IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received September 1, 2019, accepted September 10, 2019,
date of publication September 23, 2019, date of current version October 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2943011

Online Congestion Measurement and Control in Cognitive Wireless Sensor Networks

SHUGUO ZHUO[1], HOSSEIN SHOKRI-GHADIKOLAEI[2], (Member, IEEE), CARLO FISCHIONE[2], (Member, IEEE), AND ZHI WANG[1], (Member, IEEE)

[1]State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 311121, China
[2]Division of Network and Systems Engineering, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 10044 Stockholm, Sweden

Corresponding author: Zhi Wang (wangzhi@iipc.zju.edu.cn)

This work was supported in part by the Key Projects of Intergovernmental International Scientific and Technological Innovation Cooperation under Grant 2017YFE0101300, in part by the National Natural Science Foundation of China under Grant 6177020565 and Grant U1509215, in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LZ19F010003, in part by the Open Research Project of the State Key Laboratory of Industrial Control Technology under Grant ICT1900322, Grant ICT1900353, Grant ICT1900363, and Grant ICT1900364, and in part by the State Key Laboratory of Robotics under Grant 2017O-05.

ABSTRACT A lightweight distributed MAC protocol is proposed in this paper to regulate the coexistence of high-priority (primary) and low-priority (secondary) wireless devices in cognitive wireless sensor networks. The protocol leverages the available spectrum resources while guaranteeing stringent quality of service requirements. By sensing the congestion level of the channel with local measurements and without any message exchange, a novel adaptive congestion control protocol is developed by which every device independently decides whether it should continue operating on a channel, or vacate it in case of saturation. The proposed protocol dynamically changes the congestion level based on variations of the non-stationary network. The protocol also determines the optimal number of active secondary devices needed to maximize the channel utilization without sacrificing latency requirements of the primary devices. This protocol has almost no signaling and computational overheads and can be directly implemented on top of existing wireless protocols without any hardware/equipment modification. Experimental results show substantial performance enhancement compared to the existing protocols and provide useful insights on low-complexity distributed adaptive MAC mechanism in cognitive wireless sensor networks.

INDEX TERMS Congestion control, CSMA, channel utilization, latency, optimization, cognitive wireless sensor networks.

I. INTRODUCTION

To address the increased demand of spectrum for wireless communication services, the cognitive radio (CR) technology has been proposed and continuously studied in the recent decade to substantially increase the spectrum utilization [1], [2]. Inspired by this technology, the concept of cognitive wireless sensor networks (C-WSNs) has emerged as a promising solution to enhance communication performance in wireless systems and internet of things (IoT) [3]–[6]. For instances, in many WSNs/IoT applications like environment monitoring, smart building or vehicle-to-infrastructure communications, C-WSNs improves WSNs by allowing low-priority devices (e.g., secondary devices) to coexist with high-priority devices (e.g., primary devices) on a wireless channel which is originally dedicated to the latter. In this way, C-WSNs enhance the utilization of the dedicated channel, called as primary channel, and provide more stable communication links to the secondary devices, improving the performance of the whole system. However, although having a higher number of transmitters (or equivalently a higher data rate per transmitter) may enhance the channel utilization, C-WSNs also need to guarantee the high quality of service of the primary devices, which must not be violated by extra communication interferences due to the secondary devices [7], [8].

Spectrum sensing is a vital function to realize coexistence of the primary and secondary devices in C-WSNs. Spectrum sensing provides an up-to-date picture of the spectrum usage. If the channel is detected as congested by many active

The associate editor coordinating the review of this manuscript and approving it for publication was Xianfu Lei.

devices or primary devices activities are detected, secondary devices may trigger channel handoff and leave the primary channel, so as to reduce the latency of the primary devices for accessing the channel. However, since spectrum sensing occupies radio and computational resources, communications of the secondary devices may be periodically interrupted in many cognitive systems. This results in significant reduction in the transmission reliability and energy efficiency of the cognitive networks. How to jointly smooth the transmissions of secondary devices (e.g., reducing unnecessary spectrum sensing and channel handoffs), increase the utilization of the channel, and meet strict latency requirements of the primary devices remains as an important research topic in C-WSNs.

A. LITERATURE REVIEW

With the evolutions of WSNs and IoT, new techniques have been continuously proposed to enhance the throughput and efficiency of communication protocols for sensor networks [9]–[13] and the CR technology is among the most promising directions. In the literature, to realize spectrum sharing in cognitive radio networks (CRNs) and C-WSNs, secondary devices should periodically sense the primary channel, and they are allowed to utilize the channel only when the primary device activities are absent [14]–[21]. HC-MAC [22] is a typical contention-based MAC protocol for cognitive networks that models the sensing process as an optimal stopping problem for determining how long a secondary device should stay in the sensing stage to optimize its expected throughput.

In [23], a cognitive adaptive MAC (CAMAC) protocol is proposed to have on-demand spectrum sensing and adaptive duty-cycling capabilities. Similar to [22], CAMAC adopts an adaptive sensing step which utilizes a variable sensing period adjusted between the limits of ‘fast’ and ‘fine sensing’ periods. For a sensing SU, the sensing period duration increases upon failed transmissions and decreases upon successful transmissions. The protocol also reduces energy consumption by sharing one SU’s sensing results to nearby non-sensing sensors, by utilizing prior location knowledge (the devices are all equipped with GPS). In [24], the authors propose a MAC (medium access control) scheme for clustered C-WSNs in which sensor nodes send their individual spectrum sensing results to the cluster head to make a combined decision. With energy consumption analysis, the cluster will decide to access the licensed channel for data transmission in case packet loss rate over the license-free channel increases. In [25], SD-MAC is proposed for spectrum database-driven cognitive machine-to-machine networks (CM2M) in which secondary devices adaptively select an ASM (available spectrum map) or a local sensing scheme for devices to obtain spectrum information. In [26], DynMAC is proposed based on GinMAC for WSNs and uses CR techniques to select the best channel for the network. The sink of the network collects son nodes’ alert information and decides to re-evaluate a global best channel when needed, while the channel searching and re-evaluation mechanism lead to a certain recovery time, which in turn delays the transmission. In [27], a spectrum sharing scheme based on the IEEE 802.15.4 superframe structure is proposed to regulate coexistence of primary and secondary devices. A coordinator device periodically carries out spectrum sensing in the sleeping period to identify transmission opportunities for the secondary devices. Some works propose using multi radios to aid spectrum sensing for secondary devices, like in [14], [28]. Reference [14] proposed a contention-based MAC protocol in which secondary devices contend in a synchronized control channel for picking a data channel by using a scheme like in slotted-ALOHA. The system requires a specific sensing device that carries out spectrum sensing, detects the absence of primary devices and broadcasts the sensing results to secondary devices through beacons. In both [14] and [28], secondary devices are equipped with two radios, one for channel management on the control channel and another for data transmission.

In a nutshell, although the sensing accuracy and efficiency may have been greatly enhanced, it should be noted that most of the above mentioned solutions still provide rather preemptive protections for the primary devices that may unnecessarily decrease the performance of the secondary devices. Specifically, they ask secondary devices to vacate the channel once primary device activities are detected even if the channel is unsaturated,[1] communications of secondary devices get frequently broken, leading to a limited throughput for the secondary devices. Like typically in [24], [25], the primary user (PU) traffic is modeled as a ON/OFF random process in which the ON state indicates that channel is occupied by PUs and the OFF state implies that the channel is idle. The ON/OFF model can only tell the appearance of PU, while it fails to describe the channel utilization and congestion level and to tell whether the channel is saturated or not. When the channel is far from saturation, secondary devices should be continuously allowed to utilize more potential bandwidth without breaking down their communications. To this end, there is still room for improvement.

To smooth secondary devices communication with fewer unnecessary spectrum handoffs, some solutions start to allow secondary devices to coexist with primary devices on the primary channel simultaneously. In [29], a channel access approach based on the carrier sense multiple access (CSMA) protocol is proposed that enables channel access with different priorities. Every secondary device uses a longer idle sensing period than primary devices before transmission. This gives higher priorities to the primary one for accessing the channel. The cost is an unnecessarily large latency for the secondary devices even under light traffic demands of the primary devices. References [30] and [31] develop mechanisms that ask the secondary devices to vacate the channel only if they cannot coexist with the primary devices, to reduce unnecessary channel handoffs. However, the secondary devices need to conduct a proactive spectrum sensing

[1]Saturation level of a channel can be defined based on its PU QoS requirement, as formally defined in Section III-B.

scheme by periodically sending a sounding signal, which consumes extra energy. Generally, spectrum sensing, that requires frequent and complex radio operations, is a heavy and energy consuming task, which is not in harmony with low energy battery-powered IoT devices. Besides, when the payloads are small, spectrum sensing duration may be comparable with the packet transmission time, and this huge sensing overhead decreases the efficiency of CRNs, as studied in [32].

To simplify radio operations in spectrum sensing and smoothen secondary devices communication, reference [33] proposed a MAC protocol for single-hop cellular network that detects interference at the MAC layer, aiming to ensure target latency and reliability requirements. Specifically, the network coordinator runs a Markov Process to estimate and rank the availability of each channel, and sends the feedback to all nodes in each cycle. By using error information of the MAC layer (e.g., lack of Data and ACKs in determined slots) for signaling an interference event, the network can turn to another channel (predefined by the Markov Process) to re-establish the connectivity in a period of time equal to the cycle frame length. In this paper, we show that CSMA, a contention-based channel access scheme that is employed in most existing standards for IoT, provides enough information about the channel availability. This information can be used to reduce the need for heavy spectrum sensing tasks. As a part of the popular CSMA mechanism, a device senses the spectrum occupancy level by measuring the backoff index (also called backoff numbers). That is, since the backoff index usually has high value when the channel is congested and has low value while the channel is unsaturated, this CSMA parameter reflects the congestion condition at a certain level. Thus, with careful calculations, the current channel congestion level can be quantified by using the CSMA backoff index instances, which are continuously produced along with the ongoing CSMA transmission procedure, at no extra cost. By utilizing the backoff index for congestion assessment, transmission and channel availability detection can be combined and carried out simultaneously without the need for any extra sensing period, which is much more efficient than having a dedicated spectrum sensing period. Moreover, a fast on-line congestion detection enables secondary devices to share the channel(s) with the primary devices as long as a desired network congestion level (and consequently latency and throughput of the primary devices) is met. This motivated our work in this paper for enhancing the spectrum utilization based on the CSMA congestion detection.

B. OUR CONTRIBUTIONS

In this paper, we propose a novel CSMA-based adaptive and distributed congestion control (ACC-CSMA) MAC protocol for significantly improving the quality of service in terms of transmission opportunities and latency for secondary devices in the network by assigning these devices to the most reliable channel (the primary channel) in C-WSNs. Generally, to guarantee the high performance of the primary devices, every secondary device in ACC-CSMA independently utilizes its local CSMA backoff numbers that are generated along with packet transmissions to measure/quantify the congestion level online. Once a high congestion level is detected by the online congestion measurement, each secondary device individually vacates the primary channel to reduce the congestion level. When the traffic demands decrease on the primary channel, ACC-CSMA can sense this change and insert secondary devices back onto the channel, so as to fully utilize the precious channel resources. Notably, this paper extends the work in [34] by including a duty-cycle scheme for conserving power and a stochastic subgradient-based optimization to guarantee maximum channel utilization under latency constraints of the primary devices with some noisy measurements. We also show how the optimality can be preserved with dynamic and bursty traffic demands of the primary devices in non-stationary wireless environments.

Compared to the literature mentioned above and to the best of our knowledge, this is the first paper to 1) use CSMA backoff index for conducting online congestion measurement for C-WSNs, 2) develop a simple yet efficient CSMA-based protocol capable of maximizing the channel utilization with latency guarantees for primary devices in non-stationary environments, 3) implement the proposed protocol on the real-world STM32W108 chips that offer IEEE 802.15.4 standard communications and validate the protocol with extensive experiments, and 4) show that both primary and secondary networks benefit from some level of controlled interference in C-WSNs, as intuited in other wireless domains, such as in [35].

It must be pointed out that, the proposed ACC-CSMA protocol provides a statistical guarantee on the latency of the primary devices. In general, when applying a contention-based wireless technology (for having its advantages of no global synchronization, low complexity and easy implementation), it is impossible to provide absolute latency guarantees due to randomness in the wireless channel and traffics of the devices.

C. PAPER STRUCTURE

The remainder of this paper is organized as follows. Section II illustrates the considered system model. Section III describes the proposed ACC-CSMA protocol and we show the feasibility verification of ACC-CSMA in section IV. In Section V, we optimize the performance of ACC-CSMA. Section VI presents the performance evaluation and the paper is concluded in Section VII.

II. SYSTEM MODEL

As a premier attempt to carry out congestion detection and spectrum sharing based on CSMA in C-WSNs, in this paper, we first consider a single hop, cellular wireless network, and leave the multi-hop network for a future study. Fig. 1 illustrates the considered model in which there is a sink (coordinator) that is responsible for monitoring local system and for collecting data packets from all devices in its communication range. We assume that all devices use the

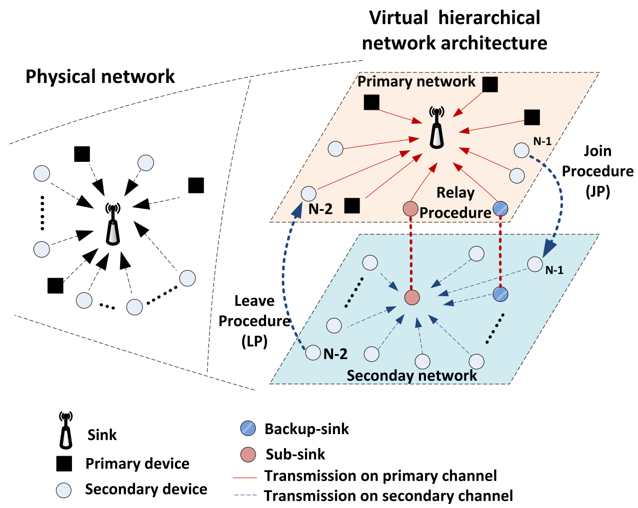


FIGURE 1. Illustration of a physical single hop network and its virtual two-hierarchy primary and secondary networks.

well-known CSMA mechanism to send their packets to the sink node for monitoring and control tasks. Without loss of generality, we consider a common scenario wherein all the devices, including the network sink, are equipped with only one transceiver. Therefore, the protocol we propose can be directly applied in most of the existing applications without any hardware/equipment modification. Depending on the sensitivity/importance of their data, devices are labeled as primary and secondary devices, coexisting in the same network, as shown in left side of Fig. 1. For instance, in the condition monitoring applications, gas detection sensors can tolerate latency of the order of milliseconds which are normally considered as primary devices, whereas the temperature sensors can tolerate latency of the order of seconds which are considered as secondary devices [36].

We consider two different types of wireless channels: one primary and a group of secondary channels. The primary channel is a channel with guaranteed interference level, e.g., licensed channel or spectrum band that is commonly not free for accessing; whereas the secondary channels are non-licensed channels that may probably have high interference,[2] and are considered as backup channels, e.g., the free 2.4 GHz band. Both the primary and secondary channels are used as data channels for data transmissions. Also, similar to most existing works, there is a control channel in the system. In our design, the control channel is mainly used for transmissions of downward command/configuration packets to secondary devices, and based on which, the secondary devices carry out the duty-cycle scheme to conserve power. Note that the developed protocol can be a baseline framework for a C-WSNs with multiple primary channels. In that scenario, we should apply the proposed protocol on an ordered set of the primary channels. In the implementation and experiments of this paper, for the convenience of evaluation and without loss of generality, we use one of the clean IEEE 802.15.4 standard defined channels to simulate the primary channel, and use the others to simulate secondary channels and the control channel. Also, we assume that all devices contend to access the channel with the un-slotted and non-persistent CSMA mechanism.

[2]Besides co-channel interference, the secondary channels may be subject to inter-system interference in multi-vendor sites.

III. THE ACC-CSMA PROTOCOL

The main goal of this paper is to develop an efficient protocol that maximizes the primary channel utilization, while guaranteeing high QoS levels for the primary devices. The protocol should have a good performance in dynamic environment of wireless networks, due to variable sources of interference, dynamic traffic demands of the primary devices, and fluctuations of the wireless channel. Moreover, we are interested in implementing a distributed algorithm with light computational and signaling overhead and without the need for a global synchronization.

A. GENERAL DESCRIPTION OF THE ACC-CSMA PROTOCOL

We divide the original physical network (left side of Fig. 1) into two virtual networks (right side of Fig. 1):

- *primary network*, composed of devices (both primary and secondary devices) operating on the primary channel; and
- *secondary network*, composed of only secondary devices that are working only on the secondary channels.

The secondary network has a sink and a backup sink, called sub-sink and backup-sink, respectively. The sub-sink and the backup-sink collect the packets from the secondary network and relay them to the sink on the primary channel, respectively. Generally, as shown in Fig. 1, to guarantee the performance of the primary devices, the proposed ACC-CSMA protocol monitors/quantifies the congestion level and maintains it by asking the secondary devices to join the secondary network (join procedure, JP) once the primary network is congested. It also fully utilizes available channel bandwidth by inserting secondary devices back to the primary channel (leave procedure, LP) once the traffic demands on the primary network are decreased.

B. ON-LINE CONGESTION MEASUREMENT

The ultimate goal of this paper is to leverage the precious primary channel by allowing as many devices (both primary and secondary) on such a channel as possible, while meeting QoS demands of the primary devices. To this end, the proposed ACC-CSMA protocol allows secondary devices to utilize the primary channel for communication even after primary devices have shown up. Fig. 2 illustrates the basic channel access scheme of secondary devices in ACC-CSMA. When the network is in a low congestion phase (we will show later how the congestion level is measured and indicated to the secondary devices), secondary devices keep their transmissions and do not vacate the primary channel even after primary devices' activities have emerged, thus reducing unnecessary channel switches. On the other hand, increasing the traffic or the number of the devices in the

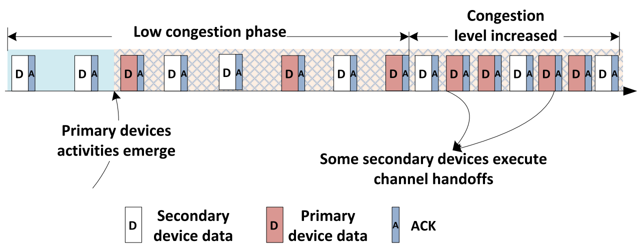


FIGURE 2. Access scheme for secondary devices on the primary channel.

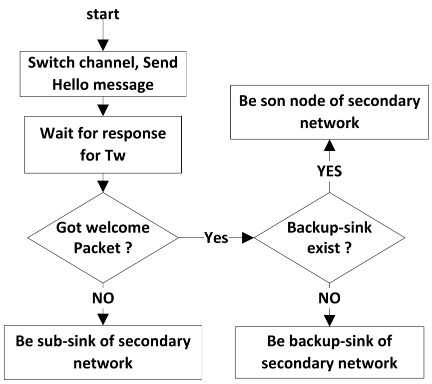


FIGURE 3. Procedure of joining the secondary network used by each secondary device of the primary network.

primary network increases its congestion level, which may decrease the reliability of the communications of the primary devices and increase their latency. In this case, to minimize the harmful interference to the primary devices and thereby guarantee their QoS demands, the secondary devices should measure the congestion level on the primary channel in real-time and vacate the channel before the congestion level can cause harmful interferences to the latency-sensitive primary devices, as shown in the right side of Fig. 2. In ACC-CSMA, we propose a simple yet effective mechanism that utilizes CSMA backoff index to achieve a fast, reliable, and on-line congestion assessment.

The backoff index b is an important parameter in CSMA that represents the number of backoff executions. More importantly, it is common in CSMA-based communication systems that the backoff index b usually has high values for congested channels and low values for unsaturated ones (as shown in Fig. 3 of [34]). Therefore, the backoff index reflects the channel congestion level, and it can be used as an on-line measure of the congestion level. However, due to the randomness in the traffic of the devices, the backoff index is a random variable. There is a possibility that a sequence of random backoffs occasionally meets a relatively long message, resulting in a temporary high b value while the channel is actually underutilized. Hence, the instantaneous b is not a reliable measure of the channel congestion level, and a more reliable measure should be adopted to filter out false alarms and accurately assess the channel congestion.

Let $\{b_{(i)}\}_{i=1}^{L}$ be the latest L values of backoff indices. To reliably quantify the congestion level, each secondary device independently records $\{b_{(i)}\}_{i=1}^{L}$ from its recent transmission attempts and calculates the congestion indicator η of the primary channel using

$$\eta = \frac{1}{L}\sum_{i=1}^{L} b_{(i)} \approx \mathbb{E}\{b\}\,, \tag{1}$$

where $\mathbb{E}\{b\}$ is the mean value of the backoff indices that are extracted from tests under a given congestion level. As shown in (1), the congestion indicator η is an estimation of $\mathbb{E}\{b\}$ with L instances. Notably, the value of L should be carefully chosen, since a) with too small L, the estimated congestion indicator η will be strongly affected by the random nature of b; and b) with too large L, the calculated η heavily depends on the historical information that may not be valid for the current congestion level of the channel (outdated information), given dynamics of the primary network and the traffic of the secondary devices. In this paper, the length L is selected based on extensive experimental observations in Section IV and VI.[3] In short, by using the backoff indices recorded from transmission attempts, a secondary device continuously measures the congestion level when it is transmitting on the primary channel.

C. ADAPTIVE CONGESTION CONTROL

Given the on-line congestion measurement described above, the proposed ACC-CSMA protocol continuously runs two procedures (JP and LP) to maintain a proper congestion level on the primary channel and allow secondary devices to fully utilize the precious communication resources.

ACC-CSMA defines an unique congestion threshold λ_c in the primary network, which is maintained by the sink and informed/updated to secondary devices by embedding the current λ_c value into every ACK packet that is replied to secondary devices upon each data transmission. When transmitting on the primary channel, each secondary device independently and continuously measures the congestion level by using the backoff indices of its own packets. In particular, after each transmission attempt, the secondary device calculates a new congestion indicator η and compares it to λ_c. If $\eta \leq \lambda_c$, the secondary device regards the primary channel as unsaturated and stays in the primary network. If $\eta > \lambda_c$, the secondary device considers that the primary channel is already congested, and it immediately vacates the primary channel and triggers the join procedure (JP), so as to join the secondary network and reduce the congestion level.

Fig. 3 illustrates JP. First, a secondary device tunes its radio to the secondary channel and broadcasts a `Hello` message for announcing a join request. Then, it waits for a `Welcome` message from the sub-sink, shown in Fig. 1, for a predefined period of time. If the `Hello` message expires, the secondary

[3] We have specifically explored the influence of L on the protocol's performance (e.g., network throughput) with real-world experiments and found that, when L is set to 5, it yields the best performance for ACC-CSMA. Due to space limit, detailed experimental results and discussions about the selection of L are omitted here.

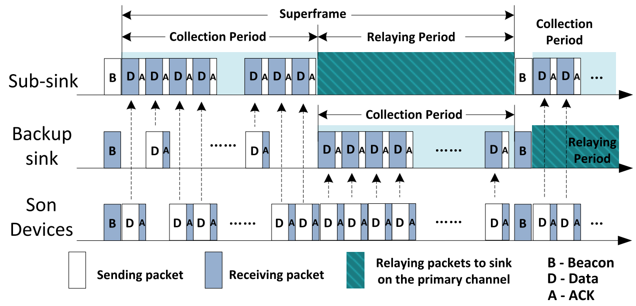


FIGURE 4. **Superframe structure of the secondary network. The sub-sink and the backup-sink alternately collect packets of the son devices and relay them to the primary sink.**

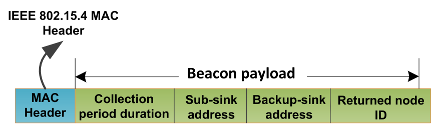


FIGURE 5. **Packet structure of the secondary network beacon.**

device considers itself as the first one joining this secondary channel, and turns to be the sub-sink itself. However, upon receiving the `Welcome` message, the secondary device operates as either a backup-sink or a son node of the secondary network, as described in Fig. 3. By this procedure, normally the first two secondary devices that join the secondary channel will be the sub-sink and backup-sink, while the following secondary devices that join the secondary network will be son nodes. Notice that the procedure can be easily modified to provide fairness in assigning backup-sink role to all devices inside the secondary network. Notably, each secondary device individually executes congestion detection and JP based on its own observation, i.e., in a distributed manner.

We define a superframe for the secondary network, synchronized among all the devices in this network using beacon signals of the sub-sink, consisting of two periods: collection and relaying, as illustrated in Fig. 4. During the collection period (its duration is indicated in the beacon, as illustrated in Fig. 5), all the son devices, including the backup-sink, transmit their packets to the sub-sink. During the relay period, the sub-sink switches to the primary channel to relay all the collected packets (including its own packets) to the sink in the primary network. When the sub-sink is relaying packets on the primary channel (i.e., out of the secondary network), the backup-sink acts as the temporary sink node in the secondary network, and it is responsible for collecting packets from the son devices, as illustrated in Fig. 4. The sub-sink switches back to the secondary channel after the relaying procedure and starts a new superframe to collect the packets. Meanwhile, the backup-sink switches to the primary channel to relay its collected packets, like what sub-sink did. In other words, the sub-sink and backup-sink, alternately, collect and relay the packets of the secondary network to the sink in the primary network. As the sub-sink or backup-sink may be responsible to relay massive packets of many secondary devices, they may block the primary channel, which substantially decrease the QoS of the primary devices. To avoid this problem, ACC-CSMA gives lower priority to the sub-sink and back-up sink compared to other devices in the primary network. Specifically, the sub-sink and backup-sink use larger CSMA backoff exponent for each individual packet transmission when transmitting on the primary channel. That is, after each single packet transmission on the primary channel, either the sub-sink or backup-sink will drop the medium occupation immediately and uses CSMA again (with a larger backoff exponent) to compete for the transmission of the next relayed packet. In other words, the sub-sink and backup-sink will not keep the medium for a long time. Instead, they relay the collected packets in many single trials.

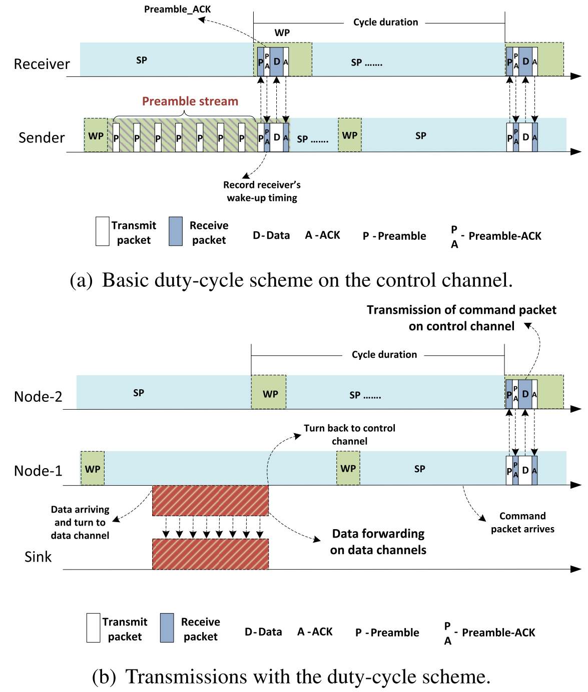
When the congestion level drops on the primary channel due to the decreased traffic demands, the sub-sink can recognize this change through the congestion indicator when it is relaying packets in the primary network. If a consecutive sequence of low η values is observed, the sub-sink considers that the primary channel is under-utilized and triggers the leave procedure (LP) to return some secondary devices back to the primary channel, so as to fully utilize the precious communication resources. This is done by using sub-sink's beacon to insert *only one* secondary device to the primary channel in each superframe (the beacon contains the ID of the node that will be returned to primary network). This avoids the ping-pong effect, i.e., returning all devices from the secondary network may introduce an overwhelming traffic to the primary channel, which, in return, causes an impeded channel for the primary devices, triggering JP again.

In short, the proposed ACC-CSMA protocol maintains congestion level λ_c in the primary channel using on-line congestion measurement and on-demand executions of the JP and LP. Clearly, the threshold λ_c has a great impact on the performance of the proposed protocol. If λ_c is set to have a too small value, secondary devices will be easily expelled from the primary channel, resulting in a low channel utilization. While if λ_c is set to have a too large value, the targeted quality of service of the primary devices may be violated due to both channel saturation and strong interference. Moreover, dynamics of the primary network requires adaptation of λ_c over time. In Section V, we develop a low overhead solution to dynamically adjusts λ_c.

D. DUTY-CYCLE OPERATION

As noted earlier, ACC-CSMA uses an extra control channel to exchange downward control packets (possible network configuration commands) between secondary devices (including communication between sink and secondary devices). When there is no data packets to forward, secondary devices switch to the control channel to periodically listen for possible coming packets. To extend the system lifetime, ACC-CSMA adopts a duty-cycle scheme on the control channel to conserve power.

Fig. 6(a) illustrates ACC-CSMA's duty-cycle scheme. In short, the duty-cycle scheme is like in X-MAC [37], which



(a) Basic duty-cycle scheme on the control channel.

(b) Transmissions with the duty-cycle scheme.

FIGURE 6. Duty scheme of ACC-CSMA.

is based on phase-lock manipulations. The scheme doesn't require global synchronization and each secondary device maintains its own superframe structure with different wake-up period (WP) timing phase on the control channel. On the control channel, all the secondary devices adopt the same cycle duration. When a node wants to send a control packet to another one for the very first time, it sends out a preamble stream to probe the receiver's wake-up phase. After phase-locked (recording the receiver's wake-up timing), when the next time the sender tries to send control packet to the same receiver, it just wait until the receiver is in its WP period, as shown in Fig. 6(a). The broadcast procedure on the control channel is similar to the preamble procedure, except that each broadcast will fully last for one cycle duration, which is popular in many duty-cycled WSNs MAC protocols.

Fig. 6(b) shows how ACC-CSMA's duty-cycle scheme works with data transmissions on both data and control channels. As noted earlier, when there is no data packets to send, secondary devices stay on the control channel and wake up periodically. While when data packets arrive, secondary devices individually turn to the data channel to execute ACC-CSMA's data forwarding scheme as illustrated in Section III-A to Section III-C. Note that, when transmitting on the data channel, a secondary device will not adopt any duty-cycle scheme and only turns back to the control channel when the data transmission is finished. In case secondary devices are in the secondary network for data transmissions (as illustrated in Section III-C), they will only return to the control channel when they have left the secondary network (be re-inserted onto the primary channel). Any transmission attempt to reach the secondary device (while it is on the data channel) on the control channel will fail, and will lead

TABLE 1. Key parameters of ACC-CSMA.

λ_c	L	BE-min	BE-max	BE-min-RP	BE-max-RP
8	5	2	5	5	8

to retrials with backoffs with a random number of cycle durations.

IV. FEASIBILITY VERIFICATION OF ACC-CSMA

As a premier attempt to carry out online congestion control based on the CSMA backoff index in C-WSNs, careful investigations are required to explore the feasibility of such a manipulation. Most importantly, we are interested in observing how quickly the proposed scheme can adapt to environmental changes and whether the performance of primary device can be well guaranteed. To this end, before optimizing its performance, we first implemented a prototype of ACC-CSMA on the real-world IEEE 802.15.4 standard STM32W108 chips to experimentally explore its feasibility.

A real-world test-bed containing 31 devices has been set up. In the experimental verification, we use one primary and one secondary channel. For the convenience of evaluation and without loss of generality, we use one of the IEEE 802.15.4 standard defined channels to simulate the primary channel, and use the others to simulate secondary channels and the control channel (while in a more realistic scenario, the primary channel could be a truly dedicated channel outside the ISM band). Key parameters of the algorithm that are adopted in the implementation are given in Table 1. The congestion threshold λ_c and the length L of the congestion estimation array are set to fixed values of 8 and 5 in this experiment. Note that λ_c is a constant value in this section. In the next section, we show how to optimize λ_c to meet the latency requirements of the primary devices in non-stationary environments. For communications on the same channel (on primary channel or secondary channel), key parameters of the CSMA mechanism, which are the minimum backoff exponent (BE-min) and the maximum backoff exponent (BE-max), are set to 2 and 5 respectively. Note that the backoff exponent BE is a value that varies between BE-min and BE-max that determines the size of the contention window, whereas backoff index b is the number of backoff times which can be larger than BE. During the relaying period on the primary channel, the sub-sink and the backup-sink of the secondary channel adopt higher backoff exponent parameters, where the minimum-backoff exponent of the relaying period (BE-min-RP) is set to 5, and the maximum backoff exponent (BE-max-RP) is set to 8, aiming to give higher access priority to primary devices. Data packets are generated by a Poisson distribution, and each generated data packet contains 120 bytes for all the devices.

Bursty traffic is among the most critical situations for most C-WSNs. So, we firstly conducted an experiment to evaluate the effectiveness of ACC-CSMA over bursty traffic conditions. We established a cellular testbed in an office room environment with 1 sink, 5 primary devices, and 25 secondary

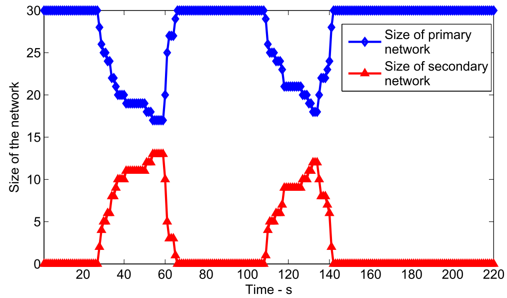


FIGURE 7. Variation of network size in the verification experiment with bursty data traffic.

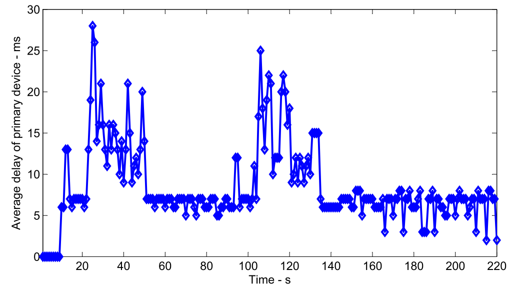


FIGURE 8. Average delay of primary device in the verification experiment with bursty data traffic.

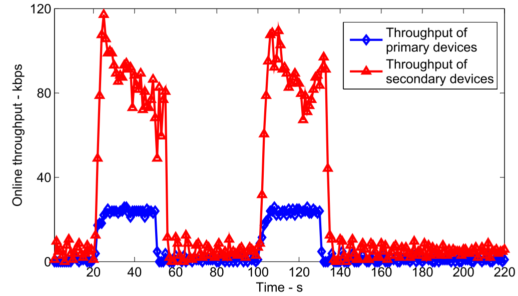
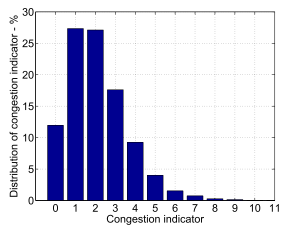
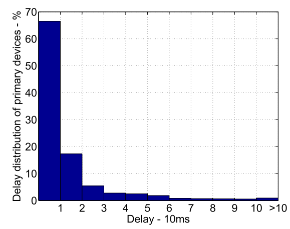


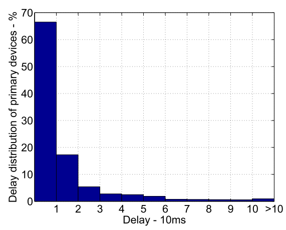
FIGURE 9. Throughput of devices (on the primary channel) in the verification experiment with bursty data traffic.



(a) η_s distribution.



(b) b distribution of primary devices.



(c) Primary devices delay distribution.

FIGURE 10. Distributions of key parameters in the bursty verification experiment.

devices, all in one-hop transmission range of each other. During non-bursty periods, all the devices adopt a low data rate of 1 packet per 5 seconds (0.192 kbps per device), while two bursty traffics are introduced at 20s and 100s respectively and each lasts for 20 seconds. During the bursty periods, a much higher data rate of 5 packets per second (4.8 kbps per device) is adopted by all the devices.

Fig. 7 shows the size variations of the primary network and the secondary network. With the adaptive congestion control of ACC-CSMA, the size of the primary network decreases at the beginning of the bursty period while the size of the secondary network increases. When the primary network is getting increasingly congested due to the bursty traffic loads, high congestion indicator η will be observed by secondary devices which triggers the JP operation to ask secondary devices to vacate the primary channel and join the secondary channel. This operation decreases the size of the primary network and increases size of the secondary network, as can be observed in Fig. 7. After the bursty periods, the primary channels becomes underutilized. The sub-sink of secondary network senses this situation and triggers the LP operation to insert some secondary devices back to the primary network so as to improve primary channel utilization. Consequently, the size of the primary network increases and and the size of the secondary network decreases.

Fig. 8 shows the average packet delay of primary devices. On the one hand, due to the bursty traffic loads, a fluctuation of latency can be observed for the primary devices. On the other hand, it can be noticed that the latency of primary devices can be maintained on a relatively low level ($< 30ms$) over the whole experiment and the fluctuation is just in a minimal scale. This is due to that the algorithm has maintained an acceptable congestion level on the primary channel to guarantee the high quality of service level of primary devices.

Fig. 9 shows the combined throughput of all primary devices and all secondary devices on the primary channel. Notice that we are not comparing their throughput in this figure (since it is meaningless) but showing the utilization gain (due to secondary devices' transmissions) of the primary channel due to the flexible access scheme. From this figure, although the throughput of secondary devices has been decreased during the burst periods, it is still maintained on a relatively high level. In other words, the proposed protocol not only guarantees the performance of primary devices, it can also provide a high throughput for the secondary devices without totally closing the channel for them.

To have a network-wide picture of the contention level, we let the sink of the system to calculate the congestion indicator of the primary channel, η_s, by using the piggybacked b from received packets, similar to the congestion indicator η calculated by secondary devices in (1). Fig. 10(a) depicts the distribution of the congestion indicators. From Fig. 10(a),

η_s indicator is less than 5 with probability 93% and less than 8 with probability 99.5%. These results imply that the proposed algorithm can maintain the congestion on an acceptable level. Fig. 10(b) shows the distribution of the backoff indices (b) of the primary devices during the bursty periods. From Fig. 10(b), the backoff index of primary devices b is 0, so these transmissions are successful at the first CSMA attempt, with a probability more than 40%. Moreover, b is less than 8 with probability 94%, indicating the high performance of primary devices for channel access. Finally, Fig. 10(c) depicts the distribution of the packet delays of the primary devices during the bursty periods. We can see that over 70% of the packet delays are lower than 10ms, and almost 95% of the packet delays are within 50ms.

So far, we have shown feasibility and effectiveness of realizing C-WSNs using backoff index of the CSMA protocol in the presence of bustry traffics and non-stationary environment. The latency can be very small, though some spontaneous large delays in the communications of the primary devices are observed. In the next section, we optimize the proposed protocol to provide statistical guarantee on the latency performance of the primary devices.

V. OPTIMIZING THE PROPOSED PROTOCOL

A. PROBLEM FORMULATION

The main goal of the ACC-CSMA is to maximize the primary channel utilization while maintaining predefined QoS levels for the primary devices. The value of λ_c directly affects the tradeoff between higher primary channel utilization and lower latency of the primary devices. In the following, we develop an optimization framework to maximize this tradeoff by choosing a proper value for λ_c.

Let $f_1(\lambda_c)$ and $f_2(\lambda_c)$ denote the primary channel utilization and average latency to successfully transmit a message of the primary devices for a given threshold λ_c. Here, we define the channel utilization as the proportion of the feasible primary channel bandwidth that is finally used for transmissions. The channel utilization has a great impact on the transmission reliability of the network since it determines the probability that a certain amount of messages are successfully delivered in a time unit, which is reflected in the throughput. Let $d_{\max}$ be the maximum tolerable delay/latency for the primary devices. Now, we can formulate the following optimization problem:

$$\underset{\lambda_c}{\text{minimize}} \quad -f_1(\lambda_c), \tag{2a}$$

$$\text{subject to } f_2(\lambda_c) \le d_{\max}, \tag{2b}$$

$$\lambda_c \ge 0. \tag{2c}$$

Remark 1: The functions of optimization problem (2) are quasi-convex on the region of decision variables of interest.

Due to the formidable complexity of an analytical formalization of the functions properties, we give evidence of Remark 1 by extensive Monte Carlo simulations. This approach is common for establishing convexity of functions when their difficult analytical structure or nonlinearities does not allow explicit derivation of the derivatives [38]. Given that optimization problem (2) is quasi-convex, it can be solved by using efficient algorithms of low computational complexity. In the next subsections, we propose a solution based on the stochastic subgradient method.

B. THRESHOLD OPTIMIZATION

We divide the time into consecutive mini-slots of Δ seconds, and call them estimation interval. The functions of optimization problem (2) are estimated at the sink. Each device embeds the system time onto every new packet. Then, the sink node calculates the latency of serving every packet, and compute the average channel utilization f_1, by counting the number of received bits, and packet latency f_2 over Δ seconds.

To solve optimization problem (2) with noisy measurements, we use stochastic subgradient method [38]. In particular, we observe that increasing λ_c statistically enhances f_1 at the expense of higher f_2, which may be intolerable for the primary devices. Similarly, reducing λ_c decreases the latency of the primary devices, at the expense of lower channel utilization. Therefore, the sink adjusts the threshold λ_c in each Δ seconds such that f_1 increases while constraint f_2 is met.

Let λ_c^k be the threshold value at mini-slot k. Let $\hat{f}_1^k$ and $\hat{f}_2^k$ be the unbiased estimation of $f_1{}^k$ and $f_2{}^k$, computed at the sink based on λ_c^k. The sink continuously monitors the primary channel utilization and average latency of the primary devices over the time, updates the optimization parameter using $\hat{f}_1^k$ and $\hat{f}_2^k$ for all k, and piggybacks the updated value in the `acknowledgement` packets replied to secondary devices. Then, the secondary devices operate with the updated parameter λ_c^{k+1}, which will affect the primary channel utilization and the average latency of the primary devices. Once a reduction in the primary channel utilization is detected ($\hat{f}_1^k < \hat{f}_1^{k-1}$) by a decrease of the threshold ($\lambda_c^k < \lambda_c^{k-1}$), the transmitter changes the update direction and adopt a higher value for the next congestion threshold (namely, $\lambda_c^{k+1} > \lambda_c^k$) in order to avoid further reduction of the number of devices in the primary channel. Otherwise, the transmitter is in the right direction of decreasing the threshold. Let $A_k = \{\hat{f}_1^k \ge \hat{f}_1^{k-1}\} \cap \{\hat{f}_2^k \le d_{\max}\}$ define the event of having a higher channel utilization while meeting the latency requirements of the primary devices, and $B_k = \{\lambda_c^k \le \lambda_c^{k-1}\}$ define the event of decreasing the threshold. Let C_k be the result of exclusive OR operation (XOR) between A_k and B_k, i.e., $C_k = A_k \oplus B_k$, taking true if A_k and B_k do not have the same value. Let α_k be the step size of the update procedure at mini-slot k, which are assumed to be square-summable but not summable to ensure convergence of the stochastic gradient method [38]:

$$\alpha_k \ge 0, \quad \sum_{k=1}^{\infty} \alpha_k^2 < \infty, \quad \sum_{k=1}^{\infty} \alpha_k = \infty. \tag{3}$$

By computing C_k after each iteration, the optimization parameters will be updated based on stochastic gradient method:

$$\lambda_c^{k+1} = \left[\lambda_c^k - \mathbb{1}_{C_k}\alpha_k\right]^\dagger, \tag{4}$$

where $[\cdot]^\dagger$ represents the simple projection operation of the decision threshold onto the feasible region $\lambda_c^k \geq 0$, and

$$\mathbb{1}_{\bullet} = \begin{cases} -1, & \text{if } \bullet \text{ is true} \\ +1, & \text{otherwise.} \end{cases} \tag{5}$$

Following a similar analysis as the one provided in [38], it can be proved that the iterations of (4) converge to the optimal solution of optimization problem (2) almost surely, which is omitted here due to lack of space. The convergence rate, however, may hinder practicality of the algorithm. To solve this problem, the ACC-CSMA protocol defines that the algorithm converges if the relative change of the channel utilization, denoted by γ_k, is negligible. That is, the update procedure terminates if

$$\gamma_k = \left|\frac{\hat{f}_1^{k-1} - \hat{f}_1^k}{\hat{f}_1^{k-1}}\right| \leq \varepsilon_2, \tag{6}$$

where $\varepsilon_2 > 0$ is a predetermined threshold.

C. THRESHOLD ADAPTATION

Although the optimal congestion threshold λ_c can be obtained by running iteration (4) in systems with stationary network conditions, some C-WSNs can be highly dynamic. Thus, λ_c should be adaptively tuned when necessary to continuously keep the optimality of the system. To this end, the sink tracks the changes in the optimality of the current threshold. That is, it monitors the primary channel utilization over every mini-slot and then compares the relative change to a predetermined threshold ε_1, where $\varepsilon_1 \gg \varepsilon_2$. Upon detection of a sudden change in the primary channel utilization ($\gamma_k \geq \varepsilon_1$), the update procedure will be recalled to find new optimal λ_c for the new network state; because the current solution is not optimal anymore. Notice that the threshold for terminating update procedure ε_2 should be small enough to mitigate early termination due to measurement noise, whereas ε_1 should be large enough to allow detection of real changes in the network status and to avoid false alarm due to measurement noise, thus mistakenly triggering the update iterations. Clearly, the update procedure has to be recalled also once the latency requirements of the primary devices are not met. Defining $\varepsilon^k = \varepsilon_1$ for $k = 1$ and $\varepsilon^k = \varepsilon_2$ for $k \geq 2$, Fig. 11 shows the proposed adaptation algorithm.

Note that the proposed adaptive algorithm adjusts the congestion threshold according to the variation of the statistics describing the wireless environment. More specifically, the optimization is based on the primary channel utilization estimated by the sink, which is directly related to the average signal-to-interference-plus-noise ratio, traffic intensity of the primary devices, the number of secondary devices, and the number of primary devices. These parameters either do not change or change very slowly, depending on the specific application, which leads to considerably low rate

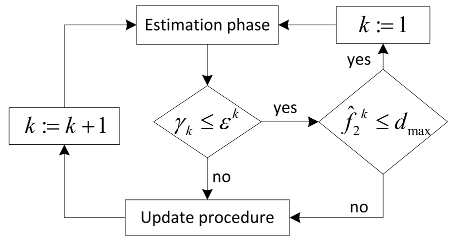


FIGURE 11. Flowchart of dynamic threshold adaptation. Every iteration of the algorithm may take Δ seconds.

of the change in the optimal λ_c. This is very important, since it provides enough time for iteration (4) to converge. More importantly, the solution is valid for a relatively long time. This low adaptation rate meets the light computational complexity requirements of many applications. For smart grid applications, as an example, the statistics of the network remains unchanged for few hours, while the subgradient iterations converges after around 30 seconds, as verified by extensive experimental results in the next section.

Our experimental results in the next section show that the adaptively optimized threshold improves the network performance with a minimal computational cost of running iteration (4). On the other hand, one can try to approximate the objective and constraint functions by using regression models and solve the corresponding optimization problem, which we have left as a future direction.

VI. EXPERIMENTAL EVALUATION

The ACC-CSMA protocol has been successfully implemented on real-world STM32W108 chips. To further evaluate the performance of the proposed protocol, a real-world testbed containing 31 STM32W108 nodes has been created. In all the experiments, without loss of generality, we use one primary channel and one secondary channel. Table 2 shows the key parameters used in the experiments. The collection period of the secondary network of ACC-CSMA is set to be 500 ms for all scenarios. Data packets are generated in Poisson distribution, and each data packet has a size of 120 bytes. The sink measures the primary channel utilization and average latency of primary devices every 1 second. Notice that due to latency sensitivity, the sink discards outdated packets of the primary devices, i.e., those with latencies higher than d_{max}. The maximum tolerable latency d_{max} of the primary devices is set to 70 ms. Actually, we have also tried other d_{max} with either lower or higher values, all came to similar results, which are omitted from this paper.

For performance comparison, we have also implemented a special version of CRNs protocol that is based on Fast spectrum Sensing and Fast channel Handoff upon primary devices detection, by which we call FSFH in this paper. The implementation of FSFH in this paper is to mimic/simulate an ideal version of the existing popular solutions that use ON/OFF random process for modeling primary device activities and execute fast spectrum handoff right upon the emergence of

TABLE 2. Key parameters of ACC-CSMA.

Parameter	Description	Value
λ_c	Initial congestion threshold	10
L	Backoff index estimation array length	5
BE-min	Minimum backoff exponent	2
BE-max	Maximum backoff exponent	5
Δ	Estimation interval duration	1 s
α_k	Update procedure step size	$5/k$
ε_1	Update procedure initialization threshold	10
ε_2	Update procedure termination threshold	0.01
$d_{\max}$	Maximum tolerable delay of primary devices	70 ms

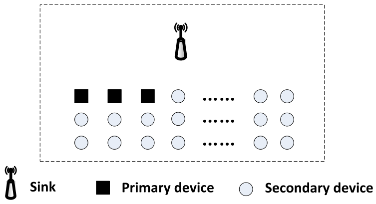
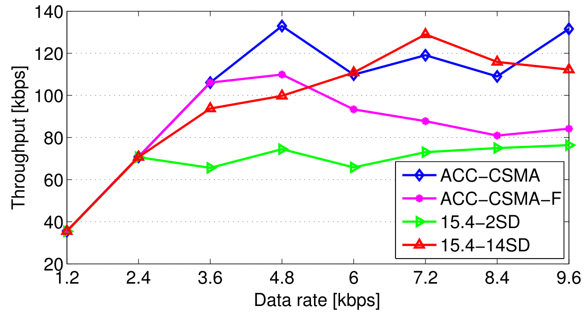


FIGURE 12. Experimental test-bed topology.

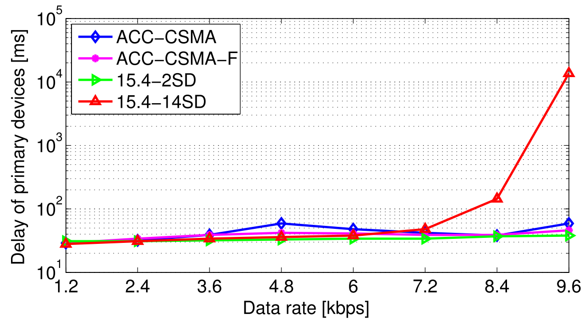
primary ones, like typically in [23]–[25]. To realize fast sensing in FSFH, and also for the convenience of implementation, we idealized the spectrum sensing scheme of FSFH by *pre-programming* the active timing of the primary devices into the secondary devices in the experiments. Once activities of the primary devices emerge on the primary channel, secondary devices of FSFH will be noticed immediately and they execute channel handover to leave the primary channel. On the other hand, once primary devices are silent on the primary channel, through pre-programming, secondary devices of FSFH immediately turn back to the primary channel. In other words, our idealized FSFH implementation has error-free spectrum sensing and instantaneous spectrum handoff capabilities.

Fig. 12 shows the network topology of the real-world test-bed on which we run experiments with 1 sink, 3 primary devices and 27 secondary devices. The node devices were deployed in normal office environment, i.e., in one room of our office building, with existence of WiFi signal. The transmit power gain of all node devices is set to 0 dB and all devices are in one-hop transmission range of each other. The test-bed in Fig. 12 simulates a real-world cellular application as shown in Fig. 1.

In the experiments, we firstly verify ACC-CSMA's performance over the legacy IEEE 802.15.4 MAC protocol which is widely used in WSNs and IoT, in scenarios where different numbers of secondary devices are initiated on the primary channel. Then, we compare ACC-CSMA with the idealized FSFH implementation with critical bursty traffic conditions. We also study how the active ratio of the primary devices affect the performance of both ACC-CSMA and FSFH, and show how ACC-CSMA outperforms FSFH with higher channel utilization. As power conserving is crucial for IoT, we finally evaluate ACC-CSMA's duty-cycle scheme with typical application traffic settings. Power consumption rates in terms of radio duty-cycle are shown.



(a) Primary channel utilization.



(b) Delay of primary devices.

FIGURE 13. Illustration of the effect of the transmission rate on the performance metrics. $d_{\max}$ is 70 ms.

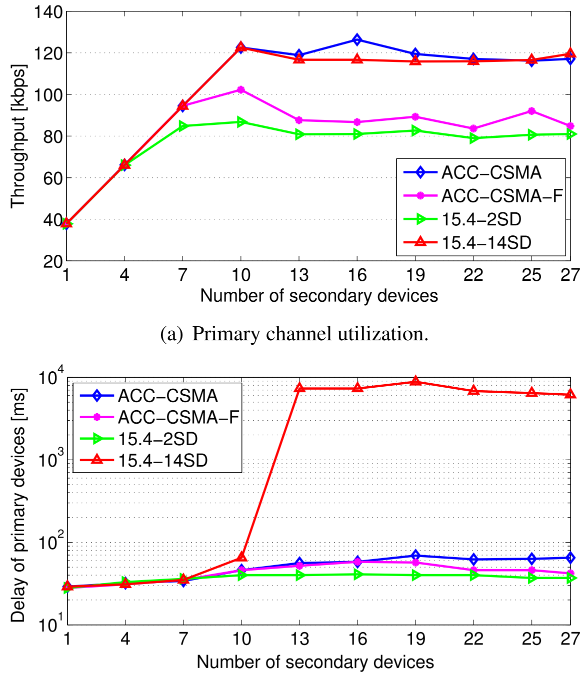
A. COMPARISON WITH STATIC STRATEGY

We firstly conduct a set of experiments to compare the performance of our proposed protocol to the static, legacy IEEE 802.15.4 solution. We apply IEEE 802.15.4 MAC protocol on the real-world test-bed as shown in Fig. 12 (containing 1 sink, 3 primary devices and 27 secondary devices) and have tested two scenarios of the protocol wherein there are either too small or too large number of secondary devices in the primary network. For the former, labeled as 15.4-2SD, we consider only 2 secondary devices. For the latter, labeled as 15.4-14SD, 14 secondary devices are introduced to the primary channel. The rest of the secondary devices are on the secondary channel. Similar to the ACC-CSMA protocol, the coordinator of the secondary channel of the IEEE 802.15.4 MAC (sub-sink) relays the collected packets to the primary sink at the end of its superframe. Moreover, to evaluate the performance gain in ACC-CSMA due to the optimal threshold λ_c design, we have implemented another version of the ACC-CSMA protocol with fixed $\lambda_c = 10$, labeled as ACC-CSMA-F. This λ_c is constant irrespective of the network condition. All other settings of ACC-CSMA-F are the same as those of ACC-CSMA.

First, we studied the effect of transmission rate on the performance metrics. In this experiment, the transmission rate of all devices was increased from 1.2 kbps (1.25 packets per second) to 9.6 kbps (10 packets per second) in seven steps, while each experimental scenario lasts for 300 seconds. Fig. 13 shows the averaged primary channel utilization in terms of throughput and latency of the primary devices. When the data rate increases, all the four protocols have increasing throughput at the beginning. Notably, when the data rate is beyond 4.8 kbps, there are some fluctuations in the throughput of all the protocols, mostly due to the randomness of multiple accesses and the dynamics of the channel. Overall, when the data rate exceeds 4.8 kbps, all protocols saturate at a lower throughput than our proposed ACC-CSMA. ACC-CSMA achieves the highest throughput while meeting the primary latency guarantee, thanks to the adaptive threshold (λ_c) selection. Both ACC-CSMA-F and 15.4-2SD protocols result in comparatively low channel utilizations when more intensive traffic rates are adopted. The main reason is due to the lack of adaptive congestion control, whose importance will be increased with the traffic rates. By introducing many secondary devices in the primary channel, 15.4-14SD also achieves high throughput. The price is high latency of the primary devices due to serious channel congestions (after the data rate has gone beyond 7.2 kbps, as shown in Fig. 13(b)), since 15.4-14SD does not dynamically maintain the congestion level of the network.

We also investigated the impact of increasing the number of secondary devices on the performance metrics. By using the same test-bed, we changed the number of secondary devices in 9 steps. In each step, all (primary and secondary) devices transmit with the same data rate of 9.6 kbps. We allow a maximum of 2 and 14 secondary devices in the primary network in 15.4-2SD and 15.4-14SD, respectively, whereas all devices are initiated to be on the primary channel in the ACC-CSMA and ACC-CSMA-F protocols.

Fig. 14 illustrates the averaged primary channel utilization (in terms of throughput) and latency of the primary devices. All protocols show similar performance when only a few secondary devices were added to the network. The main reason is that the primary network still remains unsaturated with those additional secondary devices, however they increase the primary channel utilization with no significant latency penalty. By adding more secondary devices to the system, both ACC-CSMA-F and 15.4-2SD start to have bounded channel utilization, compared to other protocols. In ACC-CSMA-F, the increased contention level expels some devices from the primary network while there are still rooms in the primary channel. In 15.4-2SD, the protocol simply forces some devices to operate in the secondary channel due to the lack of congestion adaptation capability. Once the primary network cannot tolerate more traffic demands (with 10 secondary devices in this example), ACC-CSMA maintains an appropriate congestion level in the primary channel by dynamically moving extra secondary devices to the secondary channel, which guarantees the strict latency requirements of the primary devices, as shown in Fig. 14(b). However, 15.4-14SD still allows more secondary devices to join the primary network when the congestion level is already high (after the number of secondary devices has gone beyond 10), leading to the worst performance for the primary devices (the primary device latency is high and outdated packets were dropped). In this respect, the high primary channel utilization of 15.4-14SD is only due to the large number of latency-tolerant secondary devices, and does not show a good performance. Generally, in this experiment, ACC-CSMA achieves the highest channel utilization while ensuring the low latency for the primary devices. It also shows that the proposed optimization algorithm, with some minimal computations to run iteration (4), improves the network performance (when especially compared to ACC-CSMA-F). For instance, when the network is with 13 secondary devices, ACC-CSMA respectively enhances the channel utilization by 36% and 47% compared to ACC-CSMA-F and 15.4-2SD, while meeting the latency requirements. When the network is with 16 secondary devices, these performance gains are respectively increased to 46% and 56%.



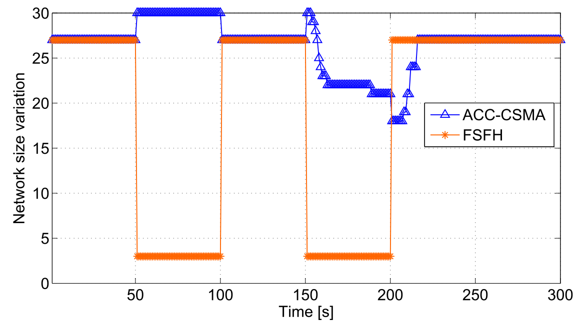
(a) Primary channel utilization.

(b) Delay of primary devices.

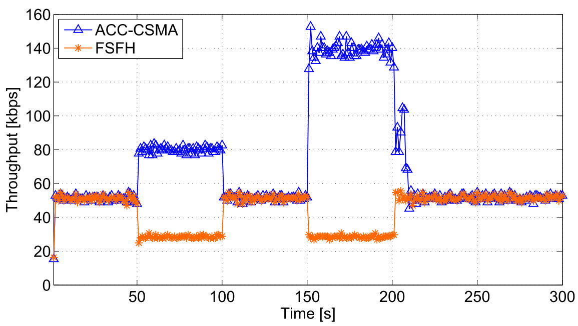
FIGURE 14. Illustration of the effect of the network size on the performance metrics.

B. EVALUATION WITH BURSTY TRAFFIC

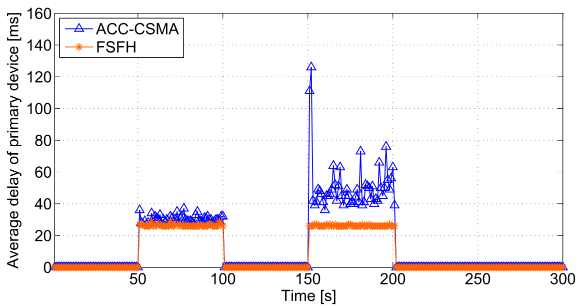
In this experiment, by using the same test-bed, we evaluate and compare the performances of ACC-CSMA and the idealized FSFH with bursty traffic loads of the primary devices, which is the most critical traffic pattern that can be occasionally introduced due to changes of system status. At the beginning of the experiment, all the secondary devices adopt a data rate of 1 packet per second (0.96 kbps per device) while all the primary devices are silent. Two bursty traffics are generated at 50 s and 150 s, respectively, each lasting 50 seconds. During the first bursty period, only the primary devices adopt an intensive data rate of 10 packets per second (9.6 kbps). The second bursty period is more critical, during which both primary and secondary devices adopt intensive data rates of 10 packets per second (9.6 kbps) and 6 packets per second (5.76 kbps), respectively. The primary devices are silent outside the bursty periods. The sink collects payloads,



(a) The number of devices in the primary network.



(b) Average throughput of the primary network.



(c) Delay of the primary devices.

FIGURE 15. **Performance under bursty traffics.**

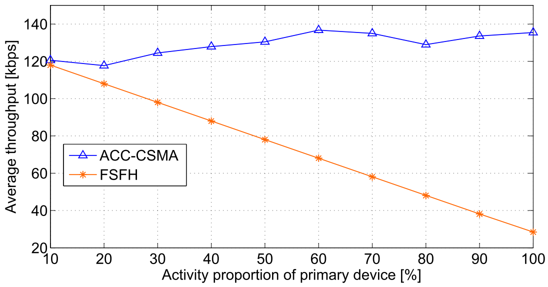
extracts the backoff index, and calculates the latency of each received packet from the primary devices. It then computes the average of these values over every 1 second.

Fig. 15 compares the performance of ACC-CSMA and FSFH. In particular, Fig. 15(a) shows the number of devices (size) in the primary network. In FSFH, all the secondary devices leave the primary channel during the bursty periods (right upon primary devices detection), thus the primary network contains only 3 primary devices in those two periods. This overprotection of the primary devices substantially reduces the untilization of the primary channel, as can be observed in Fig. 15(b) which shows the online throughput. On the contrary, the proposed ACC-CSMA protocol keeps the optimal number of the secondary devices in the primary network (Fig. 15(a)) to maximize the primary channel utilization (Fig. 15(b)) without violating the promised QoS levels of the primary devices (Fig. 15(c)). During the first bursty period in which only the few number of primary devices adopt the intensive data rate (i.e., the primary channel is underutilized), the ACC-CSMA protocol allows all the secondary devices to operate in the primary network, reducing unnecessary handoff among the channels. In the second bursty period, the primary channel gets saturated by aggressive secondary devices, and the ACC-CSMA protocol continuously forces a portion of the secondary devices to leave the primary network and join the secondary channel (to reduce the congestion), leading to a decreased size of the primary network, as shown in Fig. 15(a). Notably, ACC-CSMA keeps some secondary devices in the primary channel, unlike FSFH. Once the latency requirements of the primary devices are not met, additional secondary devices automatically leave the primary channel until a balance between leveraging the precious spectrum resource of the primary channel and guaranteeing the primary devices performance appear, as verified in Figs. 15(a)–(c). From Fig. 15(b), the proposed ACC-CSMA can provide substantially higher channel utilization than FSFH (can be as high as 4.6 times higher) without violating the latency requirement of the primary devices. Notice that the latency is higher than the threshold at the very beginning of the second bursty period in Fig. 15(c), which is natural as the ACC-CSMA protocol is highly utilizing the primary channel before this period. Once the busrty traffic period starts, the algorithm automatically changes the value of λ_c so as to temporarily expel some secondary devices from the primary channel.

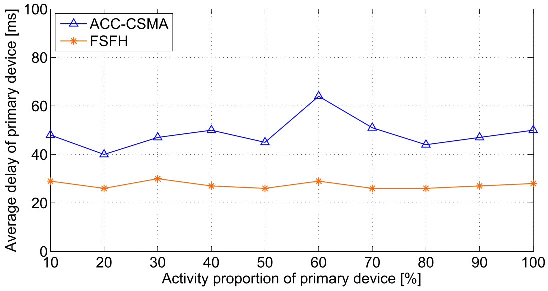
C. EVALUATION WITH TRAFFIC VARIATIONS OF PRIMARY DEVICES

In the next experiment, we test the impact of the traffic variations of the primary devices on the performance of the ACC-CSMA and FSFH. By using the same test-bed as in Fig. 12, all the primary devices adopt a data rate of 9.6 kbps (10 packets per second) and we vary the proportion of the active period of the primary devices in the experiment from 10% to 100% in ten steps. In all test scenarios, all the secondary devices adopt a fixed data rate of 4.8 kbps (5 packets per second). Each experiment scenario lasts for 300 seconds.

Fig. 16 compares the network throughout and latency performance of ACC-CSMA and those of FSFH under variable activity proportions of the primary devices. It shows that ACC-CSMA maintains the maximum primary channel utilization for all scenarios; whereas the channel utilization linearly decreases with the increase of primary devices' activities in FSFH. It is because, in FSFH, longer primary devices' activities linearly reduces the average time over which the secondary devices can operate in the primary network. On the contrary, ACC-CSMA balance the size of the primary network based on the traffic demands and managed to maintain a well-controlled congestion level on the primary channel. Before the primary channel gets saturated, some secondary devices in ACC-CSMA are allowed to utilize the channel once the performance of the primary devices is met, thus reducing unnecessary switches between secondary and primary channels and achieving higher primary channel utilization.

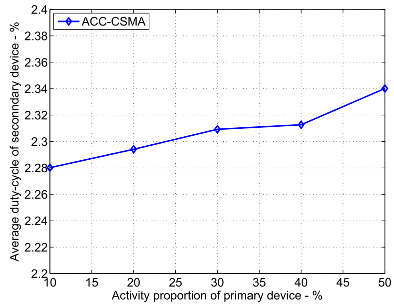


(a) Average throughput of the primary network.

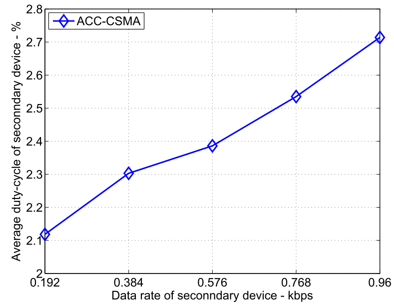


(b) Delay of the primary devices.

FIGURE 16. Performance under variable traffics of the primary devices.



(a) Average duty-cycle of secondary devices with different primary activity ratio.



(b) Average duty-cycle of secondary devices with different data rate.

FIGURE 17. Performance of energy-efficiency of ACC-CSMA.

D. EVALUATION OF THE DUTY-CYCLE SCHEME

Energy efficiency is crucial for IoT devices, in this section, we specifically evaluate ACC-CSMA's performance of conserving power with its designed duty-cycle scheme. By using the same test-bed as shown in Fig. 12, we carry out two common test experiments in which we respectively vary the active ratio of the primary devices and the date rate of the secondary devices, and check how these factors affect the power consumption rate of secondary devices in terms of radio duty-cycle.

In the first experiment, by fixing secondary devices' traffic, we check how the active ratio of the primary devices can influence the power consumption rate of secondary devices. All secondary devices are set to continuously transmit with a common fixed data rate of 1 packet per 2 seconds (0.48 kbps). The primary devices are transmitting with a more intensive data rate of 10 packets per second (9.6 kbps), while we vary the active ratio of the primary devices from 10% to 50%, to simulate different scenarios. Each test scenario lasts for 200 seconds. Notably, in all the test scenarios of this experiment, the network on the primary channel is not saturated, as we nearly observe no network handoffs among secondary devices. Fig. 17(a) summarizes the averaged radio duty-cycle of the secondary devices.

As shown in Fig. 17(a), with the increase of primary devices' active ratio, the radio duty-cycle of secondary devices raise slightly from about 2.28% to 2.34%. This shows that ACC-CSMA can well maintain low power consumption rate for secondary devices under the given test environment (a typical data rate of 1 packet per 2 seconds for C-WSNs applications), with the traffic variation of primary devices. Actually, before the network is saturated, most of the secondary devices work on control channel with the described duty-cycle scheme in Fig. 6(a). Since the primary network is not saturated and there is little chance to join the secondary network in which nodes always keep their radios on. A secondary device will only turn on its radio upon executing the rare CSMA/CA data transmissions on the primary channel and the periodic wake-up on the control channel, which contribute to low wake-up duration sum. The raise of the duty-cycle are mostly due to the extensions of CSMA/CA transmission procedures upon more frequent primary devices activities.

In the second experiment, we check how increasing activities of the secondary devices can influence their power consumption rate in ACC-CSMA. In this experiment, we fix the active ratio of the primary devices to 30%, with the same data rate of 10 packets per second (9.6 kbps) as in the previous experiment. We increase the data rate of secondary devices from 0.192 kbps (1 packet per 5 seconds) to 0.96 kbps (1 packet per second). Notably, in this experiment, the primary network experiences saturations with increasing traffic, as we have observed a few JP procedures of secondary devices during the tests. Fig. 17(b) summarizes the averaged radio duty-cycle of secondary devices in this experiment.

Fig. 17(b) shows that the averaged radio duty-cycle of secondary devices increases from about 2.1% to above 2.7%, with increasing secondary devices traffic rate. The increase of the duty-cycle is more intensive than that in the previous experiment, as we have also observed secondary network join procedures. When the primary network is saturated, some of the secondary devices start to join the secondary network in which they need to keep the radio on for maintaining the communication scheme inside the secondary network. This leads to higher power consumption, thus raise the averaged radio duty cycle. However, it can still be seen that, with the given test scenarios in which the data rates of both primary and secondary devices (e.g., around and above 1 packet per second) are relatively higher than most real application rates (e.g., 1 packet per minutes), ACC-CSMA can well maintain low duty-cycle of secondary devices (less than 3% in all our tests), verifying the energy efficiency of the proposed protocol.

Notably, we have also tested ACC-CSMA with a fully saturated network setting in which the data rate of all secondary devices is set to be 4.8 kbps (5 packets per second) and 9.6 kbps for primary devices, which are extremely intensive reporting rates. During this test, 12 (out of 27) secondary devices have been observed to join the secondary network, and the averaged radio duty-cycle of secondary devices raise to 39.12%.

VII. CONCLUSION

In this paper, a lightweight distributed adaptive congestion control protocol (ACC-CSMA) was proposed for cognitive wireless sensor networks (C-WSNs) that substantially enhances the channel utilization while meeting the requirements of the latency-sensitive (primary) devices. By using the CSMA backoff number, every secondary device locally monitors the congestion level of the network and decides on its operating channel. The proposed protocol adaptively balances the congestion level of the network and optimizes the utilization-latency tradeoff in non-stationary applications (with harsh wireless environment, dynamic traffic variations, and inter-system interference). Specifically, once room opens in the primary channel (e.g., due to decreased traffic demands), more latency-tolerant devices are automatically allowed to coexist with the latency-sensitive devices to fully utilize all transmission opportunities. Otherwise, some latency-tolerant devices locally decide to vacate this channel so as to reduce the congestion level and thereby ensure quality of service of the primary devices. We completely implemented it on the real-world STM32W108 chips and run intensive evaluation experiments. The superior performance of the proposed protocol has been verified in various scenarios (including stationary and non-stationary environments) for providing the highest channel utilization, while guaranteeing the performance of primary devices. Furthermore, we also showed that the proposed ACC-CSMA can achieve high energy efficiency in terms of radio duty-cycle with its designed duty-cycle scheme. As a future direction of this research, we will study how to apply the proposed protocol to multi-hop network and investigate the possibility of using reinforcement learning for learning and hyper-parameter tuning to enhance ACC-CSMA's performance.

REFERENCES

[1] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mahonen, "Cognitive radio networking and communications: An overview," *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 3386–3407, Sep. 2011.

[2] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Comput. Netw.*, vol. 50, pp. 2127–2159, Sep. 2006.

[3] O. B. Akan, O. Karli, and O. Ergul, "Cognitive radio sensor networks," *IEEE Netw.*, vol. 23, no. 4, pp. 34–40, Jul./Aug. 2009.

[4] G. Vijay, E. B. A. Bdira, and M. Ibnkahla, "Cognition in wireless sensor networks: A perspective," *IEEE Sensors J.*, vol. 11, no. 3, pp. 582–592, Mar. 2011.

[5] R. Etkin, A. Parekh, and D. Tse, "Spectrum sharing for unlicensed bands," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 3, pp. 517–528, Apr. 2007.

[6] T. M. Chiwewe, C. F. Mbuya, and G. P. Hancke, "Using cognitive radio for interference-resistant industrial wireless sensor networks: An overview," *IEEE Trans. Ind. Informat.*, vol. 11, no. 6, pp. 1466–1481, Dec. 2015.

[7] W.-Y. Lee and I. F. Akyildiz, "A spectrum decision framework for cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 2, pp. 161–174, Feb. 2011.

[8] A. Bonivento, C. Fischione, L. Necchi, F. Pianegiani, and A. Sangiovanni-Vincentelli, "System level design for clustered wireless sensor networks," *IEEE Trans Ind. Informat.*, vol. 3, no. 3, pp. 202–214, Aug. 2007.

[9] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[10] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[11] J. Wang, Y. Gao, W. Liu, W. Wu, and S.-J. Lim, "An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks," *Comput., Mater. Continua*, vol. 58, no. 3, pp. 711–725, 2019.

[12] J. Tan, W. Liu, T. Wang, S. Zhang, A. Liu, M. Xie, M. Ma, and M. Zhao, "An efficient information maximization based adaptive congestion control scheme in wireless sensor network," *IEEE Access*, vol. 7, pp. 64878–64896, 2019.

[13] Q. Tang, K. Yang, J. Wang, Y. Luo, K. Li, and F. Yu, "Wireless sensor network MCDS construction algorithms with energy consideration for extreme environments healthcare," *IEEE Access*, vol. 7, pp. 33130–33144, 2019.

[14] S. Debroy, S. De, and M. Chatterjee, "Contention based multichannel MAC protocol for distributed cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 12, pp. 2749–2762, Dec. 2014.

[15] C. Cormio and K. R. Chowdhury, "A survey on MAC protocols for cognitive radio networks," *Ad Hoc Netw.*, vol. 7, no. 7, pp. 1315–1329, Sep. 2009.

[16] D. Xu, E. Jung, and X. Liu, "Efficient and fair bandwidth allocation in multichannel cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 8, pp. 1372–1385, Aug. 2012.

[17] A. Lertsinsrubtavee and N. Malouch, "Hybrid spectrum sharing through adaptive spectrum handoff and selection," *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2781–2793, Nov. 2016.

[18] A. S. Zahmati, S. Hussain, X. Fernando, and A. Grami, "Cognitive wireless sensor networks: Emerging topics and recent challenges," in *Proc. IEEE Toronto Int. Conf. Sci. Technol. Humanity (TIC-STH)*, Sep. 2009, pp. 593–596.

[19] R. Deng, J. Chen, C. Yuen, P. Cheng, and Y. Sun, "Energy-efficient cooperative spectrum sensing by optimal scheduling in sensor-aided cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 61, no. 2, pp. 716–725, Feb. 2012.

[20] F. Capar, I. Martoyo, T. Weiss, and F. Jondral, "Comparison of bandwidth utilization for controlled and uncontrolled channel assignment in a spectrum pooling system," in *Proc. IEEE Veh. Technol. Conf. Spring*, vol. 3, May 2002, pp. 1069–1073.

[21] H. Shokri-Ghadikolaei and R. Fallahi, "Intelligent sensing matrix setting in cognitive radio networks," *IEEE Commun. Lett.*, vol. 16, no. 11, pp. 1824–1827, Nov. 2012.

[22] J. Jia, Q. Zhang, and X. Shen, "HC-MAC: A hardware-constrained cognitive MAC for efficient spectrum management," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 1, pp. 106–117, Jan. 2008.

[23] G. A. Shah and O. B. Akan, "Cognitive adaptive medium access control in cognitive radio sensor networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 2, pp. 757–767, Feb. 2015.

[24] J. Ren, Y. Zhang, N. Zhang, D. Zhang, and X. Shen, "Dynamic channel access to improve energy efficiency in cognitive radio sensor networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3143–3156, May 2016.

[25] Y. Liu, R. Yu, M. Pan, Y. Zhang, and S. Xie, "SD-MAC: Spectrum database-driven MAC protocol for cognitive machine-to-machine networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 1456–1467, Feb. 2017.
[26] L. H. A. Correia, T.-D. Tran, V. N. S. S. Pereira, J. C. Giacomin, and J. M. S. Silva, "DynMAC: A resistant MAC protocol to coexistence in wireless sensor networks," *Comput. Netw.*, vol. 76, pp. 1–16, Jan. 2015.
[27] C.-M. Wong and W.-P. Hsu, "Short paper: Study on cognitive radio in IEEE 802.15.4 wireless sensor networks," in *Proc. IEEE WF-IoT*, Mar. 2014, pp. 179–180.
[28] W. S. Jeon, J. A. Han, and D. G. Jeong, "A novel MAC scheme for multichannel cognitive radio ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 922–934, Jun. 2012.
[29] S.-Y. Lien, C.-C. Tseng, and K.-C. Chen, "Carrier sensing based multiple access protocols for cognitive radio networks," in *Proc. IEEE ICC*, May 2008, pp. 3208–3214.
[30] S. D. Nguyen, T.-L. Pham, and D.-S. Kim, "Dynamic spectrum handoff for industrial cognitive wireless sensor networks," in *Proc. 11th IEEE INDIN*, Jul. 2013, pp. 92–97.
[31] M.-P. Tran, T. P. T. Minh, H.-M. Lee, and D.-S. Kim, "Effective spectrum handoff for cognitive UWB industrial networks," in *Proc. 20th IEEE ETFA*, Sep. 2015, pp. 1–4.
[32] C.-C. Wu, S.-H. Wu, and W.-T. Chen, "On low-overhead and stable data transmission between channel-hopping cognitive radios," *IEEE Trans. Mobile Comput.*, vol. 16, no. 9, pp. 2574–2587, Sep. 2017.
[33] P. M. Rodriguez, A. Lizeaga, M. Mendicute, and I. Val, "Spectrum handoff strategy for cognitive radio-based MAC for real-time industrial wireless sensor and actuator networks," *Comput. Netw.*, vol. 152, pp. 186–198, Apr. 2019.
[34] S. Zhuo, H. Shokri-Ghadikolaei, C. Fischione, and Z. Wang, "Adaptive congestion control in cognitive industrial wireless sensor networks," in *Proc. 13th IEEE INDIN*, Jul. 2015, pp. 900–907.
[35] B. Rong and A. Ephremides, "Method and system for cooperative transmission in wireless multi-hop networks," U.S. Patent 8 542 579, Sep. 24, 2013.
[36] J. Åkerberg, M. Gidlund, and M. Björkman, "Future research challenges in wireless sensor and actuator networks targeting industrial automation," in *Proc. 9th IEEE INDIN*, Jul. 2011, pp. 410–415.
[37] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. ACM SenSys*, 2006, pp. 307–320.
[38] S. Boyd, L. Xiao, and A. Mutapcic, "Subgradient methods," Stanford Univ., Stanford, CA, USA, Lecture Notes EE392o, 2003, pp. 2004–2005.

SHUGUO ZHUO received the B.E. and Ph.D. degrees in automation from Zhejiang University, China, in 2010 and 2015, respectively. His research interests include wireless sensor networks, medium access control protocols, and cognitive industrial wireless sensor networks.

HOSSEIN SHOKRI-GHADIKOLAEI received the B.Sc. degree in electrical engineering from the Iran University of Science and Technology, in 2009, the M.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, in 2011, and the Ph.D. degree in electrical engineering and computer science from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2018.

His research interests include distributed optimization and machine learning, with applications in communication networks. He was a recipient of the IEEE Communications Society Stephen O. Rice Prize, in 2018, the Premium Award for the Best Paper in *IET Communications*, in 2014, the Program of Excellence Award from KTH, in 2013, and the Best Paper Award from the Iranian Student Conference of Electrical Engineering, in 2011. He was selected as an Exemplary Reviewer for the IEEE Transactions on Communications, in 2017 and 2018.

CARLO FISCHIONE received the Laurea degree *(summa cum laude)* (5/5 years) in electronic engineering and the Ph.D. degree (3/3 years) in electrical and information engineering from the University of L'Aquila, Italy, in April 2001 and May 2005, respectively. He has held research positions at the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2015, as a Visiting Professor; Harvard University, Cambridge, MA, USA, in 2015, as an Associate Professor; and the University of California at Berkeley, CA, USA, from 2004 to 2005, as a Visiting Scholar, and from 2007 to 2008, as a Research Associate. He is currently a Full Professor with the Division of Network and Systems Engineering, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden. Meanwhile, he also has offered his advice as a Consultant to numerous technology companies, such as ABB Corporate Research, the Berkeley Wireless Sensor Network Lab, Ericsson Research, Synopsys, and the United Technology Research Center. He is the Co-Founder and the Scientific Director of MIND Music Labs. He has coauthored over 180 publications, including a book, book chapters, international journals and conferences, and international patents. His research interests include optimization with applications to networks, wireless and sensor networks, the Internet of Things, and digital systems. He is an Ordinary Member of the Italian Academy of history Deputazione Abruzzese di Storia Patria (DASP). He received or co-received a number of awards, such as the IEEE Communication Society S. O. Rice Best Paper Award of 2018 for the Best IEEE Transactions on Communications article, the Best Paper Award of the IEEE Transactions on Industrial Informatics, in 2007; the Best Paper Awards at the IEEE International Conference on Mobile Ad-hoc and Sensor System 05 and 09 (the IEEE MASS 2005 and the IEEE MASS 2009); the Best Paper Award of the IEEE Sweden VT-COM-IT Chapter, in 2014; the Best Business Idea Awards from VentureCup East Sweden, in 2010, and from Stockholm Innovation and Growth (STING) Life Science in Sweden, in 2014; the Ferdinando Filauro Award from the University of L'Aquila, in 2003; the Higher Education Award from the Abruzzo Region Government, Italy, in 2004; the Junior Research Award from the Swedish Research Council, in 2007; and the Silver Ear of Wheat Award in history from the Municipality of Tornimparte, Italy, in 2012. He is an Editor of the IEEE Transactions on Communications and an Associate Editor of *IFAC Automatica*.

ZHI WANG is currently a Ph.D. Advisor with the College of Control Science and Engineering, Zhejiang University. As a Principal Investigator and a Co-Investigator, he has investigated more than 20 international collaborations and national research grants, including nine China–Germany, China–France, China–U.K., and China–Portugal collaboration projects, eight National Natural Science Foundation of China projects, and one China 863 High-Tech Program. He has authored/coauthored more than 100 SCI/EI indexed journal articles, including more than 20 IEEE Transactions and journals (the IEEE TMC, the IEEE TPDS, the IEEE TSP, the IEEE TIM, the IEEE IoT, the IEEE TII, and IEEE Access) and more than 20 top international conference articles (the IEEE INFOCOM and the IEEE SECON). He has also published a book *Wireless Sensor Networks*. He is currently preparing two books titled *Initial Investigation of Mobile Phone Positioning Technology* and *Sound Sensing and Localization Technologies*. His research interests include acoustic signal and array signal processing, acoustic SLAM and robot positioning and navigation, indoor localization and temporal–spatial data mining, device-free human activity recognition, compressed sensing and deep learning, data fusion of multiple sensing systems, industrial Internet of Things (IIoT), and privacy protection in crowdsourcing systems. He was also a recipient of the 2018 Microsoft Indoor Localization Competition in sound group.