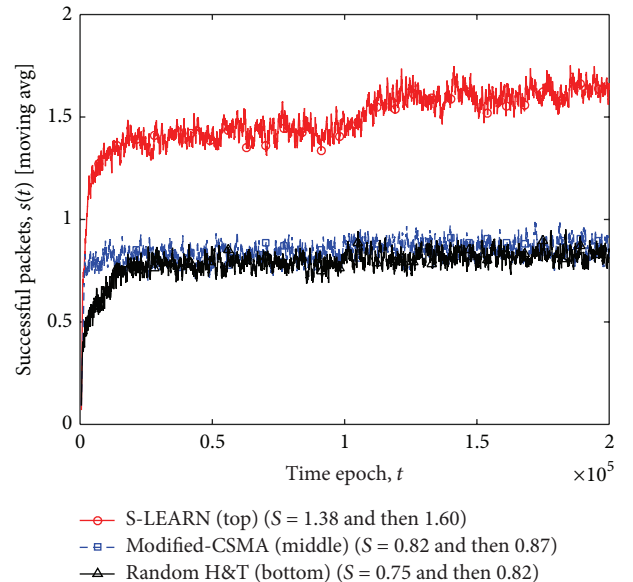


FIGURE 14: Number of successfully transmitted packets by sensor nodes at each time epoch in the second network setup when $\lambda = 1.95 \times 10^{-3}$ packets/epoch.

nodes are activated. This brings the total number of sensor nodes to 900, which is similar to what we had in our first experiment. Hence, the results of the earlier section should provide a reference for comparison.

Figure 14 shows the number of packets successfully transmitted by nodes versus time. It is evident that the random and modified-CSMA techniques cannot adopt properly to this 20% increase in load on the system. On the other hand, our S-LEARN MAC protocol progressively learns about the new situation and gradually adjusts the schedules at the different sensor nodes to accommodate the newcomer population, thus increasing the overall system throughput, albeit on the

expense of some increase in collisions, which is natural at such high level of load (see Figure 15).

In Figure 16 we show the learning time T_l the S-LEARN technique requires for different loads on the system (i.e., different packet arrival rates). To find T_l we count the number of consecutive epochs it takes for the system to move from the old S value (e.g., $S = 1.38$ in Figure 14) to the new S value (e.g., $S = 1.60$ in Figure 14). We notice that as the load on the system increases, so does the learning time, which is expected. This continues until the load becomes high enough that the system is saturated, which means the throughput of

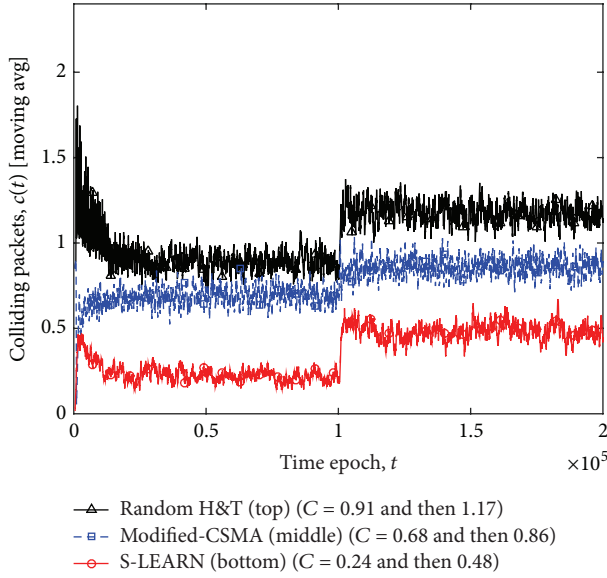


FIGURE 15: Number of colliding packets from sensor nodes at each time epoch in the second network setup when $\lambda = 1.95 \times 10^{-3}$ packets/epoch.

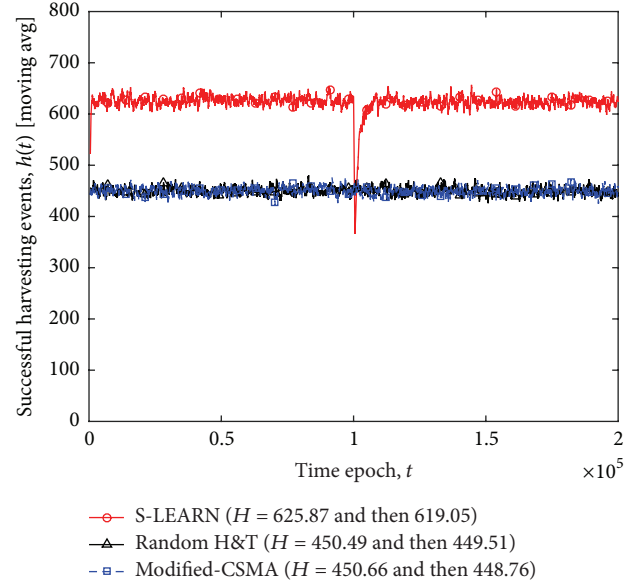


FIGURE 17: Number of sensor nodes successfully harvesting energy at each time epoch in the third network setup when $\lambda = 1.95 \times 10^{-3}$ packets/epoch.

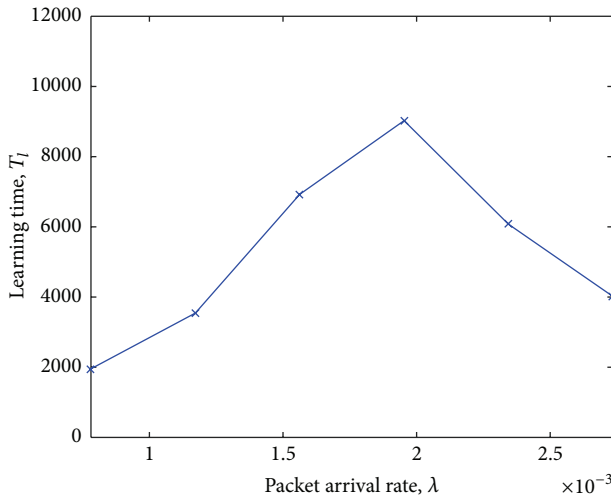


FIGURE 16: Learning time for sensor nodes versus arrival rate (system load) in the second network setup. Results are averaged over multiple simulation runs.

the system can no longer increase even with more offered load (see also Figure 11). At that point, the learning process cannot do anything to improve the system throughput and its interval starts diminishing.

5.3. PU Disruption. In the final test we disturb the process of energy harvesting at the sensor nodes by abruptly changing the activity probability of the most active and the least active PUs in the system at the middle of simulation. Instead of $\mu_5 = 0.9$, the most active PU switches to $\mu_5 = 0.1$, and the least active PU, which used to have $\mu_1 = 0.1$, continues the rest of the simulation with $\mu_1 = 0.9$.

Both the random harvest and transmit along with the modified-CSMA algorithms are not affected by this change because they do not make any assumptions about the location of the PU; rather they randomly and uniformly pick a band; if a PU is active in that band they harvest its energy. However, this behavior is not very useful because it can only harvest a small amount of wireless energy as seen in Figure 17. However, our S-LEARN technique not only can harvest more energy by predicting where the most active PUs are, but also can adjust once it notices that it is not harvesting enough energy, and recalibrate its harvesting scores to move to the spectrum band that corresponds to the most active PU (see Figure 17).

During the adjustment time, of course, there is a slight drop in the harvested energy and also a slight drop in the number of successful packets (see Figure 18), which is affected because of lack of sufficient energy. It is worth mentioning here that a larger battery reservoir (which we did not use) can help in such cases because it can accommodate continuing data transmission while adjusting the harvesting process to the new system conditions, albeit with the disadvantage of adding extra cost to the system.

It is also interesting to mention that the adjustment time T_a is not dependent on the system load as the equations governing the harvesting process (see (3) and (4)) are decoupled from the data transmission process (system load). This is evident in Figure 19, which shows the adjustment time versus average packet arrival rate.

6. Tradeoffs

The control parameters in our S-LEARN technique can be quite useful to tune the behavior of the CRSN to steer it

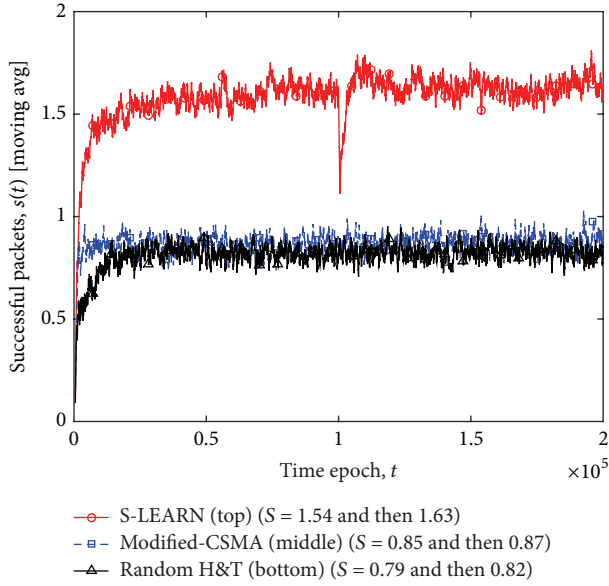


FIGURE 18: Number of successfully transmitted packets by sensor nodes at each time epoch in the third network setup when $\lambda = 1.95 \times 10^{-3}$ packets/epoch.

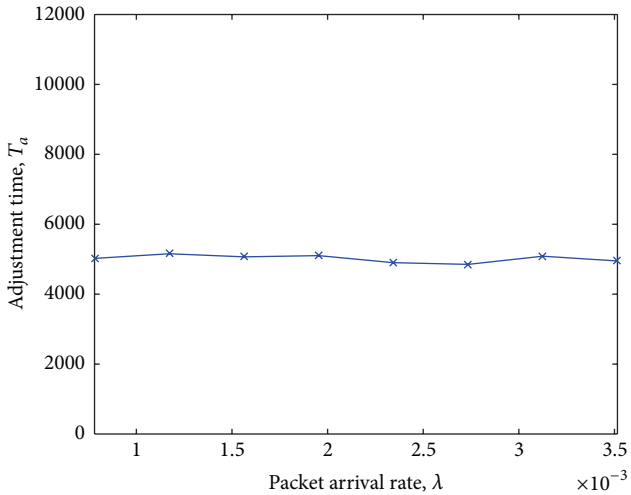


FIGURE 19: Adjustment time for sensor nodes versus arrival rate (system load) in the third network setup. Results are averaged over multiple simulation runs.

in the direction most desirable for the network designer. For example, the network can be optimized for overall throughput and minimum delay, or on the other hand, it can be optimized to quickly react to changes in the network setup, such as when new nodes are introduced to the system or when PU activity changes.

To show some examples on how the different parameters can control the performance of the powerful S-LEARN protocol, we rerun our simulations based on different values of the control parameters when $\lambda = 2.344 \times 10^{-3}$ packets/epoch.

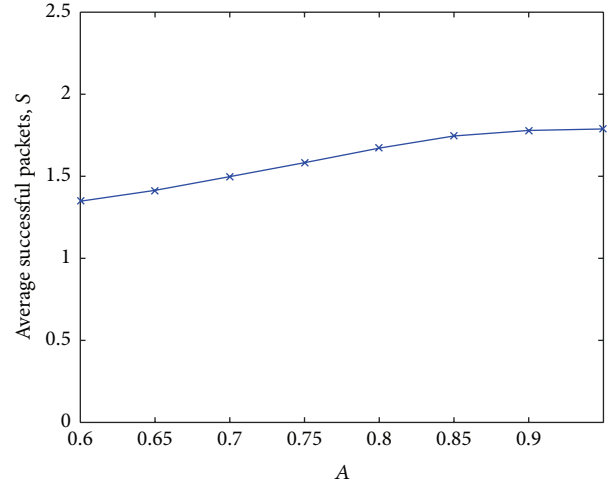


FIGURE 20: Average number of successfully transmitted packets by sensor nodes versus different values of the aging parameter A .

For example Figures 20–23 show the behavior of the network throughput, collisions, learning time, and adjustment time versus different possible values of the aging parameter A . All other parameters are retained to the values in Table 1.

It is easy to see that a larger A value (closer to the maximum of 1.0) results in higher throughput and smaller number of collisions. This is because the sensor nodes start retaining the knowledge they learn from the network and use that to produce better schedules to avoid colliding with each other. However, when the sensor nodes cannot forget old information, they cannot adapt quickly to changes in the network configuration as evident by progressively longer learning time and adjustment time required when A is closer to 1.0. The designer can balance both requirements by choosing a midpoint value for the A parameter or, if he/she so desires, can prefer higher throughput compared to quick response, at which case a higher A value is more befitting. Or the designer might prefer better response to changes by reducing the value of the aging factor A , albeit on the cost of losing some throughput.

The transmit score parameter a_c balances a tradeoff between throughput and learning time. Higher values of a_c force the sensor nodes to quickly leave contended bands and search for different ones, which reduces their aggressiveness in obtaining spectrum, thus reducing throughput (as shown in Figure 24). However, it also allows the newcomer nodes to avoid aggressively disturbing the incumbent nodes but substitute that with them looking for other empty bands. This results in a smoother joining process and shorter learning time (see Figure 25).

Similar observations can be made about the second transmit parameter a_p (see Figures 26 and 27), though for a different reason. Higher values of a_p force sensor nodes to leave bands that have been used successfully in the past for harvesting, which allows the sensor nodes to avoid spectrum bands that are commonly used by active PUs. However, this also means that some transmission opportunities in empty bands are missed (remember that no PU is active for

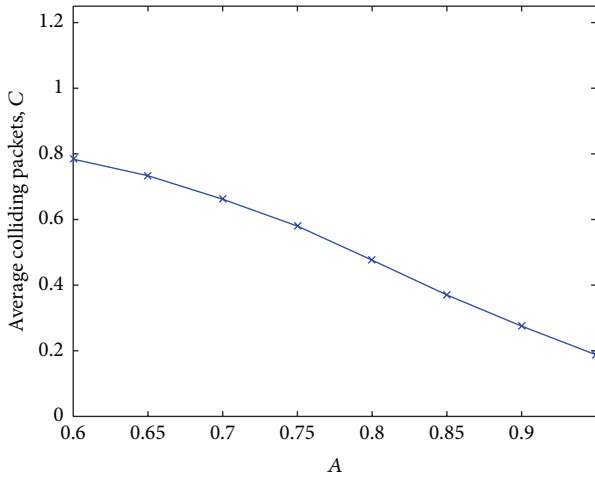


FIGURE 21: Average number of colliding packets from sensor nodes versus different values of the aging parameter A.

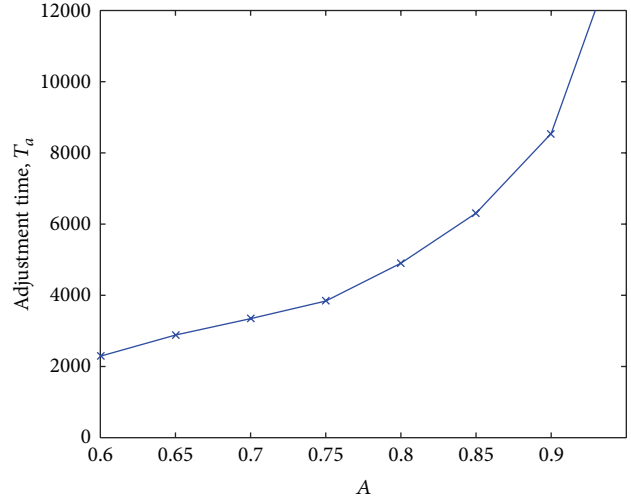


FIGURE 23: Adjustment time for sensor nodes versus different values of the aging parameter A.

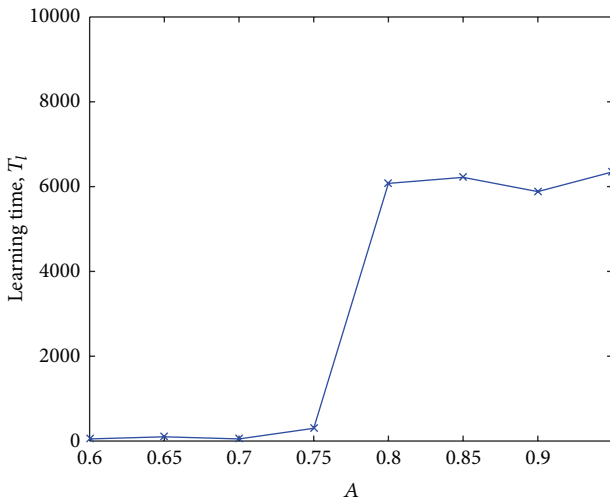


FIGURE 22: Learning time for sensor nodes versus different values of the aging parameter A.

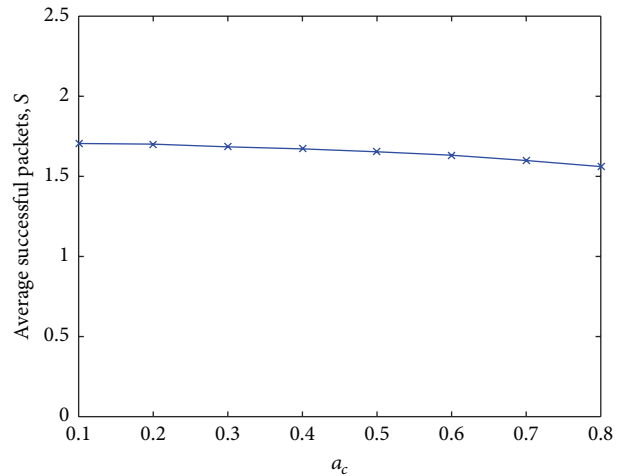


FIGURE 24: Average number of successfully transmitted packets by sensor nodes versus different values of the transmit parameter a_c .

100% of the time), which means that throughput will suffer. Conversely, newcomers (who are starved of bandwidth) will have an easier time finding empty slots (that incumbents avoided) and hence will have a shorter learning time.

Finally, the harvesting parameter a_h has an effect on adjustment time and the amount of energy harvesting possible. When a_h is small, that is, when $(1 - a_h)$ is large, the second parameter in (3) becomes more dominant, allowing a sensor node to quickly detect changes in the PU activity probability and move quicker to a more active PU for harvesting, which translates into a small adjustment time. This is clearly evident from Figure 28. However, a higher a_h value allows a node to stick more closely to the spectrum where it has harvested more energy compared to other spectrum bands, which when PU activity is stable allows the node to harvest much more energy in the long run (as shown in Figure 29).

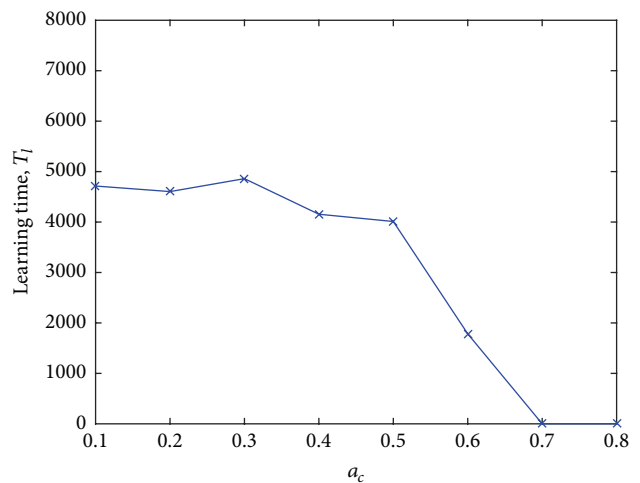


FIGURE 25: Learning time for sensor nodes versus different values of the transmit parameter a_c .

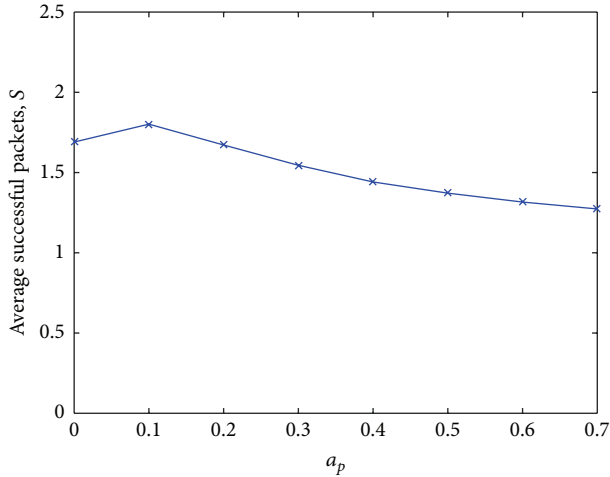


FIGURE 26: Average number of successfully transmitted packets by sensor nodes versus different values of the transmit parameter a_p .

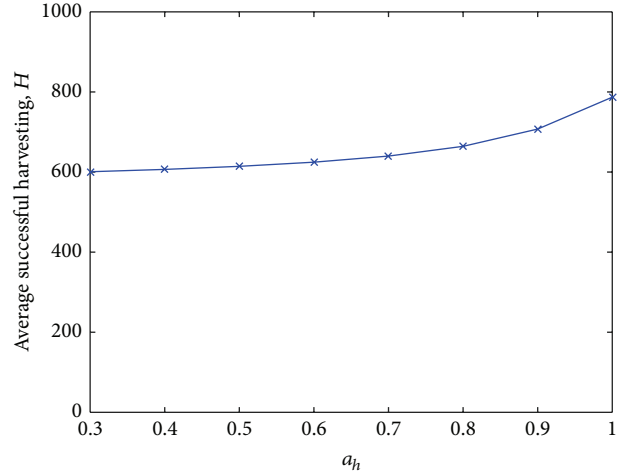


FIGURE 29: Average number of sensor nodes successfully harvesting energy versus different values of the harvesting parameter a_h .

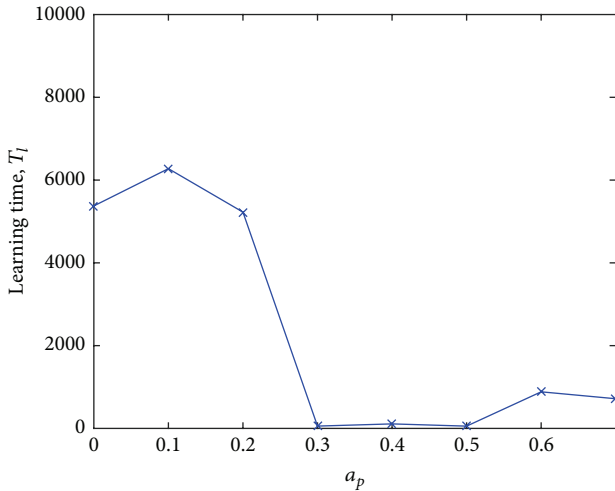


FIGURE 27: Learning time for sensor nodes versus different values of the transmit parameter a_p .

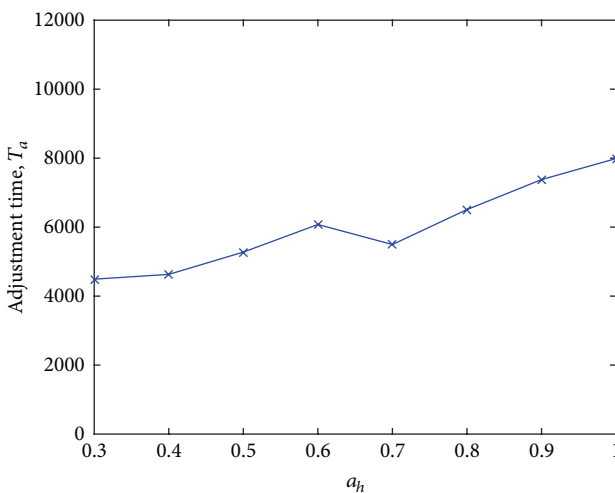


FIGURE 28: Adjustment time for sensor nodes versus different values of the harvesting parameter a_h .

Hence, for stable networks we recommend a value of $a_h = 1.0$ to increase the level of energy harvesting, but for nodes where PUs can change their behavior frequently, we recommend a value of $a_h \leq 0.7$ to keep the adjustment time small.

7. Conclusions

A novel MAC protocol was introduced for CRSNs. Each node in this protocol develops a schedule to coordinate its energy harvesting and data transmission activities. The schedule is built based on the perceived environment of PUs and other sensor nodes. This is achieved by simply maintaining four counters and using them intelligently to infer the surrounding conditions. We proved, via simulation, that the performance enhancements are remarkable, reaching almost twice that of random harvest/transmit and modified-CSMA techniques.

In addition, the S-LEARN protocol is quite robust and can easily adapt to new sensor nodes joining the network or to PUs changing their activity behavior. Not only that, but such robustness and performance are controllable by a set of parameters that can be changed to suit the objective of end users of the system.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

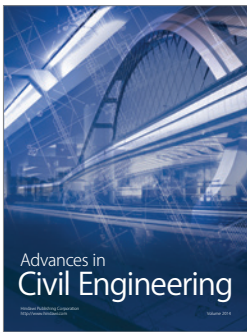
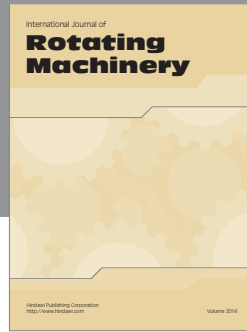
Acknowledgment

This work was supported in part by the Deanship of Academic Research at The University of Jordan.

References

- [1] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, "Wireless sensor networks: a survey on recent developments and potential synergies," *The Journal of Supercomputing*, vol. 68, no. 1, pp. 1–48, 2014.
- [2] A. A. A. Shabaneh, A. M. Ali, C. K. Ng, N. K. Noordin, A. Sali, and M. H. Yaacob, "Review of energy conservation using duty cycling schemes for IEEE 802.15.4 wireless sensor network (WSN)," *Wireless Personal Communications*, vol. 77, no. 1, pp. 589–604, 2014.
- [3] J. Chen, J. Jia, E. Dai, Y. Wen, and D. Zhao, "Bicriteria optimization in wireless sensor networks: link scheduling and energy consumption," *Journal of Sensors*, vol. 2015, Article ID 724628, 11 pages, 2015.
- [4] A. R. Swain, R. C. Hansdah, and V. K. Chouhan, "An energy aware routing protocol with sleep scheduling for wireless sensor networks," in *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA '10)*, pp. 933–940, IEEE, Perth, Australia, April 2010.
- [5] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "PW-MAC: an energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 1305–1313, IEEE, Shanghai, China, April 2011.
- [6] Z.-T. Li, Q. Chen, G.-M. Zhu, Y.-J. Choi, and H. Sekiya, "A low latency, energy efficient MAC protocol for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 946587, 9 pages, 2015.
- [7] M. Hawa, S. Taifour, M. Qasem, and W. Tuffaha, "A dynamic cross-layer routing protocol for Mobile Ad hoc Networks," *AEU—International Journal of Electronics and Communications*, vol. 66, no. 12, pp. 996–1005, 2012.
- [8] Abdul Razaque and K. M. Elleithy, "Low duty cycle, energy-efficient and mobility-based border node—MAC hybrid protocol for wireless sensor networks," *Journal of Signal Processing Systems*, vol. 81, no. 2, pp. 265–284, 2015.
- [9] G. AL-Sukkar, I. Jafar, K. Darabkh, R. Al-Zubi, and M. Hawa, "Cooperative energy efficient routing for wireless sensor networks in smart grid communications," *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, vol. 7, no. 4, pp. 409–416, 2013.
- [10] N.-T. Le and Y. M. Jang, "Energy-efficient coverage guarantees scheduling and routing strategy for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 612383, 12 pages, 2015.
- [11] A. S. M. Z. Kausar, A. W. Reza, M. U. Saleh, and H. Ramiah, "Energizing wireless sensor networks by energy harvesting systems: scopes, challenges and approaches," *Renewable and Sustainable Energy Reviews*, vol. 38, pp. 973–989, 2014.
- [12] H. Nishimoto, Y. Kawahara, and T. Asami, "Prototype implementation of ambient RF energy harvesting wireless sensor networks," in *Proceedings of the IEEE Sensors*, pp. 1282–1287, Kona, Hawaii, USA, November 2010.
- [13] L. Xie, Y. Shi, Y. T. Hou, and A. Lou, "Wireless power transfer and applications to sensor networks," *IEEE Wireless Communications*, vol. 20, no. 4, pp. 140–145, 2013.
- [14] G. Papotto, F. Carrara, A. Finocchiaro, and G. Palmisano, "A 90-nm CMOS 5-Mbps crystal-less RF-powered transceiver for wireless sensor network nodes," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 2, pp. 335–346, 2014.
- [15] P. Nintanavongsa, M. Y. Naderi, and K. R. Chowdhury, "Medium access control protocol design for sensors powered by wireless energy transfer," in *Proceedings of the 32nd IEEE International Conference on Computer Communications (INFOCOM '13)*, pp. 150–154, IEEE, Turin, Italy, April 2013.
- [16] Y. Yao, X. Song, C. Yin, and S. Huang, "Opportunistic energy harvesting and energy-based opportunistic spectrum access in cognitive radio networks," in *Cognitive Radio Oriented Wireless Networks: 10th International Conference, CROWNCOM 2015, Doha, Qatar, April 21–23, 2015, Revised Selected Papers*, vol. 156 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pp. 187–198, Springer, Basel, Switzerland, 2015.
- [17] R. E. Ali, F. F. Digham, K. G. Seddik, M. Nafie, A. El-Keyi, and Z. Han, "A probabilistic MAC for cognitive radio systems with energy harvesting nodes," in *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP '14)*, pp. 143–147, IEEE, Atlanta, Ga, USA, December 2014.
- [18] V.-D. Nguyen, H. V. Nguyen, and O.-S. Shin, "Wireless energy harvesting for cognitive multihop wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2015, Article ID 613165, 9 pages, 2015.
- [19] W. Guibène and D. Slock, "Cooperative spectrum sensing and localization in cognitive radio systems using compressed sensing," *Journal of Sensors*, vol. 2013, Article ID 606413, 9 pages, 2013.
- [20] S. Homayouni and S. A. Ghorashi, "Sub-banding the secondary users' channel in cognitive radio networks considering unreliable spectrum sensing," *Wireless Personal Communications*, vol. 77, no. 2, pp. 1129–1143, 2014.
- [21] R. Al-Zubi, M. Hawa, G. Al-Sukkar, and K. A. Darabkh, "Markov-based distributed approach for mitigating self-coexistence problem in IEEE 802.22 WRANs," *The Computer Journal*, vol. 57, no. 12, pp. 1765–1775, 2014.
- [22] B. R. Franciscatto, V. Freitas, J. M. Duchamp, C. Defay, and T. P. Vuong, "High-efficiency rectifier circuit at 2.45 GHz for low-input power RF energy harvesting," in *Proceedings of the IEEE European Microwave Conference*, pp. 507–510, Nuremberg, Germany, October 2013.
- [23] S. Agrawal, S. K. Pandey, J. Singh, and M. S. Parihar, "Realization of efficient RF energy harvesting circuits employing different matching technique," in *Proceedings of the 15th International Symposium on Quality Electronic Design (ISQED '14)*, pp. 754–761, IEEE, Santa Clara, Calif, USA, March 2014.
- [24] M. Stoopman, S. Keyrouz, H. J. Visser, K. Philips, and W. A. Serdijn, "Co-design of a CMOS rectifier and small loop antenna for highly sensitive RF energy harvesters," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 3, pp. 622–634, 2014.
- [25] T. Le, K. Mayaram, and T. Fiez, "Efficient far-field radio frequency energy harvesting for passively powered sensor networks," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 5, pp. 1287–1302, 2008.
- [26] A. P. Sample, A. N. Parks, S. Southwood, and J. R. Smith, "Wireless ambient radio power," in *Wirelessly Powered Sensor Networks and Computational RFID*, J. R. Smith, Ed., pp. 223–234, Springer, New York, NY, USA, 2013.
- [27] S. Shrestha, S.-K. Noh, and D.-Y. Choi, "Comparative study of antenna designs for RF energy harvesting," *International Journal of Antennas and Propagation*, vol. 2013, Article ID 385260, 10 pages, 2013.
- [28] A. Aziz, A. Mutalib, and R. Othman, "Current developments of RF energy harvesting system for wireless sensor networks,"

- Advances in Information Sciences and Service Sciences*, vol. 5, no. 11, pp. 328–338, 2013.
- [29] X. Chen, X. Wang, and X. Chen, “Energy-efficient optimization for wireless information and power transfer in large-scale MIMO systems employing energy beamforming,” *IEEE Wireless Communications Letters*, vol. 2, no. 6, pp. 667–670, 2013.
- [30] S. Sayilir, W.-F. Loke, J. Lee et al., “A –90 dBm sensitivity wireless transceiver using VCO-PA-LNA-switch-modulator co-design for low power insect-based wireless sensor networks,” *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 996–1006, 2014.
- [31] X. Huang, A. Ba, P. Harpe, G. Dolmans, H. de Groot, and J. R. Long, “A 915 MHz, ultra-low power 2-tone transceiver with enhanced interference resilience,” *IEEE Journal of Solid-State Circuits*, vol. 47, no. 12, pp. 3197–3207, 2012.
- [32] H. Ju and R. Zhang, “Throughput maximization in wireless powered communication networks,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 1, pp. 418–428, 2014.
- [33] X. Kang, C. K. Ho, and S. Sun, “Full-duplex wireless-powered communication network with energy causality,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, pp. 5539–5551, 2015.
- [34] J. Kim and J.-W. Lee, “Energy adaptive MAC protocol for wireless sensor networks with RF energy transfer,” in *Proceedings of the 3rd International Conference on Ubiquitous and Future Networks (ICUFN ’11)*, pp. 89–94, Dalian, China, June 2011.
- [35] G. Yang, C. K. Ho, R. Zhang, and Y. L. Guan, “Throughput optimization for massive MIMO systems powered by wireless energy transfer,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 8, pp. 1640–1650, 2015.
- [36] K. Huang and E. Larsson, “Simultaneous information and power transfer for broadband wireless systems,” *IEEE Transactions on Signal Processing*, vol. 61, no. 23, pp. 5972–5986, 2013.
- [37] D. W. K. Ng, E. S. Lo, and R. Schober, “Energy-efficient resource allocation in multiuser OFDM systems with wireless information and power transfer,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’13)*, pp. 3823–3828, Shanghai, China, April 2013.
- [38] D. Niyato and P. Wang, “Competitive wireless energy transfer bidding: a game theoretic approach,” in *Proceedings of the IEEE International Conference on Communications (ICC ’14)*, pp. 1106–1111, IEEE, Sydney, Australia, June 2014.
- [39] D. T. Hoang, D. Niyato, and D. I. Kim, “Cooperative bidding of data transmission and wireless energy transfer,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’14)*, pp. 1597–1602, IEEE, Istanbul, Turkey, April 2014.
- [40] D. Niyato, P. Wang, H. Tan, W. Saad, and D. I. Kim, “Cooperation in delay tolerant networks with wireless energy transfer: performance analysis and optimization,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 8, pp. 3740–3754, 2015.
- [41] V. Sharma, U. Mukherji, V. Joseph, and S. Gupta, “Optimal energy management policies for energy harvesting sensor nodes,” *IEEE Transactions on Wireless Communications*, vol. 9, no. 4, pp. 1326–1336, 2010.
- [42] M. Gatzianas, L. Georgiadis, and L. Tassiulas, “Control of wireless networks with rechargeable batteries,” *IEEE Transactions on Wireless Communications*, vol. 9, no. 2, pp. 581–593, 2010.
- [43] X. Lu, P. Wang, D. Niyato, and E. Hossain, “Dynamic spectrum access in cognitive radio networks with RF energy harvesting,” *IEEE Wireless Communications*, vol. 21, no. 3, pp. 102–110, 2014.
- [44] S. Park, J. Heo, B. Kim, W. Chung, H. Wang, and D. Hong, “Optimal mode selection for cognitive radio sensor networks with RF energy harvesting,” in *Proceedings of the 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC ’12)*, pp. 2155–2159, IEEE, Sydney, Australia, September 2012.
- [45] L. Gavrilovska, D. Denkovski, V. Rakovic, and M. Angjelichinoski, “Medium access control protocols in cognitive radio networks: overview and general classification,” *IEEE Communications Surveys and Tutorials*, vol. 16, no. 4, pp. 2092–2124, 2014.
- [46] Z. A. Eu, H.-P. Tan, and W. K. G. Seah, “Design and performance analysis of MAC schemes for wireless sensor networks powered by ambient energy harvesting,” *Ad Hoc Networks*, vol. 9, no. 3, pp. 300–323, 2011.
- [47] IEEE, “IEEE standard for local and metropolitan area networks part 11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications,” IEEE Standard IEEE 802.11-2012, 2012.
- [48] IEEE, “IEEE standard for local and metropolitan area networks part 15.4: low-rate wireless personal area networks (LR-WPANs),” IEEE Standard 802.15.4-2011, 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

