An Approach to Detect Anomalous Degradation in Signal Strength of IEEE 802.15.4 Links

Songwei Fu*, Matteo Ceriotti*, Yuming Jiang†, Chia-Yen Shih*, Xintao Huan*, Pedro José Marrón*
*Networked Embedded Systems, University of Duisburg-Essen, Germany
†Norwegian University of Science and Technology (NTNU), Norway

***Abstract*—Accurate detection of the channel quality degradation is crucial for applying effective remedial actions to ensure the reliability of IEEE 802.15.4 links. Without knowing the channel quality is degraded, remedial actions may lead to more packet losses, e.g., increasing transmission power may cause even more interference. In this work, we aim to detect the channel quality degradation that turns a good link into a bad one, based on the received signal strength of radio links. The detection should be accurate and robust to diverse channel characteristics and dynamic environmental changes. To achieve this, we propose RADIUS, a lightweight approach that lays its foundation on a thresholding technique based on Bayesian decision theory and combines it with techniques for adapting to environmental changes. Extensive evaluation of RADIUS on a testbed shows that the employed Bayes thresholding technique outperforms two relevant state-of-the-art thresholding techniques by providing a higher accuracy consistently for all links across the network. Besides, RADIUS is able to keep a low error rate of detection (5.78% on average) in a 72-hour experiment, adapting to environmental changes. Furthermore, we developed an exemplary application of RADIUS to show how an existing transmission power tuning scheme can benefit from using RADIUS as an accurate and robust trigger for taking remedial actions.**

I. Introduction

The performance of Wireless Sensor Networks (WSNs) can be heavily affected by packet losses over radio links due to different causes, e.g., external interference [1], packet collisions [2], and degraded channel quality [3]. Applying appropriate remedial actions [4] [5] on the lossy links may enhance link performance. However, actions performed without diagnosis or based on erroneous diagnosis may have no effect or even negative effects, e.g., increasing output power can cause even more interference. Therefore, it is essential for a WSN to accurately diagnose link packet losses for taking effective actions in order to ensure link and network reliability.

In this paper, we focus on one of the major causes for the packet losses of IEEE 802.15.4 links, that is the degradation in channel quality experienced by radio links. We aim to provide an approach to detect the channel quality degradation based on a direct measurement of the channel at the radio hardware – Received Signal Strength Indicator (RSSI). Many previous studies revealed the degradation in both RSSI and link Packet Reception Rate (PRR) under the impact of temperature [6], human presence [7], climate condition and terrains [8], etc. To differentiate from the inherent random fluctuation in RSSI, we refer the RSSI degradation that turns a good link (high link PRR) into a bad one (low link PRR) as *anomalous RSSI degradation* or *RSSI anomaly* throughout the paper.

Detecting such anomalous RSSI degradation of radio links with a high accuracy is not as simple as one might imagine. Many empirical studies [3] [9] show that RSSI is not well correlated with PRR, and hence, not an accurate indicator of PRR. Other works [4] [5] prove the existence of RSSI thresholds, over which the RSSI results in a PRR of good links. This allows us to use RSSI to estimate whether a link is a good link. However, RSSI dropping below those thresholds does not necessarily mean that a good link turns into a bad link. Finding the RSSI thresholds to accurately indicate a transition from a good link into a bad one is indeed a challenging task.

To demonstrate this challenge, we evaluate an intuitive approach, in which the runtime averaged RSSI of each link over a sliding window is compared with a predefined RSSI threshold to make decisions about RSSI anomalies. We apply such an approach to real data traces and vary the RSSI threshold in a wide range. The resultant detection error rates are presented in Figure 1. The figure shows the typical performance of such an approach: independent of the sliding window size, while a predefined RSSI threshold may work well for one link (e.g., -85 dBm for Link 2-1), the same threshold may not work well for different links (e.g., Link 1) or the same link at a different time (Link2-2).

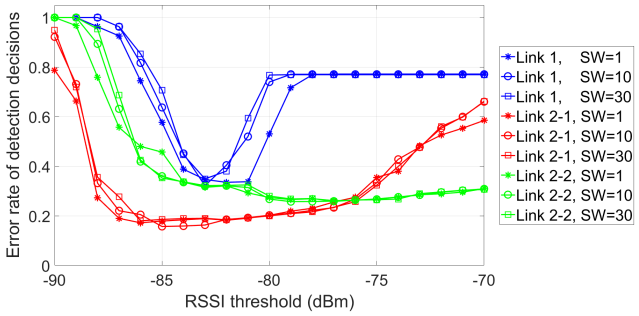

Fig. 1: The performance of an intuitive approach based on predefined thresholds. SW refers to sliding window size.

The reason for this is that the channel characteristics experienced by radio links (e.g., RSSI mean and variance) may differ from link to link and also vary over time due to environmental changes [10]. Hence, the best RSSI threshold that discriminates between RSSI values from a good link and those from a bad link may vary both spatially and temporally, depending on the specific channel characteristics of each link. In fact, there is still a lack of such an approach that takes the spatio-temporal characteristics of the radio channel into account to detect anomalous RSSI degradation accurately.

The main question investigated in this paper is therefore: Is it possible to design an approach to detect anomalous RSSI degradation of radio links, which fulfills the following requirements? First, the approach shall be lightweight due to resource-constrained sensor nodes. Second, it should provide a high detection accuracy. Third, the high accuracy should be consistent for all links across the network and robust over time. In other words, manually tuning the performance of the approach for each link or at different times should be avoided.

To answer this question, we propose RADIUS, a lightweight yet accurate and robust approach to detect locally at sensor nodes the anomalous RSSI degradation of inbound radio links. Central to the design is a thresholding method based on the Bayesian decision theory to achieve high detection accuracy, generating an RSSI threshold for each individual link according to its specific channel characteristics. In addition, RADIUS employs techniques to capture accurate channel characteristics within a short time, when the performance of the network satisfies the user requirements. Furthermore, RADIUS employs techniques for accurate detection and also the techniques for continuously updating the thresholds to adapt to environmental changes.

We implement RADIUS and evaluate its performance and overhead on an indoor testbed of 32 sensor nodes. We compare the employed Bayes thresholding method with two relevant state-of-the-art thresholding methods from the literature [11] [12]. The comparison results show that the Bayes thresholding provides the highest accuracy among all three methods. Unlike the other two methods, the accuracy for Bayes thresholding is consistent for all links across the network without the need of fine-tuning of its performance for each link. In a 72-hour experiment, the RADIUS system keeps a low error rate robust to environmental changes over time. Overall, it achieves an averaged error rate of 5.78% for all links of the testbed.

Finally, RADIUS can be used as an accurate and robust trigger for remedial actions to restore link and network performance. To demonstrate this, we developed an exemplary application that utilizes RADIUS as a trigger to adjust the transmission power of sensor nodes to maintain link PRR. The outcome is compared with a transmission power tuning scheme from the literature [4] to show an example of how existing works can benefit from the capability of RADIUS.

The rest of the paper is organized as follows. Section II reviews the related work. Section III presents the RADIUS approach in details. Section IV presents the experimental evaluation of different thresholding methods, the overall system and an exemplary application of RADIUS. Finally, Section V concludes the paper and discusses future work.

II. Related Work

Most existing diagnosis approaches in sensor networks focus on network-wide diagnosis [13] [14] [15]. They are able to detect and localize lossy links by monitoring metrics such as link PRR or other software-based LQEs [3]. However, such approaches are unable to determine the underlying cause of link packet losses, e.g., whether a loss is due to degraded channel quality or the internal interference caused by sensor nodes themselves. Without truly knowing the cause, applying the most direct remedial action such as increasing transmission power [4] may create more interference, leading to more packet losses. In order to determine the cause of low link PRR, e.g., the degradation in channel quality, we need information from the PHY layer. In RADIUS, we use RSSI, a direct channel quality attribute resident within every received packet.

The accuracy of detecting the channel quality degradation is also important for the effectiveness of remedial actions. E.g., for a transmission power tuning scheme, a wrong detection decision indicates it either fails to improve the PRR or wastes energy. Many previous studies [3] [6] [7] [8] [9] analyzed the relation between RSSI and link PRR in different scenarios. They revealed that RSSI is not well correlated with PRR, especially in the transitional region. This fact together with the random nature of RSSI [3] makes our problem a challenging task. Data smoothing may mitigate the problem, but it is not very effective in our case, as showed in Figure 1. In RADIUS, we tackle this problem from a different angle, that is to find the best possible thresholds tailored to each link. As Figure 1 shows, a good threshold (or thresholding method) is very effective to increase the detection accuracy.

Problems like detecting anomalous behaviors (in our case, anomalous degradation in RSSI) are typically solved using anomaly detection techniques. Powerful machine learning-based techniques such as clustering [16], neural networks [17] and support vector machine [18] are resource-hungry and thus do not fit the resource-constrained sensor nodes. Decision tree classifiers, on the other hand, consist of a set of simple rules once the classifiers are trained. In [1], sensor nodes identify the type of interference based on a decision tree. Nevertheless, for numeric attribute such as RSSI, we still need to find the optimal threshold for classification purposes, which is exactly what we aim to solve with this work.

Measure-based statistical techniques (e.g., mean, variance, maximum) are the most widely used techniques for thresholding in WSNs due to their low overheads. Two popular techniques of this category are Percentile-based and Chebyshev inequality-based thresholding techniques. An example of the former technique is Memento [19], which uses an empirical CDF of missing heartbeat numbers to detect sensor failures. Another example is RASID [12], which defines a threshold at a given percentile after the density function is estimated. In other cases, when the underlying probability distribution is not known a priori, the Chebyshev thresholding technique has often been applied. For instance, it is used in [11] to troubleshoot the network performance issues and in [20] for target detection in WSNs. However, our investigation, as to be shown later in this paper, reveals that both Percentile and Chebyshev thresholding methods do not achieve high accuracy when links experience diverse channel characteristics.

Bayesian decision theory [21] has been found useful in many scientific and engineering fields. A classical example is the detection of binary digits “0” and “1” in a noisy channel. In RADIUS, we employ the Bayesian decision theory to derive

the optimal RSSI thresholds for each monitored link, through which a minimal error rate is mathematically guaranteed with a low overhead. Additionally, RADIUS employs different techniques to improve the accuracy and maintain it by adapting to environmental changes.

III. The RADIUS Approach

In this section, we give the details of the RADIUS approach. We start with an overview of the approach, followed by the details of the system modules.

A. RADIUS Overview

Figure 2 shows an overview of the RADIUS system architecture. Such a system is implemented on each sensor node to monitor the inbound links and report to high-level applications the detected degradation in channel quality experienced by the monitored links, based on the RSSI values of the messages received from each link. RADIUS handles the message stream of each inbound link independently. However, it can process different types of messages received from the same link, e.g., the application packets either originated from direct child nodes in the routing tree or forwarded by them, beacon messages of routing protocols, and if needed, the probing messages initiated by RADIUS itself.

The RADIUS approach runs in two phases: (1) A short *training* phase, during which RADIUS constructs a normal profile of RSSI values for each inbound link when the performance of the deployed network (e.g., end-to-end PRR) satisfies the user requirements. An RSSI threshold is generated accordingly for each link based on its specific normal profile. (2) A *detection* phase, in which RADIUS compares the runtime smoothed RSSI with the generated threshold to decide whether there are RSSI anomalies or not. It also updates the stored normal profiles so that it can adapt to environmental changes.

During the training phase, the *Profile Construction Module* records for each inbound link the RSSI values of incoming packets and estimates the number of RSSI values required by each link to capture the various channel characteristics within a short time. At the end of this phase, this module will create for each link a *normal profile* consisting of the average and standard deviation of the recorded RSSI values. The profiles of all links are constructed concurrently in this short training phase. Based on the generated normal profiles, the *Thresholding Module* computes for each monitored inbound link an RSSI threshold that is used later by other modules to detect RSSI anomalies with a minimal error rate.

During the detection phase, the *Detection Module* examines the RSSI values of each inbound link and decides whether there are RSSI anomalies or not. This operation is applied to each link independently. If needed, the module notifies the detected RSSI anomalies to the network administrator or higher-level software modules. The *Profile Update Module* updates the normal profiles constructed in the training phase. The RSSI thresholds are updated accordingly by the Thresholding Module in order to adapt to environmental changes.

We now introduce the core module of RADIUS – the Thresholding Module, followed by other system modules.

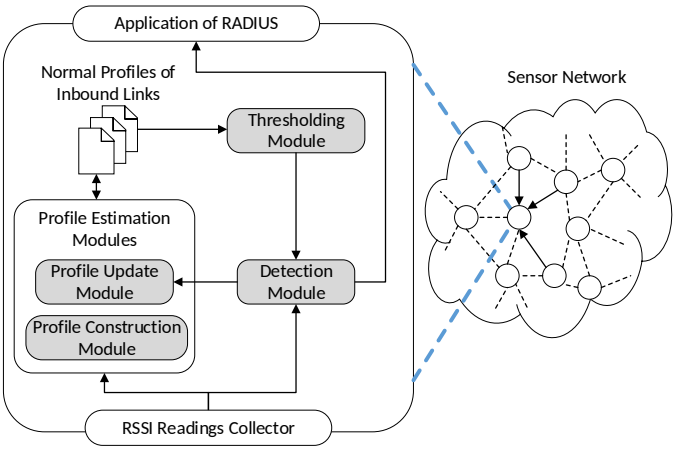

Fig. 2: The RADIUS approach.

B. RSSI Thresholding: The Bayes Threshold

The purpose of the Thresholding Module is to generate for each inbound link the optimal RSSI threshold such that the rate of making wrong decisions about whether there are RSSI anomalies is minimized. Such error rate is expressed in terms of a *false positive* rate (FPR) and a *false negative* rate (FNR). The former is the rate of false alarms, i.e. the ratio of detecting RSSI anomalies while the link is still a good link (PRR above a threshold); the latter is the rate of missing events, i.e. the ratio of no-anomaly decisions while the link has turned into a bad link (PRR below a threshold).

Mathematically, minimizing the error rate of such a decision problem has been comprehensively studied under the Bayesian decision theory. Let H_g and H_w respectively denote a link being a good link and a bad link. Let E denote an error (either a false positive or a false negative). Then, based on the Bayesian decision theory, $P(E)$, the probability of error, can be expressed in terms of conditional probabilities as follows:

$$P(E) = P(E|H_g)P(H_g) + P(E|H_w)P(H_w), \qquad (1)$$

where $P(H_g)$ is the *a priori* probability of a link being a good link, and $P(H_w) = 1 - P(H_g)$. Consequently, $P(E|H_g)$ is the probability of error when the link is a good link, i.e. the false positive rate, and $P(E|H_w)$ is the probability of error when the link is a bad link, i.e. the false negative rate.

We assume that RSSI follows a normal distribution $N(\mu, \sigma)$, which has been experimentally validated for low-power radio links of WSNs [5] [22]. We further assume for simplicity that the RSSI distributions of the same link being as a good link and as a bad link, while with different means μ_g and μ_w respectively, have similar standard deviation σ. With such assumptions, the probability density functions of RSSI for the good link $f_g(x)$ and for the bad link $f_w(x)$ are as follows:

$$f_g(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left\{-(x-\mu_g)^2/2\sigma^2\right\} \qquad (2)$$

$$f_w(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left\{-(x-\mu_w)^2/2\sigma^2\right\} \qquad (3)$$

Based on Equations 2 and 3, the Bayes error $P(E)$ can be

expressed as a function of the threshold τ:

$$P_E(\tau) = \int_{-\infty}^{\tau} f_g(x)dx \cdot P(H_g) + \int_{\tau}^{\infty} f_w(x)dx \cdot P(H_w) \quad (4)$$

An example of the above error rates is plotted in Figure 3, where the false positive rate and the false negative rate are marked as shaded areas with respect to an RSSI threshold τ.

We minimize $P(E)$ by letting $d\Big(P_E(\tau)\Big)/d\tau = 0$. We can then obtain the optimal RSSI threshold T_{Bayes} and the resultant minimized error rate P_E as follows:

$$T_{Bayes} = \frac{1}{2}(\mu_g + \mu_w) + \frac{\sigma^2 ln(P(H_w)/P(H_g))}{\mu_g - \mu_w} \quad (5)$$

$$P_E(\alpha) = Q\left[\alpha - \frac{1}{2}\alpha^{-1} ln\Big(P(H_w)/P(H_g)\Big)\right] P(H_g) + Q\left[\alpha + \frac{1}{2}\alpha^{-1} ln\Big(P(H_w)/P(H_g)\Big)\right] P(H_w) \quad (6)$$

where $Q(x)$ is a Q-function [23] and α is defined as:

$$\alpha = (\mu_g - \mu_w)/2\sigma. \quad (7)$$

We refer the RSSI threshold that is computed using Equation 5 as the Bayes threshold. The Thresholding Module computes the Bayes threshold for each inbound link to achieve minimized error rates of detecting RSSI anomalies.

If we set μ_w in Equation 5 to the RSSI value of the border of the transitional region (e.g., -89 dBm) reported in previous experimental studies [4] [24], the Bayes threshold then simply depends on two factors: (1) the normal profile of the link, i.e. (μ_g, σ), and (2) a thresholding parameter $P(H_g)$. While the normal profile of each link is generated locally at sensor nodes, the setting of $P(H_g)$ is a user-defined value for all sensor nodes across the network. In Section IV-B, we will study the impact of the thresholding parameter $P(H_g)$ on the detection performance and compare the Bayes threshold with two other state-of-the-art thresholding techniques.

C. Profile Construction: Estimating Training Set Size

The purpose of the Profile Construction Module is to capture, within a short time, the RSSI characteristics of each link while the link is still a good link. Specifically, it constructs a normal profile (μ_g, σ) for each link based on a set of RSSI values collected in the training phase in which we assume all links are good links directly after on-site deployment of the network. The user can decide when the training phase shall start by checking e.g., the end-to-end PRR.

The size of the training set for profile construction has a direct impact on the *training delay* (i.e. the time needed for each link to construct the normal profile) and the estimation error of μ_g and σ, and hence a great impact on the system accuracy. In addition, such training set size shall be link dependent, e.g., a small training set may suffice for a stable link while a larger size is required for links with highly fluctuating RSSI readings.

To address these challenges, the Profile Construction Module utilizes the confidence interval to allow sensor nodes themselves to compute for each inbound link an appropriate training set size, depending on the different RSSI characteristics.

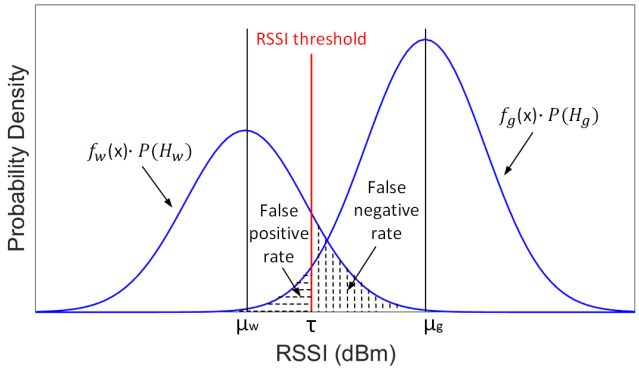

Fig. 3: The false positive rate and false negative rate with respect to a given RSSI threshold τ in a Gaussian channel. μ_w and μ_g are the means of the RSSI distributions of a bad link and a good link, respectively.

According to the *Central Limit Theorem*, for an attribute x with any type of underlying distribution, the margin error of the confidence interval for the mean $\bar{x}$ is $e_\mu = z \cdot \sigma_p/\sqrt{n}$, where z is the z-score ($z = 2.58$ for a confidence level of 99%), n is the number of samples and σ_p is the population standard deviation. We transform the equation of e_μ, apply it to our problem and obtain the training set size as follows:

$$N_{ts} = \left(\frac{z \cdot \sigma_p}{E_\mu}\right)^2 \quad (8)$$

where E_μ is a user-defined error of the estimated RSSI mean μ_g. The population standard deviation σ_p can be approximated by the standard deviation of a small set whose size has to be larger than 30 [25]. We set this number to 50 in our case.

When the training phase starts, the Profile Construction Module computes the RSSI standard deviation σ_p for each inbound link after collecting the first 50 RSSI measurements. Then, the module uses Equation 8 to estimate the specific training set size N_{ts} for each link with a user-defined setting of E_μ. In Section IV-C, we will show the impact of the parameter E_μ on the training delay and detection accuracy.

D. Anomaly Detection: Data Smoothing

The Detection Module runs during the detection phase after the profile construction of the monitored link is complete. As RSSI is random in nature, transient RSSI fluctuations may increase the error rate of RADIUS. To overcome this limitation and make RADIUS more robust, the Detection Module applies a sliding window of size l to compute a short-term average of RSSI and compares the l-averaged RSSI with the Bayes threshold to trigger a detection of RSSI anomaly.

The choice of l has a trade-off on the system performance. A smaller l makes the detection more responsive, but it may not be sufficient to clean the noise. On the other hand, a larger l may be a better choice for data cleaning, but overly smoothing may fail to capture the real anomalous degradation in RSSI. In Section IV-C, we will evaluate the impact of the sliding window size l on the system performance and discuss the *detection delay*, i.e. the time between the degradation in link PRR and the detection of the RSSI anomaly.

E. Profile Update: Adapting to Environmental Changes

Due to the dynamic changes in the environment, e.g., transition between day and night or change of the surrounding obstacles, the stored normal profiles of inbound links may not capture the real normal state. Therefore, the Profile Update Module needs to update the normal profile (μ_g, σ) during the detection phase. The technique we use to update the profiles is to continuously update the training set with the "normal" RSSI readings measured during the detection phase.

To identify whether an RSSI value is normal or not, RADIUS assigns an anomaly score s for it, indicating the significance of the anomalous behavior. s is calculated by $s = \mu_l / T_{Bayes}$, where μ_l is the l-averaged RSSI value and T_{Bayes} is the Bayes threshold. During the detection phase, RADIUS collects consecutive RSSI readings in disjoint groups of size l_{update} and also their anomaly scores to compute the average anomaly score. The group of RSSI readings with an average anomaly score of less than one is added to the training set. Through this, the training set is updated, and accordingly, the normal profile and the Bayes threshold are updated. In Section IV-C, we will evaluate the profile update technique and study the effect of l_{update} on the system performance.

To avoid increasing memory usage during the process of profile construction and update, we implemented this in a memory-friendly way, i.e., to compute μ and σ with a single pass without storing the previous measurements of RSSI. To do so, we reformulate μ and σ in the following way:

$$\mu = \frac{s}{n}, \quad \sigma = \sqrt{\frac{1}{n-1}\left(q - \frac{s^2}{n}\right)} \tag{9}$$

where s and q are defined as follows:

$$s = \sum_{i}^{n} x_i, \quad q = \sum_{i}^{n} x_i^2 \tag{10}$$

in which x_i is the i-th RSSI reading. In this way, RADIUS stores only 2 counters (s and q) for each inbound link to compute and update μ and σ, rather than storing the entire training set. By doing so, the RAM consumption of RADIUS has a complexity of $O(m)$, where m is the number of inbound links, remaining independent from the number of RSSI values.

IV. Experimental Evaluation

In this section, we evaluate the performance of the Bayes threshold on an indoor testbed and compare it with two state-of-the-art thresholding techniques, followed by the evaluation of the overall RADIUS system. Furthermore, we demonstrate an exemplary application to show how existing approaches, e.g., a transmission power tuning scheme from the literature [4], can benefit from RADIUS.

A. Testbed Setup

We implement RADIUS based on TinyOS 2.1.2 and evaluate its performance on an indoor testbed that consists of 32 TelosB nodes mounted on the walls and ceilings, spreading over an entire office floor (Figure 4). The experiment environment is a typical indoor environment where packet losses over radio links may be caused by factors such as people walking around, shadowing effects of furnitures and doors, resulting in anomalous degradation in signal strength of radio links.

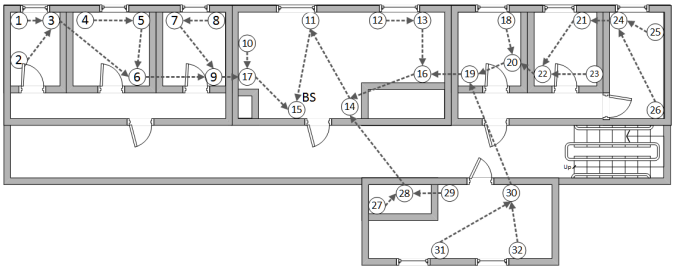

Fig. 4: Indoor testbed. Node 15 is the base station.

For evaluation purposes only, we minimize in all experiments the packet losses caused by factors other than channel quality degradation, so that we know the ground truth on the cause. All experiments use TDMA MAC for transmissions to reduce collisions, even though RADIUS is independent from the employed MAC (as to be shown in Section IV-D). In addition, all sensor nodes operate at the default channel of CC2420 (Channel 26), which is known to have minimal interferences from WiFi signals.

B. Evaluation and Comparison of the Thresholding Methods

We now evaluate the performance of the Bayes threshold and compare it with two state-of-the-art thresholding techniques: Percentile-based and Chebyshev inequality-based thresholds. Besides their popularity in the literature [11] [12] [19] [20], we choose these two thresholding methods for comparison because they compute thresholds based on the same RSSI statistics (average and standard deviation) as the Bayes threshold, implying that all three thresholding techniques incur similar computation and memory overhead.

Specifically, the Percentile-based threshold is defined as the *x-th percentile* of the underlying RSSI distribution of a good link while the Chebyshev threshold is defined as follows:

$$T_{cheby} = \mu_g + \sigma * \sqrt{\frac{1 - P_{target}}{P_{target}}}, \tag{11}$$

where μ_g and σ are the mean and standard deviation of the RSSI values collected in the training phase when the link is a good link and P_{target} is a user-defined target probability.

Comparison perspectives. We evaluate the detection accuracy of all three thresholding methods from two perspectives.

First, we study the impact of the thresholding parameter on the detection accuracy, namely the *x-th percentile* for the Percentile threshold, P_{target} for Chebyshev threshold and $P(H_g)$ for the Bayes threshold. How the accuracy is influenced by the thresholding parameter determines whether the thresholding method requires a fine-tuning process of its thresholding parameter for each radio link. An ideal thresholding method can choose a parameter setting flexibly from a wide range and yet achieve good accuracy for all links across the network.

Then, we investigate the performance of the thresholding methods when the RSSI distribution of a good link overlaps with that of a bad link. The more the two distributions overlap, the more difficult is to identify whether RSSI values are from

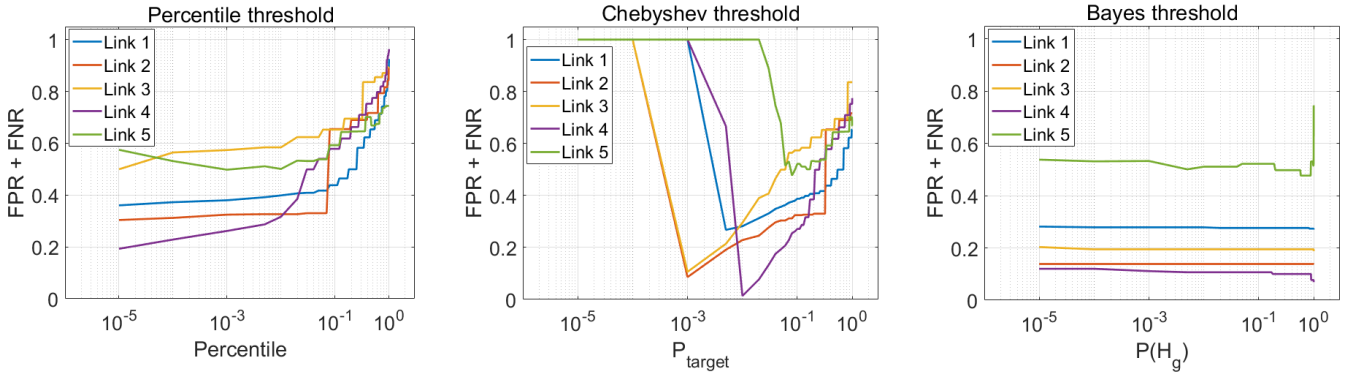

Fig. 5: Comparison of three thresholding methods: Impact of thresholding parameters. The choice of the thresholding parameter has a significant impact on the performance of Percentile or Chebyshev threshold while it has minimal impact on the Bayes threshold.

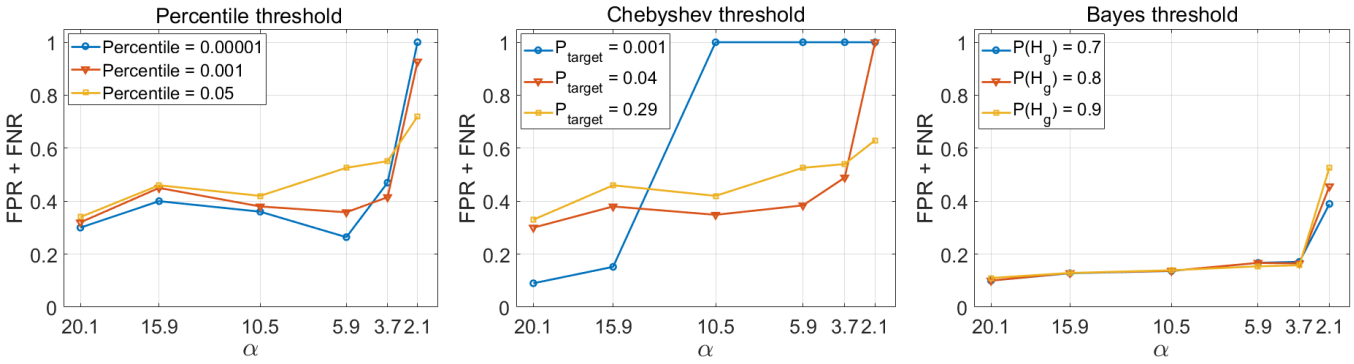

Fig. 6: Comparison of three thresholding methods: Performance under different detection difficulty levels, with typical thresholding parameter values. A smaller α indicates more overlapping of the two RSSI distributions, hence a higher difficulty level.

the distribution of a good link or a bad one. We quantify the degree of overlapping with a so-called *detection difficulty level* α defined in Equation 7. A smaller α indicates that the RSSI mean of a good link (μ_g) gets closer to the RSSI mean of a bad link (μ_w) and/or the standard deviation σ increases, implying a higher detection difficulty level to identify RSSI anomalies.

Impact of the thresholding parameters. For this purpose, we collected RSSI and PRR traces from exemplary radio links. We apply all three thresholding methods with varying thresholding parameters individually to the same data traces, i.e., the normal profiles and the RSSI anomalies experienced by each link are the same for all three methods. The PRR threshold of a good link, used as a reference to decide whether a detection decision is a false positive or a false negative, is set to 80%. The resultant error rates (sum of FPR and FNR) for 5 exemplary links using the three thresholding methods are presented in Figure 5.

We can clearly see from Figure 5 that the accuracy for both Percentile and Chebyshev thresholds varies dramatically with their thresholding parameters. The error rate for the Percentile threshold is in general higher than the other two methods. This is most likely because the Percentile threshold tends to provide tight RSSI bounds of good links, easily resulting in a high FPR in case of RSSI fluctuations. The Chebyshev threshold can achieve a very low error rate only if its thresholding parameter P_{target} is carefully selected. However, such optimal P_{target} varies significantly from link to link, implying that fine-tuning of P_{target} for each link is necessary for the Chebyshev threshold to achieve a high accuracy across the network.

In contrast, the accuracy achieved by the Bayes threshold is insensitive to the value of $P(H_g)$ (unless $P(H_g)$ is approaching the extreme, e.g., 1) and is consistently robust for every link. Furthermore, even with a coarse setting of $P(H_g)$, Bayes threshold provides close to the minimal error rate among all three methods for the analyzed links. Similar results are observed for all testbed links. Thanks to such features, RADIUS can employ a parameter setting of $P(H_g)$ from a wide range (e.g., between 0.7 and 0.9) for all links and achieve high accuracy for all links across the network without the need of tuning its thresholding parameter for every individual link.

Performance under different detection difficulty levels. We now examine how each threshold performs under various detection difficulty levels indicated by α (Equation 7), especially for small α values when the RSSI distributions of the good link and the bad link highly overlap with each other.

We select from the testbed representative links with different α values ranging from a low difficulty level ($\alpha = 20.1$) to an extremely difficult one ($\alpha = 2.04$). We apply each thresholding method with three typical thresholding parameter settings, based on the analysis of the Figure 5, to the RSSI and PRR traces collected from these links. The resultant error rates under different α values are plotted in Figure 6.

Figure 6 shows that when α deceases, the error rates increase in general for all three methods, as expected. Among the three thresholding methods, the Bayes threshold provides

TABLE I: Testbed experiment: Error rates of all three thresholding methods with typical thresholding parameter settings. (P1=0.00001, P2=0.001, P3=0.05, C1=0.001, C2=0.04, C3=0.29, B1=0.7, B2=0.8, B3=0.9)

Techniques	Average	Deviation	Minimum	Maximum
Percentile–P1	38.79%	23.69%	11.50%	100%
Percentile–P2	36.39%	17.68%	9.19%	92.70%
Percentile–P3	38.99%	14.27%	16.34%	83.81%
Percentile–Avg	38.06%			
Chebyshev–C1	77.95%	38.10%	4.90%	100%
Chebyshev–C2	44.11%	29.45%	10.12%	100%
Chebyshev–C3	39.46%	18.83%	16.64%	86.26%
Chebyshev–Avg	53.84%			
Bayes–B1	15.20%	6.63%	9.69%	23.68%
Bayes–B2	12.78%	4.77%	9.23%	16.67%
Bayes–B3	11.83%	4.69%	6.26%	16.13%
Bayes–Avg	13.27%			

TABLE II: Overhead of all three thresholding techniques.

Techniques	RAM (bytes)	ROM (bytes)	Computation (ms)
Percentile	66	2850	8.3
Chebyshev	66	3052	8.9
Bayes	68	4688	10

the best performance. The error rate for the Bayes threshold increases only slightly with smaller α except a notable increase for the most difficult case ($\alpha = 2.04$). Furthermore, the Bayes threshold provides the lowest error rate among the three thresholding methods for every difficulty level.

Testbed experiment. Finally, we evaluate how all three thresholding methods perform in real implementations running on sensor nodes. Specifically, we install on each sensor node the modules for thresholding, detection and a partial profile construction (using fixed training set size).

We ran a data collection application on each sensor node to periodically send data at a 2 second interval to the base station following the topology depicted in Figure 4. Note that RADIUS can also be applied to dynamic routing with several modifications. We will show in Section IV-D how RADIUS can be extended for a state-of-the-art routing protocol.

We let each experiment run for about 3 hours, in which only one of the three thresholding methods is employed. In all experiments, the PRR traces of radio links and the alarms that report the detected RSSI anomalies are recorded for analysis purposes. The resultant error rates for a total of 9 experiments are listed in Table I. The results in Table I confirm our previous analysis. The Bayes threshold provides the lowest averaged error rate (13.27%) among all three thresholds. Furthermore, the small standard deviations of the error rates (between 4.69% and 6.63%) indicate that the performance of the Bayes threshold is robust for all links across the network with diverse channel characteristics.

The overheads of the three thresholding methods are presented in Table II. The RAM consumption required by all three thresholds does not increase with the number of RSSI values needed for profile construction, due to the memory-friendly implementation (Section III-E). The extra ROM overhead for the Bayes threshold is due to the additional mathematic library required by the logarithmic operation (Equation 5).

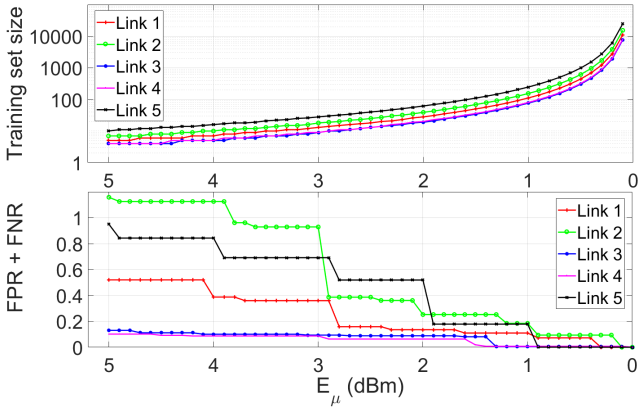

Fig. 7: Impact of system parameter E_μ of the Profile Construction Module. $E_\mu = 1$ dBm provides a good tradeoff between the training set size (i.e. training delay) and the error rate.

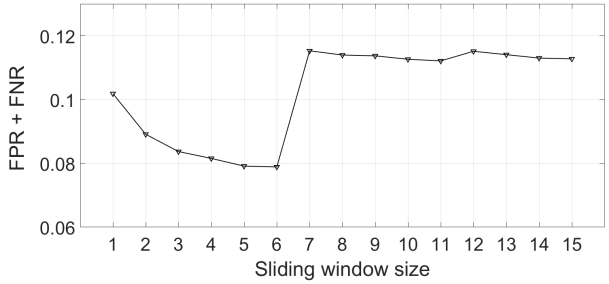

Fig. 8: Impact of system parameter l (sliding window size) of the Detection Module. $l = 3$ packets provides a fast response time to RSSI anomalies and a low error rate.

C. Evaluation of the Overall System

From the evaluation of the Bayes threshold, we have observed that with a coarse setting of $P(H_g)$ from a wide range, the Thresholding Module can achieve a low error rate for all links across the network. We set $P(H_g) = 0.9$ for our system. In this section, we investigate the impact of system parameters of the remaining RADIUS modules. Then, we evaluate a complete implementation of RADIUS on the testbed in a 72-hour experiment. Finally, we analyze the system overhead.

System parameters. We now study the impact of the system parameters required by other RADIUS modules, namely, the RSSI error E_μ for estimating the training set size in the Profile Construction Module, the sliding window size l for data smoothing in the Detection Module, and the update window size l_{update} for profile updating in the Profile Update Module.

In Figure 7, we plot the training set sizes estimated using different E_μ values and the corresponding error rates for 5 exemplary links. The impact of parameter E_μ shows a tradeoff: smaller E_μ indicates higher estimation accuracy of the normal profiles and thus higher system accuracy while it may also increase the training set size significantly. To achieve a good tradeoff, we choose $E_\mu = 1$ dBm. With such setting, the resultant size of the RSSI training set is typically between 100 and 200, i.e., a training delay of 3 to 6 minutes for an inter-packet interval of 2 seconds. In practice, RADIUS can utilize the packets from different traffic, e.g., packets forwarded by direct child nodes, to reduce the training delay significantly.

Furthermore, we choose sliding window size $l = 3$ to smooth RSSI values during the detection phase, which gives RADIUS a short response time to RSSI anomalies and a low error rate, as depicted in Figure 8. Finally, we notice that, by setting the update window size l_{update} to 50, the profile updating technique can significantly reduce the error rate, i.e., the error rate can be reduced of 3% to 8% in all experiments.

System performance. We configure the RADIUS system with the aforementioned parameter settings and install it on every sensor node in the testbed. In the experiments, each node sends a data packet to the base station every 10 seconds as depicted in Figure 4. We ran the experiment for 72 hours.

Figure 9 demonstrates how the RADIUS system behaves during the 72 hours for one exemplary link (from node 6 to node 9). From the PRR trace depicted in Figure 9(b), we can observe that the link experienced high packet losses in three time periods during the experiment. This is most likely caused by people crossing the link, resulting in anomalous RSSI degradations (see Figure 9(a)). The logged alarms that reported such RSSI anomalies are marked in red in Figure 9(d). Figure 9(c) shows that the RSSI threshold is updated during the whole experiment adapting to the environmental changes (e.g., transition between day and night, with and without human activities). Consequently, RADIUS keeps a low error rate for this link over the whole experiment period, as shown in Figure 9(e), with an overall error rate of 8.9%.

Then, we plot the error rate for every link in the testbed in Figure 10. Among all the links, the link from node 30 to node 19 shows the highest error rate (13.6%). This is most likely due to a small α (Equation 7) caused by three walls and a wide corridor between the sender and the receiver node, making it difficult to identify RSSI anomalies. Overall, RADIUS achieves a low error rate for every link in the network (5.78% on average). Based on these results, we demonstrate that RADIUS can achieve high accuracy in detecting anomalous RSSI degradation of all links across the network while such high accuracy is robust to changes in the environment.

Impact of data rate. We repeat the experiment with different date rates, i.e. with inter-packet interval of 5 seconds and 30 seconds, respectively. From the experimental results, we observe that there is no clear impact of data rates on the detection accuracy. The averaged error rates for 5 seconds and 30 seconds intervals are 6.32% and 5.96%, respectively.

The training delay, on the other hand, clearly increases with the inter-packet interval. In the experiments, we allow RADIUS to incorporate forwarding packets for both training and detection, significantly reducing the training delay and also the detection delay for those nodes close to the base station.

We also observe that the detection delay is affected by not only the data rate but also the packet losses of monitored links. During the detection phase, RADIUS makes a decision for every newly arrived packet based on a short-term (over 3 packets) RSSI average. From the experimental results, we observe that the detection delay increases with the link loss rate. For a link of 70% loss rate, we observed a detection delay equal to the time required for the transmission of 8 packets.

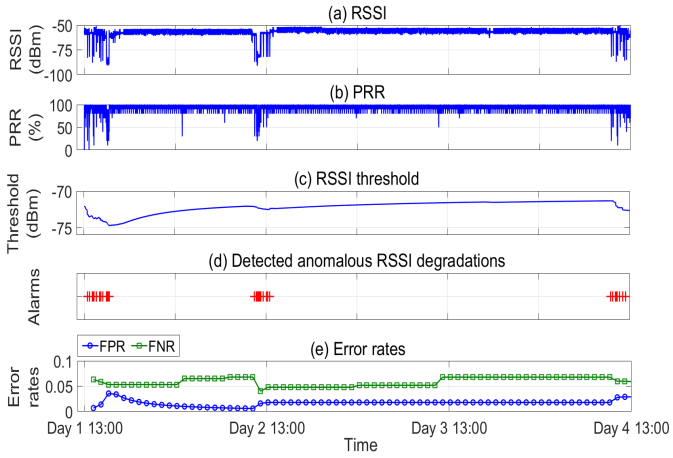

Fig. 9: 72-hour experiment: Results for the link from node 6 to node 9 (Figure 4). Experiment started at 13:00 on Day 1. Day 3 is a public holiday (no human activities).

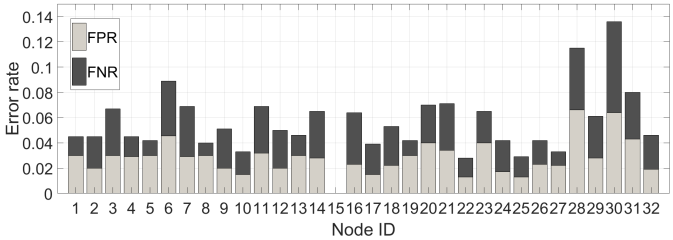

Fig. 10: 72-hour experiment: Error rates for every link in the testbed. Node ID refers to the sender node ID of each link and node 15 is the base station (Figure 4).

Overhead. Finally, we analyze the overhead of the RADIUS implementation. Due to the memory-friendly implementation of profile construction and update (Section III-E), the memory consumption of RADIUS only increases with the number of monitored links. For two inbound links, the whole RADIUS system consumes 176 bytes RAM and approximately 6 KB ROM (in comparison to 10 KB RAM and 48 KB ROM in a TelosB device). As all operations of RADIUS are performed locally on sensor nodes, RADIUS incurs no communication overhead unless it requires additional probing messages to construct or update profiles (more discussion on this in Section IV-D). The main computation overhead of RADIUS comes from the processing of the Bayes threshold, which we have already presented in Table II.

D. Application of RADIUS: A Case Study

Applicability. RADIUS is a system to detect the degradation in channel quality experienced by inbound links. Monitoring a link in both directions requires RADIUS running on nodes at both sides of the link. Besides that, RADIUS is neither limited to a specific communication topology (e.g., tree-based or mesh) nor to a MAC protocol (e.g., TDMA or CSMA).

Nevertheless, RADIUS requires several modifications to accommodate dynamic routing protocols. In the training phase before the application starts, we let each sensor node broadcast probing messages one after another so that each node can construct profiles for all neighbors in the communication

TABLE III: Literature-based vs. RADIUS-assisted Tx-power tuning

Schemes	Avg. PRR	Avg. Energy (μJ/bit)
CTP + medium Tx-power	78.2%	0.55
CTP + literature-based tuning	81.4%	0.61
CTP + RADIUS-assisted tuning	89.1%	0.63

range. During the detection phase, we can utilize the beacon messages of the routing protocols to update thresholds for those links that are used in the current routing paths.

We notice that the performance of RADIUS may degrade due to excessive link packet losses. However, we consider such case as another type of failure (disconnected link), which can be detected by checking e.g., inter-packet arrival time [19].

An exemplary application. To demonstrate the potential of this work, we present an example of how existing approaches can benefit from using RADIUS as an accurate and robust trigger to perform remedial actions.

We take the transmission power (Tx-power) tuning scheme from [4] as an example. In [4], Tx-power adjustment is triggered when the runtime RSSI is below a predefined threshold (-89 dBm). In our exemplary application, we employ RADIUS as the trigger instead. We implement both the literature and RADIUS-assisted Tx-power tuning schemes on sensor nodes with standard TinyOS CSMA at the MAC layer and the Collection Tree Protocol (CTP) [26] at the routing layer.

We ran on the testbed the following experimental configuration individually for 24 hours: (1) CTP and fixed medium Tx-power (-5 dBm) for all nodes, (2) CTP and the literature-based Tx-power tuning scheme, and (3) CTP and RADIUS-assisted Tx-power tuning scheme. The training procedure in (3) takes about 5 minutes in total when probing messages for the profile construction are sent at a 10 ms interval. The resultant averaged end-to-end PRR and energy consumption per transmitted bit for each experiment are listed in Table III.

From the table, we can see that the literature-based tuning scheme improves the averaged end-to-end PRR by only 3.2% while the RADIUS-assisted tuning scheme improves the PRR by 10.9% with slightly more energy consumption. The reason for this is that RADIUS provides a higher accuracy of detecting RSSI anomalies than a predefined RSSI threshold (-89 dBm in this case). As presented in Figure 1, a predefined RSSI threshold may not work well for all network links nor in the cases when the RSSI characteristics of links change over time. As a consequence, the Tx-power tuning scheme from the literature may suffer from a high FNR (i.e. miss to detect RSSI anomalies and thus fail to improve link PRR) and/or a high FPR (i.e. increase Tx-power for good links and thus achieve only limited improvement of link PRR).

V. Conclusion

This paper presents RADIUS, an approach to detect anomalous degradation in signal strength of IEEE 802.15.4 links, which enables sensor nodes to decide locally whether the experienced channel quality degradation is the cause for link packet losses. To achieve this, RADIUS (1) lays its foundation on a Bayes thresholding scheme, integrated with dedicated techniques for (2) fast yet accurate profile construction, (3) accurate detection with data smoothing, and (4) profile update to adapt to environmental changes. Extensive testbed evaluation shows that RADIUS has fulfilled its design requirements: lightweight, accurate and robust to a diversity of channel characteristics and dynamic environmental changes. To demonstrate the potential of this work, we showed that an existing transmission power tuning scheme can benefit from the capability of RADIUS as an accurate and robust trigger for taking remedial actions. Currently, we are investigating further how existing works, e.g., a cross-layer tuning scheme in [5], can benefit from RADIUS. Furthermore, we plan to study how RADIUS can be combined with existing techniques to detect external interference [1] and packet collisions [2], allowing a comprehensive diagnosis of link-level packet losses.

References

[1] F. Hermans *et al.*, "SoNIC: Classifying interference in 802.15.4 sensor networks," in *Proc. of IPSN*, 2013.

[2] D. Son *et al.*, "Experimental study of concurrent transmission in wireless sensor networks," in *Proc. of SenSys*, 2006.

[3] N. Baccour *et al.*, "Radio link quality estimation in wireless sensor networks: A survey," *ACM Transactions on Sensor Networks*, 2012.

[4] S. Lin *et al.*, "ATPC: Adaptive transmission power control for wireless sensor networks," in *Proc. of SenSys*, 2006.

[5] S. Fu *et al.*, "Experimental study for multi-layer parameter configuration of WSN links," in *Proc. of ICDCS*, 2015.

[6] C. A. Boano *et al.*, "Hot packets: A systematic evaluation of the effect of temperature on low power wireless transceivers," in *Proc. of ExtremeCom*, 2013.

[7] S. Lin *et al.*, "Toward stable network performance in wireless sensor networks: A multilevel perspective," *ACM Transactions on Sensor Networks*, 2015.

[8] R. Marfievici *et al.*, "How environmental factors impact outdoor wireless sensor networks: A case study," in *Proc. of MASS*, 2013.

[9] K. Srinivasan *et al.*, "An empirical study of low-power wireless," *ACM Transactions on Sensor Networks*, 2010.

[10] C. U. Bas *et al.*, "Spatio-temporal characteristics of link quality in wireless sensor networks," in *Proc. of WCNC*, 2012.

[11] A. A. Mahimkar *et al.*, "Towards automated performance diagnosis in a large iptv network," in *Proc. of ACM SIGCOMM*, 2009.

[12] A. Kosba *et al.*, "RASID: A robust wlan device-free passive motion detection system," in *Proc. of PerCom*, 2012.

[13] N. Ramanathan *et al.*, "Sympathy: a debugging system for sensor networks," in *Proc. of SenSys*, 2004.

[14] X. Miao *et al.*, "Agnostic diagnosis: Discovering silent failures in wireless sensor networks," *IEEE Trans. on Wireless Comm.*, 2013.

[15] K. Liu *et al.*, "Self-diagnosis for detecting system failures in large-scale wireless sensor networks," *IEEE Trans. on Wireless Comm.*, 2014.

[16] S. Rajasegarar *et al.*, "Distributed anomaly detection in wireless sensor networks," in *Proc. of ICCS*, 2006.

[17] H. Wang *et al.*, "Intrusion detection for wireless sensor networks based on multi-agent and refined clustering," in *Proc. of CMC*, 2009.

[18] S. Rajasegarar *et al.*, "Quarter sphere based distributed anomaly detection in wireless sensor networks," in *Proc. of ICC*, 2007.

[19] S. Rost *et al.*, "Memento: A health monitoring system for wireless sensor networks," in *Proc. of SECON*, 2006.

[20] M. Zhu *et al.*, "Fusion of threshold rules for target detection in wireless sensor networks," *ACM Transactions on Sensor Networks*, 2010.

[21] J. L. Melsa *et al.*, *Decision and estimation theory*. McGraw-Hill Inc., 1978.

[22] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Prentice Hall PTR, 2001.

[23] "Q-function," Url:https://en.wikipedia.org/wiki/Q-function.

[24] K. Srinivasan *et al.*, "Understanding the causes of packet delivery success and failure in dense wireless sensor networks," in *Proc. of SenSys*, 2006.

[25] J. Pitman, *Probability*. Springer, 1993.

[26] O. Gnawali *et al.*, "Collection tree protocol," in *Proc. of SenSys*, 2009.