IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received March 31, 2019, accepted April 16, 2019, date of publication April 29, 2019, date of current version May 14, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2913679

QoS Aware Routing Protocol Through Cross-layer Approach in Asynchronous Duty-Cycled WSNs

PEIZHONG SHI[1,2], CHUNSHENG GU[1], CHUNPENG GE[1], AND ZHENGJUN JING[1]
[1]School of Computer Engineering, Jiangsu University of Technology, Changzhou 213001, China
[2]Department of Information System, City University of Hong Kong, Hong Kong

Corresponding author: Peizhong Shi (spz0812@jsut.edu.cn)

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61602216, Grant 61672270, and Grant 61702236.

ABSTRACT The growing usage of wireless sensor networks (WSNs) in different scenarios makes the Quality-of-Service (QoS) a paramount issue in WSN-based applications. We are especially interested in the QoS aware of asynchronous duty-cycled WSNs in the light of designing routing protocol, whereby waiting for latency and malicious packet dropping are critical to the network performance. We first propose an optimized detection mechanism for malicious packet dropping attack. Then, based on sleep latency and queue length, a method for congestion degree measurement is proposed under asynchronous duty-cycled low power listening (LPL) modes. Last but not least, we introduce related QoS aware metrics for the design of QoS aware routing protocol, which can enhance the QoS performance in throughput, delay, and packet losses (TDL). Experimental results demonstrate that the QoS performance of our TDL-based routing protocol is better than those of the CTP-Watchdog protocol and the CTP-Optimized protocol, especially in terms of higher-throughput, lower-delay, and less-loss rate. By optimized threshold, the detection accuracy of our detection mechanism is improved much more than that of the traditional watchdog technology. Furthermore, we evaluate that the preferred optimization factor can help to make a tradeoff between the false negative rate and the false positive rate. Our TDL-based routing protocol is implemented under the component-based architecture by a cross-layer approach, which provides a practicable solution for the design of QoS-aware routing protocol in asynchronous duty-cycled WSNs.

INDEX TERMS Wireless sensor networks, QoS aware, routing protocol, asynchronous, duty-cycled, malicious packet dropping, congestion degree, cross-layer.

I. INTRODUCTION

The last decades have witnessed advances in multiple wireless sensor networks in both the academic and industrial world. A wireless sensor network (WSN) is composed of a set of distributed hardware-constrained wireless devices in charge of monitoring targeted areas. The applications of WSNs are numerous and range from environmental monitoring to urban health monitoring through healthcare, logistic applications, and smart grids, to name a few. And in each application, information may request different Quality-of-Service (QoS) processing (regular monitoring messages vs. alarm messages) [1]. However, due to the unreliable characteristics of the wireless medium and the hardware limitations of devices, providing QoS for applications remain a challenging task, especially in asynchronous duty-cycled WSNs. Each deployed sensor node independently schedules itself to be active briefly and then stays dormant for a long time [2], [3], which usually operates in a duty-cycled manner (e.g., 5% or less) to bridge the gap between limited energy supplies and network lifetime. The definition of QoS can slightly vary depending on the application area [4]. For example, QoS can indicate the capability to provide assurance that the performance requirements of a specific application can be met. QoS is measured to estimate the presentation of network performance [5], [6], like-bandwidth, delay, jitter, and packet losses, are establish to mainly control the QoS issues [7]. Compared with traditional WSNs, specific features in asynchronous duty-cycled WSNs (resource constraints in terms of power, processor, and memory; unreliable links increase the rate of packet drops, delays, and energy consumption; waiting latency of independent sleep scheduling) make the designers

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

and developers resort to optimization approaches and techniques to provision QoS support. To this end, the issue of QoS provisioning in WSNs has gained a growing attention by the research community due to the wide spectrum of applications. There are numerous studies that thoroughly address the issue of providing QoS in general, and the performance parameters at every layer of the protocol stack should be adjusted for QoS performance. Research on interaction of different layers called cross-layer mechanisms are receiving an outstanding attention to provide better QoS support [4], [8].

This paper addresses the QoS aware routing problem in asynchronous duty-cycled WSNs. Based on link quality parameters, we exploit the security analysis in packet dropping attack and propose the method of malicious node detection, which can distinguish the malicious packet dropping introduced by the attacks from natural packet loss due to the collisions and channel errors. Further more, the detection threshold is deduced for minimizing optimization, practically with a preference factor. Since waiting latency aggravates the occurrence of network congestion except for limited wireless channel, limited buffer queue, burst traffic and so on. Congestion degree is defined to measure the degree for both node-level congestion and link-level congestion, which helps to preferably measure the local congestion degree at each intermediate node. In order to make routing decisions dependent on QoS awareness, the metric Expected Transmission Time (ETT) is introduced and recognized as the primary routing criteria, whereby the algorithm of reasonable parent node selection and switching is given and described by a case study. The cross-layer approach of our solution for QoS aware routing protocol is designed and implemented under the component-based architecture of our prior work. The particular contributions of this paper can be summarized as follows.

(1) We study the QoS aware routing protocol in asynchronous duty-cycled WSNs, and model the problem base on the related parameters including throughput, delay and packet loss. Then, we refine the QoS performance and optimal conditions for the definition of problem description.

(2) Under asynchronous duty-cycled Low Power Listening (LPL) modes, because waiting latency and malicious packet dropping are critical to the to the network performance, we propose an optimized detection mechanism for malicious packet dropping attack, and propose a method for congestion degree measurement based on sleep latency and queue length.

(3) We introduce related QoS aware metrics for the design of QoS aware routing protocol, where the routing paths have the features of less-cost, lower-delay and less-loss rate. As far as we know, this is the first work that implements a QoS aware routing protocol for duty-cycled WSNs under the component-based architecture.

(4) Experimental results demonstrate that the QoS performance of our TDL-based routing protocol is better than those of the CTP-Watchdog protocol and the CTP-Optimized protocol, especially in terms of higher-throughput, lower-delay and less-loss rate. By optimized threshold, the detection accuracy of our detection mechanism is improved much more than that of the traditional watchdog technology. Furthermore, we evaluate that the preferred optimization factor can help to make a tradeoff between the false negative rate and the false positive rate. Therefore, our TDL-based routing protocol can be recognized as a practicable solution for the design of QoS-aware routing protocol in asynchronous duty-cycled WSNs.

The remainder of this paper is organized as follows. Researches based on QoS routing in WSNs are related with our proposed method in Section II. Section III presents the network model and our problem description. The design of QoS aware routing protocol based on malicious packet dropping detection, congestion degree measurement, QoS aware routing metrics and algorithm, and the cross-layer QoS architecture are presented in Section IV. By conducting several experiments, results of our proposed approach are discussed in Section V. Finally, we concluded the paper in Section VI.

II. RELATED WORK

In the literature, the routing in wireless sensor networks is classified according to the structure of the network in linear routing, hierarchical routing and location-based routing [9]. Also, the protocols can be classified according to their operation in routing based on consistency, multi-paths routing techniques, by negotiation, based on requests or based on the QoS [10]. Overall, QoS support in WSNs is a fairly important research problem with many remaining issues to investigate. Here, we continue to study the QoS optimization problem and forcefully on approaches related literature that can be grouped into three categories as follow.

The first category, identifying key QoS requirements and metrics, is a feasible QoS management scheme by establishing a QoS model based on a specific application scenario, which can potentially involve trade-offs can be derived. In [11], using collective intelligence of artificial ants as intelligent agents, the authors proposed an approach for QoS routing algorithm of WSNs based on Ant Colony Optimization (ACO). It is the tradeoff between a certain guaranteed QoS requirements and acceptable computational complexity. In [12], Zhang *et al.* developed a guaranteed QoS model, such as the upper bounds on buffer queue length/delay/effective bandwidth, and single-hop/multi-hops delay/jitter/effective bandwidth. Efficient Media Access Control (MAC) scheme was proposed to provide autonomous QoS to the sensor nodes in one-hop QoS retransmission group [13]. In [14], Ababneh *et al.* studied the problem of routing, bandwidth and flow assignment in wireless body area networks (BANs). An adaptive joint routing and bandwidth allocation protocol was proposed for traffic streaming, which maximized the network utility while satisfying the QoS requirements. In order to achieve both security and QoS requirements, the authors in [15] discussed the packet based attacks and studied the Optimal Inspection Points (OIP) problem, which was required to find a subset of nodes in a given network to perform the deep packet inspection so as to maximize the number of scanned packets while satisfying the delay constraints. On the basis of predictability of TDMA schedule, the authors in [16] proposed a self-stabilizing hop-constrained energy-efficient (SHE) protocol for constructing minimum energy networks for hard real-time routing, which helped to meet the QoS requirements while prolonging the network lifetime. In [17], El Hammouti *et al.* presented a game theory based approach to maximize quality of service of the aggregate frame success rate, while optimizing power allocation. In [9], Deepa and Suguna proposed a routing protocol RPAR (Greedy Realtime Power Aware Routing) that was adaptive to the QoS optimization in terms of end-to-end delay, packets delivery ratio and energy conservation. In [18], an Optimized QoS-based Clustering with Multipath Routing Protocol (OQoS-CMRP) was proposed for WSNs to reduce the energy consumption in sink coverage area by applying the Modified Particle Swarm Optimization (PSO)-based clustering algorithm. El Hammouti *et al.* [19] proposed a novel link quality based opportunistic routing method, where its algorithm selected the relay nodes based on OR theory to enhance the lifetime of whole network and feasibly create the ideal transmission distance for energy saving. In [20], multi-hop gateway node was implemented with both protocol to achieve maximum lifetime and energy efficiency, whereby the optimal algorithm respectively computed least possible adequate dominating sets and constantly preserved QoS requirements across iterations. Investigating the effects of multi-hop communication on a traffic system model designed with a Markov discrete-time M/M/1 queuing model, in [21], Hasan *et al.* presented a mathematical model for a new-generation of forwarding QoS routing determination that enabled allocation of optimal path to satisfy QoS parameters. In wireless sensor and actuator networks, Yahiaoui *et al.* [22] designed a delay and energy sensitive routing protocol based on-demand routing approach that provided QoS in terms of delay and energy consumption.

The second category, based on cross-layer design approach, is widely used to improve the network performance that generally includes two aspects of design methods: theoretical mathematical modeling and practical protocol design [23]. In [4], Al-Anbagi *et al.* presented a survey on the state of the art of cross-layer QoS approaches to achieve delay and reliability bounds in critical applications. Due to QoS constraints, in [24], Ruiz *et al.* proposed a cross-layer architecture to achieve the QoS guarantee, and saved great energy by eliminating collisions and considerably reducing idle listening. Shi and Fapojuwo [25], utilizing the concept of cross-layer optimization, proposed an iterative reuse factor (IRF) approach to efficiently solve the problem. However, this approach is centralized in nature because of the knowledge of the whole network topology. In [26], a traffic and QoS-aware cross-layer MAC protocol for wireless multimedia sensor networks namely urgMAC was proposed to provide continuous QoS support with video quality trade-off at the application layer dynamically for applications. To improves network reliability and reduces excessive packets retransmissions, a dynamic clustering based energy efficient and QoS-aware routing protocol (called EQRP) has been proposed [23], which is inspired by the real behavior of the bird mating optimization (BMO).

The third category consists of integrated QoS solutions that introduces the sleep scheduling for energy efficiency, especially in asynchronous duty-cycled WSNs. In [27], Kim *et al.* presented a QoS aware energy efficient priority based receiver-initiated asynchronous MAC protocol for energy harvesting WSNs. In [28], Kim *et al.* proposed a new reliable protocol termed Cross-layer Channel Access and Routing (CCAR), which simultaneously supported both MAC and routing operations for medical-grade QoS provisions. In addition, CCAR introduced an effective route maintenance scheme to avoid link failures in bottlenecked intermediate nodes, which prevented unnecessary packet drops and route rediscovery evocations.

From the above, QoS is one of the most important topics in WSNs, which is related to the network performance adaptable to the requirement of applications. The major focus of the aforementioned works is optimizing QoS for the throughput, energy efficiency, lifetime and delay. Others are primarily why cross-layer design for supporting QoS performance has been a focus of much recent work, and the joint optimization of control over two or more layers can yield significantly improved performance. However, different from traditional WSNs, there are multi-constrained QoS requirements that have to be jointly satisfied by the cross-layer approach in asynchronous duty-cycled WSNs. The importance challenges that have to be taken into account during the design of cross-layered solution is the impact of QoS performance that comes with sleep scheduling. Especially in the case that system performance estimations in different layers have to be jointly considered and optimized in terms of throughput, delay and packet loss rate.

III. SYSTEM MODEL AND PROBLEM DESCRIPTION

A. SYSTEM MODEL

We assume a network with N nodes, where all of them operate under the duty-cycled asynchronous Low Power Listening (LPL) modes [29]. Each sensor only has two possible working states: the active state, in which the sensor can perform all the functions of sensing, listening, transmitting, and receiving; and the dormant state, in which the sensor turns off all the functional modules except for a wake-up timer. Specifically, when a dormant sensor wakes up, it either switches to the active state, or transmits packets and then switches back to the dormant state. In other words, a sensor can transmit a packet at any time but can receive a packet only when it is active. So, as shown in Figure 1 for node n_i and n_j, nodes under asynchronous LPL operation modes

TABLE 1. Notations and their descriptions for problem formulation.

Notation	Description
$N^S_{j,k}$	Number of packets sent by node n_j to node n_k
$N^R_{i,w}$	Number of packets received (listened) by node n_w from node n_i
$q_{j,w}$	The link quality indicates the average packet reception ratio between node n_j to node n_w
P^D_j	The probability of packet drop in node n_j
P^F_j	The probability of packet forward in node n_j
P^{MD}_j	The probability of malicious packet drop in node n_j
MN_j	Whether node n_j is a malicious node decided by watchdog node
τ	The threshold to make a decision for malicious nodes

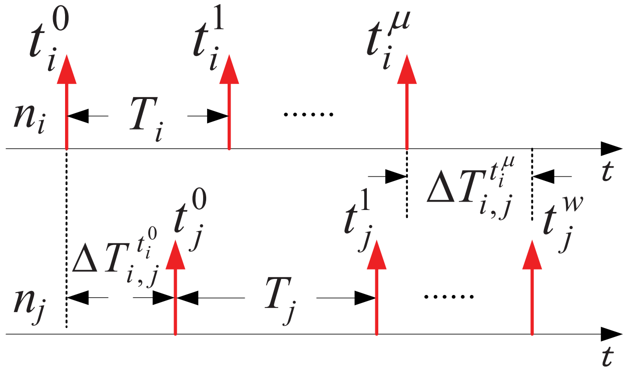


FIGURE 1. System model under asynchronous LPL operation mode.

will be fractionally awake in every slot, and listen to the channel for a short period of time, periodically polling the radio channel to check if there are any incoming packets. We define the duration of the LPL sleep interval for node n_i and n_j as T_i and T_j, respectively. It is possible that $T_i \neq T_j$ (ALPL) for two nodes n_i and n_j. The wake-up time expansion of each node n_i is modeled as discrete and infinite, which is expressed as $\{t_i^0, t_i^1, t_i^2, \cdots, t_i^M\}$, M is $+\infty$, and $t_i^M - t_i^{(M-1)} = T_i$.

B. PROBLEM DESCRIPTION

In asynchronous duty-cycled WSNs, different applications demand different QoS requirements. Specifically, it needs to exploit QoS-based routing metrics not only helps in providing the throughput performance, it is essential to ensure the different QoS aware for the application with delay constraint, the application with packet loss rate constraint or without. This paper addresses QoS aware routing problem and provides an approach to design the routing protocol to satisfy QoS constraints.

Definition 1: In asynchronous duty-cycled WSN, R stands for the routing path sets, where any path $Path(s, d)$ transmits from source node s to destination node d. Suppose $E(r)$ is the edge set of routing path r that $r \in R$, its QoS parameters can be described as follows:

(1) $throughput(r) = max\{throughput(e(i,j)), e(i,j) \in E(r)\}$.
(2) $delay(r) = \sum_{e(i,j)\in E(r)} delay(e(i,j))$.
(3) $loss(r) = min\{loss(e(i,j)), e(i,j) \in E(r)\}$.

Definition 2: The QoS aware routing problem of asynchronous duty-cycled WSN: In asynchronous duty-cycled WSN, an approach is needed to find a routing path r^* that $r^* \in R$ to satisfy the follow QoS performance and optimal conditions.

(1) $throughput(r^*) \geq T_{min}$.
(2) $delay(r^*) \leq D_{max}$.
(3) $loss(r^*) \leq L_{max}$.
(4) $cost(r^*)$ *is minimum.*

Here, T_{min}, D_{max} and L_{max} separately stands for lower limit throughput, upper limit delay and upper limit packet loss of the end-to-end QoS constraints. The $cost(r^*)$ is the cost of path, including the expected transmissions and the waiting latency. The formula (1) and (2) emphasis a high throughput and low latency solution for packets forwarded through the path while formula (3) requires the reliability and security. The formula (4) brings a kind of optimization target, where the reasonable path can meet multiple QoS constraints.

IV. PROTOCOL DESIGN

A. OVERVIEW

The QoS aware approach aim to design a new routing protocol to detect packet dropping attack for reliability and security while provide good network performance in terms of throughput and delay. The general operating process of this approach consists of malicious packet dropping detection, congestion degree measurement, QoS based routing metrics and algorithm, design of cross-layer QoS architecture, and optimization for control packet. Table 1 contains symbols and notations that are used throughout the entire paper.

B. MALICIOUS PACKET DROPPING DETECTION

To address network performance with link quality parameters, we have applied security analysis in packet dropping attacks. That is, the behaviour of malicious dropping attacks must be differentiated from others which are really legitimate due to the collisions and channel errors. Take Figure 2 as an example, it is assumed that the routing path transmits packets from source node S to destination node D, where node n_i, node n_j and n_l are intermediate nodes. For detecting malicious dropping behavior, let node n_w as a watchdog node to monitor the forwarding events of node n_i and node n_j for malicious decision. As for watchdog node n_w, the number of packets

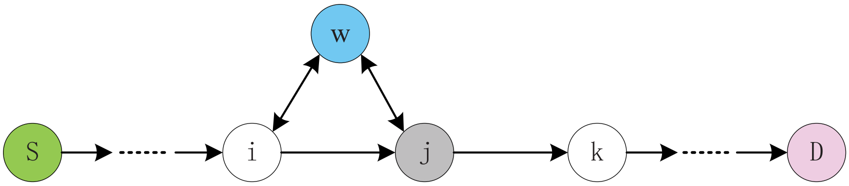


FIGURE 2. Detection model for packet dropping attacks.

$N^S_{j,k}$ forwarded by node n_j is computed as follow:

$$N^S_{j,k} = \frac{N^R_{j,w}}{q_{j,w}} \tag{1}$$

where $q_{j,w}$ is the link quality that indicates the average packet reception ratio between node n_j to node n_w, and $N^R_{j,w}$ is the number of packets received (listened) by watchdog node n_w while node n_j is forwarding to node n_k. Meanwhile, the number of packets $N^R_{i,j}$ actually received by node n_j from node n_i can be computed as follow:

$$\begin{aligned} N^R_{i,j} &= N^S_{i,j} \cdot q_{i,j} \\ &= \frac{N^R_{i,w}}{q_{i,w}} \cdot q_{i,j}. \end{aligned} \tag{2}$$

With the above computed $N^S_{j,k}$ and $N^R_{i,j}$, the packet loss rate P^D_j of node n_j is computed as follow.

$$\begin{aligned} P^D_j &= 1 - P^F_j \\ &= 1 - \frac{N^S_{j,k}}{N^R_{i,j}} \\ &= 1 - \frac{N^R_{j,w} \cdot q_{i,w}}{N^R_{i,w} \cdot q_{j,w} \cdot q_{i,j}} \end{aligned} \tag{3}$$

Take advantage of this watchdog technology, for node n_j, the received packets from node n_i and forwarded packets to node n_k will be monitored under the considering of link quality. In order to detect the malicious packet dropping, MN_j is defined to decide whether node n_j is a malicious node as follow:

$$MN_j = \begin{cases} 1, & P^D_j \geq \tau; \quad (malicious\ node) \\ 0, & otherwise. \quad (normal\ node) \end{cases} \tag{4}$$

where τ is the threshold, and $MN_j = 1$ means node n_j is a malicious node if $P^D_j \geq \tau$, otherwise it is the normal (legitimate) node. Obviously, the threshold τ is a decisive factor. Then the most important thing is that how to get the appropriate value for the threshold τ.

Monitoring by the selected watchdog nodes, the natural packet loss rate X_1 of the forward nodes in the routing path can be estimated, which we assumes that they obey the normal distribution. Their probability distributions are $X_1 \backsim (\mu_1, \sigma_1)$. When they are compromised by attackers and the total packet loss rate estimated will be X_2, and their probability distributions are the normal distribution that $X_2 \backsim (\mu_2, \sigma_2)$. If the detected node is a malicious node, its malicious packet dropping rate is assumed as P^{MD}. The total packet loss rate is attributed to either a natural packet loss event or a malicious dropping packet loss event or both of them that are independent events. Then, the total packet loss rate X_2 of the malicious node is as follow:

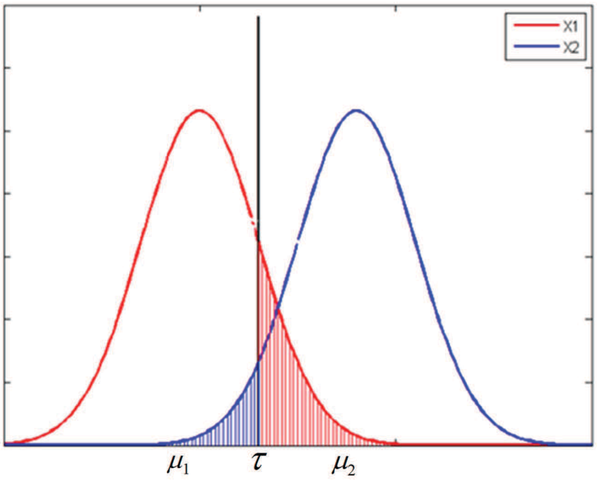


FIGURE 3. The impact of the setting for the threshold value τ.

$$\begin{aligned} X_2 &= 1 - (1 - P^{MD})(1 - X_1) \\ &= X_1 + P^{MD}(1 - X_1). \end{aligned} \tag{5}$$

The above Equation (5) shows that the total packet loss rate X_2 is equal to the natural packet loss rate X_1 plus the malicious packet dropping rate when the natural packet loss event did not happen. Absolutely, the total packet loss rate X_2 is greater than that of the natural packet loss rate X_1. The greater the malicious packet dropping rate P^{MD} is, the greater the gap of packet loss rate between X_2 and X_1 is. In other words, the greater P^{MD} becomes, the easier it is to distinguish malicious packet dropping from the natural packet dropping, which can be decided by the Equation (4) under the certain threshold value τ.

As shown in Figure 3, both of the natural packet loss rate X_1 and the total packet loss rate X_2 obey normal distribution. Take Figure 2 as an example, according to Equation (3) and Equation (4), the packet loss rate P^D_j can be computed by the watchdog node n_w and make a malicious decision for node n_j. And, more remarkable, once the threshold value τ is given fixed, false negative will occur, and its rate will rise with the increasing of the natural packet loss rate X_1, which can be shown as the red shaded area in Figure 3. The false negative rate $P_{f_N(\tau)}$ is calculated as follow:

$$P_{f_N(\tau)} = \int_{\tau}^{+\infty} f_{X_1} dx \tag{6}$$

where f_{X_1} is the probability density function of the natural packet loss rate X_1. Similarly, false positive will occur, and its rate will rise with the decreasing of the natural packet loss rate X_1, which can be shown as the blue shaded area in Figure 3. The false positive rate $P_{f_P(\tau)}$ is calculated as follow:

$$P_{f_P(\tau)} = \int_{-\infty}^{\tau} f_{X_2} dx \tag{7}$$

where f_{X_2} is the probability density function of the natural packet loss rate X_2.

Confront with the red shaded area $P_{f_N(\tau)}$ and the blue shaded area $P_{f_P(\tau)}$ shown in Figure 3, the problem for our detection mechanism is that, when the malicious packet dropping rate is small, $P_{f_N(\tau)}$ and $P_{f_P(\tau)}$ are not sufficiently separated, leading to large overlap between each other. This observation implies that when malicious packet dropping rate is relatively small, it is not easy to accurately differentiate between the malicious dropping behavior and the natural packet loss, which can be verified by Equation (5) that $X_2 \approx X_1$ if and only if P^{MD} tends to 0. Additional, under the time-varying network traffic, if the natural packet loss rate X_1 is dynamic and always changing, malicious dropping attacks can use this phenomenon to enhance their hiding features. For such cases, that is the reason why we should use the link quality estimation to correctly detect the malicious dropping nodes, as shown in Equation (1), Equation (2) and Equation (3).

Intuitively, the false negative rate $P_{f_N(\tau)}$ and the false positive rate $P_{f_P(\tau)}$ is contradictory each other. Compared with the actual packet loss rate, if the smaller the threshold value τ is, the bigger $P_{f_N(\tau)}$ will become, and the smaller for $P_{f_P(\tau)}$ even that is $P_{f_P(\tau)} = 0$. As shown in Figure 3, the red shaded area of $P_{f_N(\tau)}$ will increase while the blue shaded area of $P_{f_P(\tau)}$ will reduce. Conversely, the $P_{f_P(\tau)}$ will become bigger, and $P_{f_N(\tau)}$ will be the smaller or equal to that $P_{f_N(\tau)} = 0$. As shown in Figure 3, the blue shaded area of $P_{f_P(\tau)}$ will increase while the red shaded area of $P_{f_N(\tau)}$ will reduce.

Although there is no way to reduce $P_{f_N(\tau)}$ and $P_{f_P(\tau)}$ at the same time, meeting the actual requirements, it makes more sense to prefer one of the two rates for objective detection, or to make the tradeoff between both of each other. For example, to eliminate the malicious nodes as many as possible, it is better to reduce the threshold τ to an appropriate value, which can lead to some normal nodes identified as malicious. At the expense of these nodes, this will not affect the connectivity of network and the existence of routing paths because of the redundancy of WSNs.

For the appropriate threshold τ, we introduce a function $f(\tau)$ calculated by the sum of $P_{f_N(\tau)}$ and $P_{f_P(\tau)}$ with Equation (6) and Equation (7) respectively. Then,

Algorithm 1 Threshold Optimization Algorithm - Reasonable τ / τ^* Calculation for Accuracy Detection

```
Require: Network initialization and running process
         with no malicious node.
Input  : The probability distribution of natural packet
         loss $X_1$ that $X_1 \backsim (\mu_1, \sigma_1)$.
Output : Optimized Threshold $\tau$ / $\tau^*$.
$\tau_{init} \leftarrow min_{\tau}[P_{fN}(\tau) \leq P_{init}]$;
Malicious Detecting;
$X_2 \leftarrow X_2 \backsim (\mu_2, \sigma_2)$;
Case 1: Optimized Threshold;
$\tau \leftarrow$ Calculation by Equation (9);
Case 2: Preferred Optimization;
$\tau^* \leftarrow$ Calculation by Equation (10);
Return $\tau$ / $\tau^*$;
```

$$\begin{aligned} f(\tau) &= P_{f_N(\tau)} + P_{f_P(\tau)} \\ &= \int_{\tau}^{+\infty} f_{X_1} dx + \int_{-\infty}^{\tau} f_{X_2} dx. \end{aligned} \tag{8}$$

The optimization objective is to get threshold τ for minimizing the function $f(\tau)$, which can be expressed as follow:

$$\tau^* = min_{\tau} f(\tau). \tag{9}$$

The above optimization threshold τ^* is the derivatives of the objective function $f(\tau)$ that $f'(\tau) = 0$.

In practice, different applications may have different security requirements. Some ones give a preference for $P_{f_N(\tau)}$ while others are apt to $P_{f_P(\tau)}$. The former pursue the higher security by detecting the malicious nodes as much as possible, including the suspected normal nodes. The latter ensures that the protocol is available secure and can tolerate the malicious behavior of packet dropping. So, the optimization objective in Equation (8) and Equation (9) can be optimized with a preferred threshold τ as follows:

$$\begin{aligned} g(\tau) &= \alpha \cdot P_{f_N(\tau)} + (1-\alpha) P_{f_P(\tau)} \\ &= \alpha \cdot \int_{\tau}^{+\infty} f_{X_1} dx + (1-\alpha) \int_{-\infty}^{\tau} f_{X_2} dx \end{aligned} \tag{10}$$

$$\tau^* = min_{\tau} g(\tau) \tag{11}$$

where α is a preference factor and its value range is $[0, 1]$. The above optimization threshold τ^* is the derivatives of the objective function $g(\tau)$ that $g'(\tau) = 0$. If the factor α is set to 0.5, the preferred optimization threshold τ of Equation (10) and Equation (11) is degraded to Equation (8) and Equation (9) respectively, where $P_{f_N(\tau)}$ and $P_{f_P(\tau)}$ are considered just as important as each other.

According to the analyses above, we further give the calculation of threshold τ and τ^* based on Equation (9) and Equation (10) respectively, as shown in Algorithm 1.

Algorithm 2 Malicious Node Detection Algorithm - Based on Link Quality Estimation

```
Require: The watchdog is running in promiscuous mode
         to capture the forwarding packets.
Input  : $N^{R}_{j,w}$, $q_{j,w}$, $N^{S}_{i,j}$, $q_{i,j}$, $q_{i,w}$.
Output : $MN_j$.
1 $N^{S}_{j,k}$ ← Calculation by Equation (1);
2 $N^{R}_{i,j}$ ← Calculation by Equation (2);
3 $P^{D}_{j}$ ← Calculation by Equation (3);
  Detection by optimized threshold $\tau$ /
  $\tau^*$ in Algorithm 1;
4 $MN_j$ ← Calculation by Equation (4);
5 Return $MN_j$;
```

Any node acknowledged as watchdog node will run this algorithm for accuracy detection based on optimized τ / τ^*. After deployment, during the network initialization phase, the watchdog nodes perform analysis and statistics on the probability distribution of natural packet loss X_1 that $X_1 \backsim (\mu_1, \sigma_1)$, which is used as the input of the algorithm. Then, the initialized threshold τ_{init} can be calculated by minimizing the false negative rate $P_{f_N(\tau)}$ that $P_{fN}(\tau) \leq P_{init}$ since there is on malicious node (line 1). In the malicious packet dropping detection phase, the probability distribution of total packet loss rate X_2 will be attained that $X_2 \backsim (\mu_2, \sigma_2)$ (line 2). The optimized threshold τ (Case 1) and preferred optimization threshold τ^* (Case 2) with factor α can be calculated by Equation (9) and Equation (10), respectively (line 3 and line 4). Compared with fixed threshold set for all watchdog nodes in the network, the returned optimized threshold (line 5) help to improve the detection accuracy for each watchdog node independently. Furthermore, malicious node detection algorithm based on threshold is shown in Algorithm 2 according to Equation (1)-(4).

C. CONGESTION DEGREE MEASUREMENT

Due to the memory restrictions of the sensor nodes and limited capacity of a shared wireless medium, network congestion is a rather frequent occurrence. However, congestion may lead to a plethora of malfunctions such as packet loss, lower throughput and energy inefficiency, potentially resulting in reduced deployment lifetimes and under-performing applications, even seriously causing congestion collapse.

Different from traditional WSNs, under asynchronous duty-cycled LPL modes, waiting latency introduced will aggravate the occurrence of network congestion since much more time is spent in the limited queue. Its analysis plays an important role in measuring the congestion. Without loss of generality, suppose there are two adjacent node n_i and n_j on the transmission link $l_{i,j}$, where node n_i is the sender and node n_j is the receiver. As shown in Figure 1, the varying waiting latency under the single-hop delay model under asynchronous LPL operation mode should be calculated. Suppose at time t_i^0, the neighbor discovery latency (sleep latency) between node n_i and node n_j is $\Delta T^{t_i^0}_{i,j}$. Then at time $t_i^\mu = t_i^0 + \eta * T_i$, the neighbor discovery latency can be expressed as:

$$\Delta T^{t_i^\mu}_{i,j} = T_j - (t_i^\mu - t_i^0 - \Delta T^{t_i^0}_{i,j})\ mod\ T_j, \tag{12}$$

where $\Delta T^{t_i^0}_{i,j}$ can be measured in the following way: during the process of neighbor discovery, node n_i wakes up at t_i^0 and sends out a wake-up beacon which contains the node ID of n_i. Then node n_i keeps listening to the channel until it receives the wake-up beacon from node n_j. After receiving the wake-up beacon from n_j, node n_i sets the receiving interval from t_i^0 to the time at which the wake-up beacon from node n_j is received to $\Delta T^{t_i^0}_{i,j}$. Once the neighbor discovery latency $\Delta T^{t_i^0}_{i,j}$ at t_i^0 is measured, $\Delta T^{t_i^\mu}_{i,j}$ which is called sleep latency for data transmission can be calculated by Equation (12).

Based on the queue length and its variations, as shown in Figure 4, we present two cases to analysis the waiting latency $\tau^{t_i^\mu}_{i,j}$ over link (i, j), including the receiving time τ_{rcv} from node n_i's descendant nodes and the sleep latency $\Delta T^{t_i^\mu}_{i,j}$ for the parent node n_j to wake up. In case I, node n_j wakes up at time t_j^w that $t_i^\mu < t_j^w \leq t_i^\mu + \tau_{rcv}$, node n_i will wait at least one additional sleep interval T_j of node n_j; In case II, the wake-up time t_j^w of node n_j is that $t_j^w > t_i^\mu + \tau_{rcv}$, the waiting latency equals to $\Delta T^{t_i^\mu}_{i,j}$. So, the waiting latency $\tau^{t_i^\mu}_{i,j}$ for data transmission at time t_i^μ can be expressed as follows:

$$\tau^{t_i^\mu}_{i,j} = \begin{cases} \Delta T^{t_i^\mu}_{i,j} + \beta \cdot T_j, & t_i^\mu < t_j^w \leq t_i^\mu + \tau_{rcv}; \\ \Delta T^{t_i^\mu}_{i,j}, & t_j^w > t_i^\mu + \tau_{rcv}. \end{cases} \tag{13}$$

where $\beta \geq 1$. We assume that the sender n_i will try to wait for its receiver n_j to wake up and send the packets in the sending queue since it has finished receiving process from its child nodes. If the packet is not successfully received by the receiver during the wake-up time of the current sleep period, the sender will retransmit the packet up to maximum transmission times for reliability. This prevents a sender from keeping using the channel for a long time, so that other nodes cannot get the channel during their wake-up times, especially when the link quality is low.

Congestion in WSNs is classified into two categories. The first type is node-level congestion, which is caused by the overflow of buffers in a node, and results in packet loss and increasing queuing delays. The second called link-level occurs because of shared wireless channels in WSNs, where multiple active sensor nodes try to seize the channel at the same time. Furthermore, due to highly dynamic and poor wireless links, packets, more than often, are sent more than once, which results in more incoming packets requiring buffering in the sending queue.

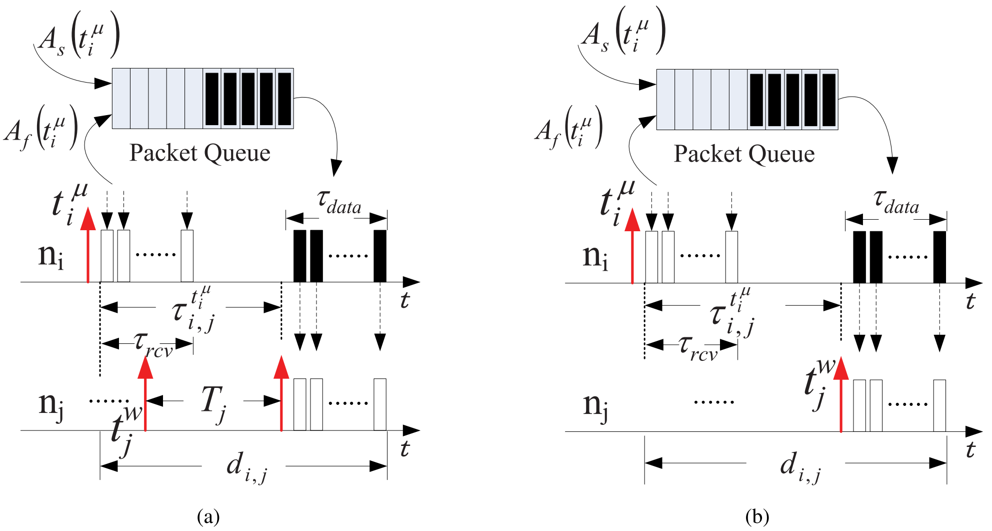


FIGURE 4. Varying waiting latency under the single-hop delay model under asynchronous LPL operation mode. (a) Case 1. (b) Case 2.

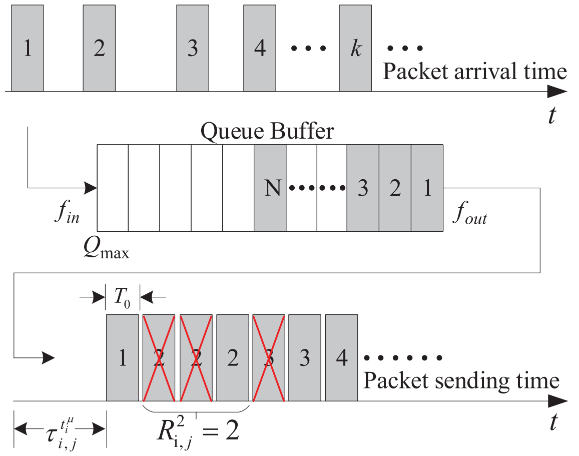


FIGURE 5. Congestion degree analysis based on queue model.

In order to preferably measure the local congestion degree at each intermediate node, congestion degree (*CD*) is defined to measure the degree for both node-level congestion and link-level congestion, which considers the size of the queue, the waiting latency and the retransmissions because of imperfect link quality. As shown in Figure 5, between node n_i and node n_j, the measurement of $CD_{i,j}$ is the transmission time for all the packets based on queue model, which consists of the waiting latency transmission delay, propagation delay, processing delay and queuing delay. Here, the propagation delay and processing delay are ignored since there are considerable differences in magnitude compared with the others.

Suppose the queue is full with packets, for the packet at the end of the queue, its maximum transmission time $T_{i,j}^{max}$ can be calculated as follow:

$$T_{i,j}^{max} = \tau_{i,j}^{t_i^\mu} + R_{i,j}^{max} \cdot Q^{max} \cdot \bar{T}_0 \tag{14}$$

where $\tau_{i,j}^{t_i^\mu}$ is calculated by Equation (13), $R_{i,j}^{max}$ is the maximum number of retransmissions which is related to the link quality $q_{i,j}$ and $q_{j,i}$, Q^{max} is the maximum length of packet queue while $\bar{T}_0$ is the time spent transmitting for each packet in the queue (which is related to the link bandwidth and packet size). Similarly, for the packet at the head of the queue that there is only one packet, its minimum transmission time $T_{i,j}^{min}$ with none of the retransmissions is calculated as follow:

$$T_{i,j}^{min} = \tau_{i,j}^{t_i^\mu} + \bar{T}_0. \tag{15}$$

Assuming that there are N packets in the queue, the average transmission time $\bar{T}_{i,j}$ of them can be expressed as follows:

$$\begin{aligned}\bar{T}_{i,j} &= \frac{\sum_{k=1}^{N}(\tau_{i,j}^{t_i^\mu} + R_{i,j}^k \cdot \bar{T}_0)}{N}\\ &= \tau_{i,j}^{t_i^\mu} + \frac{\bar{R}_{i,j}\sum_{k=1}^{N} k \cdot \bar{T}_0}{N}\end{aligned} \tag{16}$$

where $\bar{R}_{i,j}$ is the average number of retransmissions for the packets in the queue that $\bar{R}_{i,j} = \sum_{k=1}^{N} R_{i,j}^k / N$. Then, $CD_{i,j}$ metric reflecting the current link-level and node-level congestion degree between node n_i and node n_j can be calculated as follow:

$$CD_{i,j} = \frac{\bar{T}_{i,j} - T_{min}}{T_{max} - T_{min}}. \tag{17}$$

The above equation can make $CD_{i,j}$ to be changed to the dimensionless expression, where its value fall into the interval [0,1] for being compared and weighted. The larger the value of $CD_{i,j}$, the higher the congestion degree is.

D. QOS AWARE ROUTING METRICS AND ALGORITHM

The *ETX* metric in CTP (Collection Tree Protocol) implicitly minimizes radio power consumption. Since our routing mechanism aims at making routing decisions dependent on

Algorithm 3 Routing Algorithm - Reasonable Parent Node Selection for Routing Update

```
    Phase 1: Parent node Selection;
    Suppose node v's current parent node is node u;
1   $P_{etx}^{min} = MAX\_METRIC$;
2   $PNode = u$, $CNode = NULL$;
3   foreach Entry entry in routing table do
4       $w \leftarrow nodeID$;
5       $ETX_{v,w} \leftarrow$ Calculated by Equation 18 ;
6       $ETT_{v,w} \leftarrow$ Calculated by Equation 19 ;
7       $P_{ett}(v, w) \leftarrow ETT_{v,w} + P_{ett}(w, k)$;
8       if $w == PNode$ then
9           $P_{ett}(v, u) \leftarrow P_{ett}(v, w)$;
10          continue;
11      end
12      if (w's C bit is set) || $(MN_w == 1)$ then
13          continue;
14      end
15      if $P_{ett}(v, w) < P_{ett}^{min}$ then
16          $P_{ett}^{min} \leftarrow P_{ett}(v, w)$;
17          $CNode \leftarrow w$;
18      end
19  end
    Phase 2: Parent node switching;
    Conditions for switching is given in Table 2;
20  if $P_{etx}^{min}$ != $MAX\_METRIC$ then
21      if one of the conditions is established then
22          $PNode \leftarrow CNode$;
23      end
24  end
```

QoS awareness, related metrics that are direct functions of the aspects are introduced to constitute node state. The *ETX* metric, estimated by the bi-directional way using the 4bit LQE in CTP protocol, is computed as follow:

$$ETX_{i,j} = \frac{1}{q_{i,j} \cdot q_{j,i}}. \tag{18}$$

To enhance network performance under asynchronous duty-cycled LPL modes, for a single packet, the network delay between each hop (such as node n_i and node n_j) includes the waiting latency $\tau_{i,j}^{t_i^{\mu}}$ and the transmission delay T_0 under retransmission scheme. However, the retransmission cost is contained in *ETX* metric, and we ignore the propagation delay and processing delay. Then, the metric Expected Transmission Time (*ETT*) is computed as follow:

$$\begin{aligned} ETT_{i,j} &= (\tau_{i,j}^{t_i^{\mu}} + T_0) \cdot ETX_{i,j} \\ &= T_{i,j}^{min} \cdot ETX_{i,j}. \end{aligned} \tag{19}$$

In our proposed protocol, the routing metrics are updated at each intermediate node and the computed *ETT* value is embedded (piggybacked) into the wake-up beacon.

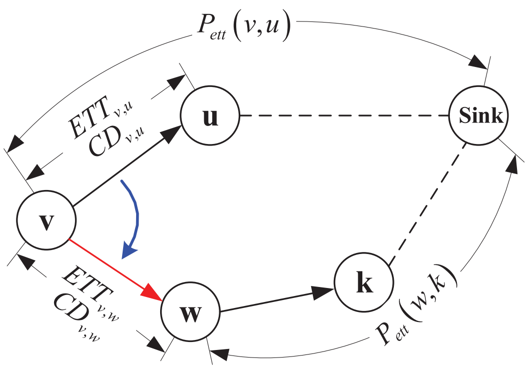


FIGURE 6. An illustrative example for reasonable parent node selection and switching.

The procedure of reasonable parent node selection and switching is given in Algorithm 3, which includes two phases. To facilitate presenting our algorithm, the related notations are given as follow:

- P_{ett}^{min}, which indicates the minimum path *ETT* metric from node v to the sink by the candidate node. The default value is MAX_METRIC;
- *PNode*, which indicates the current parent node;
- *CNode*, which indicates the selected candidate node;
- *entry*, which indicates the entry in the routing table, and it includes the *nodeID* of the node, the waiting latency, the link quality and so on;
- $P_{ett}^{v,w}$, which indicates the path *ETT* metric from node v to the sink by node w.

We consider an illustrative example shown in Figure 6, where node v needs to select the best candidate node w from the routing table and compares it with the current parent node u for switching as needed. In phase 1 (lines 1-19), node v firstly initializes some variables with certain values (lines 1-2). For each entry *entry* in node v's routing table, *ETX* and *ETT* are calculated by Equation (18) and Equation (19), respectively (lines 5-6). Then node v calculates its path *ETT* value $P_{ett}(v, w)$ by adding the $ETT_{v,w}$ value with the path *ETT* value $P_{ett}(w, k)$ of its neighbor w, where $P_{ett}(w, k)$ can be obtained from the state exchange using beacon and saved in the routing table (line 7). If node w is the current parent node *PNode*, node v will update its path *ETX* $P_{etx}(v, w)$ to the sink from node w of the current parent node *PNode* (lines 8-11). In order to control congestion and avoid packet dropping attack, some nodes must be ignored in the routing table, especially when their congestion bits (C) are set or their forwarding behaviour are annunciated as malicious by the watchdog nodes (lines 12-14). Importantly, their accurate detection mechanisms are running based on optimized threshold τ or preferred optimization threshold τ^* according to Algorithm 1 and Algorithm 2. For the rest of the neighbor nodes, such as node w, its path *ETT* value $P_{ett}(v, w)$ is compared with that of minimum variable P_{ett}^{min} (line 15). If $P_{ett}(v, w)$ is less than P_{ett}^{min}, P_{ett}^{min} will be updated and node w is temporarily recognized as the best candidate

TABLE 2. Conditions for candidate parent node switching.

Constraint Condition	NONE	$P_{etx}^{min} < P_{etx}(u,k) + \Delta ETX \cdot \bar{T}_0$
$P_{etx}(v,u) == MAX\_METRIC$	Condition 1	/
u's C bit is set	Condition 2	/
$CD_{v,u} \geq CD_{Threshold}$	Condition 3	/
$MN_u == 1$	Condition 4	/
$P_{ett}^{min} < P_{ett}(v,w)$	/	Condition 5

(lines 16-18). With the execution of this phase, node *v* can select a reasonable candidate parent node based on a comprehensive consideration of the expected transmission cost and the expected transmission delay under asynchronous duty-cycled LPL modes.

In phase 2 (lines 20-24), by properly selecting the candidate parent node, each node changes its current parent node to the selected node when one of the five decision conditions (Condition 1 ∼ Condition 5) is established as defined in Table 2. In Condition 1, the path *ETT* value $P_{ett}(v, u)$ is *MAX_METRIC*, which means that node *v*'s current parent node *u* is invalid. That usually happens when the deployed network is initialized with default value or the new node is first joined before neighbor discovery, especially in the beginning. Besides, at the running time, the broken of the link between node *v* and node *u* will reset the $P_{ett}(v, u)$ value to *MAX_METRIC*. Condition 1 indicates that node *v* must switch to the selected candidate parent node. Conditions 2 and Condition 3 validate whether node *v*'s current parent node *u* is congested due to the setting of a congestion bit and the exceeding value of the congestion degree $CD_{v,u}$ for the pre-determined threshold value $CD_{Threshold}$, respectively. Condition 4 indicates that the current parent node *u* identified as malicious, its threats can be completely eliminated by simply being excluded from the routing path, which helps to resist the packet dropping attacks. Condition 6 indicates that the routing metric *ETT* value of the selected candidate parent node *CNode* is less than that of the current parent node *u*, and node *v* will switch to candidate node *CNode* as its parent node if and only if there is an appreciable benefit in doing so. In order to avoid forming loops, node *v* try to select a node which is not a descendent of the current parent node *u*. So, according to Equation (19), we restrict the switching condition under $P_{ett}^{min} < P_{ett}(u, k) + \Delta ETX \cdot \bar{T}_0$, where $T_{i,j}^{min} \approx \bar{T}_0$ (which does not take into account the waiting latency) and $\Delta ETX = 10$ (which is a defined reference value in CTP protocol).

Above all, the main characteristic of our routing protocol is that we introduce related QoS aware metrics, where the metric *ETT* is used to find the routing path with the features of less-cost (*ETX*), lower-delay (waiting latency and transmission delay), while the metric *CD* helps to enhance congestion avoidance ability and the metric *MN* is adopted to detect malicious packet dropping attack and eliminate malicious node from the routing path. As for problem definition in Definition 2, the optimal conditions can be established by these three metrics, where *CD* is for condition (1), *ETT* is for condition (2) and (4), *MN* is for condition (3).

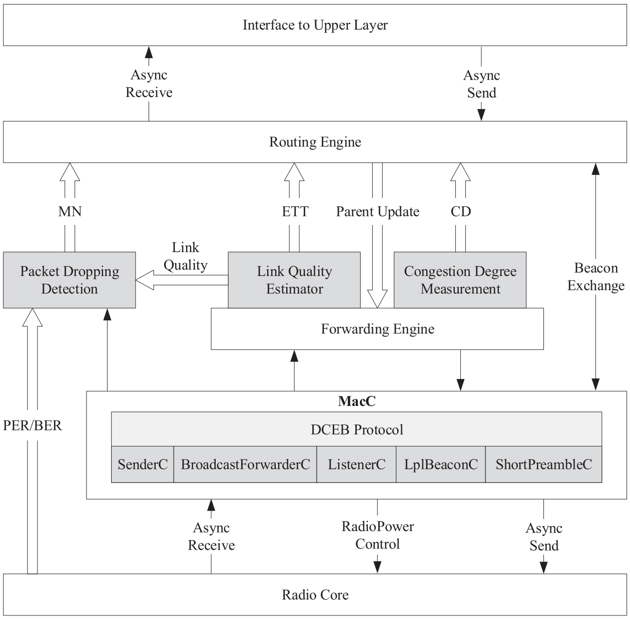


FIGURE 7. Cross-layer architecture for QoS aware TDL-based routing protocol.

E. CROSS-LAYER QOS AWARE ARCHITECTURE

In our prior work [30], under the component-based architecture [31], we designed and implemented a delay-constrained and energy-balanced multi-hop broadcast protocol called DCEB in low duty-cycled WSNs. Based on this architecture adopted cross-layer approach, as shown in Figure 7, we exploited DCEB to disseminate neighbor information through the network, and designed new components for reasonable parent node selection and good QoS aware, including *Link Quality Estimator*, *Congestion Degree Measurement* and *Packet Dropping Detection*.

In TinyOS, switching the power state of the radio between active and sleep, duty-cycled asynchronous LPL modes are provided by *Radio Core* component using the *RadioPowerControl* interface. Packets are sent and received asynchronously by *Radio Core* component through *AsyncReceive* and *AsyncSend* interfaces, which can be captured and filtered by upper layer for further analysis. Various components are composed together inside of a more general *MacC* configuration. Our previously proposed DCEB defines a *MacC* configuration that composes *SenderC*, *BroadcastForwarderC*, *ListenerC*, *LplBeaconC* and *ShortPreambleC* components, which can support our developed routing protocol for efficient unicast and broadcast in duty-cycled WSNs. In our TDL-based routing protocol, we implemented both *Link Quality Estimator* component and *Congestion Degree Measurement* component based on the *Forwarding Engine* component. The *Forwarding Engine* component is responsible for notifying the *Link Quality Estimator* component to update the link *ETX* and calculate the *ETT* according to Equation (19), and helping *Congestion Degree Measurement* component to calculate the *CD* according to Equation (17). Both *ETT* and *CD* are subsequently collated and fed to the *Routing Engine* component for reasonable parent nodes selection. In order to detect malicious packet dropping attacks, the *Packet Dropping Detection* component monitors the forwarding events of detected nodes by exploiting the broadcast characteristics of wireless channels, whereby the packets being forwarded is monitored and analyzed from physical Layer and MAC layer. In order to ensure the accuracy of the natural packet loss rate as mentioned in Section IV, link quality and PER (Packet Error Rate) / BER (Bit Error Rate) are respectively acquired by *Link Quality Estimator* component and *Radio Core* component. Eventually, malicious packet dropping attacks can be identified by Equation (4).

The *Routing Engine* component is responsible for updating routing task to provide the *Forwarding Engine* component with a reasonable parent node based on an overall routing metrics, including *ETT*, *CD* and *MN*. In addition, a *Timer* is equipped in this component. When the *Timer* expires, the operation of routing table update will be triggered. This routing mechanism in Algorithm 3 helps to find the path with the features of less-cost (*ETX*), lower-delay (waiting latency and transmission delay), which is capable of detecting malicious packet dropping attacks by the metric *MN* and enhancing congestion avoidance ability by the metric *CD*.

F. OPTIMIZATION FOR CONTROL PACKET

Beacons play an important role in maintaining routing topology. First, since beacons are broadcasts, it is the only way that nodes advertise their presence and build their neighboring tables. Second, on occurrence of significant events, for example, dramatic changes to route quality, the occurrence of congestion, or the detection of malicious behaviour, beacons are sent to inform other nodes. Our proposed DCEB protocol, designed for multi-hop broadcasts in asynchronous duty-cycled WSNs, is used for information exchange between neighbors and routing dynamically refresh. Moreover, DCEB can help to reduce control packet for beacon broadcast by its optimized forwarding mechanism.

Optimization for Reducing control packet are made not only by broadcast quantity, but also by broadcast timing according to the adjustable intervals, which refer to the routing updates and maintenance. Fortunately, the CTP protocol adjusts the communication rate of its beacons based on the expected importance of the beacon information. It sends routing packets (beacons) using a variant of the Trickle algorithm [32]. It maintains a beacon communication interval that varies between the minimum and maximum value, such as [128, 512000] in the CTP protocol. Whenever the timer expires, the interval is doubled, up to the maximum value. Therefore, in our proposed protocol, once one of the following five events is detected, the timer of the beacon communication interval in node v is reset to the minimum for real-time routing update.

1) Node v is asked to forward a data packet from a node whose *ETT* is lower than its own. This means that the nodes around node v have a significantly out-of-date estimate of routing cost.
2) Node v's routing cost decreases significantly. This event occurs because node v finds that it might provide lower-cost routes to nearby nodes.
3) Node v receives a packet with the P bit set. The P bit in the CTP protocol denotes that a node wishes to hear beacons from its neighbors.
4) Node v checks congestion degree *CD* that $CD_{v,u} \geq CD_{Threshold}$. This case, for congestion avoidance, requires node v to inform its child nodes to update the routing by Algorithm 3.
5) Node v checks that the variation of *CD* exceeds a certain threshold sent by the last beacon, namely $CD_{Update\_Threshold}$.
6) Watchdog node detects that node u is a malicious node, and immediately advertises for node v with a beacon identified that $MN_u = 1$.

The first three events are given in the CTP protocol. In the first event, node v needs to advertise its lower cost because other may wish to use it as a forwarder. In the second event, node v needs to advertise its higher cost because others may wish to stop using it as a forwarder. The third event is to enable rapid node bootstrapping and network joins. The forth event is used to advertise its higher *CD* to help neighboring nodes select their reasonable parent nodes for congestion avoidance by Algorithm 3. The fifth event indicates that the *CD* values stored in a routing table may be inconsistent with the actual value of the respective node. Therefore, if the *CD* value on a node is modified by $CD_{update\_threshold}$, node v triggers the beacon sending and helps its neighbor update the current *CD* values. The sixth event is used for watchdog node to advertise the malicious node, and the informed node v will perform Algorithm 3 for routing update, which helps to protect against the packet dropping attack.

V. PERFORMANCE EVALUATION

A. EXPERIMENT SETUP

This section we present the evaluation of our proposed QoS aware TDL-based routing protocol, which is implemented under the component-based architecture as shown in Figure 7. It runs on a TinyOS platform consisting of 20 deployed nodes

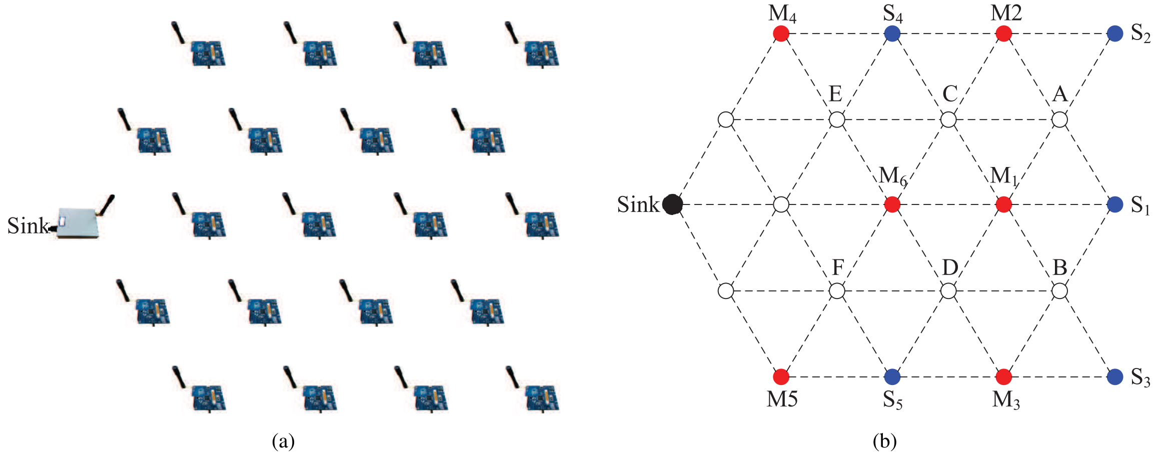


FIGURE 8. The deployed network under in-door testbed. (a) Deployed wireless sensor network. (b) Illustration of topological connections.

(EZ240) that remain connected to the sink (EZ520v2) on an in-door testbed, as shown in Figure 8. This experimental development kit (IOT-EZ240) is developed for wireless sensor network (WSN) and Internet of Things (IoT) by Institute of Computing Technology (ICT) Chinese academy of sciences.

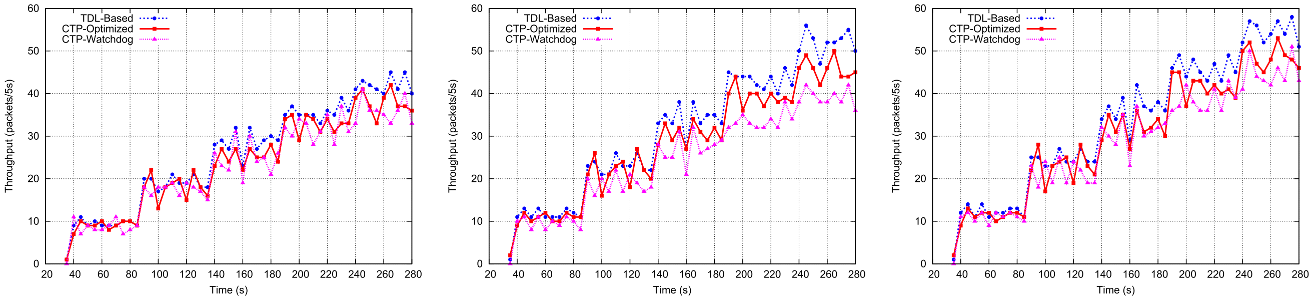
- EZ240, it is equipped with low power processor chip MSP430, radio frequency communication chip CC2420, Flash memory chip M25P80 and related sensor units, including humidity & temperature sensor and light sensor.
- EZ520v2, it serves as a gateway to help PC collect data from deployed network, where it can communicates with the deployed EZ240 nodes.

Because of the practical experiment constraints, such as lack of a limited number of experimental nodes and large-scale network for redundancy, we make the deployed network easy to satisfy topological connections as shown in Figure 8(a) and Figure 8(b), respectively. So that there exits at least one watchdog for the detection of forwarding events that occur on arbitrary forwarding links. For example, forwarding link from S_1 to M_1 in the routing path, both A and B can be the watchdogs. In the same way, as for the links from S_2 to M_2, S_3 to M_3, S_4 to M_4 and S_5 to M_5, there are A, B, E and F, respectively. Considering the case of continuous malicious nodes, such as M_1 and M_6, C or D can be the watchdog for M_6, which is independent of the detection M_1 by A. We set this scenario to show that the detection mechanism in our approach is adaptive to the continuous malicious dropping attacks in routing paths. Without loss of generality, by exploiting the redundancy of deployed nodes in large scale WSNs, more than one watchdog can be found and used to detect the each forwarding link in the routing paths.

The settings of experiment parameters are summarized in Table 3. The experiment is divided into five phases: (i) Neighbor discovery, each node in the network is keeping awake and maintaining its neighboring table according to the received beacons, including waiting latency and link quality estimation. (ii) Routing table construction, each node in the network constructs its routing table from the neighboring table, where the remains of the updating and maintaining for reasonable parent node selection are running by Algorithm 3. In order to make all the malicious nodes to be detected for verification and comparison, we set the fixed next routing nodes for S_1, S_2, S_3, S_4 and S_5 as M_1, M_2, M_3, M_4 and M_5, respectively. Then, the remaining routing selections of the paths to Sink are based on Algorithm 3. (iii) Network traffic generation, in order to generate different traffic load, five sensor nodes (S_1, S_2, S_3, S_4 and S_5) are arranged as source nodes, which will independently and periodically generated packet according to the different time. Then, 5 packets will periodically generated by each source node when each time it wakes up. Apparently, the network will be up against varying degrees of congestion. This is precisely what scenario we

TABLE 3. Experiment parameter settings.

Parameter	Setting
Number of Nodes/Sink	20 / 1
Deployment	In-door Testbed
Network Architecture	Homogeneous, Flat
Application Type	Continuous Traffic
LPL Sleep Interval	[1.5, 2.5] s
CC2420_DEF_RFPOWER	3 Level
MAX_RETRIES	5
Packet Size	64 Byte
Sending Queue Buffer Size	10
Beaconing Interval	[128, 512000] ms
Routing Update Interval	256 ms
$\triangle$Etx (Default Value in CTP)	10
Range of Preference Factor α	[0, 1]
$CD_{Update_Threshold}$	0.3
$CD_{Threshold}$	0.7



(a) (b) (c)

FIGURE 9. Comparison of throughput performance under different malicious packet dropping rate.

need to verify the advantage of our proposed approach. (iv) Malicious packet dropping attacks, during the forwarding for generated traffic, as forwarders in the routing paths, malicious nodes will perform packet dropping attacks according to setup time, such as M_1,M_2,M_3,M_4,M_5 and M_6 in Table 4. In the meantime, our proposed detection mechanism will take effect on routing update for bypassing detected malicious nodes. (v) Experimental results collection, in order to evaluate our proposed protocol and analyze the QoS performance, related results are required to compare with baseline approach.

TABLE 4. Network traffics and malicious dropping attacks.

Event	Node ID	Start Time	End Time
Traffic	S_1	30 s	280 s
	S_2	80 s	
	S_3	130 s	
	S_4	180 s	
	S_5	230 s	
Attack	M_1,M_2,M_3,M_4,M_5	30 s	280 s
	M_6	180 s	

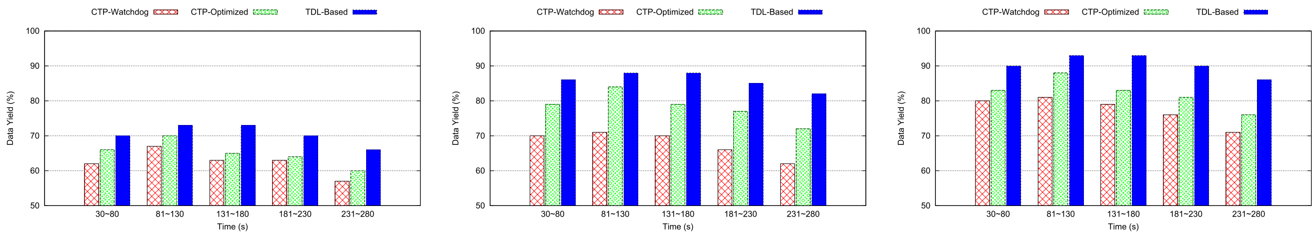
B. BASELINE AND PERFORMANCE METRICS

Since our work aims at proposing an QoS aware routing protocol for enhancing the network performance, we use the CTP protocol under the traditional watchdog technology with fixed threshold (called CTP-Watchdog) as our comparison baseline. As an improved approach for CTP-Watchdog, the proposed detection mechanism with optimized threshold (called CTP-Optimized) is used to verify the ability for detecting malicious packet dropping attacks. Also, based on CTP-Optimized, we introduce the QoS aware routing algorithm (called TDL-Based) to further verify the QoS performance. In those three approaches, the MAC protocol XMAC [33] designed for unicast transmitting is implemented and integrated into our DCEB protocol [30]. That can help support asynchronous duty-cycled LPL modes. For a comprehensive evaluation, four performance metrics are used for comparation. (1) **Throughput**: Confronted with the packets generated by source nodes, this metric is used to measure the number of packets arrived at the sink node in the network during certain time. (2) **Data Yield**: a metric that further measures data throughput is the amount of data successfully delivered to destinations, which presents the quantity of data received at the sink with respect to the total amount of data generated by source nodes in the network during certain time. (3) **E2E Delay**: the End-to-End (E2E) delay of data flows from the source node to the sink node, which related to the waiting latency, the hops from the source node to the sink node, the packet transmission delay and so on. (4) **Detection Accuracy**: this experimental results are used to evaluate the accuracy of the detection mechanism against malicious nodes, false negative rate, false positive rate and overall detection correctness are evaluated under different malicious packet dropping rate. (5) **Impact of Preference Factor**: in order to verify the applicability of our proposed approach, false negative rate and false positive rate are required to be analyzed under different values of the preference factor.

C. COMPARATIVE ANALYSIS

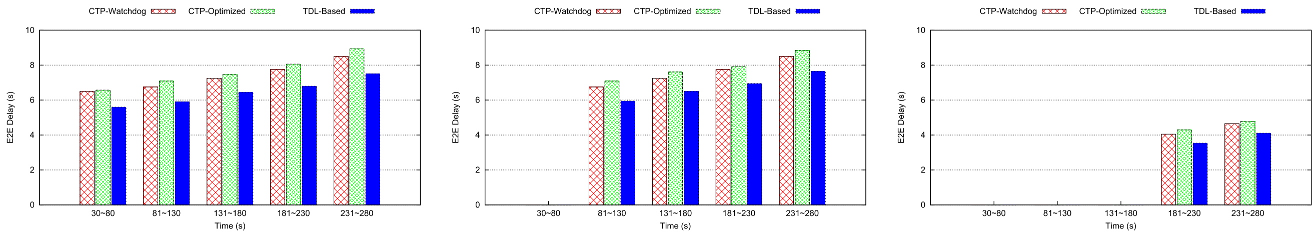
1) REGARDING THROUGHPUT

To better understand the performance of throughput, we show its statistics on each 5 seconds. Figure 9 illustrate the throughput comparison of TDL-Based, CTP-Optimized and CTP-Watchdog under different malicious packet dropping rate, such as $P^{MD} = 0.1$, $P^{MD} = 0.2$ and $P^{MD} = 0.3$. Generally speaking, the throughput of the three protocols increases with the increasing number of data flows (i.e., see the traffic settings in Table 4). It can be observed that at the time from 180s to 280s in Figure 9(a), 9(b) and 9(c), the proposed TDL-Based protocol achieves significantly better throughput performance than CTP-Watchdog protocol and CTP-Optimized protocol for all tested malicious packet dropping rate. This comparison results can be attributed to the reasons: the CTP-Watchdog protocol uses *ETX* as its routing metric while utilizing the watchdog technology with fixed threshold; Based on the same routing metric, the CTP-Optimized protocol improves the detection mechanism with optimized threshold, which can eliminate malicious node from the routing path by detect malicious packet dropping



(a) (b) (c)

FIGURE 10. Comparison of data yield performance under different malicious packet dropping rate.



(a) (b) (c)

FIGURE 11. Comparison of E2E delay performance for different data flows. (a) Date flow $S_1 \rightarrow Sink$. (b) Date flow $S_2 \rightarrow Sink$. (c) Date flow $S_4 \rightarrow Sink$.

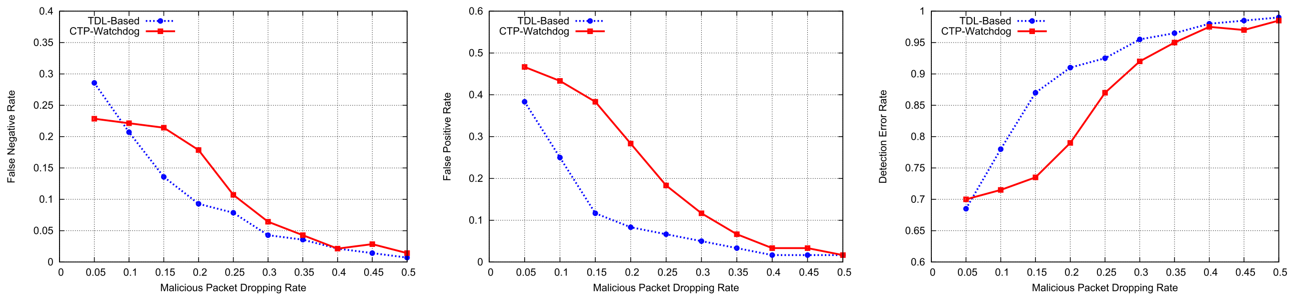
attack more precisely (i.e., see the comparison of detection accuracy in Figure 12); As for congestion control, both of these protocols set the C bit in the next routing frame and/or data frame based on CTP protocol; However, the TDL-Based protocol introduces the overall QoS aware metrics including ETT, CD and MN. The metric ETT is used to find the routing path with the features of less-cost (ETX), lower-delay (waiting latency and transmission delay); the metric CD helps to enhance congestion avoidance ability, and the metric MN is adopted to identify the malicious nodes by the detection mechanism like the CTP-Optimized protocol. These are the reasons why the throughput performance of the CTP-Optimized protocol is better than that of the CTP-Watchdog protocol, and the TDL-Based protocol achieves the best adaptability with the traffic load increasing. As shown in Figure 9(c), we give the tested results under $P^{MD} = 0.3$, where the detection accuracy of all three protocols are very close (i.e., see the comparison of detection accuracy in Figure 12). So, ignoring the performance differences of malicious packet dropping, it can be observed that the throughput performance of the TDL-based protocol is better than those of other two protocols. This in turn reflects the advantage of the overall QoS aware metrics of our TDL-Based protocol.

2) REGARDING DATA YIELD

Comparisons of data yield under different malicious packet dropping rate are given in Figure 10, which can help to demonstrate the performance of packet loss. As we analyze the experimental results, expect for the collisions and channel errors, the main reasons for packet loss are due to malicious packet dropping and congestion problem. So, it can be observed that the data yield first increases and then decreases with the increasing number of data flows (i.e., see the traffic settings in Table 4), since the congestion is become more and more serious from 131s to 280s. In contrast, the data yield under $P^{MD} = 0.1$ (i.e., see the Figure 10) is much more less than those under the other two (i.e., see the Figure 10(b) and 10(c)). This is explained by the accuracy of the detection mechanism against malicious nodes (i.e., see the detection accuracy in Figure 12 under $P^{MD} = 0.1$), where the overall detection correctness rate is 0.72 and 0.78 because of the false negative rate and false positive rate. So, the higher data yields in Figure 10(b) and 10(c) are attributed to the much more less false positive rate that more than 90% of malicious nodes can be detected (i.e., see the false positive rate in Figure 12(b)). Noticeably, the data yield of TDL-Based is higher than that of the other two protocols, where the growth values are about 20% and 10% when the traffic load occurs at the time from 230s to 280s under $P^{MD} = 0.2$. Those growth values become less obvious with increasing detection accuracy when the malicious packet dropping rate is that $P^{MD} = 0.3$ (i.e., see the false positive rate in Figure 12(a), the false positive rate in Figure 12(b) and the overall detection correctness rate in Figure 12(c). Therefore, the TDL-based routing protocol achieves less loss-rate than the CTP-Watchdog protocol and CTP-Optimized protocol.

3) REGARDING E2E DELAY

For the sake of comparison under different probable routing hops, three different data flows at different start time (i.e., see the traffic settings in Table 4), such as $S_1 \rightarrow Sink$, $S_2 \rightarrow Sink$ and $S_4 \rightarrow Sink$ in Figure 11, are selected to show the performance of E2E delay of these three protocols. As we known,



(a)

(b)

(c)

FIGURE 12. Comparison of detection accuracy under different malicious packet dropping rate. (a) False negative rate. (b) False positive rate. (c) Overall detection correctness rate.

the sleep interval is set to [1.5, 2.5]s (i.e., see the experiment parameter settings in Table 3), which is the main cause to the E2E delay. It can be observed that the E2E delay of the TDL-Based protocol is less than those of other two protocols, and the CTP-Optimized protocol has a close performance to that of the CTP-Watchdog protocol. These are explained by the different routing metrics, both the CTP-Watchdog protocol and the CTP-Optimized protocol use the metric *ETX* to find paths that minimize the expected transmissions while the metric Expected Transmission Time (*ETT*) is adopted to consider not only the expected transmissions but also the waiting latency that is recognized as one of the main causes of E2E delay in asynchronous duty-cycled WSNs. This can help the TDL-Based protocol find the routing path with the features of less-cost and lower-delay in the meantime.

4) REGARDING DETECTION ACCURACY

We verify the detection accuracy achieved by the proposed approach with the optimized threshold algorithm, which exploits the link quality estimation based on watchdog technology. We consider the type of malicious packet dropping as random dropping attack, where packets are dropped at the malicious node with a certain probability. In our experiment, we are interested in the following three accuracy metrics: false negative rate (P_{fN}), false positive rate (P_{fP}) and the overall detection correctness rate (P_{CN}), which their experimental results will be given under different malicious packet dropping rate. Let the number of all nodes and the malicious nodes be N and M, respectively, and the number of nodes because of P_{fN} and P_{fP} is e and m. Then, these three accuracy metrics are calculated as follows:

$$\begin{aligned} &\mathit{Accuracy\ Metrics} \\ &= \begin{cases} P_{fN} = \dfrac{e}{N-M} \\ P_{fP} = \dfrac{m}{M} \\ P_{CN} = \dfrac{(N-M-e)+(M-m)}{N}, \end{cases} \end{aligned} \tag{20}$$

where $N - M - e$ is the number of normal nodes correctly detected as normal while $M - m$ is that of malicious nodes correctly detected as malicious.

The detection accuracy is shown in Figure 12 under different malicious packet dropping rate. In each subfigure, there are two sets of curves, representing the proposed detection mechanism with optimized threshold (TDL-Based) and the traditional watchdog technology (CTP-Watchdog), respectively. In general, the detection accuracy of both them improves with malicious packet dropping rate (i.e., the overall detection correctness rate increases with the increasing of malicious packet dropping rate). This is obviously true that malicious packet dropping detection becomes more statistically distinguishable as the malicious node starts to drop more packets, while comparing with the estimated error of natural packet losses. As shown in Figure 12, when the malicious packet dropping rate is 0.05, the proposed TDL-Based approach provides higher false negative rate (subfigure 12(a)) but lower false positive rate (subfigure 12(b)) than the Watchdog scheme. These results imply that the TDL-Based approach leads to more false negative rate when the malicious packet dropping rate is less. Essentially, in our approach, the false positive rate is reduced at the expense of the false negative rate, which is reasonable if the malicious packet dropping rate is relatively small. The slightly higher false negative rate should not be a problem because of the node redundancy in large scale network. On the contrary, a low false positive rate is very desirable to resist attack, because it means a malicious node can be detected with a higher probability. Most importantly, the overall detection correctness rate of TDL-Based approach is approximate to that of CTP-Watchdog as shown in subfigure 12(c). When the malicious packet dropping rate is larger than 0.05, the false negative rate and the false positive rate of TDL-Based approach are both lower than those of CTP-Watchdog, which makes our approach achieve a much better detection accuracy, especially when the malicious packet dropping rate is 0.15 and 0.2. Meanwhile, when the malicious packet dropping rate is larger than 0.25, we note that the overall detection correctness rate of CTP-Watchdog is getting closer to that of TDL-Based approach since the other two rates is getting closer to each other. Obviously, the slightly higher is the malicious packet dropping rate, the significantly higher the overall detection correctness rate could perform. Also, this makes the

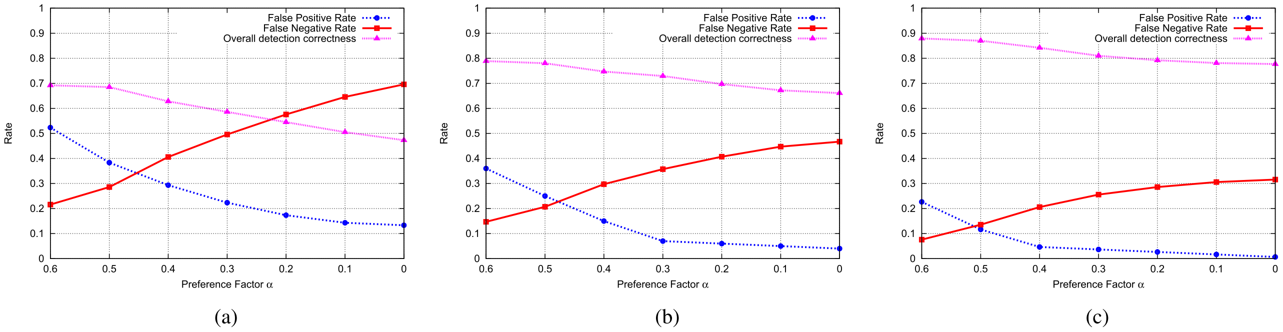


FIGURE 13. Analysis of detection accuracy under different preference factor α.

malicious nodes revealed and detected which helps in identifying the real cause of packet dropping more accurately.

5) IMPACT OF PREFERENCE FACTOR

In order to further verify the applicability of our proposed approach, false negative rate and false positive rate are required to be analyzed under different values of the preference factor α. As shown in Figure 13, with the decreasing of α, our proposed approach provides decreasing in the false negative rate and increasing in the false negative rate. This can be explained by noting that the preference factor α decides the balance between these two rates according the Equation 10 and 11. Also, as we known, the sum of these two rates will be minimized in theory when the preference factor α is 0.5, which make the overall detection correctness rate reach maximum value. However, in actual experimental results, this occurs only when the malicious packet dropping rate is 0.15 (subfigure 13(b)). The reason is that the optimized threshold of our approach can not completely eliminate the estimated error from the effect of dynamic natural packet losses due to shared wireless channel and changing network traffic. Confront with the values of α, decreasing from 0.5 to 0, the false positive rate is decreasing at the expense of the increasing in false negative rate, which leads to the slightly decreasing in the overall detection correctness rate. This evaluation results show that the preference factor α can help to make a tradeoff between the false negative rate and the false positive rate, and it does not appear to greatly affect the decreasing in the overall detection correctness rate. Our approach suggests that the larger the network, the less the preference factor α will be to detect the malicious nodes as much as possible. Inevitably, it detects falsely normal nodes as malicious ones, which will have no impact on the network connectivity because of the redundancy due to the large number of nodes. Otherwise, in the case that α is larger than 0.5, the variation trends of the false negative rate and the false positive rate are different from that of the case that α is less than 0.5. We consider it is not reasonable to sacrifice the decreasing in false positive rate for the increasing false negative rate. The original intention of our approach is to make effort for detecting all the malicious nodes, which requires the false positive rate to be as less as possible.

VI. CONCLUSION AND FUTURE WORK

QoS aware is an important requirement of asynchronous duty-cycled WSNs since deployed sensor node independently schedules itself under LPL modes. QoS routing technology mainly requires to calculate the feasible path under the related routing metrics, and also to optimize the many possibly existing paths. In this paper, we are especially interested in the QoS aware of asynchronous duty-cycled WSNs in the light of designing routing protocol, whereby waiting latency and malicious packet dropping are critical to the to the network performance. To improved the QoS performance through cross-layer approach, we propose a QoS aware routing protocol for enhanced performance in Throughput, Delay and packet Losses (TDL), where the path has the features of higher-throughput, lower-delay and less-loss rate. Besides the routing algorithm in TDL-Based protocol, an optimized detection mechanism is proposed for malicious packet dropping attack, and a method for congestion degree measurement is proposed based on sleep latency and queue length. Experimental results demonstrate that the QoS performance of our TDL-based routing protocol is better than those of the CTP-Watchdog protocol and the CTP-Optimized protocol, especially in terms of higher-throughput, lower-delay and less-loss rate. By optimized threshold, the detection accuracy of our detection mechanism is improved much more than that of the traditional watchdog technology. Furthermore, we evaluate that the preferred optimization factor can help to make a tradeoff between the false negative rate and the false positive rate. Overall, because our TDL-based routing protocol is implemented under the component-based architecture by cross-layer approach, we are convinced that it will be recognized as a practicable design of QoS-aware routing protocol in asynchronous duty-cycled WSNs.

Some open issues remain to be explored in our future work. we plan on addressing the mobility of the deployed sensor nodes as a future work. In such cases, the frequent update of the position will pose increased challenges to QoS aware performance, such as the detection mechanism for malicious packet dropping attack and how to handle the overhead of mobility and topology changes. As far as our current work is concerned, based on proposed QoS aware approach, we will further study QoS parameters and requirements in asynchronous duty-cycled WSNs, where the related QoS metrics may be used to guarantee the QoS performance. If possible we will do the future experiment in a larger network since the size of current experimental platform in this paper is relatively small. Additional, it is worth noting that collusive attack is beyond the scope of our approach, and the related defense mechanism is also our future work.

ACKNOWLEDGMENT

The authors would like to thank all of the anonymous reviewers of this work for their valuable comments.

REFERENCES

[1] G. A. Shah, S. Sohail, and F. B. Hussain, "QoS in wireless sensor networks," in *Intelligent Quality of Service Technologies and Network Management: Models for Enhancing Communication*. Hershey, PA, USA: Information Resources Management Association, 2010, pp. 53–74.

[2] J. Li and G. AlRegib, "Network lifetime maximization for estimation in multihop wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2456–2466, Jul. 2009.

[3] F. Liu, C.-Y. Tsui, and Y. J. A. Zhang, "Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 7, pp. 2258–2267, Jul. 2010.

[4] I. Al-Anbagi, M. Erol-Kantarci, and H. T. Mouftah, "A survey on cross-layer quality-of-service approaches in WSNs for delay and reliability-aware applications," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 525–552, 1st Quart., 2016.

[5] G. M. Shafiullah, A. Gyasi-Agyei, and P. J. Wolfs, "A survey of energy-efficient and QoS-aware routing protocols for wireless sensor networks," in *Novel Algorithms and Techniques in Telecommunications, Automation and Industrial Electronics*. Dordrecht, The Netherlands: Springer, 2008, pp. 352–357.

[6] S. Kumar, M. Dave, and S. Dahiya, "ACO based QoS aware routing for wireless sensor networks with heterogeneous nodes," in *Emerging Trends in Computing and Communication*. New Delhi, India: Springer, 2014, pp. 157–168.

[7] P. T. Kasthuribai and M. Sundararajan, "Secured and QoS based energy-aware multipath routing in MANET," *Wireless Pers. Commun.*, vol. 101, no. 4, pp. 2349–2364, 2018.

[8] A. Masoum, N. Meratnia, A. Dilo, Z. Taghikhaki, and P. J. M. Havinga, "Cross-layer analyses of QoS parameters in wireless sensor networks," in *Advances in Networks and Communications*. Berlin, Germany: Springer, 2011, pp. 595–605.

[9] F. Semchedine, N. A. Saidi, L. Belouzir, and L. Bouallouche-Medjkoune, "QoS-based protocol for routing in wireless sensor networks," *Wireless Pers. Commun.*, vol. 97, no. 3, pp. 4413–4429, 2017.

[10] C. Basaran and K.-D. Kang, *Quality of Service in Wireless Sensor Networks*. London, U.K.: Springer, 2009, pp. 305–321.

[11] W. Cai, X. Jin, Y. Zhang, K. Chen, and R. Wang, "ACO based QoS routing algorithm for wireless sensor networks," in *Ubiquitous Intelligence and Computing*. Berlin, Germany: Springer, 2006, pp. 419–428.

[12] L. Zhang, J. Yu, and X. Deng, "Modelling the guaranteed QoS for wireless sensor networks: A network calculus approach," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, no. 1, pp. 1–14, 2011.

[13] M. Kumaraswamy, K. Shaila, V. Tejaswi, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Efficient retransmission QoS-aware MAC scheme in wireless sensor networks," in *Networks and Communications*. Cham, Switzerland: Springer, 2014, pp. 31–42.

[14] N. Ababneh, N. Timmons, and J. Morrison, "A cross-layer QoS-aware optimization protocol for guaranteed data streaming over wireless body area networks," *Telecommun. Syst.*, vol. 58, no. 2, pp. 179–191, 2015.

[15] S. Mishra, T. N. Dinh, M. T. Thai, J. Seo, and I. Shin, "Optimal packet scan against malicious attacks in smart grids," *Theor. Comput. Sci.*, vol. 609, pp. 606–619, Jan. 2016.

[16] D.-R. Chen, "An energy-efficient QoS routing for wireless sensor networks using self-stabilizing algorithm," *Ad Hoc Netw.*, vol. 37, pp. 240–255, Feb. 2016.

[17] H. El Hammouti, L. Echabbi, and Y. Ben Maissa, "Maximizing QoS in heterogeneous wireless sensor networks using game theory and learning algorithms," in *Advances in Ubiquitous Networking*. Singapore: Springer, 2016, pp. 225–236.

[18] O. Deepa and J. Suguna, "An optimized QoS-based clustering with multipath routing protocol for wireless sensor networks," *J. King Saud Univ.-Comput. Inf. Sci.*, pp. 1–12, Dec 2017.

[19] B. R. T. Bapu and L. C. S. Gowd, "Link quality based opportunistic routing algorithm for QoS: Aware wireless sensor networks security," *Wireless Pers. Commun.*, vol. 97, no. 1, pp. 1563–1578, 2017.

[20] K. Selvakumar and R. M. Arieth, "Contrast for QOS based clustered energy efficient protocol with PSO and multi-hop gateways in wireless sensor network," *Cluster Comput.*, pp. 1–8, Dec. 2017.

[21] M. Z. Hasan, F. Al-Turjman, and H. Al-Rizzo, "Analysis of cross-layer design of quality-of-service forward geographic wireless sensor network routing strategies in Green Internet of Things," *IEEE Access*, vol. 6, pp. 20371–20389, 2018.

[22] S. Yahiaoui, M. Omar, A. Bouabdallah, E. Natalizio, and Y. Challal, "An energy efficient and QoS aware routing protocol for wireless sensor and actuator networks," *AEU-Int. J. Electron. Commun.*, vol. 83, pp. 193–203, Jan. 2018.

[23] M. Faheem and V. C. Gungor, "Energy efficient and QoS-aware routing protocol for wireless sensor network-based smart grid applications in the context of industry 4.0," *Appl. Soft Comput.*, vol. 68, pp. 910–922, Jul. 2018.

[24] J. Ruiz, J. R. Gallardo, D. Makrakis, L. Villasenor-Gonzalez, and H. T. Mouftah, "Cross-layer medium access control protocol with quality-of-service guarantees for wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2011, no. 1, p. 179, 2011.

[25] L. Shi and A. O. Fapojuwo, "Minimizing power cost in QoS constrained wireless sensor networks," *Int. J. Wireless Inf. Netw.*, vol. 20, no. 1, pp. 13–26, 2013.

[26] Y. Ozen and C. Bayilmis, "urgMAC: A new traffic and QoS-aware cross-layer MAC protocol for wireless multimedia sensor networks," *Comput. J.*, vol. 61, pp. 1460–1467, Dec. 2017.

[27] J. H. Kim, S. C. Jeon, and H. J. Park, "QoS aware energy-efficient (QAEE) MAC protocol for energy harvesting wireless sensor networks," in *Convergence and Hybrid Information Technology*. Berlin, Germany: Springer, 2012, pp. 41–48.

[28] Y.-D. Kim, K.-R. Cho, H.-S. Cho, and D. Kim, "A cross-layer channel access and routing protocol for medical-grade QoS support in wireless sensor networks," *Wireless Pers. Commun.*, vol. 77, no. 1, pp. 309–328, 2014.

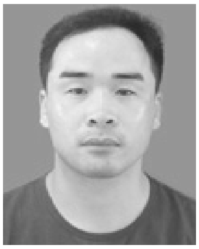
[29] D.-S. Vu, T.-N. Dao, and S. Yoon, "DDS: A delay-constrained duty-cycle scheduling algorithm in wireless sensor networks," *Electronics*, vol. 7, no. 11, p. 306, 2018.

[30] P. Shi, C. Gu, Z. Jing, and Z. Yu, "A constraint-based forwarding for multi-hop broadcasts in asynchronous duty-cycled WSNs," *Ad Hoc Sensor Wireless Netw.*, vol. 37, nos. 1–4, pp. 1–34, 2017.

[31] K. Klues, G. Hackmann, O. Chipara, and C. Lu, "A component-based architecture for power-efficient media access control in wireless sensor networks," in *Proc. 5th Int. Conf. Embedded Netw. Sensor Syst.*, 2007, pp. 59–72.

[32] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proc. 1st USENIX/ACM Symp. Netw. Syst. Design Implement. (NSDI)*, 2004, pp. 15–28.

[33] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst.*, 2006, pp. 307–320.



PEIZHONG SHI was born in Kunshan, Jiangsu, China, in 1982. He received the Ph.D. degree from the School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu, in 2014. He is currently with the School of Computer Engineering, Jiangsu University of Technology, Changzhou, Jiangsu. He was a Visiting Scholar with the Department of Information System, City University of Hong Kong, from 2018 to 2019. He is an In charge of the National Natural Science Foundation of China (NSFC). His current research interests include wireless sensor networks, QoS guarantee based on cross-layer approach, the Blockchain security, and its applications based on the consortium Blockchain architecture.

CHUNSHENG GU was born in Wuhu, Anhui, China, in 1971. He received the Ph.D. degree from the School of Management, University of Science and Technology of China, in 2005. He is currently a Professor with the School of Computer Engineering, Jiangsu University of Technology, Changzhou, Jiangsu, China. He is also in charge of the National Natural Science Foundation of China (NSFC). His current research interests include lattice-based cryptography, cloud computing and security, data security, and privacy protection.

CHUNPENG GE was born in Yancheng, Jiangsu, China, in 1987. He received the Ph.D. degree in Computer Science from the Nanjing University of Aeronautics and Astronautics, in 2016. He is currently with the School of Computer Engineering, Jiangsu University of Technology, Changzhou, Jiangsu. He is a Research Fellow with the Singapore University of Technology and Design. He is also an In charge of the National Natural Science Foundation of China (NSFC). His current research interests include cryptography, information security and privacy preserving for Blockchain, public key encryption with keyword search, proxy re-encryption, identity-based encryption, and techniques for resistance to CCA attacks.

ZHENGJUN JING was born in Danyang, Jiangsu, China, in 1978. He received Ph.D. degree in information and security from the Nanjing University of Posts and Telecommunications, in 2015. Since 2016, he has been an Associate Professor with the School of Computer Engineering, Jiangsu University of Technology, Changzhou, Jiangsu, China. His current research interests include public key cryptography scheme design and application, cloud computing and security, data security and privacy protection, and intelligent information systems.