Frame Counter: Achieving Accurate and Real-Time Link estimation in Low Power Wireless Sensor Networks

Daibo Liu¹, Zhichao Cao², Mengshu Hou¹, and Yi Zhang³
dbliu.sky@gmail.com, {caozc,yz}@greenorbs.com, mshou@uestc.edu.cn
¹School of Computer Science and Engineering, University of Electronic Science and Technology of China
²School of Software, TNUST, Tsinghua University
³School of Computer Science and Engineering, Hong Kong University of Science and Technology

Abstract
Link estimation is a fundamental component of forwarding protocols in wireless sensor networks. In low power forwarding, however, the asynchronous nature of widely adopted duty-cycled radio control brings new challenges to achieve accurate and real-time estimation. First, the repeatedly transmitted frames (called wake-up frame) increase the complexity of accurate statistic, especially with bursty channel contention and coexistent interference. Second, frequent update of every link status exhausts the limited energy supply due to long duration of beacon broadcast.

In this paper, we propose meter (Distributed Frame Counter), which takes the opportunities of link overhearing to update link status in real time. Furthermore, meter does not only depend on counting the successfully decoded wake-up frames, but also counts the corrupted ones by exploiting the feasibility of ZigBee identification based on short-term sequence of the received signal strength. We implement meter in TinyOS and further evaluate the performance through extensive experiments on indoor and outdoor testbeds. The results demonstrate that meter can significantly improve the performance of the state-of-the-art link estimation schemes.

Categories and Subject Descriptors
C.2.1 [Network Architecture and Design]: Wireless communication

Keywords
Wireless Sensor Networks; Link Estimation; Real-time; Accuracy; Low Power

1 Introduction
In wireless sensor networks (WSNs), due to the deep coupling with up-layer forwarding protocols [24] [33] [29] [34], link estimation [4] [5] [6] [7] [8] [9] has fundamental impact on network performance, especially the accuracy and timeliness are two of the most important concerns. The accurate and real-time estimation ensures the optimal forwarding is consistent. In the existing link estimation protocols [12] [26] [31], each node usually utilizes the statistic of passively received packets/acknowledgements to achieve the accuracy of link estimation. Additionally, active broadcast beacon provides the timely opportunities of packets reception over those light traffic links.

Moreover, due to the unattended deployment and limited energy resource of each node, the radio usually works in low duty-cycle mode. With excellent energy efficiency, LPL (Low Power Listening) based media access control (MAC) [10] [11] is widely adopted in low power WSNs. As shown in Figure 1, A and B are two typically nodes in low power WSNs. They periodically turn on their radio to do carrier sense, but their wake-up schedule is asynchronous. If the channel is clear, the radio will be immediately turned off. Otherwise, the radio will be kept on for an extra duration (called wake-up duration) to receive the potential packets. When A is sending a packet to B, the unknown wake-up schedule of B, A has to repeatedly send the same packet (called wake-up frame or frame) to wait for B to wake up. A will not stop the transmission of wake-up frames until it receives the acknowledgement of B or a preconfigured timer expires. As the destination of a unicast packet, B will send the acknowledgement back when it successfully decodes any wake-up frame. In broadcast, B will not acknowledge any received wake-up frame.

With LPL, however, few of the existing protocols have carefully considered the new challenges to achieve accuracy and real-time link estimation. First, each node can hear multiple wake-up frames during wake-up duration, but not all of them can be successfully decoded. However, based on the only information of decoded wake-up frames, the missed wake-up frames are hard to be precisely counted. As the example shown in Figure 1, B is hard to distinguish how many wake-up frames are missed before the successfully decoded wake-up frame. The inaccurate counting may get even worse when bursty channel contention and coexistent interference appear. Furthermore, in LPL, the transmission of broadcast wake-up frames must last for a whole sleep period to ensure
every neighbor has at least one chance to hear the wake-up frame. The long broadcast duration [21] incurs large energy consumption and increases the probability of channel contention. Thus the cost of frequent broadcast beacons used for attaining real-time link estimation is extremely huge in low-power WSNs.

In this paper, we propose meter, a distributed wake-up frame counter, which exploits the feasibility of ZigBee identification and the opportunities of link overhearing to address the above challenges. First, meter samples a sequence of the received signal strength indicator (RSSI). Then according to the different characteristics with coexistent interference, meter identifies potential ZigBee sub-sequences. Finally, meter infers the belonging of ZigBee sub-sequences based on diverse characteristics matching with decoded wake-up frames or historical records. Thus nodes can precisely count all wake-up frames when they opportunistically overhear the on-going transmission of all links, even coexistent interference exists. In this way, meter can provide accurate and real-time link estimation. The main contributions of this work are as follows:

- We propose meter to achieve accurate and real-time link estimation in low-power WSNs. To our best knowledge, meter is the first work to carefully study the influence of asynchronous duty-cycle media access on link estimation.

- We develop a distributed and light weight method to precisely count the number of both decoded and corrupted wake-up frames and determine the corresponding transmitters, even with the bursty channel contention and coexistent interference.

- We implement meter and integrate it into the widely adopted link estimator (4-bits). We evaluate it on indoor and outdoor testbed. The experimental results show that meter can significantly improve the accuracy and timeliness of link estimation.

The rest of the paper is organized as follows. The preliminary of 4-bit link estimation is given in the next section. Section 3 presents the empirical studies. Section 4 introduces the detailed design of meter. The evaluation results and additional discussion are presented in Section 5 and Section 6, respectively. Section 7 surveys the related works. We conclude this paper in Section 8.

2 Preliminary of 4-bit Link Estimation

4-bit [12] is the state-of-the-art link estimator, which has been widely applied to long-term large-scale deployments of WSNs, like GreenOrbs [13] and CitySee [14]. 4-bit uses ETX (expected transmission count) [22] to depict the link quality. The number of passively received packets/acknowledgements are accumulated to update the ETX of each link. The acknowledgements are from the parent neighbor, who is the destination of unicast routing packets. The packets are from the active broadcast beacon of each neighbor.

Figure 2 illustrates the link estimation of 4-bit with LPL-based MAC protocols. In the topology as shown in Figure 2(a), node S has five neighbors (A, B, C, D, and R). R is the parent neighbor of S. Figure 2(b) demonstrates the transmission of unicast routing packets from S to R. For each routing packet, S repeatedly transmits the wake-up frames until one R’s acknowledgement is received. According to the statistic of the number of transmitted and acknowledged packets, S continuously updates the ETX of the link towards R. Moreover, Figure 2(c) illustrates the transmission of active broadcast beacon from S to all its neighbors. For each broadcast beacon, S transmits the wake-up frames for a whole sleep period so that all neighbors have opportunity to receive the broadcast beacon. Each neighbor counts the number of successfully received beacons and the sequence number of the latest beacon to calculate the ETX of the links towards S. However, 4-bit sits on MAC layer so that overlooks the influence of asynchronous transmission of LPL-based MAC protocols. With LPL, each neighbor can hear multiple wake-up frames in wake-up duration. As shown in figure 2(b), although R fails to receive the first wake-up frame, the second wake-up frame is successfully decoded and acknowledged. Figure 2(c) also shows A, B, C and R have successfully received the broadcast beacon, but B and C do not receive all arrived wake-up frames. Since 4-bit only concerns whether any wake-up frame of unicast packet or broadcast beacon is successfully acknowledged or received, it may overestimate the link quality.

In 4-bit, each node periodically broadcasts the beacons. With long duration of LPL broadcast, the period of broadcast beacon is usually kept large (several minutes) [24] to reduce the energy consumption. As shown in figure 2(b), although the neighbors A, C and D that wake up earlier than R can overhear the unicast of S, 4-bit does not take these relatively frequent data packets to update the non-routing links. Thus 4-bit cannot timely update the status of all links.

In figure 2(b), we observe that C misses the first two wake-up frames. However, depending on the only informa-
tion of decoded wake-up frames, \( C \) is hard to know the exact number of lost wake-up frames since it wakes up. Furthermore, if \( D \) only depends on the transmitter information of decoded wake-up frames, it is hard to infer the transmitter of these lost wake-up frames. Thus more delicate method is needed to accurately count the number of both lost and received wake-up frames.

3 Empirical Study

In this section, we conduct empirical studies to show that:

- 4-bit overestimates link quality with LPL-based MAC.
- The non real-time link estimation of 4-bit may seriously degrade network performance.
- It is practical to exploit the real-time RSSI to count the wake-up frames.
- It is feasible to distinguish ZigBee from other 2.4GHz technologies with RSSI features.
- It is feasible to achieve accurate and real-time link estimation in asynchronous duty-cycled WSNs.

3.1 Inaccuracy of 4-bit Link Estimation

We carry out an experiment to show the overestimation of link quality by 4-bit. We use 3 pairs of TelosB [3] nodes with diverse link quality. With different MAC layer protocols, each node broadcasts 200 packets at 10 ms interval and calculates the packet delivery ratio (PDR) of 4-bit. Figure 3 shows that the PDR of all links is overestimated with X-MAC [11] (widely adopted LPL-based MAC protocol in TinyOS) than with always-on CSMA. In the experiment, a node keeps awake for 20ms when it detects busy channel. The packet length is 43 bytes. The transmission time of each frame packet is about 1.4ms. Thus, a receiver may hear over 10 copies of the same packet. As a result, when the link quality is very poor, the PDR without X-MAC is as low as lower than 0.3, while the PDR with X-MAC can be close to 1 in X-MAC. Misled by such overestimates of link quality, a node tends to make improper decisions for other network protocols.

3.2 Non Real-Time Link Estimation

Besides the inaccuracy of link estimation, 4-bit is not able to timely obtain the real-time status of all links. We conduct an experiment in a controlled dynamic scenario to demonstrate the performance degradation incurred by non real-time link estimation of 4-bit. As shown in Figure 4, node \( S \) periodically generates data packets and forwards them to \( \text{Sink} \) via multi-hop relay. The period is 10s. Total 3000 packets are generated during the whole experiment. The routing path is determined by CTP (Collection Tree Protocol) [24]. In CTP, the period of active broadcast beacon is adjusted by trickle timer [25]. With the default setting in TinyOS, the maximum beacon period is eight minutes. The shadow region denotes the area that suffers from the controllable bursty interference (another coexistent wireless network) resulting in link dynamics. During the transmission of data packets, the range of whose sequence numbers is from 200 to 900 (about 110 minutes), the controlled interference is generated. Each node records the retransmission count, link quality and the parent neighbor. To optimize the path ETX in CTP without the interference, a is the best parent neighbor of \( S \) and \( B \) is a better parent neighbor of \( S \) than \( D \). The change of link quality of several chosen routing links is shown in figure 5, where LQI (link quality indicator) is the reverse of ETX.

The routing path of \( S \) is \( \text{S} \rightarrow \text{A} \rightarrow \text{J} \rightarrow \text{M} \rightarrow \text{Sink} \) before the interference appears. As shown in figure 5, with the degradation of \( \text{S} \rightarrow \text{A} \) according to continuous acknowledgement loss, \( B \) is selected as the new parent neighbor. However, link \( S \rightarrow B \) is also suffering from the same interference. Since the period of active broadcast beacon is large. 4-bit is unaware of the temporary degradation of the original non-routing link \( S \rightarrow B \). \( S \) has to repeatedly retransmit through the useless link until 4-bit notice enough acknowledgement loss and \( S \) further changes its parent neighbor as \( D \). Since link \( S \rightarrow D \) does not suffer from the interference, the packets can be successfully forward out. With the end of the interference, in comparison with \( S \rightarrow A \), \( S \rightarrow D \) is no longer the optimal routing link. However CTP fails to instantly know the changes because 4-bit needs long time to update the link quality of \( S \rightarrow A \) (about 350 minutes).

The experimental results demonstrate that 4-bit can not provide timely link estimation for those non routing links. The non real-time link estimation further misleads the routing selection so that it degrades the energy efficiency and network reliability.

3.3 Capture of Corrupted Wake-up Frames

With LPL, the accurate link estimation depends on the accurate counting of wake-up frames. In this section, we conduct experiment to show the opportunity to accurately count all arrived wake-up frames. The intuition is that no matter whether an arrived wake-up frame can be successfully decoded, the RSSI can be captured in real-time to identify it. In the experiment, a sender and a receiver work in LPL based asynchronous duty-cycled mode. The sleep period is set to 512ms. The sender continuously broadcasts data packets. The receiver continuously samples the RSSI and records the timestamp of successfully decoded wake-up frames. To achieve fine-grained RSSI sampling, we reimplement several.
interfaces of CC2420 [1] in TinyOS to increase the sampling rate to 31.25KHz, i.e., 32µs/sample [19].

We plot the sampled RSSI values in time-domain, as shown in Figure 6. Five wake-up frames continuously arrived at the receiver. In the figure, the shapes of RSSI sequence of all five wake-up frames have no too much difference. According to the timestamp of the three decoded wake-up frames, the other two lost frames are identified. Thus with the RSSI sequence and the timestamp of decoded wake-up frames, it is possible to accurately count the lost wake-up frames. This is the key observation to count the number of all arrived frames when the radio is turned on.

However, several technologies, such as WiFi, Bluetooth and Microwave Oven (MWO), share the same unlicensed 2.4GHz ISM band with ZigBee [16]. Can non-ZigBee transmission be effectively and efficiently filtered with common sensor nodes to avoid wake-up frame detection error? In the next section, we carefully study the short term patterns of the RSSI sequences of ZigBee. We also study some other common wireless techniques in 2.4GHz band which are usually considered in studies of interference classification [17] and co-existence [18].

3.4 Features of Zigbee RSSI Sequence

In this section, we introduce the features that can be used to distinguish Zigbee from other coexistent interference. We adopt the features of short-term RSSI sequence in Zisense [19]. The first feature is On-air time, which indicates the transmission period of individual wake-up frame. Due to the different data rate and maximum frame size of different techniques, their on-air time is usually different. The on-air time of a normal ZigBee wake-up frame of CC2420 radio is between [576, 4256]µs. As Table 1 shown, unlike ZigBee, WiFi and Bluetooth have a shorter on-air time, while microwave ovens have a longer on-air time.

The second feature is the Peak to Average Power Ratio (PAPR). PAPR is a common measure of the fluctuation of signal power. The different modulation techniques lead different PAPR. As previous studies [15] have shown, 802.11g/n has a large PAPR (> 1.9). As shown in Table 1, MWO also has a large PAPR. ZigBee adopts Direct Sequence Spread Spectrum (DSSS) so that its PAPR is relatively stable. We analyze a set of three months logs of a 300 nodes wireless sensor network [14]. The network logs record the RSSI value of each successfully decoded wake-up frame of every node. Using the recorded RSSI samples of each neighbor, we plot the standard deviation in Figure 7. The standard variance of 68% links’ RSSI values is less than 1 dBm, and 83.5% links’ RSSI standard variance is less than 1.5 dBm.

From the analysis of on-air time and PAPR features, we observe the two features can be leveraged to identify ZigBee transmission based on the short-term RSSI information [19].

3.5 Accurate and Real-time Link Estimation

In this section, we show the overall principles of our link estimation. To achieve accurate and real-time link estimation in asynchronous and duty-cycled WSNs, the following informations of all arrived wake-up frames are necessary: 1) who is the transmitter; 2) how many frames have arrived during the node’s active state; 3) how many frames have been successfully decoded by the node.

By exploiting the continuous RSSI sequence and using short-term RSSI sequence for identifying ZigBee transmission, meter can accurately count the number of arrived ZigBee frames. Furthermore, meter can exploit overhearing to know how many frames have been decoded. Finally, with the information, which includes the address of decoded frame, sender’s traffic schedule, the interval between adjacent pieces and the average signal strength of individual signal piece, meter identifies the transmitter. Hence, it is feasible to achieve accurate and real-time link estimation. In the next section, we give the detailed design of meter.

4 Design of Meter

In this section, we first give an overview of meter in Section 4.1, and then we introduce the mechanism of exploiting continuous RSSI sampling and the features of ZigBee signal to count the number of arrived frames. In addition, we determine the corresponding transmitter in Section 4.5. Finally, we integrate meter into the state-of-the-art link estimation scheme in Section 4.6.

4.1 Design Overview

The overview of meter is shown in Figure 8. S is transmitting wake-up frames towards to R, A, B, C and D are the neighbor nodes of S. During the frame transmission, A, D,
C and R wake up one after another. Until R successfully receives the frame, except B, all other neighbors have opportunity to overhear the frames. First, according to the feature of several pieces of short-term high RSSI sub-sequences, meter infers that 3, 3, 3 and 2 frames have arrived at A, D, C and R, respectively. Moreover, the number of successfully decoded frames of A, D, C and R is 2, 0, 1 and 1, respectively. Additionally, according to diverse cross-layer characteristic matching, A, D, C and R infer the source node of the overheard frames is S. Thus the instant PDR of link S→A, S→D, S→C and S→R is 0.67, 0, 0.33, and 0.5 respectively. In this way, besides B, all other neighbors can accurately estimate and update their link quality from S to them in real-time by exploiting overhearing.

4.2 Utilization of Channel RSSI

In this section, we explain the mechanism for continuously sampling channel RSSI during a node’s wake-up duration in duty-cycled WSNs.

4.2.1 Continuous Sampling

In general, radio chip, such as CC2420 [1], has a built-in RSSI (Received Signal Strength Indicator) recording the instantaneous received signal strength that can be read from an appointed 8 bit register (called RSSI register). The RSSI value is always averaged over 8 symbol periods (128 µs). Once a radio chip has been enabled for at least 8 symbol periods, the RSSI value is valid and will be recorded in the RSSI register.

In order not to miss any arrived frame, meter continuously samples channel signal strength by reading the stored value in RSSI register. As shown in Figure 9(c), by discretizing channel signal strength into RSSI values in time-domain, RSSI values can be classified into three types: low RSSI value; high RSSI value; and no RSSI value.

4.2.2 Segmentation

To detect the existence of possible frame transmission, meter sets a sampling window W to cover the wake-up duration. W can be denoted as: \( W = \{ R_1, R_2, \ldots, R_n \} \), where \( R_i \) denotes the \( i^{th} \) sampled RSSI value. Corresponding to \( R_i \), the timestamp of sampling this RSSI is marked as \( T_i \). The cumulative time starting from the node’s waking moment. Hence, \( T_1 = 0 \). As previously mentioned, the sample interval \( T_{si} \) is set to 128 µs. Meter can calculate \( T_i \) by

\[
T_i = T_1 + (i - 1) \cdot T_{si}
\]

The sampled RSSI sequence with \( n \) samples will be fed into segmentation component for further processing.

Figure 8: Overview of meter.

Figure 9: Operations of meter once it detects busy channel, involving (b) overhearing the ongoing frame transmissions; (c) frequently sampling channel RSSI; and (d) counting the number of arrived frames.
Segmentation aims at extracting useful information from the RSSI sequence. Therefore, the segmentation component outputs segments, each consisting of consecutive RSSI samples. Since an effective signal usually results in a sudden difference to noise floor, meter uses a single threshold $T_{th}$ to detect the start and the end points of each segment. If the difference between the RSSI and the noise floor (denoted as Noise) is larger than $T_{th}$, meter detects the start of a segment. Similarly, meter detects the end of a segment when the difference falls below $T_{th}$. According to ZiSense [19], $T_{th}$ is set to 3dBm. Then, the sets of start ($S$) and end ($E$) positions of segments are:

\[ S = \{ i | R_i - \text{Noise} < T_{th}, R_i - \text{Noise} > T_{th} \} \] (1)

\[ E = \{ j | R_{j-1} - \text{Noise} > T_{th}, R_j - \text{Noise} < T_{th} \} \] (2)

We sort $S$ and $E$ in ascending order according to sample time and put them in array $I_S$ and array $I_E$, respectively. Then the $k^{th}$ segment ($S_{Seg_k}$) can be represented by

\[ S_{Seg_k} = (R_{I_{S}(k)}, R_{I_{S}(k)+1}, \ldots, R_{I_{E}(k)}). \]

Visually, after segmenting the sampled RSSI sequences, meter can discretize the active period into several signal transmission stages (high level) and pure noise stages (low level) as shown in Figure 9(d). The rising edge of step pulse signal denotes the start of a segment, and the falling edge corresponds to the end of a segment. Then, the on-air time ($T_{on-air}$) of the $k^{th}$ segment ($S_{Seg_k}$) can be calculated by

\[ T_{on-air}(k) = T_{I_{E}(k)} - T_{I_{S}(k)} = (I_E(k) - I_S(k)) \cdot T_{si} , \] (3)

\[ T_{ifi}(k) = T_{I_{E}(k)} - T_{I_{E}(k-1)} = (I_E(k) - I_E(k-1)) \cdot T_{si} . \] (4)

$T_{ifi}(k)$ denotes the time interval between the $S_{Seg_k}$ and the $S_{Seg_{k-1}}$. $T_{on-air}(k)$ is the on-air time of the $S_{Seg_k}$. Meter records the on-air time and inter-frame interval of each segment in array $T_{on-air}$ and $T_{ifi}$, respectively. In addition, the PAPR of $S_{Seg_k}$ can be calculated by

\[ \text{PAPR}(k) = \frac{\max [ R_{i}^{2} | I_{S}(k) \leq i \leq I_{E}(k)] } { \overline{S_{Seg_k}} } , \] (5)

where $\overline{S_{Seg_k}}$ denotes the average of the squared values of the elements in segment $S_{Seg_k}$. In addition, the averaged RSSI of segment $S_{Seg_k}$ can be calculated by

\[ \text{Avg}_k = \frac{ \sum_{i=I_{S}(k)}^{I_{E}(k)} R_{i} }{ I_{E}(k) - I_{S}(k) + 1 } . \] (6)

### 4.2.3 Treatment of No RSSI Sample and ACK Segment

As previously mentioned, the SPI preemption may result in no RSSI sample. Hence, in some pre-appointed sample time, meter may not be able to read the value of RSSI register. To enhance the segmentation component, we set the default value of no RSSI sample to the value of noise floor, and the value of no RSSI sample could be appropriately readjusted by satisfying the following rule.

**Rule:** $R_i$ is a no RSSI sample. If $R_{i+1}$ belongs to $S_{Seg_k}$ and the on-air time $T_{on-air}(k)$ is less than the minimum time of ZigBee transmission, meter sets $R_i$ to $\text{Avg}_k$.

Because meter will continue to sample channel RSSI until the channel is free, the no RSSI sample will never follow a high level sample.

In duty-cycled sensor networks, as shown in Figure 8, the sender transmits adjacent frames with an interval ($T_{ifi}$) for receiving the potential ACK. Once the appointed receiver successfully decodes the sender’s frame, it replies an ACK. The interval ($T_{ack}$) between the frame and ACK is set by system designer. $T_{ifi}$ should be at least longer than $T_{ack}$. In the default TinyOS system, $T_{ack}$ ranges between 192µs and 512µs [1] (hardware acknowledgement). For the platforms integrated with CC2420, the on-air time of an ACK is 352µs [20]. Because in some radio chip, such as CC2420, the ACK sender identification (ID) is not attached in the ACK payload, meter ignores the ACK segments by removing them from the arrays of $I_S$ and $I_E$. For the other segments whose on-air time is less than the minimum ZigBee frame transmission time ($T_{min}$), meter also removes them from the two arrays. In CC2420, the on-air time of the minimum packet is 576µs. For robustness, $T_{min}$ is set to 512µs (4 samples). After removing the non-ZigBee frame segments, meter uses the RSSI features to identify whether a segment indicates a ZigBee transmission in the next section.

### 4.3 Frame Identification

Except for the previously mentioned RSSI features (On-air time and PAPR), two additional significant characteristics of frame transmission in duty-cycled WSNs can be exploited to identify ZigBee frame.

**Identical on-air time:** The repeatedly transmitted frames are the same data packet, hence the on-air time ($T_{on-air}$) of each frame is the same.

**Fixed inter-frame interval:** For duty-cycled working mode, the inter-frame interval ($T_{ifi}$) is fixed. Denote the system default inter-frame interval as $T_{ifi}^{\text{valid}}$.

By exploiting the two new characteristics and the mentioned two RSSI features in Section 3.4, meter will identify the arrived ZigBee frames. Furthermore, meter will use the sampled RSSI features to determine the corresponding transmittor if none of the arrived frames has successfully been decoded. Note that the exploitation of RSSI features for identifying ZigBee transmission is first proposed by ZiSense [19]. ZiSense uses the sampled RSSI sequence to detect the existence of ZigBee transmissions and wakes up nodes accordingly. In this paper, we use the technique of ZiSense to identify ZigBee frames. The detail about the ZigBee identification presented in [19].

### 4.4 Number of Arrived and Decoded Frames

#### 4.4.1 Arrived Frames

By identifying the potential ZigBee frames, meter could count the number of arrived frames ($N_{frame}$) during its wake-up duration. One frame corresponds to a valid segment with specific start and end positions. By removing the non-ZigBee segments and invalid segments from both $I_S$ and $I_E$, the size of $I_S$ or $I_E$ is the number of arrived frames. Hence,

\[ N_{frame} = ||I_S|| = ||I_E||. \]

The accuracy of counting the number of arrived frames is evaluated in Section 5.2.1.

#### 4.4.2 Decoded Frames

Different from the number of arrived frames, the number of successfully decoded frame is easy to obtain. Once the radio chip decoded a frame (frame or ACK), a packet receiving event will be triggered to notify meter or other components. Meter records these overheard frames and the corresponding receiving time as shown in Figure 9(b). From the overheard frames, meter will extract three types information: (1)
the number of decoded frames \( N_{\text{decoded}} \); (2) the transmitter; and (3) a segment corresponding to each of the decoded frame. Note that the decoded frame could further enhance the ZigBee identification.

### 4.5 Determination of Transmitter

By identifying ZigBee transmission, meter knows the number of arrived frames. However, for each frame, the corresponding transmitter cannot be directly obtained from the RSSI features because the frame may not be correctly decoded.

#### 4.5.1 Useful Factors

Several factors can be used to determine the transmitter. First, the decoded frame carries the transmitter ID. Once successfully decoding a frame, meter knows the transmitter. Second, in duty-cycled working mode, \( T_{i,j}^{\text{valid}} \) is almost fixed. By checking whether the actual inter-frame interval \( T_{i,j} \) approximates to the system parameter \( T_{i,j}^{\text{valid}} \), meter is more confident determining whether or not the related frames are transmitted by the same sender. Note that the congestion backoff time is usually set to be larger than \( T_{i,j}^{\text{valid}} \), hence the probability that another sender transmits a frame exactly after a \( T_{i,j}^{\text{valid}} \) interval is very small. If the constraint of Equation 7 is satisfied, meter presumes that \( \text{SEG}_k \) and \( \text{SEG}_{k+1} \) are transmitted by the same sender. \( \delta \) is the parameter for segment clustering.

\[
|T_{i,j}(k) - T_{i,j}^{\text{valid}}| \leq \delta. \tag{7}
\]

Third, the averaged RSSIs of the frames transmitted by the same sender are relatively stable, which has been demonstrated by the real deployment of 300 nodes WSN presented in Section 3.4 and will be further discussed in the following subsection. If none of the arrived frames has been decoded, by using the averaged RSSI, meter can further infer who the sender is in the next section.

#### 4.5.2 Averaged RSSI

After segmentation, meter can determine that \( n \) successive segments are the frames transmitted by the same sender. Denote the segments vector as: \( \text{SEG} = \{\text{SEG}_1, \text{SEG}_{i+1}, \ldots, \text{SEG}_{i+n} \} \). However, if none of the arrived frames has been successfully decoded, meter further uses the averaged RSSI to determine the transmitter. The average RSSI \( R_{\text{avg}} \) of these segments is computed by

\[
R_{\text{avg}} = \frac{\sum_{k=1}^{i+n} \text{avg}_k}{n+1}.
\]

By comparing \( R_{\text{avg}} \) with the recorded RSSI \( \bar{R} \) of each neighbor \( A \), if

\[
|R_{\text{avg}} - \bar{R}(A)| \leq R_{\delta}, \tag{8}
\]

meter presumes \( A \) is a reasonable candidate transmitter of these frames. \( R_{\delta} \) is the valid RSSI bias. We set \( R_{\delta} \) to 1dBm according to the empirical study presented in Section 3.4. If only one candidate exists, meter presumes the candidate is the transmitter. The accuracy of this mechanism is given in the performance evaluation of Section 5.2.2. However, if multiple matched candidates exist or no matched candidate exists, meter records the number of arrived frames and the arriving timestamps corresponding to these candidate transmitters (or no candidate), and defers the determination.

#### 4.5.3 Deferred determination

In LPL-based WSNs, although the time clock of network nodes are not synchronized, the speed of time clock is stable and almost the same. Using this characteristic, meter will adequately determine which neighbor is the transmitter of the undetermined frames.

Each node maintains a bitmap of 10 bytes to clearly indicate whether or not it transmitted a packet in the past. In meter, one bit can cover 10 duty cycles. For example, if nodes’ wake-up interval is set to 512ms and there is a transmission event during the last 5 seconds, the corresponding bit should be set to 1, otherwise 0. Hence, a 10 bytes bitmap can record the transmission events for 400 seconds. The bitmap will be shared with neighbors by attaching to the free space of data or beacon packets. Each share does not necessarily require attaching all of the bitmap. Note that the data traffic of duty-cycled WSNs is low. Using one bit to cover 10 duty cycles makes sense.

Once receiving the bitmap from a neighbor \( A \) which is a recorded candidate transmitter, meter uses the recorded timestamp and the bitmap to determine whether or not \( A \) transmitted a data packet or beacon in some period which includes the recorded timestamp. If transmitted, this candidate transmitter is valid, otherwise invalid. Finally, if meter can determine only one of the candidate transmitters is valid, it presumes the candidate is the transmitter and delivers these information (transmitter, the number of arrived frames, the number of decoded frames) to upper estimation component. Otherwise, it ignores these records. The accuracy of deferred determination is given in Section 5.2.2.

#### Algorithm 1: Determination process

**Input:** segments vector SEG transmitted by the same transmitter: \( SEG = \{\text{SEG}_1, \text{SEG}_{i+1}, \ldots, \text{SEG}_{i+n} \} \);

**Output:** determination result.

- if Existing decoded frame then
  - Extracting transmitter ID;
  - return transmitter;
- else
  - According to Equation 8:
    - if Only one match then
      - return matched neighbor;
    - else if Multiple matches OR No match then
      - return Deferred determination;

#### 4.5.4 Determination Strategy

The detail determination process is given in Algorithm 1. Meter first exploits the decoded frame to exactly know the transmitter. If none of the frames has been decoded, meter uses the averaged RSSI \( R_{\text{avg}} \) to infer the candidate transmitters. If there is only one candidate, meter presumes the candidate is the transmitter. Otherwise, meter defers the determination.

#### 4.6 Integration with Link Estimator

Meter can be integrated into the state-of-the-art link estimator, 4-bit [12]. As the basic link monitoring component, meter provides the essential information for accurate link estimation. The information consists of the number of arrived frames, the number of successfully decoded frames, and the corresponding transmitter. Meter sits beneath MAC layer to monitor each frame. It is compatible with low power listen-
The table below summarizes the system parameters used by the meter in Table 2.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{th}$</td>
<td>3dBm</td>
<td>RSSI threshold to detect the start and end points of a segment</td>
</tr>
<tr>
<td>$T_{si}$</td>
<td>128μs</td>
<td>RSSI sampling interval</td>
</tr>
<tr>
<td>$T_{ack}$</td>
<td>between 192μs and 512μs</td>
<td>Acknowledge frame wait period</td>
</tr>
<tr>
<td>PAPR_{ZigBee}</td>
<td>1.3</td>
<td>Maximum PAPR of a valid ZigBee segment</td>
</tr>
<tr>
<td>$R_b$</td>
<td>1dBm</td>
<td>Valid averaged RSSI bias for determining transmitter</td>
</tr>
<tr>
<td>$T_{valid}^{i_f_i}$</td>
<td>2ms</td>
<td>System parameter of inter-frame interval</td>
</tr>
<tr>
<td>$T_{backoff}$</td>
<td>2.5ms</td>
<td>Congestion/initial backoff time</td>
</tr>
<tr>
<td>$\delta$</td>
<td>512μs</td>
<td>Parameter for segment clustering</td>
</tr>
</tbody>
</table>

In our experiments, each node generates a packet with a fixed inter-packet interval (IPI), and wakes up periodically. If there is no specific statement, IPI is set to 1 minute and wake-up interval is 512ms. The packet payload size is randomly selected from [60,100] bytes. In MAC layer, we restrict the parameter value of congestion/initial backoff time ($T_{backoff}$) and inter-frame interval ($T_{valid}^{i_f_i}$) by

$$T_{backoff} > T_{valid}^{i_f_i}. \quad (9)$$

Each experiment lasts for at least 30 minutes and repeated at least three times.

### 5.2 Accuracy of Meter

In this subsection, we conduct experiments to evaluate the accuracy of link quality which is estimated by the meter. The accuracy of the meter involves several aspects, including the number of arrived frames and the determination of the corresponding transmitter. We conduct experiments here to demonstrate the accuracy of the meter from the three aspects.

#### 5.2.1 Number of Arrived Frames

Counting the number of arrived frames is a key important function of the meter. Here, we conduct experiment to evaluate the accuracy of counting in an indoor testbed with 22 Telosb nodes. We set each node’s radio working mode to always on. In this network, there is no data packet transmission. Each node periodically broadcasts routing beacon to maintain network topology. During each routing beacon cycle, the same node periodically broadcasts routing beacon to maintain network reliability of collection tree protocol (CTP) [24] in Section 5.4.3.

#### 5.1 Evaluation Setup

We implement the meter in TinyOS-2.1.1 [2] based on TelosB platform [3] that equips with 802.15.4-compliant CC2420 radio chip [1]. The meter integrates into 4-bit to provide accurate and timely estimation for each link. It sits beneath the default LPL protocol (X-MAC [11]). Based on LPL, CTP exploits the estimated link quality to select an optimal routing to deliver data packet. To begin with, we give the summary of the system parameters used by the meter in Table 2.
uses the above mentioned four arrays to determine whether or not an identified ZigBee frame is indeed a routing beacon wake-up frame. According to the time constraints, we can clearly detect the false negative and false positive of *meter*. For example, during the interval between two successively received frames that are attached the sequence number \( n \) and \( m \) (\( m \) is larger than \( n \)), *meter* detects \( k \) ZigBee frames. If \( k = m - n \), we think *meter* has accurately detected each frame. If \( k < m - n \), we think *meter* missed \( m - n - k \) frames, which is so called false negative (FN). Otherwise, we think *meter* falsely detected \( k - (m - n) \) frames, and this is false positive (FP). Overall, there are 484 unidirectional links, and totally 48400 computational results. We first compute the probabilities of accurately detected part, false negative ratio (FNR), and false positive ratio (FPR), which are listed in Table 3. As shown in it, 97.3% wake-up frames can be successfully identified by *meter*. The FN and FP are only account for 0.8% and 1.9%, respectively. The FN is likely caused by the lose of SPI resource, and FP is possibly caused by external interference. Furthermore, we extract the segment vectors which contain FN or FP. For a specific segment vector, we compute the FNR and FPR by using the total of FN-s/FPs to divide it by 50. We plot the cumulative distribution function (CDF) of these ratios in Figure 10. As shown in the figure, even in these segment vectors, more than 92.5% segment vectors are with less than 3 (6%) FN and 89.4% segment vectors are with less than 4 (8%) FP. The experimental results demonstrate that *meter*’s frame identification can accurately detect and identify ZigBee frames.

### 5.2.2 Determination of Transmitter

By accurately counting the arrived frames, another key important function of *meter* is determination of the corresponding transmitter. To evaluate the accuracy of transmission determination by *meter*, in this section, we conduct experiments in the indoor testbed and outdoor playground using 22 Telosb nodes. We use a center node to synchronize all network nodes by periodically broadcasting its local time using the highest power level (level 31 in CC2420). The power level of the other nodes is set to 2 in indoor testbed and 5 in the outdoor testbed to form a multiple-hop network, respectively. According to network synchronization, each node transmits a data packet during an appointed cycle in turn. One cycle lasts for 10 seconds. For example, node \( i \) will transmit its data packets and routing beacons in the \( k \)th cycle when \( k \% 22 = i \). Node 0 is the sink node in our experiments. Nodes will not forward the received data packets if the current cycle is allocated to a specific node. But in the \( k \)th cycle, if \( k \% 22 = 0 \), all network nodes will forward the recorded information to sink node through multiple-hop transmission.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Min</th>
<th>Max</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>No interference (channel 26)</td>
<td>1</td>
<td>11</td>
<td>4.7</td>
</tr>
<tr>
<td>With interference (channel 19)</td>
<td>1</td>
<td>8</td>
<td>3.6</td>
</tr>
</tbody>
</table>

Table 4: Size of segment vector after clustering in different scenarios.

For each experiment, by identifying ZigBee frame and clustering frames, *meter* determines the corresponding transmitter of each segment vector. *Meter* uses the decoded frame (DF), averaged RSSI (AR), and deferred determination (DD) to determine the transmitter. For each segment vector, the transmitter can be accurately inferred according to the synchronization mechanism and the allocated transmission cycle. Then, we will know whether or not the transmitter determination of *meter* is correct.

We first compute the minimum, maximum, and averaged number of frames in segment vectors, and list them in Table 4. Then, we plot the proportion of each information used for transmitter determination in Figure 11. As shown in the figure, the transmitter of more than 60% segment vectors can be directly determined by using the decoded frame, no matter in indoor or outdoor testbed. For the remainder of the vectors, *meter* uses averaged RSSI and deferred determination to infer the transmitter. In indoor scenario, because routing and non-routing links have a low degree of differentiation, the average RSSI can only infer a small part (about one third) of these segment vectors without decoded frame, and the other two thirds segment vectors exploit deferred determination to obtain the transmitter. In outdoor testbed, *meter* can exploit the spatial diversity of routing and non-routing links to infer transmitter. Furthermore, we compute the accuracy of using AR and DD to determine transmitter, and list them in Table 5. According to the restriction of Equation 8 and the requirement of only one satisfied candidate, the accuracy of AR can reach to 95.5%. The deferred determination also can determine transmitter with high accuracy. But note that about one fifth deferred determinations are ignored because of the lack of a definite result.

In this section, the experimental results have amply demonstrated the accuracy of *meter*. In the following section, we continue to show the timeliness of *meter*. 

![Figure 12: CDF of the time interval between two successive link updates. By integrating *meter* into 4-bit, the update rate is significantly increased.](image-url)
Figure 13: Performance of multi-hop indoor testbed experiment with 22 nodes and outdoor experiment with 30 nodes. By integrating meter into 4-bit, network performance is improved.

5.3 Timeliness of Meter
To illustrate the timeliness of meter, we conduct experiments in both indoor and outdoor testbed. In each scenario, we use the following configuration of network protocols: (1) CTP, LPL, and 4-bit; (2) CTP, LPL, and 4-bit integrated with meter. Uniformly, we set our networks’ work channel to 19, the transmission power of CC2420 is set to level 5, and each node generates a data packet each 2 minutes.

4-bit maintains an estimation window $w$ to update the quality of each link. In 4-bit, the size of $w$ is 5. We do not change the window size by integrating with meter, but the window unit is changed from a data/beacon packet to a wake-up frame. Then, each node records the time interval between two successive link quality updates for each link (routing link or non-routing link), and forwards these time intervals to sink. We plot the CDF of these time intervals in Figure 12. As shown in the figure, by integrating meter into 4-bit, the update rate of link quality is significantly increased. By setting IPI to 2 minutes in this experiment, almost all (97.9%) links can be updated within 200 seconds because meter can accurately count the arrived frames (no matter data packet or routing beacon). By monitoring the potential active links once they transmit data/beacon frames, meter will trigger fast update of link quality. On the other hand, without integrating with meter, 4-bit updates link quality slowly, so that the vast majority of (more than 50%) links can not be updated one time for eight minutes. Compared with 4-bit, meter can guarantee the timeliness of link quality estimation because of its inherent advantage: as long as a neighbor transmits data/beacon packet during the wake-up duration, meter can count the number of arrived frames and lost frames in real time.

As mentioned in the previous subsections, meter can provide accurate and timely link estimation. In the next section, we continue to evaluate the performance of meter applied in a data collection tree.

5.4 Collection Tree Protocol Performance
Link estimator has fundamental impact on the performance of data collection. In this section, we explore how well the Collection Tree Protocol (CTP) [24] performs by using meter. CTP is the default data collection protocol in TinyOS and it represents a canonical link layer client. To test its performance, we conduct experiments in both indoor testbed with 22 node and outdoor testbed with 30 nodes. In the indoor testbed, all sensor motes are hung on a vertical wall. In the outdoor testbed, all sensor motes are put on the ground with glazed tiles. The outdoor testbed is surrounded on three sides by office build where about 18 WiFi APs interfere with the network. The multi-path effect may be more serious for the outdoor testbed. Both indoor and outdoor testbeds are surrounded by many WiFi APs. The work frequency of some APs overlays with the 19th channel of ZigBee. We separately test the performance of CTP by integrating or without integrating meter into 4-bit. By collecting recorded information of each node, we compute the distribution of packet delivery ratio (PRR), energy consumption, and transmission cost.

5.4.1 One-hop Delivery Cost vs. Path Length
In both indoor and outdoor experiments, each node records the one-hop transmission count of each data packet and forwards these records to sink. We plot the CDF of single-hop transmission count in Figure 13(c). By embedding meter into 4-bit, the one-hop transmission count does have been reduced in the indoor experiments (Indoor: 4-bit and Indoor: 4-bit+meter in the figures) or outdoor experiments (Outdoor: 4-bit and Outdoor: 4-bit+meter in the figures). The average single hop transmission count of indoor experiments is reduced from 1.73 times to 1.32 times (reduced by about 23.7%) by integrating meter into 4-bit, and reduced from 3.49 times to 1.97 times (reduced about 43.5%) in outdoor testbed. The decreased transmission count makes nodes take more time in sleep state, so as to further save energy consumption which is demonstrated in the next subsection. Moreover, the reduced transmissions count can further decrease packet collisions caused by hidden terminal and reduce suppressed transmission opportunities resulting from exposed terminal.

Additionally, the distribution of path length (hop count) from each node to sink node is also plotted in Figure 13(d). As shown, the averaged path length of indoor networks is reduced by about 7% by using meter to estimate link quality, and that of the outdoor networks is reduced by about 18.3%. Because meter can accurately estimate each link, CTP uses the optimal links to construct network routing, so as to find the shortest path from each node to sink. The reduced single-hop transmission count and reduced path length further improve the network performance represented in energy consumption and network reliability, which are discussed in the following subsections in detail.

5.4.2 Energy Consumption
We also plot the distribution of energy consumption (duty cycle) across different scenarios in Figure 13(b). In the indoor experiments, there is no significant difference between the energy consumption of CTP+4-bit and that with meter, except for decreasing the probability of extremely high energy consumption. However in the outdoor experiments, by
integrating meter into 4-bit, the average duty cycle is significantly decreased from 15.1% to 9.6%. For the former, the little improvement of energy efficiency is due to the small-sized testbed that reduces the diversity of wireless links. Hence once the current routing is severely degraded, the probability that there is another high quality routing is small. This case is also indicated from the similar distributions of single hop transmission count plotted in Figure 13(c). While in outdoor experiments, by adopting meter, the reduced average single hop transmission count can save energy consumption.

5.4.3 Network Reliability

In this section, we compare the network reliability of CTP+4-bit with that of CTP+4-bit integrated with meter. By plotting the CDFs of nodes’ averaged PRR in different scenarios in Figure 13(a), we can conclude that the network reliability (PRR) is averagely improved by 2.3% in the indoor experiments, and 5.6% in outdoor experiments.

As shown in the figure, by integrating with meter, nodes’ PRRs are relatively high and distributed in a smaller range. Although channel RSSI is quite different at the location of different nodes (especially in outdoor testbed), due to the accuracy and timeliness of meter, CTP can effectively select the optimal routing to forward packets. If the currently used link is severely degraded, meter can quickly find the temporarily available link. Hence, CTP could utilize the best available routing if there exists one, so as to decrease the effect of local interference on network performance.

6 Discussion

In this section, we further discuss several technological problems of meter, and then state the possible optimization on meter.

6.1 Reliability and Efficiency

By using meter, each node maintains the links to all connected neighbors by passively and precisely counting the wake-up frames. Each node continuously samples the RSSI after it turns on the radio. Then with the sampled RSSI sequence, it counts the wake-up frames that either are successfully decoded or identified by ZigBee recognition. Our ZigBee recognition method is based on the unique characteristics of ZigBee low power transmission that is different from other coexistence interference (e.g. WIFI, Bluetooth, Microwave Oven). Furthermore, the time complexity of ZigBee identification algorithm is $O(n)$, where $n$ is the number of RSSI samples. The time overhead is very small. Additionally, the recognition accuracy is guaranteed when sampled RSSI sequences of the on-going transmission are higher than the noisy floor. Hence, in duty cycled mode, nodes can reliably and efficiently estimate each link’s quality by guaranteeing accuracy and timeliness by using meter.

6.2 Applicability

Similar to 4-bit, meter can be applied to large scaled WSNs. In a connected network, the only factor that affects the performance of meter is the node deployment density. In a very dense network, it is difficult for a node to discriminate the RSSI of neighbors’ data/beacon transmission. Hence, the probability of determination of transmitter is relatively low, resulting in more ignored frames. The case may affect the timeliness of meter. But we believe it is better than 4-bit.

Furthermore, our signal identification method is mainly based on the unique characteristics of different physical layer technologies. With more delicate classifier and fine-grained RSSI samples, our method can also identify the signal of WIFI, Bluetooth or Microwave Oven. However, most of sensor motes (e.g. TelosB, MicaZ, IRIS) adopt ZigBee (802.15.4) as the physical layer technology. Further considering the limited computation and memory resources, our method targets on distinguishing ZigBee from other coexistence interference, but does not further classify the categories of coexistence interference. Moreover, our ZigBee identification algorithm and parameter settings are device independent so that the proposed method can run on other sensor motes besides TelosB with CC2420.

6.3 Compatibility

In our paper, we just simply adopt the independent binary link model without considering the dependency of communication channel. We focus on providing a more precise and real-time counting method to provide more accurate model construction. Some existing works on link quality prediction [8] are from another angle to reduce the negative effect of link quality on network performance. Hopefully, our methods can be directly adopted by the predication based link models to improve the accuracy of model construction.

7 Related Work

Basically, the link quality estimation process involves three steps: link monitoring, link measurements, and metric evaluation [5]. The metric is mathematical expression denoting link quality within an estimation window. We refer to this metric as link quality estimator (LQE). The LQE evaluation requires link measurements. For example, to evaluate the PRR (packet receipt ratio) [22] estimator, link measurements consist in extracting the sequence number from each received packet. Link monitoring defines a strategy to have traffic over the link allowing for link measurements. Link monitoring is the basic component of link estimation scheme. The performance of link monitoring has great influence on the performance of link estimation. There are three kinds of link monitoring [5]: (1) active link monitoring, (2) passive link monitoring, and (3) hybrid link monitoring.

Active link monitoring. In active link monitoring, a node monitors the links to its neighbors by sending beacon packets. Beacon packets can be sent either by broadcast [22], or by unicast [23]. Broadcast beacon packets involve no link-level acknowledgments or retransmissions, in contrast to unicast beacon packets. Beacon packets are generally sent at a certain rate, which yields a tradeoff between energy-efficiency (low rates) and accuracy (high rates). An adaptive beaconing rate [24] [25] might provide a good balance for this tradeoff.

Passive link monitoring. Unlike active link monitoring, passive link monitoring exploits existing traffic without incurring additional communication overhead. In fact, a node listens to transmitted packets, even if these packets are not addressed to it (overhearing) [26] [12]. It can also listen to acknowledgments of messages sent by different neighbors [27].

Passive link monitoring has been widely used in WSNs due to its energy-efficiency compared to active link monitoring [28] [30] [27] [26]. However, passive monitoring incurs the overhead of probing idle links [23] found that overhearing involves expense of significant energy. In addition, when the network operates at low data rate or unbalanced traffic, passive link monitoring may lead to the lack of up-to-date link measurements. Consequently, it leads to inaccurate link quality estimation.
Hybrid link monitoring. The use of a hybrid mechanism combining both active and passive monitoring may yield an efficient balance between up-to-date link measurements and energy-efficiency [23]. For instance, Gnawali et al. [24] introduced a hybrid link monitoring mechanism for performing both link quality estimation and routing advertisements. Active link monitoring consists in broadcasting beacons with a non-fixed rate. Rather, a trickle algorithm [25] is used to adaptively tune the beaconing rate: Initially, the beaconing rate is high and decreases exponentially until it reaches a certain threshold. When the routing layer signals some problems such as loop detection, the beaconing rate resets to its initial value. Active link monitoring is coupled with passive link monitoring, which consists in hearing received acknowledgments from neighbors (that represent next hops).

It was argued by several recent studies that link quality estimation where link monitoring is based on data traffic (passive) is much more accurate than that having link monitoring based on beacon traffic (active) [24] [31] [32]. However, the passive link monitoring is exploited to estimate the link quality between routing links in existing works. The vast majority of links are measured using broadcast beacon. Due to the infrequency of broadcast beacon in a relative stable network, the quality of these links is usually outdated. Furthermore, in duty-cycled and asynchronous WSNs, both active and passive monitoring are coarse-grained. Different from existing works, meter achieves both accurate and real-time link monitoring.

8 Conclusion
In this paper, we propose meter which exploits the feasibility of ZigBee identification of short-term signal strength sequence, but does not only depends on decoding frame. First, meter obtains the signal strength sequence during the radio is on. Then meter identifies the number of ZigBee signal pieces in signal strength sequence. Finally, with the information, which include the address of decoded frame, sender’s traffic schedule, the interval between adjacent pieces and the average signal strength of individual signal piece, meter identifies the transmitter address of each signal piece. With decoded frames, meter fully counts all frames to accurately update link status in time. We implement meter in TinyOS and evaluate its performance through extensive experiments on indoor and outdoor scenarios. The results demonstrate that meter can significantly improve the performance of the state-of-the-art link estimation scheme.

Acknowledgment
The authors would like to thank the shepherd, Dr. Pei Zhang, for his constructive feedback and valuable input. Thanks also to anonymous reviewers for reading this paper and giving valuable comments.

This study is supported in part by NSFC No. 61472067, NSFC No. 61472217, National Key Technology R&D Program No. 2013BAH33F02, Key Technologies Research and Development Program of Sichuan Province No. 20130Z0006, and NSFC No. 61102444

9 References