

Interference Resilient Duty Cycling for Sensor Networks Under Co-Existing Environments

Xiaolong Zheng, *Member, IEEE*, Zhichao Cao, *Member, IEEE*, Jiliang Wang, *Member, IEEE*, Yuan He, *Member, IEEE*, and Yunhao Liu, *Fellow, IEEE*

Abstract—To save energy, wireless sensor networks often run in a low-duty-cycle mode, where the radios of sensor nodes are scheduled between ON and OFF states. For nodes to communicate with each other, low power listening (LPL) and low power probing (LPP) are two types of rendezvous mechanisms. Nodes with LPL or LPP rely on signal strength or probe packets to detect potential transmissions, and then keep the radio-ON for communications. Unfortunately, in co-existing environments, signal strength and probe packets are susceptible to interference, resulting in undesirable radio ON time when the signal strength of interference is above a threshold or a probe packet is interfered. To address the issue, we propose ZiSense, a low duty cycling mechanism resilient to interference. Instead of checking the signal strength or decoding the probe packets, ZiSense detects the ZigBee signals and wakes up nodes accordingly. On sensor nodes with limited information and resource, we carefully study and extract short-term features purely from the time-domain RSSI sequence, and design a rule-based approach to efficiently identify the existence of ZigBee. We theoretically analyze the benefit of ZiSense in different environments and implement a prototype in TinyOS with TelosB motes. We examine ZiSense performance under controlled interference and office environments. The evaluation results show that, compared with the state-of-the-art rendezvous mechanisms, ZiSense significantly reduces the energy consumption.

Index Terms—Energy efficiency, heterogeneous networks, signal detection, wireless sensor networks, wireless communication.

I. INTRODUCTION

DUE to the energy constraint on sensor nodes, it is of great importance to save energy and extend the network lifetime in wireless sensor networks. Recent studies show that radio activities are the main source of energy consumption [2]. Hence, a common approach to save energy is to make nodes working in a low duty cycle mode. The radio of a sensor node is scheduled between “sleep” (turn off the radio) and

Manuscript received August 17, 2016; revised January 12, 2017; accepted March 30, 2017. Date of publication April 12, 2017; date of current version July 13, 2017. This work was supported in part by National Basic Research Program No. 2014CB347800, NSFC Major Program under Grant No. 61190110, National Natural Science Foundation of China under Grant No. 61672320, No. 61572277, No. 61532012, and No. 61472217, National Natural Science Fund for Excellent Young Scientist No. 61422207, and China Postdoctoral Science Foundation under Grant No. 2016M601034. This paper was presented at the 12th ACM Conference on Embedded Networked Sensor Systems (SenSys 2014) [1]. The associate editor coordinating the review of this paper and approving it for publication was B. Liang. (*Corresponding author: Xiaolong Zheng.*)

The authors are with the TNLIST, School of Software, Tsinghua University, Beijing 100084, China (e-mail: xiaolong@greenorbs.com; caozc@greenorbs.com; jiliang@greenorbs.com; he@greenorbs.com; yunhao@greenorbs.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCOMM.2017.2692758

0090-6778 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

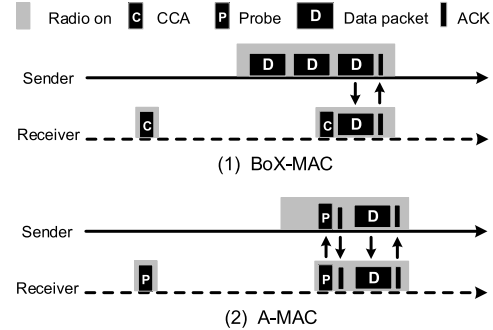


Fig. 1. The duty cycled communication flow of BoX-MAC and A-MAC.

“wake up” (turn on the radio) state. Many existing sensor network applications show the significant improvement in energy efficiency brought by the asynchronous duty-cycled media access control (MAC) protocols [3]–[5], compared to the always-on methods. A crucial issue with those protocols is to design a rendezvous mechanism, which makes the sender and receiver wake up during a same period of time to communicate with each other.

Low Power Listening (LPL) and Low Power Probe (LPP) are two well-known types of rendezvous mechanisms adopted in asynchronous duty-cycled MAC. BoX-MAC [4] and A-MAC [3] are state-of-the-art MAC protocols compatible with LPL and LPP. As shown in Fig. 1, in BoX-MAC, each receiver periodically wakes up to sample the energy level in the wireless channel, which is called CCA (Clear Channel Assessment). If the energy level is above a predefined threshold, the receiver stays awake to receive the potential packet. The sender in BoX-MAC repeats transmitting a same data packet (called preamble [4]) until an ACK (acknowledgement packet from the receiver) is received. In A-MAC, each receiver periodically wakes up to send a probe. Instead of transmitting the preamble, the sender meets the receiver by successfully decoding the probe from the intended receiver.

Both LPL and LPP can significantly reduce the energy consumption in low duty cycling mode. Nevertheless, they suffer performance degradation in noisy environments with signal interference [2]. Specifically, a node under LPL is likely to incorrectly regard the interference as interested signals and improperly keep the radio on (false wake-up), leading to considerable but unnecessary energy consumption. The probability of false wake-up significantly grows in the crowded unlicensed 2.4GHz ISM band [2], which is used by various technologies such as ZigBee [6], WiFi [7], Bluetooth [8], and microwave ovens. Similar problems also exist with LPP. If a probe from the receiver is corrupted by interference, the sender

will stay awake and wait for the intended probe for a long period of time (idle listening). In short, simply relying on the signal strength to maintain a rendezvous mechanism suffers the false wake-up problem. On the other aspect, relying on successfully decoded packets suffers the idle listening problem. The ubiquitous cross technology interference seriously degrades the efficacy and efficiency of the existing rendezvous mechanisms.

To address above issues, we propose ZiSense, an energy efficient rendezvous mechanism tailored to sender-initiated MAC protocols in noisy environments. Instead of checking signal strength or decoding the probe, ZiSense detects ZigBee transmissions based on the featured patterns of ZigBee signals. Nodes can therefore wake up only when ZigBee signals are detected.

The design of ZiSense faces several challenges in practice. First, the available information on most ZigBee-compatible devices is very limited (e.g., only RSSI on CC2420 radio). Second, the computation resource is very limited. Third, the detection time should be short to restrict the overhead. To address these challenges, we first carefully select features from time domain RSSI samples to effectively distinguish ZigBee signals from other interfering ones. We reduce the sampling time of the features and improve the RSSI sampling technique to minimize the sampling overhead. Compared to LPL, our sampling method does not incur extra overhead. Last but not least, we design a light-weight rule-based approach to distinguish ZigBee signals from interferences. The main contributions of this work are summarized as follows.

- We empirically study the performance of the existing rendezvous mechanisms and disclose that energy detection in LPL is too simple to filter the interference and probe decoding in LPP is too strict to cope with the interference
- We propose ZiSense, an energy efficient rendezvous mechanism that uses RSSI sequence patterns to recognize ZigBee and avoids false wake-ups under interference environments.
- We implement ZiSense and evaluate its performance in various environments. The results show that ZiSense significantly reduces the energy consumption of sensor nodes under interference, compared to the existing rendezvous mechanism.

The rest of this paper is organized as follows. Section II discusses the related work. Section III illuminates the motivations of this work. In Section IV, we empirically study the short-term characteristics of RSSI sequences of different 2.4GHz signals. Section V presents an overview of ZiSense and Section VI elaborates on the identification algorithm design. Section VII presents the implementation details. Section VIII shows the evaluation results of ZiSense in both controlled and real environments. We conclude this work in Section IX.

II. RELATED WORK

A. Interference Aware Rendezvous Mechanism

Most of the existing LPL compatible approaches adopt a fixed CCA threshold to detect the ZigBee transmissions. They suffer from serious false wake-up problem [4]. The

authors in [9] propose AEDP that adaptively adjusts the CCA threshold to alleviate this problem. However, AEDP may still have false wake-ups when the signal strength of interference is higher than the ZigBee. The scenario can be very common in current indoor environments with cross technology interference [2], [10]. In ZiSense, nodes keep awake by recognizing the ZigBee signal according to its time domain features, which is irrelevant with the signal strength. Therefore, ZiSense is more adaptive for the general interference situations.

B. Coexistence

Recently, many works study the coexistence between ZigBee and other interference to enhance the robustness of ZigBee transmissions. Packet delivery performance is measured in [2]. The authors in [11] study the chip error patterns under interferences. The authors in [2] use redundant headers and the forward error correction code to alleviate packet corruption. The impact of 802.11 interference on body sensor networks is studied in [12]. The authors also find that bit errors in 802.15.4 packets are temporally correlated with 802.11 traffic. Based on this correlation, they further propose an error recovery method that mitigates the effect of interference [13]. The authors in [14] propose an approach to enable ZigBee perform transmissions during the whitespace of WiFi traffic. These works concentrate on mitigating the effect of interference on receiving and decoding packets. In this paper, we study energy inefficiency incurred by interference in low duty cycle media access. Our work is complementary to above works since we reduce energy consumption from a different aspect.

C. Interference Classification

Many efforts have been made to interference classification. Airshark [15] and WiFiNet [16] leverage powerful WiFi hardware to get the spectrum information to detect and classify non-WiFi interference. DOF [17] provides the local wireless information plane, including the information of the interferers. It is proposed in [18] to scan 16 ZigBee channels to get the spectrum characteristics for classification. ZiFi [10] and ZiFind [19] recognize WiFi signal by detecting periodical beacons in WiFi. They depend on a relative long-term sampling since the default period of WiFi beacon is 100ms. SoNIC [20] proposes a method to classify non-ZigBee interference by the observation that different interference results in different corruption patterns on received packets. SoNIC needs to identify the corrupted bits and then extracts the features of the signal corresponding to the corrupted bits.

These methods aim at providing a detailed classification for non-ZigBee interference. They either rely on dedicated hardware or complicated algorithm together with long-term sampling. None of them provides a light weight feature fetching and identification algorithm to fulfill the needs of the short-term ZigBee signal detection.

III. MOTIVATION

A. Impact of Interference on LPL

To study the impact of interference on LPL, we conduct a series of experiments. We deployed three TelosB motes in an

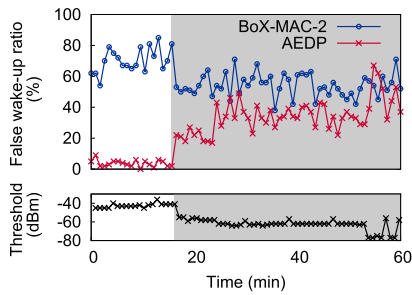


Fig. 2. False wake-up ratio of threshold-based CCA checking in an office environment.

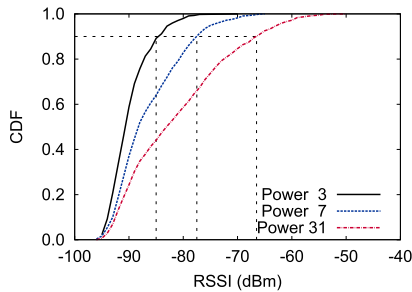


Fig. 3. CDF of RSSI on Mirage testbed with different RF powers.

office environment on channel 22. Node 1 acts as a sender and broadcasts a packet every 10 seconds; node 2 runs BoX-MAC-2 [4] as a receiver; node 3 runs the adaptive-threshold method, AEDP [9], as another receiver. The sleep interval of both receivers is set to 512ms. We measure the number of wake-ups without receiving any data as the number of false wake-ups. Then we calculate the false wake-up ratio as the number of false wake-ups to the total number of wake-ups.

Fig. 2 plots the false wake-up ratios for these two receivers. It is clear that BoX-MAC-2 experiences a very high false wake-up ratio, which is higher than 40% in most of the time. Then we inspect the performance of AEDP with an adaptive threshold. As shown in Fig. 2, AEDP has a low false wake-up ratio in the beginning. We check the beginning region and find that the RSSI of link, which varies in $[-50dBm, -36dBm]$, is higher than the interference. AEDP can therefore filter the interference from the communication link and effectively reduce the false wake-ups to achieve a low duty cycle. However, as the link RSSI decreases, the performance of AEDP degrades. When the link RSSI is between $[-77dBm, -55dBm]$, as shown in the grey region in Fig. 2, the false wake-up ratio becomes very high. We find that during the grey region, the interested signal and the interference cannot be separated based on a signal strength threshold. Thus AEDP cannot find an appropriate threshold to avoid false wake-ups.

AEDP works effectively when the link is stronger than the interference. However, a high RSSI is not always common for real low-power wireless links. We use SING [21] dataset to examine the link RSSI in a real indoor system. Fig. 3 shows that 90% of links have a RSSI lower than $-66dBm$, even though the highest transmitting power (0dBm) is adopted.

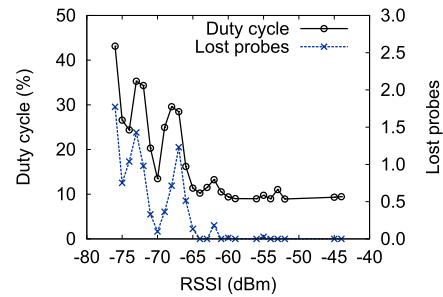


Fig. 4. Duty cycle and the number of probe loss of A-MAC under various link qualities.

Other works and experiments also show similar results of the link RSSI [22]. Considering the interference sources usually have a higher transmission power (e.g., WiFi), it is likely the signal strength of interference is higher than link RSSI [2].

B. Impact of Interference on LPP

We also conduct experiments to show the impact of interference on LPP performance. In LPP, a sender listens to the channel for probe packets to learn the existence of the interested receiver. Upon receiving a probe from the intended receiver, the sender will send its packets. However, the probe packets may get lost under interference, resulting in idle listening on the sender. We use two nodes with A-MAC [3], the most recent protocol with LPP, to investigate the impacts of interference on LPP. In the experiment, the receiver sends a probe every 512ms and the sender generates a packet every 2 seconds. Fig. 4 presents the duty cycle of the sender with A-MAC under different environments. When the link RSSI is high, (e.g., between $[-60dBm, -40dBm]$), the probe packets rarely lost and the duty cycle ratio is low. When the link RSSI becomes lower, the probe packets are not reliable. Once a probe packet is lost, the sender needs to wait for an entire sleep interval (i.e. 512ms) without sending any packet. However, corruption of probe packets does not necessarily means the data packet is also corrupted since the repeated transmissions of data packets can sufficiently increase the receiving probability. The fragile single probe packet in existing LPP wastes the transmission chances, resulting in idle listening at the sender. This idle listening time significantly increases the duty cycle.

C. Summary

To summarize, the performances of both LPL and LPP significantly degrades in presence of interference. The fundamental problem is that LPL and LPP rely on either checking the signal strength or decoding the probe packets for a rendezvous of the sender and receiver. Both of these techniques are susceptible to interference. Checking the signal strength is a loose condition that allows too much interference to wake up nodes. In contrast, decoding probe packets is a stringent condition that makes the senders ignore some transmission chances, resulting in idle listening. Our key insight is that a node should only wake up when there is a ZigBee transmission rather than a detected high energy or a decoded probe packet.

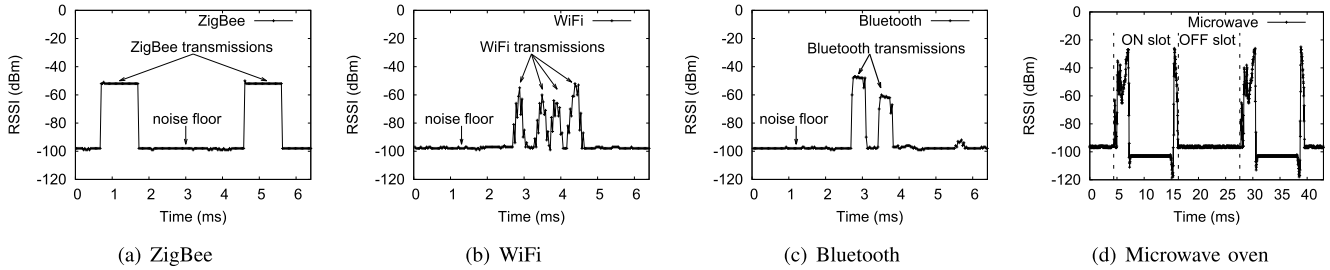


Fig. 5. RSSI patterns of different 2.4GHz technologies, collected by CC2420.

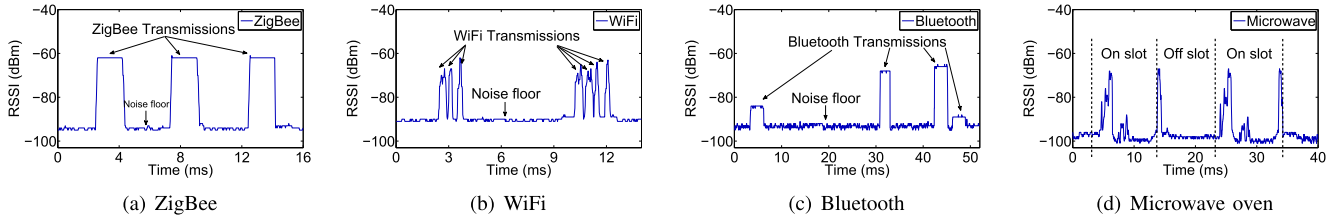


Fig. 6. RSSI patterns of different 2.4GHz technologies, collected by CC2650.

TABLE I
CHARACTERISTICS OF COMMON 2.4GHz TECHNOLOGIES

Wireless technology	On-air time	MPI	PAPR	UNF
ZigBee	[576, 4256] μ s	2.8ms or 192 μ s	≤ 1.3	FALSE
WiFi	[194, 542] μ s	$\geq 28\mu$ s	≥ 1.9	FALSE
Bluetooth	366/1616/2866 μ s	NA	≤ 1.3	FALSE
MWO	10ms	10ms	≥ 2.9	TRUE

IV. SHORT-TERM RSSI CHARACTERISTIC STUDY

We measure the RSSI sequences of common 2.4GHz wireless technologies by CC2420 and CC2650 with sampling rates 31.25KHz and 43.48KHz respectively, under the controlled environments to obtain their accurate characteristics. Fig. 5 and Fig. 6 present the RSSI sequences of different technologies sampled by CC2420 and CC2650, respectively. Noise floor is the received signal strength of the background noise when there is no wireless activity on the channel.

We carefully study the differences of RSSI sequences between ZigBee and other technologies and analyze the reasons behind the differences. We summarize the features in Table I.

A. On-Air Time

With the data rate of 250Kbps [6] and valid packet length of [18, 133]bytes [23], the on-air time of a ZigBee packet is [576 μ s, 4256 μ s]. The packet lengths and data rates specified by IEEE standard 802.11 [7] limits the on-air time of a WiFi packet in [192, 542] μ s. The residential microwave ovens work in ON-OFF mode, with 10ms ON-time and 10ms OFF-time under 50Hz power supply. Bluetooth adopts a frequency hopping technique with a hopping rate 1600 hop per second according to IEEE standard 802.15.1 [8].

Then, normally, the on-air time of Bluetooth is 366 μ s. But when transferring audio, Bluetooth usually combines three or five packets together to improve channel utilization. Then the on-air time of Bluetooth can be 366/1616/2866 μ s. Although we can filter Bluetooth by the specific three possible on-air time. But we still have the risk of filtering valid ZigBee when the on-air time of some certain-length ZigBee packets happen to be comparable. Actually, we don't need use a single feature to distinguish ZigBee from others. We rely on the intergraded information of all the features.

B. MPI

Minimum Packet Interval (MPI) is the interval between successive transmissions. The MPI of adjacent ZigBee unicast packets is 10ms by default settings in TinyOS-2.1.2, and is further reduced to 2.8ms in AEDP [9]. The MPI between adjacent ZigBee broadcast packets is 192 μ s by default settings of CC2420 [24]. MPI of WiFi is at least a DIFS time which is 28 μ s for 802.11 g/n. MPI of Bluetooth refers to the interval between successive packets transmitted on the frequency overlapped with the same ZigBee channel. It is not a fixed value and varying because Bluetooth follows a pseudorandom hopping. MPI of MWOs is defined as time between two ON slots. Hence, a specific fixed MPI of ZigBee helps to distinguish ZigBee from others.

C. PAPR

Due to different PHY modulation techniques, wireless technologies experience different received energy fluctuations. Fluctuations are observed even during one packet's transmission for WiFi Fig. 5(b). Similar results are observed during the experiments when WiFi uses BPSK, QPSK, 16-QAM and 64-QAM with different coding rates. This is because the signals from multiple sub-carriers of OFDM, modulation technique adopted by WiFi [7], can be added constructively.

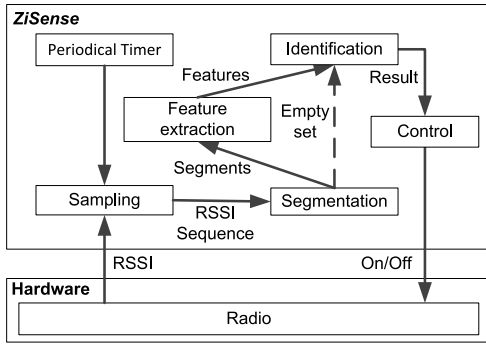


Fig. 7. Framework of ZiSense.

Then the variations from all sub-carriers will be obvious and spurious high power peaks occur [25]. DSSS in ZigBee [6] and FHSS in Bluetooth [8] have flat RSSI segments because both of them are single-carrier modulation techniques, as shown in Fig. 5. Peak to Average Power Ratio (PAPR) is a common measure of the fluctuation of signal power. As previous studies [25] have shown, 802.11g/n have a large PAPR (≥ 1.9). MWOs also have a large PAPR. In contrast, ZigBee and Bluetooth have a relatively small PAPR (≤ 1.3).

D. UNF

Under Noise Floor (UNF) is an indicator of containing RSSI lower than the minimum possible noise floor. The RSSI sequence during a microwave oven operating show fluctuations cross the noise floor, as shown in Fig. 5(d). Similar phenomenon is also observed in other studies [26]. It is due to the saturation of the intermediate frequency amplifier chain.

Above features are extracted from network standards, hardware specifications and modulation methods, etc. Meanwhile, most features can be extracted in a very short time period as observed in aforementioned measurements. The results show that leveraging those features can *efficiently* identify ZigBee transmissions without incurring additional sampling overhead comparing to existing sampling based LPL and decoding based LPP mechanisms.

V. ZISENSE OVERVIEW

A. Working Flow

The empirical results in Section IV have demonstrated the feasibility of identifying ZigBee with only RSSI information within a short period. In this section, we present ZiSense, a ZigBee sensing based rendezvous mechanism tailored to the sender-initiated duty-cycling MAC protocols, to be interference-resilient under coexisting environments.

Fig. 7 presents the framework of ZiSense. The first component is RSSI sampling that obtains RSSI samples from the radio. Then the segmentation component analyzes the input RSSI sequence to obtain a set of segments. We define a segment as a sub-sequence of the obtained RSSI sequence, which is composed of consecutive samples with RSSI readings different from the noise floor. Therefore, if no segment is found, ZiSense considers the channel as idle and directly turns off the radio. Otherwise, the feature extraction component takes the

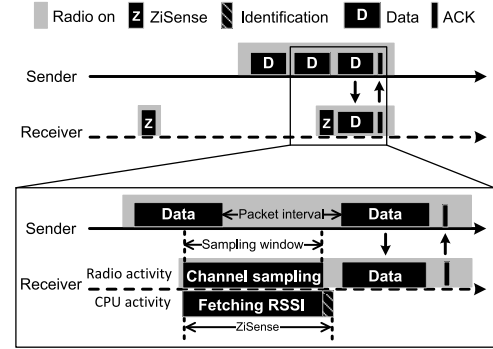


Fig. 8. The duty cycled communication flow of ZiSense compatible MAC.

detected segments as input and calculates the features for each segment. The extracted features of a segment are represented in the form of feature vector, $\mathbf{f} = (PAPR, T_{on}, MPI, UNF)$, which is provided to ZigBee identification component to determine whether a ZigBee segment exists. If yes, ZiSense keeps the radio on for T_w ms to receive the potential packets. Otherwise, the radio will be turned off immediately.

In a sender-initiated MAC protocol integrated with ZiSense, the sender initiates the transmission and repeats transmitting the data packets (called preamble) until an ACK is received. As shown in Fig. 8, the receiver with ZiSense periodically wakes up to perform channel sampling and the operations in Fig. 7. If a ZigBee signal is sensed, the node will turn on the radio to receive the potential packets. Otherwise, it turns off the radio. Through this way, comparing with LPL, ZiSense avoids the energy wasted on false wake-ups by accurate ZigBee signal identification. In ZiSense, since the receiver might overhear multiple preamble packets after it wakes up, packet transmission is more resilient to interference than LPP [27], in which the probe is only transmitted once. In section VIII-H, we investigate the influences of interference on packet retransmissions for both ZiSense and LPP.

B. Sampling and Segmentation

As shown in Fig. 8, the sender transmits adjacent packets with an interval for receiving the potential ACK since there is no need to repeat the transmission if the receiver has received the packet. Therefore, the waiting time should be at least longer than the ACK delay, T_{ACK} . To detect the existence of possible transmission in LPL, the sampling window must be longer than the minimal packet interval from the sender. The sampled RSSI sequence with W samples will be fed into segmentation component for further processing.

Segmentation aims at extract useful information from the RSSI sequence. Therefore, the segmentation component outputs segments related to the signals. Since an effective signal usually results in a sudden difference to the noise floor, ZiSense adopts a threshold method to detect the start and end points of each segment. If the difference between the RSSI and the noise floor (denote as $Noise$) is larger than a threshold th_d , it is considered as the start of a segment. Similarly, The end of a segment is detected when the difference falls below th_d . Denote the start position and end position of segment k as S_k and E_k .

C. Feature Extraction

Feature extraction component takes the set of extracted segments as input and calculates the values of features listed in Table I. Then each segment will have a feature vector $\mathbf{f} = (PAPR, T_{on}, MPI, UNF)$.

1) *On-Air Time*: The on-air time of segment k is $T_{on}(k) = (E_k - S_k) \cdot T_s$, where T_s is the sampling period.

2) *PAPR*: Denote the normalized RSSI segment as $X' = \{x'_1, x'_2, \dots, x'_W\}$, where $x' \in [0, 1]$. The PAPR of segment k can be calculated as: $PAPR(k) = \max\{x'_l{}^2 | S_k \leq l \leq E_K\} / \overline{X'_k}^2$, where $\overline{X'_k}$ denotes the average of the squared values of the elements in segment X'_k .

3) *MPI*: To calculate the MPI, we need to identify the segments belong to the same signal source. We observe that the average RSSI and on-air time of segment from the same signal source do not vary significantly during a short sampling period. Hence, we determine the segments belonging to the same signal source by the similar average RSSI and on-air time. Given a segment k , the nearest segment from the same signal source can be determined as

$$j = \arg \min_j |k - j|$$

$$\text{subject to: } \begin{cases} |T_{on}(k) - T_{on}(j)| \leq \delta, \\ |\overline{X}_k - \overline{X}_j| \leq \varepsilon, 1 \leq j \neq k \leq K \end{cases} \quad (1)$$

where \overline{X}_k and \overline{X}_j are the average RSSI readings of segment k and j , and δ and ε are error thresholds for deciding same on-air time and average RSSI readings, respectively. Then MPI of segment k is calculated as:

$$MPI(k) = (S_{\max\{k,j\}} - E_{\min\{k,j\}}) \cdot T_s \quad (2)$$

where T_s is the sampling period.

4) *UNF*: $UNF(k)$ is the Boolean indicator. $UNF(k)$ is *TRUE* if any RSSI sample in segment k has the value lower than the minimum possible noise floor, th_n . Otherwise, $UNF(k)$ is *FALSE*.

Segment k will have a feature vector \mathbf{f}_k with value $(PAPR(k), T_{on}(k), MPI(k), UNF(k))$ after extraction. Then the set of feature vectors is provided to identification algorithm to determine whether ZigBee exists.

VI. ZIGBEE IDENTIFICATION

Given a set of feature vectors, ZigBee identification component determines whether any feature vector matches the characteristics of a valid ZigBee segment. In this section, we explain our design of the identification algorithm. We first propose a set of deterministic rules based on ZigBee's underlying standard, IEEE 802.15.4. But we find the features may be corrupted in practice, resulting in the mismatch of a valid ZigBee segment. We then propose an improved algorithm which is able to identify the ZigBee segments even with some corrupted features to enhance the robustness of ZiSense. We finally compare the accuracy of different methods and discuss their application scenarios.

Algorithm 1 Deterministic Rules to Identify ZigBee

Input : feature vector $\mathbf{f} = (PAPR, T_{on}, MPI, UNF)$;
Output: whether the segment is ZigBee or not.

```

1 if  $PAPR > PAPR_{ZigBee}$  &&  $T_{on} < T_{min}$  &&  $|MPI - MPI_{valid}| > \delta$ 
  &&  $UNF == TRUE$  then return FALSE;
2 else return TRUE;

```

A. Rule Generation

We can recognize ZigBee signal by checking whether all the features meet the ZigBee standard. On this basis, we propose a set of deterministic rules, as shown in Algorithm 1.

To determine a segment is ZigBee or not, Algorithm 1 has four conditions to check: (1) $C1 : PAPR \leq PAPR_{ZigBee}$; (2) $C2 : T_{on} \geq T_{min}$; (3) $C3 : |MPI - MPI_{valid}| \leq \delta$; (4) $C4 : UNF = FALSE$. Since the sampling window is shorter than the maximum on-air time of a valid ZigBee packet (denote as T_{max}), the extracted T_{on} is impossible to be larger than T_{max} . This is why Algorithm 1 does not check whether $T_{on} > T_{max}$. Since MPI_{valid} has two values, 2.8ms for unicast and 192 μ s for broadcast, then $C3$ considered as being satisfied as long as one of the MPI_{valid} values satisfies. Algorithm 1 is a strict testing that filters out all the invalid segments with any violation to the conditions. Only the ZigBee segments with the correct feature vector can pass all the checking conditions. Hence, the condition vector of a ZigBee segment $\mathbf{C} = (C1, C2, C3, C4)$ should be (T, T, T, T) , where T is *TRUE* and F is *FALSE*.

B. Handle Corrupted Features

In Algorithm 1, if one of the conditions is violated, the segment will be treated as non-ZigBee. As a consequence, a ZigBee segment with corrupted features will be treated as a non-ZigBee signal incorrectly. This is undesirable for ZiSense's design since ZiSense follows a *conservative design principle* that tries to identify ZigBee transmissions as much as possible. Therefore, we discuss several cases with corrupted features and enhance the robustness of Algorithm 1. Based on the analysis, ZiSense adopts Algorithm 2, a more robust algorithm to identify ZigBee transmissions.

1) *Case 1*: $PAPR$ is corrupted due to the overlapped concurrent signals. When the wireless environment is very crowded, the interference signal may overlap with a ZigBee signal. Then $PAPR$ of the segment generated by this overlapped ZigBee signal may be corrupted, leading to the condition vector $\mathbf{C}_{E1} = (F, T, T, T)$. An example is shown in Fig. 9.

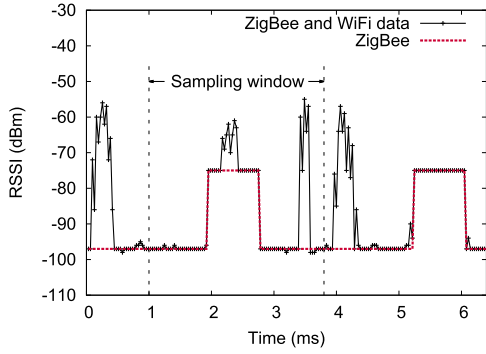
In this case, we regard the segment with \mathbf{C}_{E1} as ZigBee. This extension is safe due to following two reasons. First, a ZigBee segment with \mathbf{C}_{E1} is impossible to be regarded as other technologies since none of the interference has a condition vector with value (F, T, T, T) . Second, other technologies are hard to be identified as ZigBee. This is because to have

Algorithm 2 Robust ZigBee Identification

Input : feature vector $\mathbf{f} = (PAPR, T_{on}, MPI, UNF)$;
Output: whether the segment is ZigBee or not.

```

1 if  $PAPR \leq PAPR_{ZigBee}$  then
2   if  $(UNF == FALSE \ \&\& \ |MPI - MPI_{valid}| \leq \delta)$  then
3     return TRUE;
4 else
5   if  $T_{on} < T_{min} \ \&\& \ |MPI - MPI_{valid}| > \delta \ \&\& \ UNF == TRUE$  then return FALSE;
6   else return TRUE;
```


 Fig. 9. Example of segments with corrupted $PAPR$.

a condition vector as \mathbf{C}_{E1} , at least two conditions of the non-ZigBee segments have to be further violated.

2) *Case 2*: T_{on} can be corrupted if channel sampling begins at the packet tail and ends at the packet head, as shown in Fig. 10. Since the ZigBee transmissions are only partly detected, the extracted segments will have a shorter on-air time than expected. As a result, a ZigBee segment in this case will have the condition vector $\mathbf{C}_{E2} = (T, F, T, T)$. Therefore, we extend our algorithm to regard a segment with \mathbf{C}_{E2} as ZigBee.

This extension is safe from the perspective of identifying ZigBee segments since \mathbf{C}_{E2} is still different with the condition vectors of other co-existing technologies. However, the extension increases the probability of regarding interference as ZigBee. Among the interference sources, the Bluetooth with condition vector (T, F, F, T) is closest to \mathbf{C}_{E2} . The frequency hopping adopted in Bluetooth makes it possible for a Bluetooth device to jump back to the same channel after exact $MPI_{valid} \pm \delta$ time, resulting in a condition vector (T, F, T, T) and then false wake-ups. We argue that the probability that a Bluetooth segment has a MPI exactly equal to MPI_{valid} is low. This is also verified by our experiments in office environment in Section VI-C.

3) *Other Cases*: $C3$ and $C4$ are possible to be corrupted but with a low probability. $C3$ of a ZigBee segment is hard to be violated since MPI is corrupted only when a segment

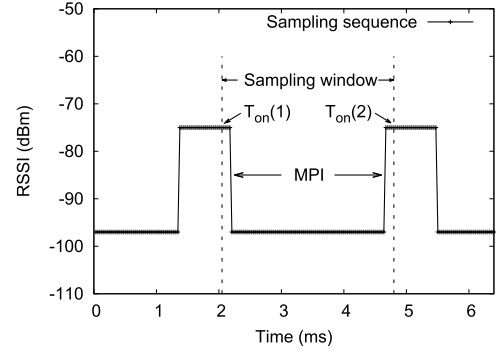
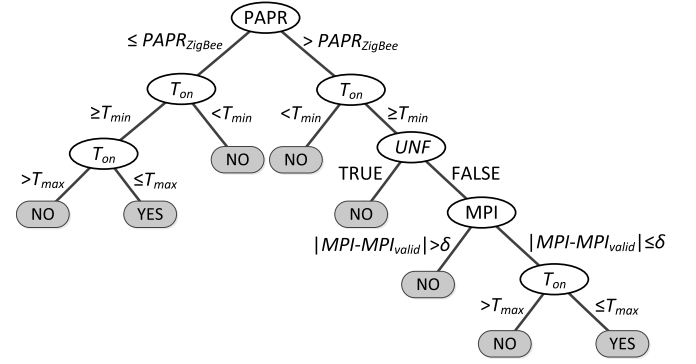

 Fig. 10. Example of ZigBee segments with $T_{on} < T_{min}$.


Fig. 11. The trained decision tree according to C4.5.

from other technology has the same average RSSI and on-air time as the valid ZigBee segments. $C4$ of a ZigBee segment is impossible to be violated since $UNF == TRUE$ is an exclusive feature of MWOs. To handle more complicated cases, ZiSense can extend the sampling period to precisely capture the correct features when too many false wake-ups occur. However, this extension increases the baseline energy. Hence, in current design, we do not extend ZiSense further to handle more complicated cases for the low baseline energy.

C. Analysis of the Identification Algorithms

To analyze the effectiveness of Algorithm 2 in practice, we collect 11621 labeled segments under a controlled office environment in presence of WiFi, Bluetooth and MWOs. We deploy a pair of TelosB motes into the environment to perform transmissions. The sender keeps sending packets with various packet sizes. The receiver collects the channel samplings. The sender and receiver are synchronized before collecting the traces. To obtain the ground truth, we label a segment as ZigBee if the sender sends a ZigBee packet at the same time. Otherwise, we label the segment as non-ZigBee.

Then the algorithms take the labeled traces as input to validate the effectiveness of our algorithms. For comparison, we also directly take the labeled segments as the training set to train a decision tree according to C4.5 algorithm [28]. The resulted decision tree is shown in Fig. 11. We adopt a 10-fold cross validation to obtain its accuracy. The identification accuracies of different algorithms are presented in Table II. False negative (FN) rate is the probability that a ZigBee signal is detected as a non-ZigBee signal, leading to retransmissions

TABLE II
IDENTIFICATION ACCURACIES OF DIFFERENT ALGORITHM

Algorithm	TP rate	FN rate	TN rate	FP rate
Algorithm 1	87.6%	12.4%	100%	0%
Algorithm 2	97.5%	2.5%	97.6%	2.4%
Decision tree trained by C4.5	97.3%	2.7%	99.1%	0.9%

at the sender in the sender-initiated MAC protocols. False positive (FP) rate is the probability that a non-ZigBee signal is detected as a ZigBee signal, resulting in false wake-ups. True positive (TP) rate is the probability that a ZigBee signal is correctly detected. True negative (TN) rate is the probability that a non-ZigBee signal is correctly detected.

The results show that Algorithm 1 can only identify 87.6% ZigBee segments. This is because the corrupted ZigBee segments fail passing the condition checking. Algorithm 1 does not consider any non-ZigBee as ZigBee, leading to 0% false wake-up. Algorithm 2 obviously improves the TP rate and identifies 97.5% ZigBee segments correctly. However, it increases the FP rate to 2.4%. The decision tree directly trained from traces achieves 97.3% TN rate and 0.9% FP rate. Compared to Algorithm 2, the TP rate is decreased by 0.2%. This is because the trained decision tree does not have any preference and sacrifices 0.2% TP rate for increasing the TN rate by 1.5%. However, Algorithm 2 prefers improving TP to reducing FP, following our conservative design principle.

D. Generality of ZiSense

ZiSense is a general method that leverages distinct RSSI sequence patterns to recognize ZigBee for energy management in MAC layer. When applying to other environments, knowledge of the long-term existences of interference can help to improve accuracy. However, sensor nodes can encounter all the interference, especially for the systems with mobile nodes [29]. As a general method, ZiSense takes all the common interference into account. When applying ZiSense to other platforms, ZiSense with Algorithm 2 can be adopted by changing the parameter values accordingly.

ZiSense is easily compatible with the low-power duty cycling MAC protocols such as BoX-MAC [4], TSCH in IEEE 802.15.4e [30] and ContikiMAC [31], [32]. As we presented, it is easy to have false wake-up problem for the energy detection method that wakes up when the channel energy is above a threshold. ZiSense mainly reduces the energy consumption by reducing the number of false wake-ups. The main design is using characteristics of RSSI sequence to recognize valid ZigBee transmissions, instead of simply checking the energy threshold. Hence, as long as the MAC protocol uses energy detection method, ZiSense can be applied. The replacement is easy because the inputs and outputs of existing energy detection method and our identification-based method in ZiSense are same. Only replacing the algorithm is fine. Hence, ZiSense is easy to transplant to other low duty-cycling MAC protocols.

ZiSense is also compatible with the routing protocols built upon duty-cycling MAC protocols. ZiSense is a duty-cycling

method in MAC layer. It doesn't rely on any specific routing protocol. In our evaluation, we provide a case study that integrating ZiSense with CTP to show that the modification of MAC layer by ZiSense won't lead to upper layer packet delivery performance degradation. Other routing protocols such as 6LoWPAN [33] and RPL [34] can also use ZiSense. In current implementations, the common way for using 6LoWPAN and RPL is adding a 2.5 layer between routing and MAC layers to translate IP packets into 15.4 packets. Namely, in MAC layer, implementations follow IEEE 802.15.4 to be compatible with the hardware. Therefore, ZiSense is still applicable in the systems with routing protocols such as 6LoWPAN and RPL.

With the development of WSNs, the radio can be more powerful than CC2420. In ZiSense, the features used for identifying ZigBee transmissions are extracted from standard specifications and default protocols. Hence, a newer hardware will not destroy the features. As long as the new hardware can obtain RSSI information with a sampling frequency not lower than CC2420, it can use ZiSense. Actually, the new radio such as TI CC2650 is usually more powerful than CC2420. In our testing, we can achieve a RSSI sampling frequency of 43.48KHz, which is much higher than 31.25KHz in our current implementation on CC2420.

With the vigorous development of wireless technologies, emerging technologies may arise. When applying ZiSense in environments with new co-existing technologies, Algorithm 2 can still be adopted to identify ZigBee since the features of ZigBee signals will not change due to the appearance of other new technologies. But the false positive rate may increase if the underlying standards of new technologies own feature vectors similar to ZigBee. Then maybe new features need to be included and new identification rules need to be made for reducing the false positive rate. However, for now, Algorithm 2 is good enough to identify ZigBee from the common wireless technologies operated on 2.4GHz.

VII. IMPLEMENTATION

We target on TelosB [35] motes to implement ZiSense under TinyOS 2.1.2 [36]. In ZiSense, a sender will take certain time to wait the potential ACK between two adjacent transmissions of preamble packets. The RSSI sampling duration of a receiver should be longer than ACK waiting period to avoid mishearing any ongoing transmissions. According to the measurement results in AEDP [9], the ACK waiting period is at least 2.8ms. In our implementation, we set the RSSI sampling duration to 2.9ms, as same as AEDP [9]. To obtain enough samples in such a short time, we increase the SPI speed and simplify the interfaces to quickly fetch RSSI readings from the register. The sampling frequency is increased to 31.25KHz. Hence, the sampling window of 2.9ms provides a RSSI sequence of $W = 90$ RSSI readings. The RSSI.RSSI_VAL register in CC22420 always has valid RSSI value when reception has been enabled at least 8 symbol periods ($128\mu s$). Hence, our implementation does not change the hardware RSSI sampling and just increase the register reading rate, which does not incur extra energy consumption. After RSSI sampling, ZiSense will take around 0.5ms to identify whether ZigBee exists.

TABLE III
SUMMARY OF PARAMETERS IN ZiSense

Notation	Value	Description
T_{ACK}	2.8ms	ACK delay, inherited from existing system implementations [9]
MPI_{valid}	2.8ms, 192 μ s	ZigBee <i>MPI</i> of unicast or broadcast, inherited from existing system implementations [9] [24]
D_s	2.9ms	Sampling duration, required to be larger than MPI_{valid} , inherited from existing system implementations [9]
T_{max}	4256 μ s	Maximum valid on-air time, specified by IEEE 802.15.4 [6]
T_{min}	576 μ s	Minimum valid on-air time, specified by IEEE 802.15.4 [6]
$PAPR_{ZigBee}$	1.3	Maximum <i>PAPR</i> of a valid ZigBee segment, empirical value consistent to previous studies [25]
δ	64 μ s	Error threshold for equivalence of time, empirical value allowing two sample errors
ε	1dBm	Error threshold for equivalence of average RSSI, empirical value
th_d	3dBm	RSSI threshold to detect effective signals, a common threshold to judge the existence of RF signal
th_n	-100dBm	The minimum possible noise floor, specified by the datasheet of radio chip [24]
T_{rx}	100ms	Active time after receiving a packet, inherited from existing system implementations [4] [9]
T_w	100ms	Active time after waking up, inherited from existing system implementations [4] [9]

We summary the implemented parameters of ZiSense in Table III. The parameters can be divided into three categories based on the principle deciding the values. (1) System parameters consistent to existing methods and previous system implementation, such as T_{ACK} , MPI_{valid} , D_s , T_{rx} , T_w . (2) Parameters decided by technical standards or hardware. For example, T_{max} , T_{min} , and $PAPR_{ZigBee}$ are decided by IEEE standards. th_n is the minimum possible noise floor. Since the sensitivity of CC2420 is -100 dBm to 0 dBm. Hence, th_n is -100 dBm. (3) Parameters empirically decided in our method. th_d is the threshold used for detecting RF signals. In existing work, it is a common way to regard a signal with RSSI 3 dBm larger than the noise as effective RF signals. Hence, we set $th_d = 3$ dBm. δ is set to 64μ s to allow two sampling errors because our sampling rate is 32μ s per sample. ε is set to 1 dBm for checking the equivalence of average RSSI. Since the RSSI of packets from the same transmitter will not change too much during a short time (tens of milliseconds), 1 dBm is large enough to tolerate the transient variance of RSSI.

For comparison, we implement AEDP according to the descriptions in the paper [9]. We also optimize the ACK waiting period from 10 ms to 2.8 ms, for BoX-MAC-2. The implementation of ZiSense takes more space to store the RSSI sequence and algorithm codes. It consumes extra 1058 bytes RAM and 6344 bytes ROM, 32.2% and 23.1% more than the default implementation. Such extra consumption is affordable for most of the sensor application programs on TelosB motes.

VIII. EVALUATION

We evaluate the performance of ZiSense from the following aspects. First, we validate its effectiveness to mitigate false

wake-ups under different interference environments. We also study the impacts of link signal strengths and data rates on false wake-up mitigation. Second, we compare the total energy consumption among several protocols under different interference environments, to show our performance gain under controlled environments. Then we integrate ZiSense with CTP, and evaluate the energy, link and routing performance in a real-world data collection application.

A. False Wake-Up

To verify that ZiSense is effective to solve the false wake-up problem, we compare ZiSense with the CCA based mechanisms, AEDP and BoX-MAC-2, under various controlled interference environments. We conduct the experiments in an empty room with maximum internal distance of 4.5 m. We use LanTraffic V2 [37] on two laptops to generate WiFi signals. One laptop continuously transmits UDP data to the other laptop via a 802.11 b/g/n AP at 5 Mbps data rate. For Bluetooth interference, we use a Bluetooth headset to listen to music on an iPhone 5 to generate the Bluetooth signal and configure two smartphones transmit an image file with size of 1 M Bytes every 5 minutes. We use a Haier MJ-1870M1 microwave oven to heat a bowl of water to build the microwave interference. Then, we put a pair of sensor nodes at different distances from the interference sources to vary the interference strength. The receiver checks channel every 512 ms. The sender transmits a packet every 10 s. The RSSI of the ZigBee signal is about -40 dBm. The channel is set to 22 .

The distributions of the false wake-up ratios are shown in Fig. 12. For different interference sources and interference strengths, ZiSense keeps a low false wake-up ratio. AEDP performs better than the default BoX-MAC. Comparing with AEDP, the false wake-up ratio is reduced by 88.9% , 49.4% and 96.3% in average under WiFi, Bluetooth and Microwave interference. The results verify the effectiveness of ZiSense to conquer the false wake-up problem. The results also reveal that the influence of WiFi signal is more serious than Bluetooth and Microwave. The frequency hopping in Bluetooth and the Faraday cage of microwave oven mitigate the impact of the interference on a certain channel.

B. Impacts of Link Signal Strength

We further explore the impacts of link signal strength on the performance of conquering false wake-up problem. We deploy a sender and three receivers in an office. The sender broadcasts a packet every 10 s. The three receivers run BoX-MAC-2, AEDP and ZiSense respectively. The receivers calculate the average packet RSSI as the link RSSI. We vary the distance between sender and receivers to get various link signal strengths. In each run, the receivers count the number of wake-ups without receiving packets as the number of false wake-ups. At each location, we conduct ten runs of experiment. We group the links into 8 buckets based on link RSSI. Each bucket adopts the average link RSSI as representative link signal strength.

The false wake-up ratios of different methods are presented in Fig. 13. BoX-MAC-2 presents a high false wake-up ratio.

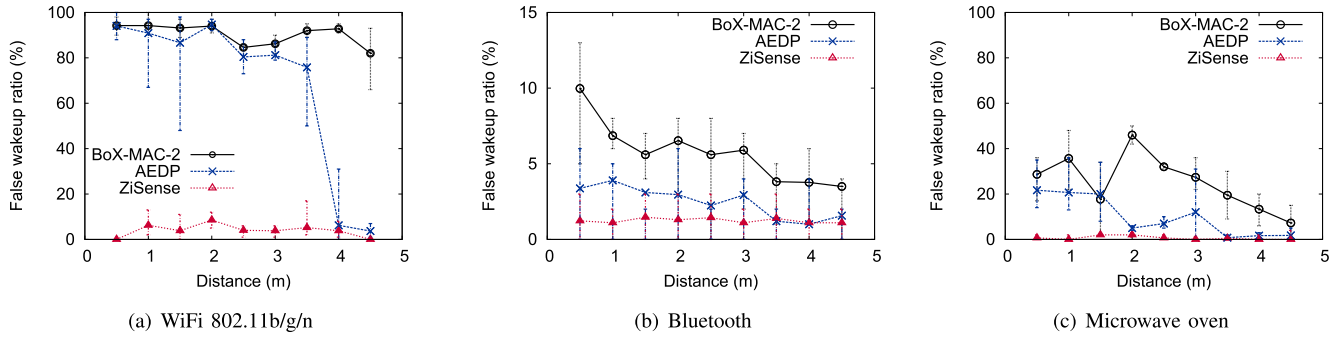


Fig. 12. The comparison of false wake-up ratio among ZiSense, BoX-MAC-2 and AEDP under the interference environments.

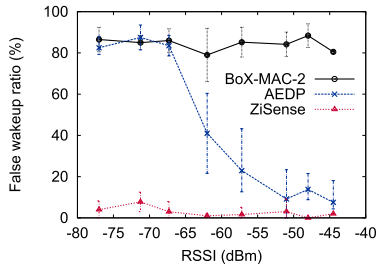


Fig. 13. Impacts of link strength on false wake-up ratio.

AEDP effectively reduces the false wake-ups when link signal strength is strong ($RSSI \in [-55, -40]$). However, when link signal strength is weak ($RSSI \in [-80, -65]$), AEDP has as many false wake-ups as that BoX-MAC-2 has. Between the weak and strong regions, a transition zone ($RSSI \in [-65, -55]$) exists. In the transition zone, AEDP can only avoid partial false wake-ups. ZiSense keeps a low false wake-up ratio under various link signal strengths. This is because ZiSense leverages the differentiable signal features to distinguish ZigBee and interference, which do not vary with the link signal strength.

C. Impacts of CCA Rate

We further explore the impacts of cca rate on the performance of ZiSense. We deploy nodes running ZiSense without data packet receiving or transmitting under the office environment same to the environment in Section VIII-B. Hence, the duty cycle is only related to the efficiency of the duty-cycling methods. When there is no packet receiving or transmitting, we can deduce the duty cycle of the optimal method that has no false wake-up. We then vary the CCA interval to compare the duty cycle of ZiSense with the optimal duty cycle. We plot the results in Fig. 14. The difference between ZiSense and the optimal duty cycle is within we can see ZiSense has a good approximation to the optimal duty cycle under a wide range of CCA intervals.

D. Impacts of Data Rate

We also explore the impacts of data rate on the performance of different mechanisms. We deploy 7 nodes and collect data with CTP in an office environment. The wake-up period of all

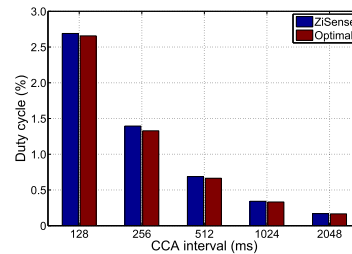


Fig. 14. Impacts of CCA rate on duty cycle.

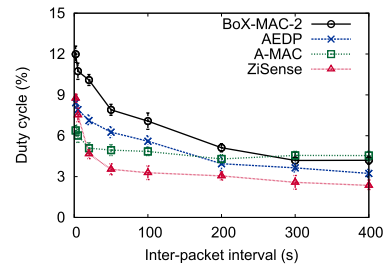


Fig. 15. Impacts of data rate on duty cycle.

nodes is 512ms for all mechanisms. 6 nodes send packets to the sink periodically. We configure inter-packet intervals (IPI) from 2s to 400s. The signal strength of routing links keeps around -70dBm . We run A-MAC, BoX-MAC-2, AEDP and ZiSense sequentially. We perform 5 experiments for each mechanism. We calculate the average radio duty cycles of all nodes.

Fig. 15 presents the results. When data rate is low ($IPI \geq 50\text{s}$), the duty cycle of A-MAC keeps around 4.5%, but the duty cycle of BoX-MAC, AEDP and ZiSense increases slowly with the same trend along the data rate gets high. The main reason is that the periodical probe-transmission/signal-detection dominates whole energy consumption. The efficient synchronized transmission in A-MAC mitigates the influence of small incremental packet transmission on duty cycle. Due to ZiSense successfully avoids false wake-up, its duty cycle is approximate 24.9% and 42.8% lower than AEDP and BoX-MAC, respectively. Since sending probe will take more energy than ZigBee identification [9] and the possible probe loss of A-MAC, the duty cycle of ZiSense is at least 28.5% less than A-MAC. When data rate is high ($IPI \leq 50\text{s}$),

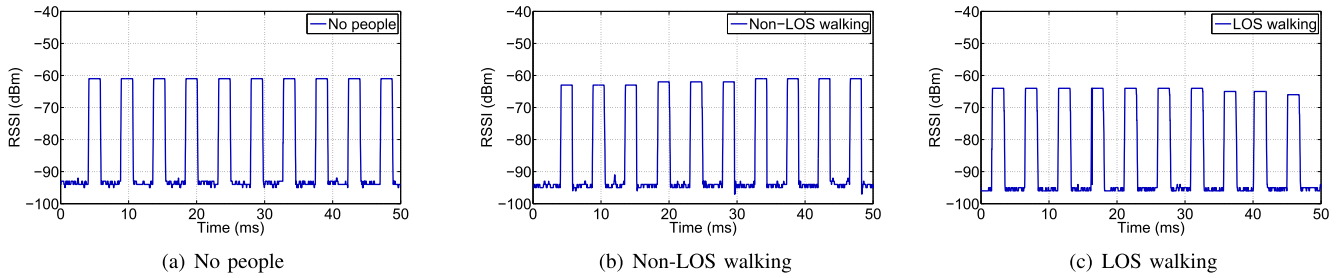


Fig. 16. The RSSI sequences of ZigBee transmissions with and without people moving around.

the duty cycle of sender-initiated mechanisms increases faster than A-MAC. The main reason is that the huge amount of packet sending dominates whole energy consumption and the synchronized transmission in A-MAC leads higher energy efficiency. Moreover, the duty cycle of ZiSense increases more quickly than AEDP and BoX-MAC. This is because when the data rate is high, a node has high probability to detect the packet transmission of others and keep awake unnecessarily. In the worst case ($IPI = 2s$), the duty cycle of ZiSense is 27.1% and 3.7% worse than A-MAC and AEDP, but it still 26.9% higher than BoX-MAC. Overall, ZiSense efficiently avoids the influence of the interference under various data rates and keeps the duty cycle low.

E. Impacts of People Movement

The people movement also affects RSSI [38]. But due to the low speed of people movements, it is not common to observe RSSI variations during one packet transmission which is in several milliseconds. Therefore, the people movement will not corrupt the ZigBee features. To verify our analysis, we conduct new experiments to analyze the RSSI variations in different environments with and without people moving around. We use a TelosB node periodically transmitting data packets and use a TelosB node as receiver to collect the RSSI sequences during the transmissions. We deploy the transmitter and receiver four meters away with Line-of-Sight (LOS) transmission path. We collect the RSSI sequences when there is (1) no people movement, (2) a person walking on the non-LOS paths, (3) a person walking in the LOS path. We also repeat the experiments when there is no LOS transmission path between transmitter and receiver. The results are similar and omitted here due to the limited space.

The experiment results are shown in Fig. 16. We can observe that the RSSI will no change when there is no people movement. When there is people movement, RSSI is affected. Different RSSI values can be observed for different packets. But during a single packet transmission, the RSSI hardly changes. Even for different packet transmissions during tens of milliseconds, the RSSI change is very small (less than 2dBm in 50ms). The features during one ZigBee transmission will not be corrupted. Namely, people movement has little impact on ZiSense.

F. Duty Cycles

Given the effectiveness of ZiSense of solving false wake-up problem, we want to explore the performance gains in terms of

TABLE IV
THE COMPARISON OF THE DUTY CYCLE AMONG ZiSense, BoX-MAC-2, AEDP AND A-MAC UNDER DIFFERENT NETWORK CONDITIONS, WHEN LINK SIGNAL STRENGTH IS -70dBm

Environment	BoX-MAC-2	AEDP	ZiSense	A-MAC
Clean environment	3.31%	3.32%	3.39%	2.97%
Office environment	10.86%	8.38%	4.21%	5.08%
Severe interference	21.80%	18.87%	5.14%	12.28%

the duty cycle. We evaluate the duty cycle of different nodes under different environments. We measure the duty cycles of A-MAC, BoX-MAC-2, AEDP and ZiSense under different environments. The link signal strength keeps be around -70dBm . The wake-up period on the receiver is 512ms. The sender generates a data packet every 10s. We take records about the radio-on time and the total time on both sender and receiver. Then we calculate the average of duty cycles on the sender and the receiver as the achieved duty cycle. For each environment, we conduct the experiment with two link signal strength settings, -70dBm and -40dBm , to evaluate the dependency between performance and link signal strength.

First, we conduct experiments in a $100 \times 50\text{m}^2$ office. 6 WiFi APs are deployed and the nearest one is 3m away from our sensor nodes. Several Bluetooth wireless earphones, keyboards and mouse are operated from 2m to 10m. Table IV shows that when link signal strength is -70dBm , the average duty cycle of ZiSense is 4.21%, which is the lowest. Compared to BoX-MAC-2, AEDP and A-MAC, ZiSense reduces the duty cycle by 61.2%, 49.8% and 17.1% separately. These results show the benefit that ZiSense experiences less false wake-ups and idle listening caused by interference.

Then we use the WiFi UDP traffic mentioned in Section VIII-A to build the severe interference environment (interference probability 0.9). We put the nodes at 3.5m away from the interference source. Table IV shows the duty cycle of ZiSense is 5.14%, which is just 22.1% larger than the office environment. However, comparing with the office environment, the serious interference leads to 100.7%, 125.2% and 141.7% more energy consumption for BoX-MAC-2, AEDP and A-MAC separately. These results further illustrate the interference resilience of ZiSense.

Finally, we move the pair of nodes in an outdoor playground without wireless interference. Table IV plots the results. A-MAC achieves the lowest duty cycle, about 2.97%. This is

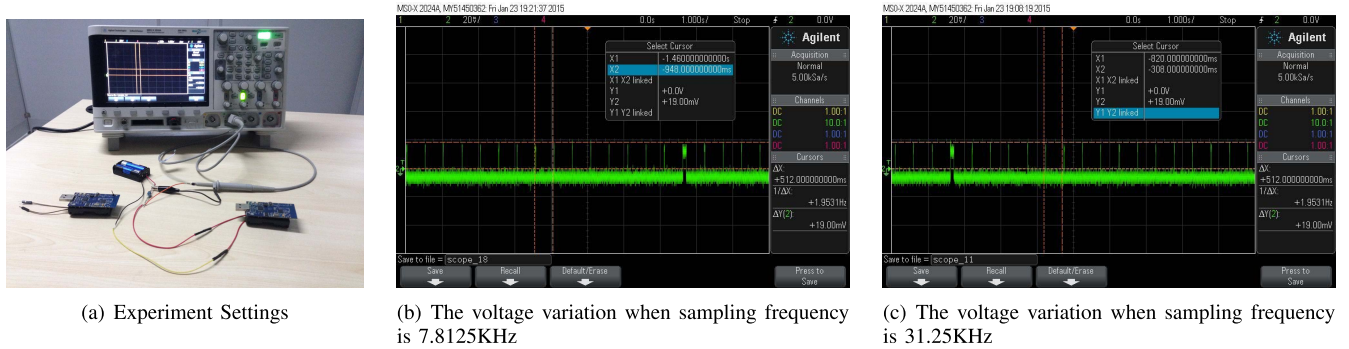


Fig. 17. Measuring the voltage variation during 1000ms under different sampling frequencies.

TABLE V
THE COMPARISON OF THE DUTY CYCLE AMONG ZiSense, BoX-MAC-2, AEDP AND A-MAC UNDER DIFFERENT NETWORK CONDITIONS, WHEN LINK SIGNAL STRENGTH IS -40dBm

Environment	BoX-MAC-2	AEDP	ZiSense	A-MAC
Clean environment	3.32%	3.32%	3.39%	2.95%
Office environment	9.60%	6.03%	3.88%	4.52%
Severe interference	17.81%	11.31%	4.34%	9.79%

because the sender with A-MAC predicts the time of probes to shorten the idle listening time. The duty cycle of BoX-MAC-2 and AEDP are comparable, about 3.31%. However, ZiSense results in 0.08% higher duty cycle than BoX-MAC-2 and AEDP. This is because ZiSense takes a longer time to process the RSSI sample sequence. Nevertheless, the baseline energy consumption of ZiSense is much smaller than its benefit.

Table V shows the performances when link signal strength is -40dBm . In the clean environment, all methods show similar performances, compared to the performances when link signal strength is -70dBm . This is because both signal strengths can guarantee the normal working flows of all methods, without the impacts from interference. When there is interference existing, we can find the performances differ under different link signal strengths. The reason behind these results is a good link signal strength can better conquer the transmission failures caused by interference.

G. Energy Consumption Under Different Sampling Rates

In ZiSense, we increase the register reading rate of RSSI.RSSI_VAL register in CC2420 from 7.8125KHz to 31.25KHz to get fine-grained RSSI information. We do not increase the actual sampling rate of the radio hardware. Hence, such implementation should not bring in additional energy consumption. We conduct experiments to evaluate the energy consumption when we read RSSI.RSSI_VAL register with different rates.

Our experiment settings are shown in Fig. 17 (a). We put a resistor, R , in series with the power supply circuit of a sensor node. We observe the variation of voltage at R , U_R . Then we can calculate the current, $I = U_R/R$, to represent the energy consumption on a sensor node.

In our evaluation, we set $R = 1\Omega$ and power the sensor node with two AA battery, with a supply voltage 3V. The

sensor nodes are configured to perform a channel assessment every 512ms. We observe the variations of the voltage at R , U_R when adopting different RSSI sampling rates. Fig. 17 (b) and Fig. 17 (c) present the experiment results when sampling rate are 7.8125KHz and 31.25KHz respectively. The impulses with intervals 512ms are caused by the channel assessments when the radio is turned on. We can find that the currents are almost the same under different sampling rates. The peak currents under both settings are around 19.0mA. The results demonstrate that the increase of RSSI sampling rate in ZiSense does not incur any additional energy overhead.

H. Integration With CTP in the Real System

We investigate the energy, link and routing performance of ZiSense in a real indoor data collection deployment. We use the default CTP in TinyOS 2.1.2 as the routing scheme. We deploy 41 TelosB nodes in our $100 \times 50\text{m}^2$ office. The deployment is shown in Fig. 18 (a). The corresponding topology of our system, with a transmission power of 0dBm, is shown in Fig. 18 (b). The sink node is labeled by a red triangle. The periodical wake-up interval is 2 seconds. Each node generates one data packet per 5 minutes. The communication channel is set to 22, which overlaps with WiFi channel used by the office APs. Bluetooth interference comes from the wireless keyboards and mouse as well as Bluetooth headsets used by the staff in the office. A microwave oven operated during meal time is deployed at the location marked by the rectangle. We compare ZiSense with A-MAC, BoX-MAC-2 and AEDP. For each mechanism, we continuously collect the data for 24 hours, 4 days in total. We conduct all the experiments in workday to keep similar interference conditions.

Table VI shows the average performance of duty cycle, packet delivery ratio (PDR), retransmission per packet (RTX), the number of wake-ups per 5 minutes, path hop count and routing ETX (the expected transmission count which is the routing metric used in CTP). The duty cycle of ZiSense is the lowest. The duty cycle of A-MAC is 68.7% higher than ZiSense. The reason is RTX of AMAC is 3.34, which is much larger than other protocols. The probe and packet loss is also verified, since the ETX is larger than hop count. The lower PDR of A-MAC reveals that single transmission of the probe is more vulnerable in co-existing environment, resulting in packet loss. However, in ZiSense, BoX-MAC-2 and AEDP,

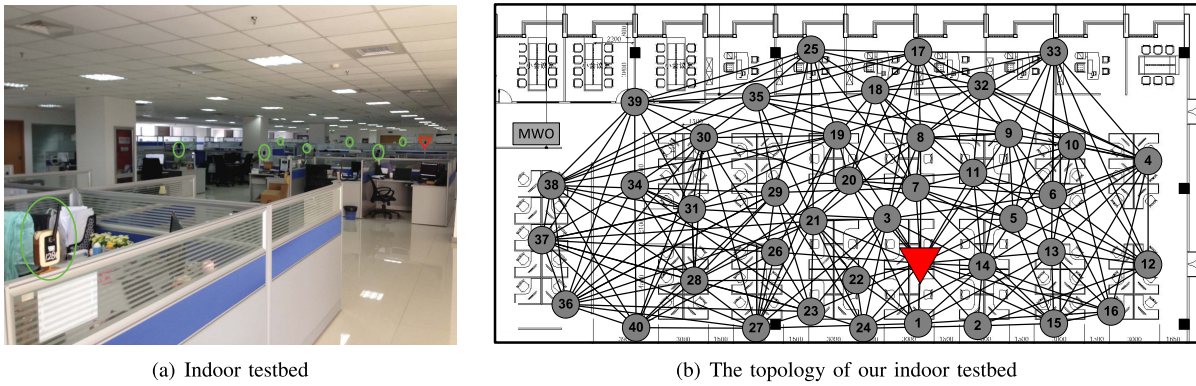


Fig. 18. Our indoor testbed in an office environment. The sink node is labeled by a red triangle.

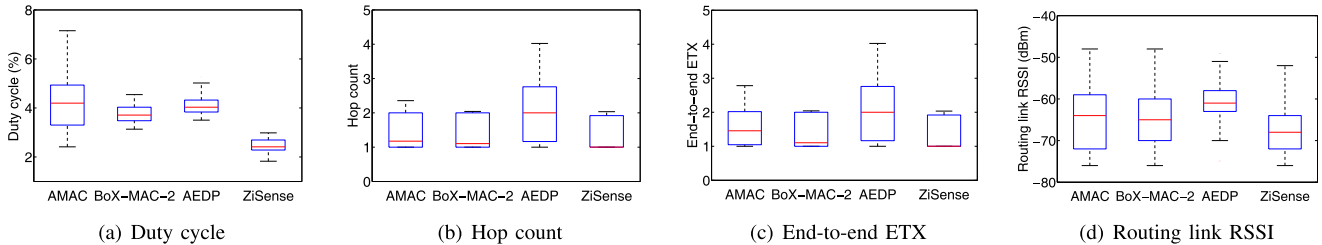


Fig. 19. Box-plot comparison of A-MAC, BoX-MAC-2, AEDP and ZiSense, presenting the median, 25th percentile, 75th percentile and the range of duty cycle, hop count, end-to-end ETX, and the RSSI of routing links.

TABLE VI
OVERALL PERFORMANCE OF DIFFERENT APPROACHES

Protocols	Duty cycle	PDR	RTX	Wake-ups per 5 min	Hop count	ETX
A-MAC	4.15%	99.26%	3.34	NA	1.33	1.44
BoX-MAC-2	3.74%	99.48%	0.10	61.61	1.42	1.43
AEDP	4.14%	99.65%	0.04	47.56	2.03	2.05
ZiSense	2.46%	99.79%	0.05	33.41	1.29	1.30

since the receiver might hear multiple preamble packets after waking up, the receiver has more chances to receive one correct packet. This is also why the RTX in these protocols is relatively small. All methods have similar PDR, illustrating that both AEDP and ZiSense will not decrease the PDR. The duty cycle reduction of ZiSense is achieved by reducing false wake-ups without sacrificing the delivery ratio of valid packets.

The duty cycle of BoX-MAC-2 is 52% higher than ZiSense since there are 84.4% more wake-ups for BoX-MAC-2 nodes. The duty cycle of AEDP is higher than BoX-MAC-2, but the number of wake-ups keeps low. Since the hop count of AEDP is larger than BoX-MAC-2, there are more packet relayed in AEDP. When the periodical wake-up interval is 2s, the relaying packets will consume more energy than wake-ups. The results infer that although AEDP could avoid the false wake-up by setting a high CCA threshold, it also increases the hop count of routing path and degrades the energy efficiency. In summary, ZiSense mitigates the energy inefficiency incurred by the interference. It also keeps the routing efficiency as much as possible.

Fig. 19 (a) presents the box-plots of duty cycles of different methods. AMAC has a high median duty cycle and a larger variation on the duty cycle, because the nodes affected by heavy interference have larger RTX which increases the duty

cycle significantly. Fig. 19 (b) and (c) present the box-plots of the average path hop count and end-to-end ETX of all nodes. Most of the nodes in AEDP have larger hop counts than the other mechanisms. But the difference between ETX and hop count is small in AEDP, indicating a small number of retransmissions. The reason is that using a higher CCA threshold in AEDP filters the links with low RSSI. Fig. 19 (d) shows the box-plots of the RSSI of the routing links. The RSSI median of the routing links is around -67dBm in A-MAC, BoX-MAC-2 and ZiSense. Since AEDP increases the CCA threshold, the RSSI median of the routing links in AEDP is -61dBm , higher than that of others.

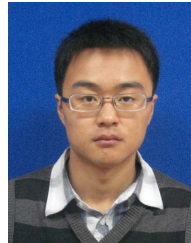
IX. CONCLUSION

We study the performance of existing rendezvous mechanisms for low duty-cycled wireless sensor networks and investigate the fundamental problems in those protocols. We observe that both the energy detection in LPL and packet probing in LPP are susceptible to interference, resulting in false wake-ups or idle listening. In this paper, we propose ZiSense, a new rendezvous mechanism for duty-cycled WSNs. Instead of energy detection and probe packets, ZiSense leverages the featured patterns of ZigBee signals that are more resilient to interference. By avoiding unnecessary wake-ups, nodes in ZiSense reduce the energy consumption in noisy environment. We theoretically validate the performance gain of ZiSense, implement it in TinyOS, and evaluate its performance on TelosB motes with extensive experiments. The results show that, compared with existing rendezvous mechanisms, ZiSense significantly enhances the energy efficiency of sensor nodes.

REFERENCES

[1] X. Zheng, Z. Cao, J. Wang, Y. He, and Y. Liu, "ZiSense: Towards interference resilient duty cycling in wireless sensor networks," in *Proc. ACM SenSys*, 2014, pp. 119–133.

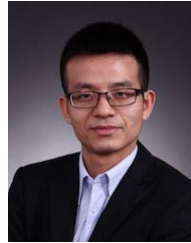
- [2] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis, "Surviving Wi-Fi interference in low power ZigBee networks," in *Proc. ACM SenSys*, 2010, pp. 309–322.
- [3] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis, "Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless," in *Proc. ACM SenSys*, 2010, pp. 1–14.
- [4] D. Moss and P. Levis, "BoX-MACs: Exploiting physical and link layer boundaries in low-power networking," Comput. Syst. Lab., Stanford Univ., Stanford, CA, USA, Tech. Rep. SING-08-00, 2008.
- [5] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. ACM SenSys*, 2004, pp. 95–107.
- [6] *IEEE Computer Society*, IEEE Standard 802.15.4, 2003.
- [7] *IEEE Computer Society*, IEEE Standard 802.11, 2012.
- [8] *IEEE Computer Society*, IEEE Standard 802.15.1, 2005.
- [9] M. Sha, G. Hackmann, and C. Lu, "Energy-efficient low power listening for wireless sensor networks in noisy environments," in *Proc. ACM IPSN*, 2013, pp. 277–288.
- [10] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma, "ZiFi: Wireless LAN discovery via ZigBee interference signatures," in *Proc. ACM MobiCom*, 2010, pp. 49–60.
- [11] K. Wu, H. Tan, H.-L. Ngan, Y. Liu, and L. M. Ni, "Chip error pattern analysis in IEEE 802.15.4," *IEEE Trans. Mobile Comput.*, vol. 11, no. 4, pp. 543–552, Apr. 2012.
- [12] J.-H. Hauer, V. Handziski, and A. Wolisz, "Experimental study of the impact of WLAN interference on IEEE 802.15.4 body area networks," in *Proc. EWSN*, 2009, pp. 17–32.
- [13] J.-H. Hauer, A. Willig, and A. Wolisz, "Mitigating the effects of RF interference through RSSI-based error recovery," in *Proc. EWSN*, 2010, pp. 224–239.
- [14] J. Huang, G. Xing, G. Zhou, and R. Zhou, "Beyond co-existence: Exploiting WiFi white space for Zigbee performance assurance," in *Proc. IEEE ICNP*, 2010, pp. 305–314.
- [15] S. Rayanchu, A. Patro, and S. Banerjee, "Airshark: detecting non-WiFi RF devices using commodity WiFi hardware," in *Proc. ACM IMC*, 2011, pp. 137–154.
- [16] S. Rayanchu, A. Patro, and S. Banerjee, "Catching whales and minnows using WiFiNet: Deconstructing non-WiFi interference using WiFi hardware," in *Proc. USENIX NSDI*, 2012, p. 5.
- [17] S. S. Hong and S. R. Katti, "DOF: A local wireless information plane," in *Proc. ACM SIGCOMM*, 2011, pp. 230–241.
- [18] K. R. Chowdhury and I. F. Akyildiz, "Interferer classification, channel selection and transmission adaptation for wireless sensor networks," in *Proc. IEEE ICC*, Jun. 2009, pp. 1–5.
- [19] Y. Gao, J. Niu, R. Zhou, and G. Xing, "ZiFind: Exploiting cross-technology interference signatures for energy-efficient indoor localization," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 2940–2948.
- [20] F. Hermans, O. Rensfelt, T. Voigt, E. Ngai, L.-Å. Nordén, and P. Gunningberg, "SoNIC: Classifying interference in 802.15.4 sensor networks," in *Proc. ACM IPSN*, 2013, pp. 55–66.
- [21] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "Understanding the causes of packet delivery success and failure in dense wireless sensor networks," Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, Tech. Rep. SING-06-00, 2006.
- [22] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low-power wireless," *ACM Trans. Sensor Netw.*, vol. 6, no. 2, p. 16, 2010.
- [23] *TinyOS TEP111*, accessed on Dec. 30, 2016. [Online]. Available: <http://www.tinyos.net/tinyos-2.x/doc/html/>
- [24] *CC2420 Datasheet*, Texas Instrum., Dallas, TX, USA, 2007.
- [25] C. Schurgers, "Systematic approach to peak-to-average power ratio in OFDM," *Proc. SPIE*, vol. 4474, p. 454, Nov. 2001.
- [26] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. Zúñiga, "JamLab: Augmenting sensor network testbeds with realistic and controlled interference generation," in *Proc. ACM IPSN*, 2011, pp. 175–186.
- [27] Z. Cao, Y. He, Q. Ma, and Y. Liu, " L^2 : Lazy forwarding in low-duty-cycle wireless sensor network," *IEEE/ACM Trans. Netw.*, vol. 23, no. 3, pp. 922–930, Jun. 2015.
- [28] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2001.
- [29] X. Wang, X. Lin, Q. Wang, and W. Luan, "Mobility increases the connectivity of wireless networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 2, pp. 440–454, Apr. 2013.
- [30] *Part 15.4: Low-Rate Wireless Personal Area Networks (WPANs), Amendment 1: MAC Sub-Layer*, IEEE Standard P802.15.4e-2012, 2012.
- [31] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. IEEE Local Comput. Netw.*, Nov. 2004, pp. 455–462.
- [32] A. Dunkels, "The ContikiMAC radio duty cycling protocol," Swedish Institute of Computer Science, Sweden, Tech. Rep. T2011:13, 2011.
- [33] J. Hui and P. Thubert, *Compression Format for IPv6 Datagrams in Low Power and Lossy Networks*, document draft-ietf-6lowpan-hc-15, 2010.
- [34] T. Winter *et al.*, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*, document Internet Draft draft-ietf-roll-rpl-06, 2010.
- [35] Crossbow Technology. *TelosB Mote Platform*, accessed on May 4, 2016. [Online]. Available: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/TelosB_Datasheet.pdf
- [36] *TinyOS*, accessed on Dec. 30, 2016. [Online]. Available: <http://www.tinyos.net/>
- [37] *LanTrafficv2*, accessed on Jan. 1, 2017. [Online]. Available: <http://www.zti-telecom.com/EN/LanTrafficV2.html>
- [38] C. Xu *et al.*, "SCPL: Indoor device-free multi-subject counting and localization using radio signal strength," in *Proc. ACM/IEEE IPSN*, Apr. 2013, pp. 79–90.



Xiaolong Zheng (M'12) received the B.E. degree from the School of Software Technology, Dalian University of Technology, in 2011, and the Ph.D. degree from the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, in 2015. He is currently a Post-Doctoral Researcher with the School of Software, Tsinghua University. His research interests include wireless networks and pervasive computing.



Zhichao Cao (M'13) received the B.E. degree from the Department of Computer Science and Technology, Tsinghua University, and the Ph.D. degree from the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. He is currently with the TNLIST, School of Software, Tsinghua University. His research interests include sensor networks, mobile networks, and pervasive computing. He is a member of the ACM.



Jiliang Wang (M'10) received the B.E. degree from the Department of Computer Science, University of Science and Technology of China, in 2007, and the Ph.D. degree from the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology. He is currently an Assistant Professor with the TNLIST, School of Software, Tsinghua University. His research interest includes wireless and mobile networks, and pervasive computing.



Yuan He (M'07) received the B.E. degree from the University of Science and Technology of China, the M.E. degree from the Institute of Software, Chinese Academy of Sciences, and the Ph.D. degree from The Hong Kong University of Science and Technology. He is currently an Associate Professor with the TNLIST, School of Software, Tsinghua University. His research interests include Internet of Things, sensor networks, pervasive computing, and cloud computing. He is a member of the ACM.



Yunhao Liu (F'15) received the B.S. degree in automation from Tsinghua University, China, in 1995, and the M.S. and Ph.D. degrees in computer science and engineering from Michigan State University, USA, in 2003 and 2004, respectively. He is currently a Cheung Kong Professor and the Dean of the School of Software, Tsinghua University, China. He is also a member of the Tsinghua National Laboratory for Information Science and Technology. His research interests include RFID and sensor network, the Internet and cloud computing, and distributed computing. He is a fellow of the ACM.