Computer Networks 62 (2014) 101–121

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Improving broadcast reliability for neighbor discovery, link estimation and collection tree construction in wireless sensor networks

Behnam Dezfouli [a,b,*], Marjan Radi [a,b], Shukor Abd Razak [a], Kamin Whitehouse [c], Kamalrulnizam Abu Bakar [a], Tan Hwee-Pink [b]

[a] *Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor, Malaysia*
[b] *Networking Protocols Department, Institute for Infocomm Research (I^2R), A*STAR, Singapore*
[c] *Department of Computer Science, University of Virginia, Charlottesville, VA, USA*

ARTICLE INFO

Article history:
Received 31 May 2013
Received in revised form 19 November 2013
Accepted 17 January 2014
Available online 27 January 2014

Keywords:
Large-scale sensor networks
MAC
Collision avoidance
Routing

ABSTRACT

Neighbor Discovery and Link Estimation (NDLE) phase and Collection Tree Construction (CTC) phase are essential for correct and efficient operation of network protocols. However, the accuracy of these phases is highly affected by packet collisions, because CSMA is used for access arbitration and it does not support collision avoidance with broadcast transmissions. To improve NDLE accuracy: (i) We propose contention window adjustment mechanisms that rely on collision detection through the capture effect. In contrast to the existing approaches that utilize a long inter-packet duration for collision avoidance, the proposed mechanisms do not depend on network configuration and can provide adaptive collision avoidance with respect to the local collision intensity. (ii) We propose a mathematical model through which the MAC protocol can be configured to achieve a desired broadcasting success probability. (iii) We investigate and show the potential benefits of exploiting partially recovered packets during the NDLE phase. To improve CTC accuracy, we propose the Geowindow algorithm, which reduces packet collisions through contention window size management and transmission prioritization. Our results show that the Geowindow algorithm can improve the efficiency of the TinyOS's Collection Tree Protocol up to 74% in terms of tree cost, without increasing duration or energy consumption. Also, it can improve the packet delivery performance up to 70% in data gathering scenarios. The proposed MAC mechanisms of this paper are not only suitable for the initialization phases, but they can also be used for NDLE and CTC updates during the regular network operation, as well as other broadcast-based traffic patterns.

1. Introduction

In contrast to the centralized wireless networks (in which all the nodes communicate directly with a base station), sensor networks are deployed and operate in a distributed manner. After network deployment, a Neighbor Discovery and Link Estimation (NDLE) protocol should be executed by all the nodes to gather neighborhood information and estimate link qualities [1,2]. The importance of NDLE can be studied from various perspectives. For example, from the network-layer point of view, since multi-hop communication is used for data transmission, and due to the unreliability of low-power wireless links, routing protocols mainly rely on link qualities to find the most

[*] Corresponding author at: Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia (UTM), Johor, Malaysia. Tel.: +60 16 639 2405.

E-mail address: dezfouli@ieee.org (B. Dezfouli).

http://dx.doi.org/10.1016/j.comnet.2014.01.002
1389-1286/© 2014 Elsevier B.V. All rights reserved.

efficient paths towards the sink node [3,4]. Also, geographic routing protocols require each node to be aware of its neighboring nodes to find the shortest paths towards the destination [5]. At the MAC layer, many protocols rely on neighborhood information to establish collision-free slot assignment [6]. After the NDLE phase, since the main observable traffic pattern in sensor networks is many-to-one (a.k.a., *converegecast*) [7,8], it is the responsibility of the Collection Tree Construction (CTC) phase to establish efficient paths from each node towards the sink [9,10]. Consequently, NDLE and CTC are essential to operationalize a wireless sensor network.

During the NDLE phase, nodes should broadcast a fixed number of beacon packets to identify and measure the link qualities to their neighbors. Similarly, CTC is a packet flooding (started from the sink node) in which every node broadcasts its minimum cost towards the sink. Therefore, NDLE and CTC utilize CSMA for channel access arbitration [9,2]. However, as these phases include a significant number of broadcast transmissions, packet collisions highly affect the accuracy of these phases [11,2]. During the NDLE phase, for those missing beacon packets caused by collision, nodes cannot distinguish between the packet losses caused by link unreliability and those caused by collision. Consequently, they cannot properly estimate their link cost to the node from which the packet has been originated. Furthermore, nodes cannot effectively discover their neighbors. Packet collisions during the CTC phase increase the number of cost update failures and raise the cost of the constructed tree. Particularly, missing a cost packet at a node not only affects that node's cost, but is also affects the path cost of the nodes which could have used this node as their ancestor. The inaccuracies introduced by the NDLE and CTC phases affect the efficiency of the higher-layer protocols. For example, inaccurate link estimations impact the efficiency of the paths used for data forwarding. Similarly, an inaccurate CTC phase results in data transmissions over non-optimal paths and causes higher energy consumption and lower delivery ratio.

Achieving reliable broadcast transmissions during the NDLE and CTC phases is a challenging problem due to the following reasons: First, as collision detection through mutual handshaking (e.g., exchanging CTS and ACK packets) is not possible with broadcast transmissions, no contention window adjustment can be applied. Second, utilizing multiple unicast transmissions instead of a broadcast transmission is not feasible, because it requires the nodes to be aware of their neighbors, which is not available at network initialization. In addition, unicast transmissions significantly increase the duration and energy consumption of the initial phases. Third, since collision detection is not supported, no retransmission can be expected. Unfortunately, although the literature proposes many MAC protocols for improving the reliability of unicast transmissions during the data gathering phase, no specific packet broadcasting protocol have been proposed for these initial phases [7]. Moreover, those MAC protocols that provide broadcast support during the data gathering phase, require the nodes to have their neighborhood information, which is not provided at network initialization [12].

Due to the challenges of achieving broadcast reliability, conservative approaches such as excessive backoff duration or fixed beaconing rate have been used to reduce collisions [2,13,9]. However, these approaches do not provide collision detection and they are not adaptive to network dynamics. Specifically, due to the influence of various parameters (such as network density, transmission power, path loss, beacon packet length and radio speed) on the number of collisions, it is hard to achieve a trade-off between accuracy and duration (or energy). This is even more challenging when no exact network density can be found for large-scale wireless sensor networks with random deployment. For example, the fixed beaconing rate approach either has been used in small-scale networks [6], or it has a very long inter-packet interval. Beside these drawbacks, since the fixed beaconing rate mechanism can only moderate hidden-node collisions, it should be accompanied with a sufficiently long contention window to avoid those packet collisions caused by identical backoff slot selection.

The contributions of this paper are therefore:

(i) Using collision detection through preamble detection, we propose adaptive contention window adjustment mechanisms for the broadcast traffic pattern of the NDLE phase. Various backoff schemes are proposed and their efficiency is investigated in terms of link estimation accuracy, number of detected neighbors and broadcast reliability. Performance evaluations show that the combination of linear and exponential backoff schemes provides fast and stable adaptation against collisions. Our results also show that while the proposed mechanisms considerably improve NDLE accuracy, they do not violate the energy efficiency requirement of sensor networks. The proposed mechanisms are independent of network size, and they can provide adaptive collision avoidance against neighborhood size and traffic intensity.

(ii) We improve NDLE accuracy through utilizing partially recovered packets. Our results confirm the benefits of employing this mechanism, which can be achieved without extra overhead. We also clarify the relationship between collision intensity and packet recovery.

(iii) We propose a mathematical model which can be used to compute the contention window size required for achieving a desired broadcasting success probability. This model can be specifically used for MAC configuration when network parameters are known.

(iv) For the CTC traffic pattern in which a broadcast flood is started from the sink and propagates throughout the network, we propose an algorithm, called *Geowindow*, that provides collision avoidance through contention window management. Whenever a node wants to broadcast a cost packet, the Geowindow algorithm assigns a specific sub-contention window (within the original contention window) from which that node can select its backoff duration. The size and priority of each sub-contention window is determined based on the contention intensity among the neighbors of the node from which the cost packet has been received. Through performance evaluation on the TinyOS's Collection Tree Protocol, our results show that the Geowindow algorithm achieves up to 74% improvement in tree construction accuracy and up to 70% improvement in data gathering efficiency, without considerable effect on CTC duration or energy consumption.

(v) We enable and investigate CTC improvement through collision detection and packet retransmission. Although this approach improves CTC accuracy, our results show that it significantly increases CTC duration, which makes it inappropriate for periodical execution during the data gathering phase.

The rest of this paper is organized as follow. Section 2 provides the background of this research. In Section 3 and 4 we present the proposed improvements for the NDLE phase and CTC phase, respectively. We conclude in Section 5.

2. Background

In the first part of this section we overview the existing NDLE and CTC mechanisms. In the second part, we review channel access and collision avoidance mechanisms for broadcast reliability. Finally, this section studies collision detection and partial packet recovery mechanisms.

2.1. Neighbor discovery, link estimation and collection tree

Current studies on low-power wireless communications have revealed significant packet reception variations around a transmitter [14,15]. In particular, three regions have been identified: connected, transitional, and disconnected. While the links in the connected region show more than 90% packet reception rate, the quality of those links in the transitional region varies between 90% and 10%. As these variations affect the performance of higher-layer protocols, various software-based, hardware-based and hybrid approaches have been proposed for link quality measurement. Meanwhile, link estimation mechanisms (especially software-based techniques) and neighbor discovery protocols rely on packet broadcasting during their operation [16–18]. In particular, the main approach is to broadcast a specific number of beacons by each node [2,11,19,13]. Consequently, each node can identify its neighbors and utilize the number of received beacons for estimating its incoming packet reception rate from its neighbors. In order to compute outgoing packet reception rates, each beacon packet should include the number of beacon packets the sender has received from its neighbors. Accordingly, every packet loss directly affects link quality estimation, because a node cannot discriminate between packet losses caused by collision and those caused by link unreliability. Among the link quality metrics, ETX (expected number of transmissions) [13] is the most widely used link quality measurement metric that reflects the expected number of MAC layer transmissions and retransmissions for a successful packet delivery over a link. ETX of a given link is defined as $1/pq$, where p and q are forward and backward packet reception rates, respectively. ETX has specifically been used in the Collection Tree Protocol (CTP) of TinyOS [20].

All of the existing CTC protocols utilize broadcast transmissions for tree construction and maintenance. For example, MintRoute [21], MultihopLQI [22] and Collection Tree Protocol (CTP) [9,20] (all belonging to the core of the TinyOS's data collection layer) use aggressive broadcasting from the sink node towards the tree leaves. This flooding traffic pattern establishes routing paths from each node towards the sink node. In addition, it employs CSMA for channel access. Consequently, CTC accuracy is affected by the packet losses caused by collision. For example, when a sample node i misses a cost packet that could have reduced its cost towards the sink node, it affects all the nodes that their cost relies or could have relied on the cost of node i. Therefore, losing a cost packet may result in higher difference between the cost of the produced tree and the optimal tree. The other drawback of the mentioned CTC protocols is their hastiness for cost broadcasting. As stated earlier, when a node updates its minimum cost value, it tries to immediately broadcast its newly computed cost. However, this node will probably receive a new cost packet in the near future, which is due to the MAC layer contention mechanism. Therefore, the new packet would be lost (since the radio is in transmission mode) or this node has to resend a new cost after receiving a cost packet.

2.2. Medium access control

Carrier Sense Multiple Access (CSMA) is a principal access arbitration mechanism which is used by the TinyOS's default MAC protocol [23,24] and IEEE 802.11 standard [25]. While variants of this protocol exist, the main idea is as follows: Before each packet transmission, a node should select a random *backoff time* from a *contention window* composed of *W time slots*. Backoff timer is decremented at each slot boundary as long as the channel is free. Transmission is commenced when the backoff counter reaches 0.

The two main collision avoidance mechanisms proposed for CSMA are *handshaking* and *exponential backoff*. The handshaking mechanism requires the sender and receiver to exchange RTS, CTS and ACK packets for channel reservation. Using the exponential backoff mechanism, a node should double its contention window size and resend the packet whenever it does not receive the expected ACK or CTS packet. Unfortunately, as these mechanisms rely on unicast transmissions, none of them can be used during the initialization phases. Nevertheless, even if the transmissions were unicast and collision detection was possible, contention window adjustment would not be accurate because the difference between the packet loss caused by link unreliability and that caused by collision cannot be detected.

Most of the MAC protocols proposed for wireless sensor networks rely on CSMA for their setup and during their operation (e.g., S-MAC [26], Z-MAC [6], TRAMA [27] and SCP-MAC [2]). For example, with S-MAC, two nodes cannot communicate if they have not identified each other during the neighbor discovery phase. Similarly, TRAMA utilizes CSMA to perform neighbor discovery, topology update and traffic information broadcasts. Therefore, while these protocols can improve transmission efficiency during the data gathering phase, their performance depend on initialization accuracy.

The most straightforward way for improving broadcast reliability is to broadcast multiple copies of each packet. Although this approach has been used in vehicular networks (e.g., [28,29]), it cannot be used in the NDLE phase of sensor networks, because this phase requires a predetermined number of transmissions. In addition, as this approach multiplies duration and energy consumption, it is not useful for the CTC phase.

With respect to the aforementioned drawbacks, the following approach has been adopted by many protocols (e.g., [13,2,11]) to achieve collision avoidance: NDLE duration is divided into N beaconing intervals, during which each node can send one or more beacon packets. Since this approach utilizes a constant interval between transmissions, it is simply referred as *Constant Interval* (CI) in this paper. Using the CI approach, collision avoidance capability depends on beacon size, beaconing interval, relative positioning of the beaconing intervals and node density. Although this approach has been widely employed [12,2], its configuration requires some information regarding the underlying network. Therefore, it cannot be considered as an adaptive approach, especially when the network size is large or node density is not uniform. Using transmission staggering for collision avoidance has also been used in D-MAC [30]. However, D-MAC has been designed for unicast transmissions, and cannot be used in the initialization phases.

Balon et al. [31] utilized packet sequence numbers to measure reception efficiency and collision intensity within a specific time interval. The measured reception rate is then used to adjust the contention window duration. This approach has the following drawbacks. Firstly, using sequence numbers for collision detection requires continuous packet reception from a single source for a minimum duration. However, the initialization phases do not satisfy this requirement. The second and more important drawback is that this approach cannot discriminate between a packet loss caused by collision and a packet loss caused by link unreliability. Jamieson et al. [32] proposed a backoff mechanism in which nodes use a non-uniform function for selecting their backoff duration. Although this mechanism reduces the probability of one-hop collisions, it has no effect on hidden-node collisions. Boano et al. [33] have evaluated the effects of various backoff mechanisms in the presence of interference. Although their evaluations confirmed the importance of backoff for collision avoidance, they do not propose any adaptive backoff mechanism based on collision intensity. Other approaches on broadcast reliability have mainly focused on the reliability of data broadcasting during the network operation [34–36]. In other words, these approaches try to make sure a broadcast packet is received by all the network nodes at least once. However, this is different from the NDLE and CTC phases in which it is required to receive the broadcast packet of each node at its neighboring node.

Due to network dynamics and inefficiency of the conservative mechanisms, some approaches have been proposed for updating neighborhood discovery, link estimation and tree structure during the data gathering phase [37,38]. However, the importance of the initial phases can be justified through two reasons: First, since low-power MAC protocols highly rely on the sleep state to reduce power consumption, nodes cannot monitor the channel continuously [39]. Second, those algorithms that rely on the initially discovered network information are costly to be updated. For example, while TDMA-based MAC protocols run a scheduling algorithm to assign collision-free slots to the nodes [12], the literature shows the high overhead of schedule update during network operation [40]. It is worth mentioning that we do not disregard the importance of updating link estimations and routing paths during the data gathering phase; rather, in this paper, we aim to propose mechanisms that can be used to improve the performance of NDLE and CTC during network initialization *and* data gathering.

2.3. Collision detection and partial packet reception

The studies of [41–43] showed that the capture effect exists in low-power transceivers, and allows the radio to be synchronized with a new stronger packet while another packet is being received. In addition, Whitehouse et al. [41] showed that the capture effect provides partial packet reception and can be used for collision detection. Yun et al. [44] proposed a collision detection mechanism that relies on exchanging transmission times, and it is used for improving the channel access parameters of CSMA. Although this collision detection approach is effective, it requires mutual packet exchange and it can only be used for unicast transmissions. Hauer et al. [45] proposed continuous RSSI sampling for collision detection through finding RSSI elevations. This mechanism is particularly useful in detecting those collisions that are shorter than packet duration, or those collisions caused by external interference. In contrast, as this paper aims to handle inter-node interference, and not external interference, collision detection is achieved through the capture effect. Jamieson and Balakrishnan [46] have highlighted the importance of packet collisions, and they developed a customized radio which allows the higher layers to receive confidence information regarding the reliability of the received bits. To the best of our knowledge, this feature is not provided by available radios and cannot be used for real-world applications.

3. Improving neighbor discovery and link estimation

The beacon packet size used during the NDLE phase should be equal to the data packet size, because link qualities should be estimated based on the data packet size utilized during the data gathering phase. If the neighborhood information is not enough for filling the payload field, additional bytes are filled with a specific byte value.

This section first describes collision detection and partial packet reception (which are widely used in this paper), then the proposed broadcast reliability mechanisms are presented.

3.1. Exploiting the capture effect for collision detection and partial packet recovery

All the transmitted data packets in a wireless network should begin with a specific bit pattern called *preamble* or *physical-layer header*. Preamble allows the receiver to obtain certain information about the incoming data packet before being enabled to receive its data bits. This operation is referred to as *synchronization*. Moreover, in order to determine the start of the data bytes, specific bytes are added after the preamble bytes. These bytes are called *sync word* or *start of frame delimiter* (*SFD*).

Fig. 1 shows the collision detection condition with the default TinyOS's packet format. We assume *Packet 1* can be successfully synchronized when *Packet 2* does not exist. We also assume *Packet 2* is stronger than *Packet 1* and it can be successfully synchronized in the presence of *Packet 1*. In Fig. 1a, *Packet 2* arrives during the preamble bytes of *Packet 1*. Therefore, since the radio does not realize these preamble bytes belong to different packets, collision detection cannot be achieved. In Fig. 1b, *Packet 2* arrives after the preamble bytes of *Packet 1*. More specifically, the radio receives the preamble bytes of *Packet 2* after receiving the sync bytes of *Packet 1*. Therefore, a collision can be detected. Assuming *Packet 2* arrives after the source address (SA) or destination address (DA) field of *Packet 1*, these fields can also be recovered and used by the higher-layer protocols.

3.2. Contention window adjustment schemes

In this section we utilize collection detection through the capture effect to adjust contention window size for achieving collision avoidance. Specifically, we try to provide access mechanisms that can adaptively change their backoff duration based on collision intensity. This is in contrast with the CI mechanism and CSMA with fixed contention window size (cf. Section 2). The proposed contention window adjustment schemes can also be used in other broadcast-based traffic scenarios.

We use the following mechanism to utilize collision detection information for backoff adjustment. Whenever the NDLE module receives a collision detection notification from the MAC layer, it sets a flag, called *collision flag*, in its next beacon packet to inform its neighbors about the detected collision. When a node receives a collision-indicative beacon packet, it should increase its contention window duration and use it for its subsequent beacon transmissions. We propose three contention window adjustment mechanisms as follows:

- **Linear (LI).** Assume that node *i* receives a collision-indicative beacon packet from node *j*. Node *i* increases its current contention window size through adding the initial contention window size to the current contention window size. Also, node *i* refers to its one-hop neighborhood table and increases the number of collision-indicative beacon packets received from node *j*. As long as a collision-indicative beacon packet is received, node *i* can increase its contention windows size unless a maximum value is reached. Whenever node *i* receives a noncollision-indicative beacon packet from node *j*, it refers to its one-hop table and evaluates the number of collision-indicative packets that has already been received from this neighbor. No action is required if the evaluated value is equal to 0. Otherwise, this value is decremented by 1 and the initial contention window size is subtracted from the current contention window size. Using this mechanism, contention window size is only reduced for those neighbors that have previously caused contention window increment.
- **Exponential (EXP).** This mechanism is similar to LI, however, it uses exponential increments and decrements. Each contention window increment doubles the current contention window size, subject to a maximum value. Similarly, each contention window reduction halves the current contention window size.
- **Linear and Exponential (LIN–EXP).** In this mechanism, contention window increments and decrements are performed exponentially and linearly, respectively.

3.3. Utilizing partially recovered packets for improving accuracy

In Section 3.1 we showed that a packet arrival can cause collision detection if it arrives after the sync bytes of the first packet. Beside collision detection, specific bytes of the first packet (i.e., the lost packet) can be recovered if the collision happens after those bytes. When the NDLE module receives a partially recovered packet, if the collision has happened after the source address of the first packet, but before the payload bytes, the NDLE module can only detect the address of the beacon sender. If this is the first time a beacon packet

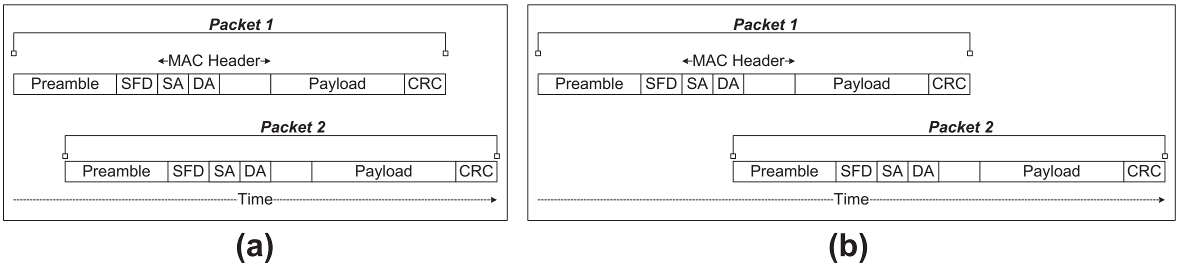

Fig. 1. Collision detection through the capture effect. (a) This collision cannot be detected because it happens before receiving the SFD bytes. (b) This collision can be detected because it happens after receiving the SFD bytes.

is received from that node, this partial packet recovery increases the number of discovered neighbors. Besides, it allows the receiver to have a better estimation about the number of received packets from that node. If the collision has happened during or after the payload bytes, in addition to the source address the partially recovered payload bytes can also be utilized for link estimation enhancement. The problem with this approach is to determine the correctness of the received bytes. A possible solution is to include a CRC byte at the end of MAC header, and few CRC bytes between payload bytes.

3.4. Mathematical modeling of collision probability during packet broadcasting

In this section we present an analytical approach through which we can determine the required contention window size for achieving a desired success probability for broadcast transmissions. When a node broadcasts a beacon packet, packet reception at a one-hop neighbor may be corrupted by the one-hop and two-hop neighbors. Therefore, the probability of successful reception at the one-hop nodes depends on the parameters such as the number of one-hop neighbors, number of two-hop neighbors, beacon packet length and contention window size. Among these parameters, the number of one-hop and two-hop neighbors is unknown. Hence, first, we analytically estimate the number of one-hop and two-hop neighbors using the channel model equations of [47,14]. These equations can provide the distance at which a specific packet reception rate can be achieved. For Mica2 motes (CC1000 transceiver [48]), the packet reception rate at a given distance (d) is given by

$$PRR = \left(1 - 0.5 \times e^{-\frac{SNR}{2} \times \frac{B}{R}}\right)^{b}, \tag{1}$$

where SNR is the signal-to-noise ratio (not in dB), B is the noise bandwidth, R is the radio bit rate and b is the number of bits in the transmitted data frame. SNR (not in dB) at a given distance d is

$$SNR = 10^{\frac{P_t - PL(d_0) - 10\eta \log(d) - P_n}{10}}, \tag{2}$$

where P_t is the transmission power, $PL(d_0)$ is the path loss at reference distance d_0, η is the path loss exponent and P_n is the noise power. Using these two equations we can compute the distance at which a specific packet reception rate (PRR) is achieved,

$$\Upsilon(PRR) = 10^{\frac{P_t - PL(d_0) - P_n - 10\log\left(\frac{-2R}{B} \times \ln\left(2\left(1 - PRR^{\frac{1}{b}}\right)\right)\right)}{10\eta}}. \tag{3}$$

We assume the minimum link quality to a one-hop neighbor is 10%. Therefore, one-hop neighbors are located within distance $\Upsilon(0.1)$. Having $\Upsilon(0.1)$, we propose the *Neighborhood Count* algorithm (Algorithm 1) through which we can estimate the number of one-hop and two-hop neighbors. Assume a node in a square network with width X_{area}. With respect to the relationship between $X_{area}/2$ and $\Upsilon(0.1)$, the Neighbor Count algorithm considers three cases for neighborhood calculations.

Algorithm 1. Neighborhood count

```
Input:
Total number of nodes in the network: $N_{node}$
The width of the area: $X_{area}$
The radius of the area in which one-hop neighbors reside: $\Upsilon(0.1)$
Output:
The average number of one-hop ($N_{one-hop}$) and two-hop ($N_{two-hop}$) neighbors per node
if ($\Upsilon(0.1) > X_{area}/2$) then
    $N_{one-hop} = N_{node}$
    $N_{two-hop} = 0$
else if ($\Upsilon(0.1) < X_{area}/2$) and ($2 \cdot \Upsilon(0.1) < X_{area}/2$)
    $N_{one-hop} = \pi \cdot \Upsilon^2(0.1) \times (N_{node}/X^2_{area})$
    $N_{two-hop} = (\pi \cdot (2 \cdot \Upsilon(0.1))^2 - \pi \cdot \Upsilon^2(0.1)) \times (N_{node}/X^2_{area})$
else if($\Upsilon(0.1) < X_{area}/2$) and ($2 \cdot \Upsilon(0.1) > X_{area}/2$) then
    $N_{one-hop} = \pi \cdot \Upsilon^2(0.1) \times (N_{node}/X^2_{area})$
    $N_{two-hop} = N_{node} - (\pi \cdot \Upsilon^2(0.1) \times (N_{node}/X^2_{area}))$
end if
```

Having the neighborhood information we can find the probability of a successful broadcast when CSMA is used as the MAC protocol. In the proposed model we assume that the contention window is slotted, and backoff counter is decremented at each slot boundary irrespective to the channel status. Since one-hop neighbors can sense each others' transmission, a broadcast transmission by node i collides with a one-hop transmission if at least one of the one-hop neighbors selects the backoff slot selected by node i. Therefore, the probability of one-hop collision is

$$\left(1 - \frac{1}{W}\right)^{N_{one-hop}}, \tag{4}$$

where W is the contention window duration in terms of slots. Node i's transmission collides with a two-hop transmission if at least one of the two-hop neighbors selects its contention slot between $[\delta_i - f_s, \delta_i + f_s]$, where δ_i is the selected slot by node i and f_s is the beacon packet transmission duration in terms of the number of backoff slots. Therefore, since the two-hop neighbors should not select $2 \times f_s$ slots of the contention window (W), the probability of two-hop collision is

$$\left(1 - \frac{2f_s}{W}\right)^{N_{two-hop}}. \tag{5}$$

However, node i may select its backoff slot within the first f_s slots of the contention window. Consequently, two-hop neighbors cannot select their slot from $[1,\delta_i]$ and $[\delta_i, \delta_i + f_s]$. Assuming that node i's backoff slot is selected within $[1,f_s]$, the average selected value is $(f_s+1)/2$. As a result, the probability of two-hop collision equals

$$\left(1 - \frac{3f_s + 1}{2W}\right)^{N_{two-hop}}. \tag{6}$$

Therefore, we can compute the probability of successful broadcasting as

$$Pr(success) = \frac{f_s}{W} \times \left(1-\frac{1}{W}\right)^{N_{one-hop}} \times \left(1-\frac{3f_s+1}{2W}\right)^{N_{two-hop}} + \frac{W-f_s}{W} \times \left(1-\frac{1}{W}\right)^{N_{one-hop}} \times \left(1-\frac{2f_s}{W}\right)^{N_{two-hop}}. \quad (7)$$

The presented mathematical model can be used for analyzing the effects of various network parameters on collision probability. In addition, it allows finding the contention window size corresponding to a specific success probability.

Using this mathematical model, Fig. 2 shows how the number of neighbors and contention window size affect the probability of collision-free broadcasting. As this figure shows, number of two-hop neighbors has higher effect than the number of one-hop neighbors. This justifies using the capture effect for collision detection because hidden-node collisions are potentially detectable (depending on the arrival times).

3.5. Simulation settings and definitions

We have implemented the protocols and algorithms of this paper using the OMNeT++ simulation framework [49]. Moreover, we have developed accurate wireless channel and physical layer models which can precisely simulate the characteristics of low-power wireless communications. To this aim: Firstly, among the interference models, we have used the signal-to-interference-plus-noise ratio (SINR) model due to its highest accuracy [50]. Secondly, since low-power transceivers support the capture effect, we have considered packet reception through the stronger-first and stronger-last captures [41]. Thirdly, we have considered the deviations of transmission power and noise floor caused by hardware heterogeneity [14]. Table 1 presents the default simulation parameters of this paper. The radio parameters have been chosen based on the characteristics of Mica2 motes with CC1000 radio. The environmental parameters have been chosen based on the reports of [14]. The MAC and packet format parameters are from the TinyOS implementation [23,51].

In addition to the send/receive interfaces between layers, cross-layer interfaces have been implemented as follows: (i) MAC layer informs the NDLE and CTC modules regarding collision detection, and delivers partially received packets to these modules; (ii) NDLE and CTC modules can directly access the CSMA module and adjust its parameters.

Table 2 shows the networks used for performance evaluations. In order to achieve a uniform distribution, each network is divided into 16 squares, and 25 nodes are selected uniformly within each square. Also, we ensured that the minimum inter-node distance is 1 m. Neighborhood size is computed as the average number of neighbors per node for which their corresponding average link quality (measured based on Euclidean distance) is higher than 10%. Each result value is the median of 10 simulation runs. Error bars represent upper and lower quartiles.

In the next section we investigate the effects of the following MAC mechanisms on NDLE performance:

- **CSMA**: The CSMA MAC protocol that employs a slotted contention window.
- **CSMA [*X*]**: An improved version of the CSMA mechanism in which *X* indicates the contention window adjustment scheme.
- **CSMA [*X*][PR]**: An improved version of the CSMA [X] mechanism in which PR indicates the use of partially recovered packets for accuracy improvement.
- **CSMA [Analytical CW]**: The CSMA protocol whose contention window duration is obtained from the mathematical model given in Section 3.4.

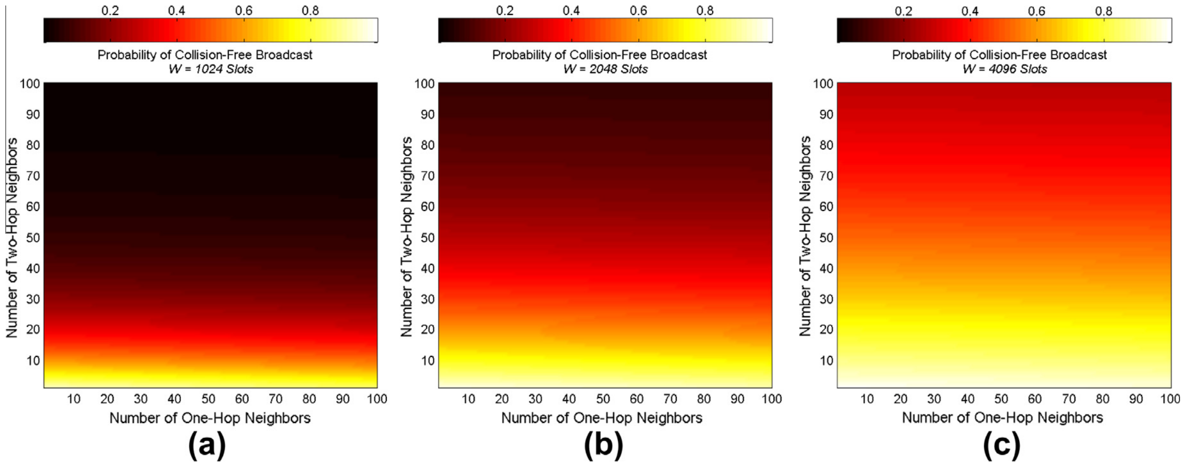

Fig. 2. Mathematical analysis of collision-free broadcasting. Each sub-figure shows how changing the number of one-hop and two-hop neighbors affect the probability of collision-free broadcasting with a given contention window size. Since the carrier sensing mechanism of CSMA can significantly avoid one-hop collisions, this figure shows that broadcasting success probability mainly depends on the number of two-hop neighbors. Therefore, as most of the collisions are due to the hidden-node problem, the capture effect can be used for collision detection and packet recovery.

Table 1
Simulation parameters.

Parameter	Value
Radio	
Average noise power (dBm)	−106
Noise figure (dB)	13
Switch to TX/RX (us)	250
Radio sampling (us)	350
Evaluate radio sample (us)	100
Noise bandwidth (B) (Hz)	30,000
Modulation	NC-FSK
Encoding	Manchester
Radio speed after encoding (R) (bits per second)	19,200
Transmission power (dBm)	0
Reference distance (d_0) (m)	1
PL (d_0) (dB)	55
Standard deviation of transmission power heterogeneity (dB)	1.2
Standard deviation of noise floor heterogeneity (dB)	0.9
Correlation of transmission power and noise floor	−0.7
TX current consumption (mA)	16.5
RX/Idle current consumption (mA)	9.6
Environment	
Ambient temperature (°C)	27
Path loss exponent (η) (outdoor)	4.7
Multipath channel variations (σ_{ch}) (outdoor)	3.2
MAC	
Contention window (slot)	32
Carrier sensing threshold (dBm)	−100
Packet format	
Physical header (byte)	10
MAC header (byte)	5
Payload NDLE/CTC (byte)	29/3
CRC (byte)	2
Battery	
Capacity (mAh)	2500
Voltage (V)	3

Table 2
Networks used in the evaluations of this paper.

Number of nodes	Area (m^2)	Average number of neighbors per node
400	85 × 85	10 (Low density)
400	60 × 60	20 (Low density)
400	45 × 45	30 (Medium density)
400	40 × 40	40 (High density)
400	37 × 37	50 (High density)

- **CI**: The fixed beaconing rate approach which defines the interval between beacon transmissions (cf. Section 2). We have employed 1 beacon per second as used in [13].
- **CI [*X*]**: An improved version of the CI mechanism in which *X* indicates the contention window adjustment scheme.
- **CI [*X*][PR]**: An improved version of the CI [X] mechanism, in which PR indicates the use of partially recovered packets for accuracy improvement.

3.6. Performance evaluations and discussions

Link Estimation Accuracy. Fig. 3 shows the performance of the MAC mechanisms in terms of link quality estimation accuracy. The accuracy of link quality estimation is measured through computing the RMSE of the average link quality (which is obtained through considering the link length) and the link quality estimation obtained from the NDLE protocol. Assume that set $\mathbf{L} = \{l_1, l_2, l_3, \ldots, l_N\}$ indicates those links for which their average link quality is higher than 10%. Also, set $\mathbf{E} = \{e_1, e_2, e_3, \ldots, e_N\}$ represents the link estimations obtained during the NDLE phase. Link estimation accuracy is computed as follows,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(l_i - e_i)^2}{N}}. \tag{8}$$

As Fig. 3 shows, CSMA achieves the lowest link estimation accuracy because using a 32-slot contention window is not suitable even in the low network densities. As stated earlier, link estimation accuracy depends on the number of packets a node receives from its neighbors. We investigate this value in Fig. 4. Based on this figure, the CSMA mechanism achieves a fixed beacon reception percentage irrespective to the number of beacon transmissions and neighborhood size. When the neighborhood size increases, the number of those close neighbors from which a packet can be successfully received increases for each node. However, this also intensifies the number of neighbors in the transitional region. Consequently, as long as the contention window size is fixed, the number of receptions remains almost unchanged. This behavior can also be observed with CSMA [Analytical CW] (because it has a fixed contention window size) and CSMA [LIN–EXP] (because its contention window size variations are smaller than that of other mechanisms). However, compare with CSMA, since CSMA [Analytical CW] and CSMA [LIN–EXP] present higher reception percentage, their link estimations are more accurate and demonstrate accuracy improvement as the number of beacon transmissions increases.

Using the exponential contention window adjustment scheme, nodes can quickly change their contention window size based on contention level. The benefit of this fast contention window adjustment mechanism is to achieve collision avoidance even when the number of received collision-indicative beacons is low. For example, as Fig. 3a and c show, CSMA [EXP] and CI [EXP] present higher accuracy over CSMA [LIN] and CI [LIN], respectively. However, the drawback of this scheme is its lower collision avoidance capability (compare with the linear scheme) when the number of beacon transmissions increases. For example, at neighborhood size 30, while in Fig. 3a CSMA [EXP] shows 14% improvement over CSMA [LIN], the situation is reversed in Fig. 3b and CSMA [LIN] shows about 21% improvement over CSMA [EXP]. Using the linear scheme, nodes perform slow increasing and decreasing variations on their contention window size. For example, assuming that the maximum contention window size is 1024 slots, when the contention window size reaches its maximum value, receiving a noncollision-indicative packet reduces the contention window size by 32 slots. In contrast, exponential scheme halves the contention window size and uses a 512-slot contention window for its next beacon transmission. In this case, since the contention window size undergoes large variations, nodes cannot choose the

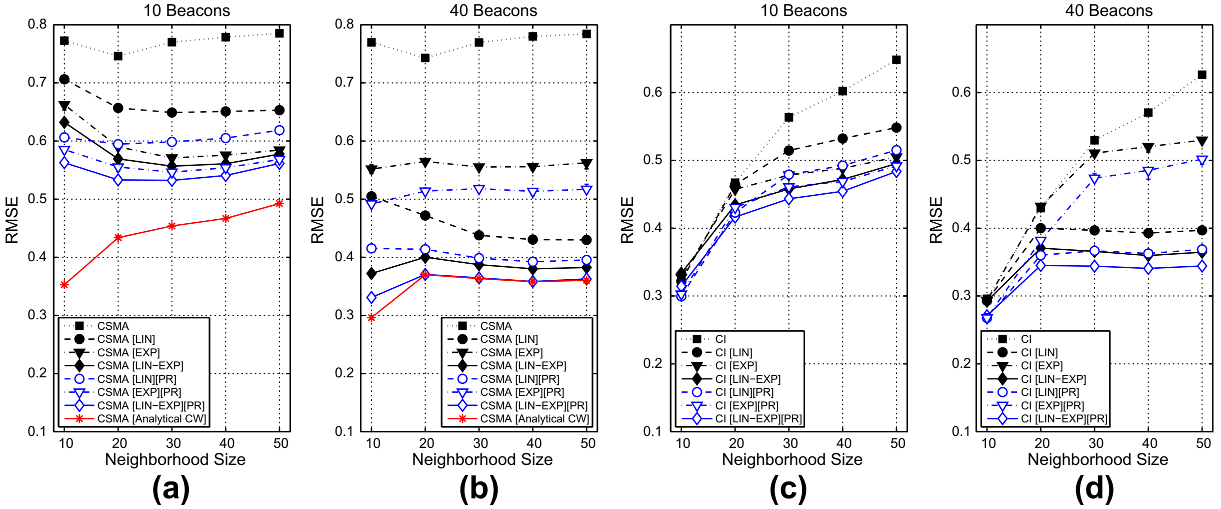

Fig. 3. Influence of the MAC mechanisms and number of beacons on link estimation accuracy. The [LIN-EXP]-based schemes provide highest performance as they benefit from the fast contention window adjustment of the exponential scheme and lower variations of the linear scheme. For those mechanisms that utilize a contention window adjustment scheme, increasing the number of beacons allows for better contention window adjustment and higher accuracy. Increasing the number of beacons also contributes to accuracy improvement as it increases the granularity of link estimation.

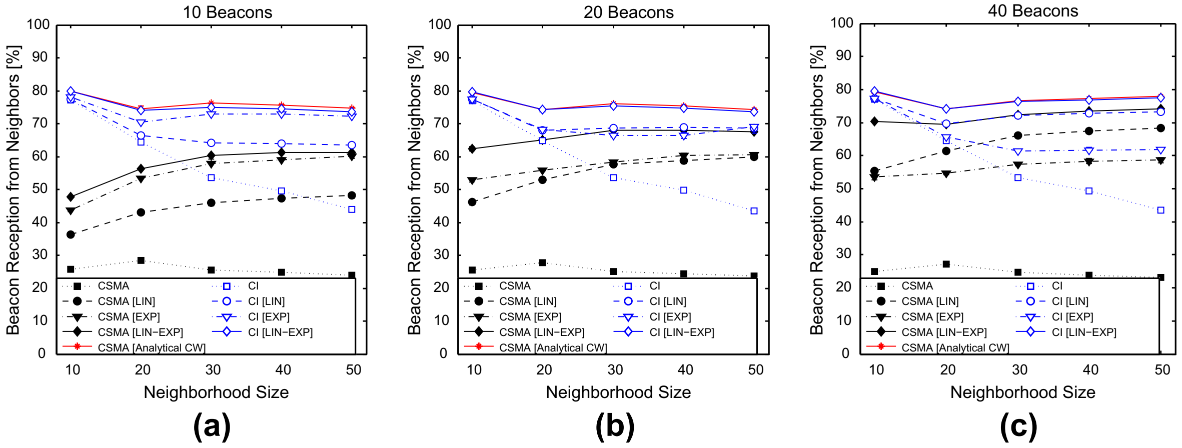

Fig. 4. The average packet reception percentage of the nodes from those neighbors that their average link quality is higher than 10%. At low network densities the CI-based mechanisms present similar performance because the inter-beacon interval can effectively avoid collisions. However, as the neighborhood size increases, the performance of these mechanisms depends on the utilized backoff scheme. This figure also shows that the performance of the linear backoff scheme depends on network density and number of exchanged beacons. Also, the potential benefits of the mixed linear and exponential schemes can be observed.

values of range [513, 1024]; hence, the maximum available contention window size is not efficiently utilized for collision avoidance. In contrast with the linear and exponential schemes, the [LIN–EXP]-based mechanisms provide fast increase and slow decrease in contention window size. The faster and more stable changes can provide higher accuracy than the other backoff schemes.

For those mechanisms that employ a fixed beacon rate (i.e., CI-based mechanisms), Fig. 3 shows a noticeable increasing trend versus neighborhood size. When the neighborhood size is low, most of the collisions are caused due to the hidden-node problem because the number of transmissions that can be sensed at each node is small. Therefore, the long inter-beacon delay of the CI-based mechanisms can significantly reduce these collisions. Moreover, as these mechanisms also employ CSMA before each beacon transmission, they can effectively avoid those collisions caused by identical backoff slot selection. Therefore, Fig. 4 shows that all the CI-based mechanisms demonstrate similar beacon reception percentage for low neighborhood densities. Increasing the neighborhood density extends the number of nodes that can sense each other's transmissions. This reduces the number of hidden-node collisions and intensifies those collisions caused by identical backoff slot selection. Consequently, the effectiveness of using a fixed beacon transmission reduces and collision avoidance efficiency mainly depends on contention window duration. For this reason, we can observe the improvements achieved through augmenting the CI-based mechanisms using the proposed backoff schemes.

For the CSMA [Analytical CW] mechanism we have considered 50% reception probability to obtain the desired contention window size. Our results show that this mechanism can effectively improve link estimation accuracy, compare with other mechanisms. However, whereas the desired reception probability is set to 50%, the beacon reception percentage of this mechanism is more than 70%, as Fig. 4 indicates. Because the mathematical model uses the maximum neighborhood density to compute broadcasting success probability, the obtained contention window size is especially valid for central nodes. Therefore, this value results in lower collision probability for those nodes that have lower number of neighbors (e.g., nodes near the network margin).

As Fig. 3 shows, increasing the number of beacon transmissions improves link estimation accuracy. While the backoff adjustment schemes benefit from the number of beacon transmissions for improving their contention window adjustment, increasing the number of beacons also improves link estimation granularity. For example, when the number of beacons is 10, if a node looses one of its neighbor's beacon packets, the estimated link quality between these nodes reflects at least 10% inaccuracy if the actual link quality is higher than 10%. However, when the number of beacons is 20, the introduced inaccuracy is 5%. One might ask how link accuracy improvement is achieved with higher number of beacons when the beacon reception percentage is almost fixed for a given mechanism (e.g., CSMA [Analytical CW] in Fig. 4). In this case, we observed that increasing the number of beacons specifically improves link estimation accuracy for those links in the connected region. This also explains the benefits of employing higher number of beacons at high neighborhood densities.

Fig. 3 also shows the effects of using partially recovered packets for improving link estimation accuracy. Notice that the improvements achieved through utilizing partially recovered packets depend on the number of those collisions in which some data bytes of the lost packet can be recovered. Although this situation depends on the number of collisions, a very high collision rate also reduces the number of packet recoveries. For example, as Fig. 5a shows, the number of experienced packet collisions with CSMA [LIN] is constantly higher than that of CI [LIN]. However, Fig. 5c indicates that for the neighborhood sizes higher than 30 the number of MAC header recoveries with the CI [LIN] mechanism is higher. This behavior is characterized by the lower inter-packet arrival times with CSMA [LIN] which are caused due to the higher number of collisions. In this case, no collision can be detected if the radio is not synchronized with the incoming packet when a new packet arrives and causes collision. Furthermore, when a collision occurs before completely receiving the MAC header of the first packet, no information of the first packet can be recovered and used by the NDLE protocol. Consequently, increasing the number of network collisions (through larger neighborhood size or more beacon transmissions) does not necessarily increase the benefits of partial packet recovery. In particular, we can argue that increasing the number of collisions per second causes higher number of collision detections and recoveries; however, when this ratio goes beyond a specific threshold, the number of collision detections and recoveries starts to fall. Beside the aforementioned issues, the followings also affect the benefits of partial packet recovery: (i) For a given packet size, reducing the ratio of preamble to packet size increases the efficiency of packet recovery; (ii) Increasing the contention window duration reduces the number of collisions caused by identical slot selection, therefore, improves the number of recoverable collisions.

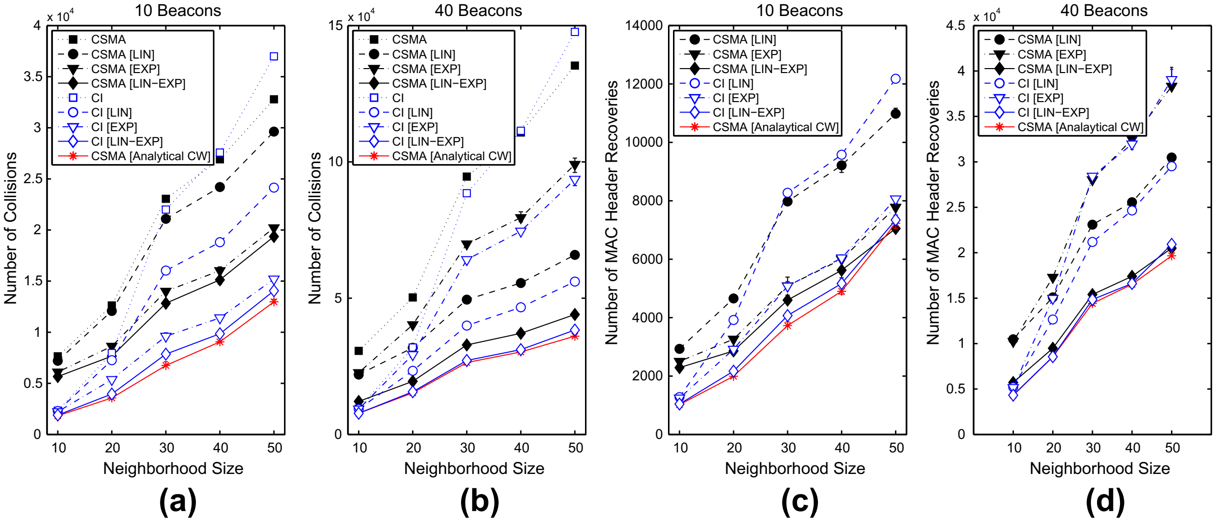

Fig. 5. (a) and (b) The number of collisions that caused packet corruption. (c) and (d) The number of MAC headers that have been recovered during the collisions. Each collision corresponds to a packet arrival that causes the packet currently being received by the radio to be lost. Delivering partially recovered packets to the NDLE module allows for higher link estimation accuracy and more number of detected neighbors. Comparing the number of collisions and number of MAC header recoveries shows that increasing the number of collisions does not necessarily indicate higher number of MAC header recoveries. Specifically, for a given network configuration, increasing the number of collisions beyond a threshold value reduces the number of recoveries.

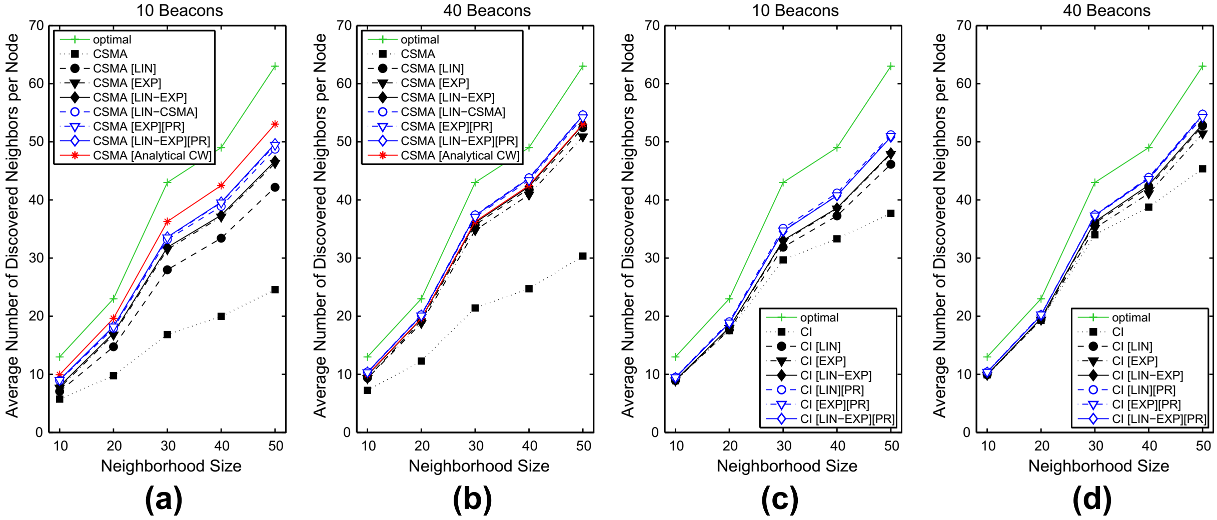

Fig. 6. Influence of the MAC mechanisms and number of beacons on neighbor discovery. Increasing the number of beacon transmissions improves the number of discovered neighbors, because: (i) the probability of beacon receptions from the neighbors increases, and (ii) link variations can better be reflected.

An interesting observation is the relationship between the number of collisions and packet reception performance. For example, although Fig. 4 shows higher beacon reception percentage for the CI mechanism compare with the CSMA mechanism, Fig. 5 shows higher number of collisions for the CI mechanism at high neighborhood densities. From the signal reception point of view, due to the intense inter-node interference at high neighborhood densities, the number of packets that provide enough SINR to be received by the radio is lower with the CSMA mechanism. Therefore, as we only count those collisions that cause packet corruption, no collision is counted when a new signal arrives at a node while the radio is not receiving any packet.

Neighbor Discovery. Fig. 6 shows the number of discovered neighbors corresponding to each mechanism. Although these results are demonstrated against neighborhood density (x axis), it cannot accurately indicate the number of discoverable neighbors per node. In particular, our measurement of neighborhood size only considers those links that their average link quality is higher than 10%. However, this is not an accurate estimation because low-power wireless links exhibit significant variations in their quality. Consequently, we utilized the *optimum* protocol that employs 100 beacons and detects almost all of the potential neighbors of each node. This protocol behaves as a baseline for measuring the efficiency of other mechanisms. The main observation of our study is that increasing the number of beacon packets improves the number of discovered neighbors. For example, when the neighborhood density is 50, the average number of discovered neighbors with the CSMA mechanism for 10 and 40 beacon transmissions equals 25 and 30, respectively. This can be described as follows: Firstly, it is intuitive that more beacon transmissions increase the chance of receiving at least one beacon from all the potential neighbors. Second, link variations can better be reflected through increasing the number of beacon transmissions because it improves the chance of beacon reception at longer distances. This figure also confirms the improvements achieved through the backoff adjustment schemes. For example, when the neighborhood size is 50, with 10 and 40 beacons the CI [LIN] mechanism detects about 9 and 8 neighbors more than the CI mechanism, respectively. The other observation is the higher number of detected neighbors when the partial packet recovery mechanism is used. In contrast to the link estimation case for which payload bytes of partially recovered packets are required for estimating the quality of outgoing links, neighbor discovery only requires receiving the source address of the beacon sender to identify a neighbor.

Duration and Energy Efficiency. Fig. 7 shows NDLE duration corresponding to various mechanisms. Note that this figure does not show the NDLE duration of those mechanisms with partial packet recovery technique, because this technique does not affect duration. It is evident that improving NDLE accuracy through collision avoidance comes at the cost of higher duration. In order to clarify how this translates into energy consumption, Fig. 8 shows the average battery consumption per node. Comparing this figure with previous results reveals that while the proposed mechanisms provide considerable NDLE improvement, the highest energy consumption is less than 0.035% of a node's battery capacity. Since the proposed mechanisms provide collision avoidance based on local collision intensity, the reported maximum value also holds for larger networks as long as the neighborhood size is 50 and the number of beacons is 40. Therefore, in addition to providing adaptive collision avoidance, these mechanisms also respect the importance of energy efficiency in wireless sensor networks. An important observation with Figs. 7 and 8 is that the energy consumption trend is very similar to that of duration. This is due to the use of CSMA, which does not employ radio duty cycling.

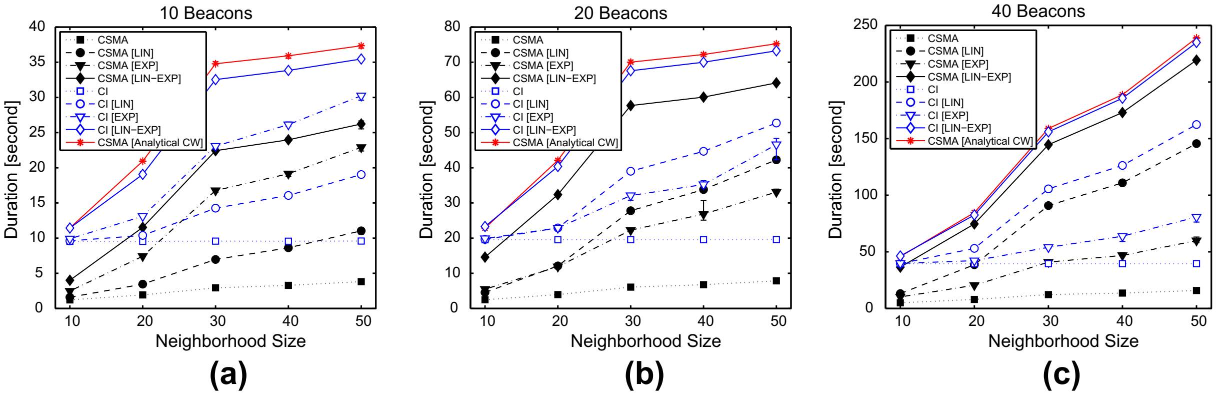

Fig. 7. Influence of the MAC mechanisms and number of beacons on NDLE duration.

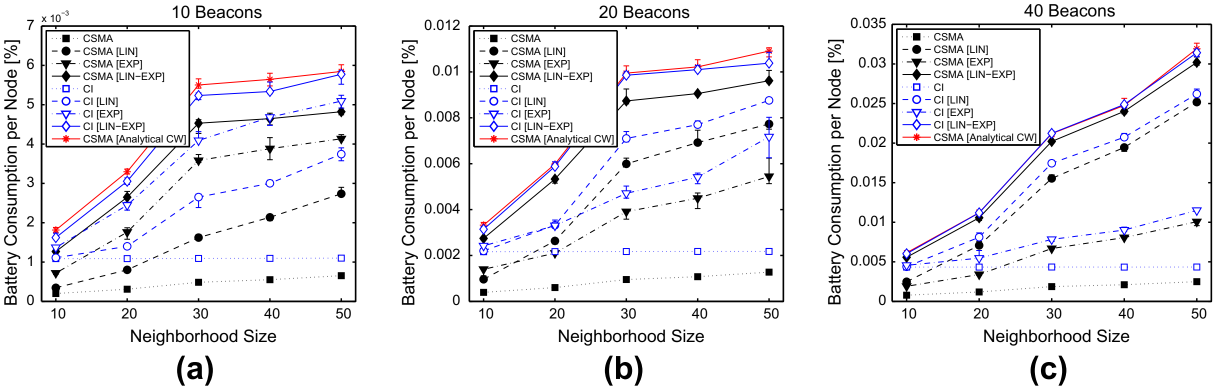

Fig. 8. Influence of the MAC mechanisms and number of beacons on the average percentage of battery consumption per node.

4. Improving collection tree construction

After the NDLE phase, sink node starts the CTC phase through broadcasting a cost packet wherein the cost field (ETX in this paper) equals zero. Initially, all the nodes set their cost to infinity, indicating that they have no path towards the sink. Whenever a node receives a cost packet, it should add the received cost to the cost of the link over which this cost packet has been received. If the result is lower than this node's current cost, the parent node should be updated and the new cost should be broadcasted. During this process, a node may be unable to find its minimum-cost path towards the sink if it loses a cost packet due to collision. In addition, during the data gathering phase, the probability of successful packet delivery to the sink node reduces as the cost of the path over which the packet is sent increases. This also increases the number of retransmissions when packet retransmission is employed at the MAC layer. Therefore, reducing CTC accuracy causes lower data delivery percentage and wastes energy resources. On the other hand, as stated earlier, CTC protocol should be periodically run during the data gathering phase to update path costs. These periodical updates are specifically required due to: (i) inherent variations of low-power wireless links, (ii) node mobility, (iii) node arrival and death, and (iv) obstacle movement. Hence, CTC execution frequency depends on network dynamics. The main aim of this section is to improve packet broadcast reliability during the CTC phase without increasing CTC duration.

4.1. Child–parent cost distribution

Using various network densities, we analyzed child-parent link costs during the CTC phase. More specifically, we analyzed the probability of cost broadcast with respect to the link cost between a sender and its parent. For each cost value we consider the maximum integer value that is lower than the cost value (i.e., floor value). Moreover, it is assumed that the maximum cost corresponding to a link is 100, which is obtainable when the forward and backward packet receptions rates are 10%. Hence, 100 intervals are considered for the child-parent link costs. Fig. 9 shows the frequency of packet transmissions with respect to the child-parent link costs. As the network density reduces, the number of transmissions shifts towards the left and the ECDF curve shows higher slope. Specifically, for neighborhood size 10, 30 and 50 the probability of cost broadcast for the link costs between 1 and 10 is 85%, 71% and 54%, respectively. This behavior can be justified as follows: Increasing network density produces more number of high-quality links and also raises the number of recep-

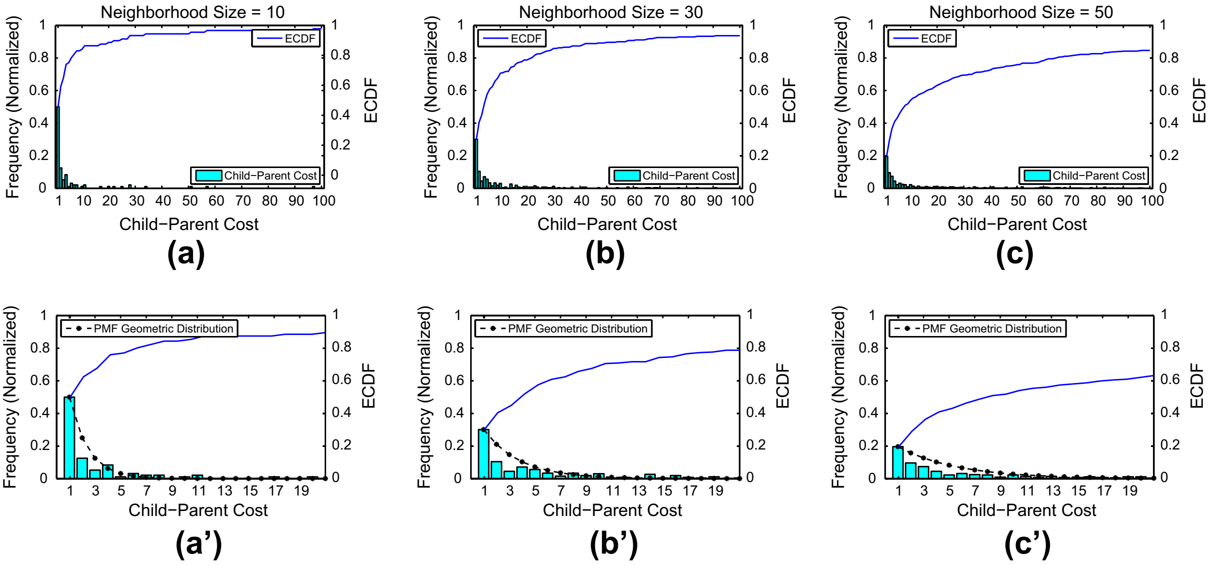

Fig. 9. The probability of cost broadcast with respect to the child-parent link costs. Lower figures present the upper figures with more details including the geometric distribution. Although increasing neighborhood size reduces the frequency of cost broadcast at lower child-parent link cost values, nevertheless, these results show that more than 50% of the transmissions occur for those child-parent link costs that are less than 10. The relationship between cost broadcast frequency and child-parent cost can be estimated through the geometric distribution.

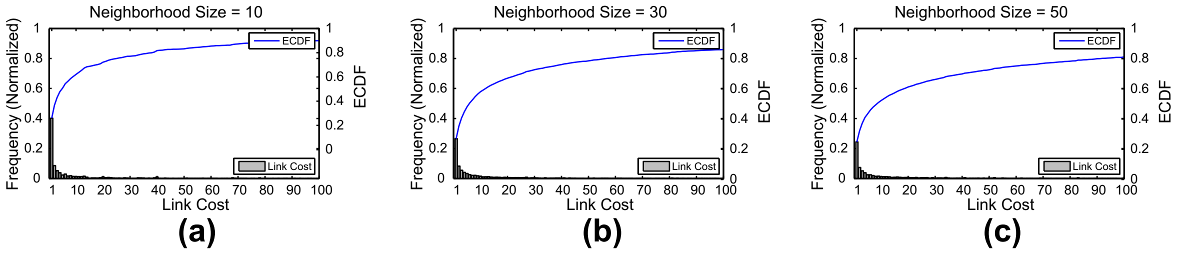

Fig. 10. Link cost distribution with various network deployments. Link cost distribution is similar to the cost broadcast distribution shown in Fig. 9. Therefore, we use link cost distribution to compute the λ parameter in Eq. 9.

tions per cost broadcast. Therefore, cost transmission probability shows higher dispersion and ECDF curve shows lower slope versus child-parent link cost. Nevertheless, these results show that even at high network densities more than 50% of the transmissions are for those child-parent costs that are below 10. As the ECDF curves show, cost broadcast distributions can be well estimated through the exponential distribution, which is equivalent to the geometric distribution due to the discrete child-parent cost values. This has been demonstrated in the lower row of Fig. 9 through adding the PDF functions of the geometric distribution. Assuming the link cost between node j and node i is $c_{j,i}$, the probability of cost broadcast by node j after receiving a cost packet from node i can be estimated as follows,

$$\Psi(c_{j,i}, \lambda) = \lambda(1-\lambda)^{\lfloor c_{j,i} \rfloor - 1}, \tag{9}$$

where λ is the parameter of the geometric distribution (a.k.a., rate parameter).

Regarding channel contention level, Eq. 9 provides a good indication of the contention among the neighbors of a cost broadcaster (node i in this case). However, it is still unclear how the λ value can be obtained. To this aim, we analyzed the link cost distribution of various network deployments (Fig. 10). These results show that the link cost distribution for a given network deployment is similar to the cost broadcast distribution of that network. Based on these results, in the next section we propose two mechanisms to compute the λ value.

4.2. Computing the λ value

In order to utilize Eq. 9 for estimating channel contention intensity among the neighbors of a cost broadcaster, it is required to have a network-wide or a per-node estimation of the λ value. To this aim, we propose and investigate two approaches.

Mathematical approach. As Fig. 9 shows, the ECDF curves of various network densities surpass 0.8 as the child-parent cost reaches 100. Moreover, according to Eq. 9, $\Psi(1, \lambda) = \lambda$. Therefore, we observed that λ can be obtained through estimating the ratio of the number of nodes with $1 \leqslant ETX < 2$ to the total number of neighbors with $\lfloor ETX \rfloor \leqslant 100$. In this approach, which provides a network-wide estimation of the λ value, the area belonging to the neighbors with a specific link quality is computed. Assuming symmetric links, since the required link quality to achieve $\lfloor ETX \rfloor = \lfloor 1/p^2 \rfloor = 1$ is to have $p > 70\%$, we compute $\Upsilon(0.7)$ that is the distance within which link qualities are higher than 70%. Similarly, we compute $\Upsilon(0.1)$ for those links with $\lfloor ETX \rfloor \leqslant 100$. Considering uniform network density, the ratio of the nodes inside the circle with radius $\Upsilon(0.7)$ to the circle with radius $\Upsilon(0.1)$ is

$$\lambda = \frac{\Upsilon^2(0.7)}{\Upsilon^2(0.1)}, \tag{10}$$

where $\Upsilon(x)$ is computed through Eq. 3. Hence, λ is obtained. Notice that using this model the λ value depends on the radio (e.g., transmission speed) and environmental (e.g., path loss) parameters, and it is independent of network density. With the parameters given in Table 1, the λ value is about 80%. Therefore, all the nodes utilize the default λ value irrespective of the λ of the node from which a cost packet has been received.

Adaptive Approach. The second approach which provides a per-node estimation of the λ value utilizes the information obtained during the NDLE phase. At each node the collection tree protocol refers to the neighbor table and computes the ETX of its links. Assume node i wants to compute its λ value. First, we define set Φ_l^u, which includes those links between node i and its neighbors that the floor of their ETX cost lays within a specific range $[l, u]$,

$$\begin{aligned} &\Phi_l^u = \left\{\varphi_{i,1}, \varphi_{i,2}, \ldots, \varphi_{i,j}, \ldots, \varphi_{i,|\Phi|}\right\}, \\ &\varphi_{i,j} \in \Phi \text{ if } l \leqslant \left\lfloor ETX(\varphi_{i,j}) \right\rfloor \leqslant u. \end{aligned} \tag{11}$$

If $l = u$, set Φ_l^u includes those links between node i and its neighbors that the floor of their ETX cost equals a specific value.

Then, we find the floor of the minimum link cost between node i and its neighbors,

$$\gamma = \left\lfloor \min\left\{ETX(\varphi_{i,1}), ETX(\varphi_{i,2}), \ldots, ETX(\varphi_{i,|\Phi|})\right\}\right\rfloor. \tag{12}$$

Using these definitions, the λ value is estimated as follows

$$\lambda = \frac{\left|\Phi_\gamma^\gamma\right|}{\left|\Phi_1^{100}\right|}. \tag{13}$$

Fig. 11 shows the λ distribution for three various network densities. Compare with the computed network-wide values, most of the node's λ value is lower than 80%. Since the proposed mathematical model assumes uniform network density, it produces valid values for those nodes that their transmission coverage area is within the network area. Therefore, the mathematical model can be specifically used for large networks.

4.3. The Geowindow algorithm

This section proposes a contention window assignment algorithm, called *Geowindow* algorithm (Algorithm 2). For a given set of input parameters this algorithm generates a sub-contention window (sub-CW) through specifying its first and last slot numbers. A generated sub-CW has the following properties: (i) To avoid incrementing the CTC duration, this algorithm does not increase the original contention window size; rather, the generated sub-CW is selected from the original contention window. (ii) The size of the generated sub-CW depends on the path cost value towards the sink. Specifically, sub-CW size increases as the child-parent link quality reduces. (iii) This algorithm tries to reduce the number of early cost broadcasts that may cause collision or may be invalid in the near future. To this aim, the generated sub-CW is shifted in time based on the child-parent link cost.

Assume node i broadcasts a cost packet. When a neighbor of node i receives this packet and decides to select node i as its parent, it runs the Geowindow algorithm to find the sub-CW from which it can select its backoff slot for sending its cost packet. In order to avoid those collisions caused by identical slot selection, the size of the assigned sub-CW increases geometrically as the link cost to the parent node reduces. Moreover, since earlier transmission should be assigned to lower link cost value, the position of the assigned sub-CW depends on the link cost to the parent node. Consider a cost packet received at node j from node i. If this packet has been received over a link with cost $\lfloor c_{j,i} \rfloor = N$, the Nth sub-CW (referred to as W_N) should be used for the transmission of this packet. The length of W_N is computed as follows,

$$W_N = \Psi(N, \lambda) \times W, \tag{14}$$

where $\Psi(N, \lambda)$ is given in Eq. 9, and W is the original contention window size. While the length of the Nth sub-CW manages channel contention among those one-hop neighbors of node i with a specific child-parent link cost (i.e., $\lfloor c_{j,i} \rfloor = N$), the relative position of the sub-CWs reflects higher transmission priority for lower child-parent costs. In order to find the start of the Nth sub-CW, the size of the previous $N - 1$ sub- CWs is required. Since,

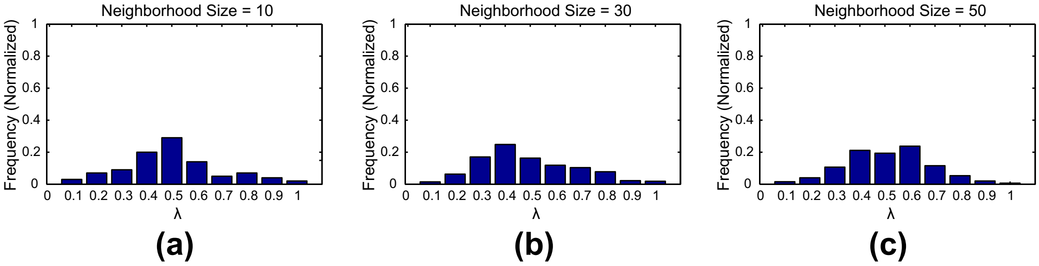

Fig. 11. The frequency of various λ values with three network densities. Each bar indicates the probability of having nodes within a given λ range. These results show the variations of the λ value, which are specifically caused by those nodes that their transmission range does not completely lie within the network area.

Algorithm 2. Geowindow algorithm

```
Input:
A link cost $c$ associated with the link to the parent node
A link cost $\gamma$ indicating the minimum cost of the parent node to its neighbors
Parameter $\lambda$ which can be a predefined constant value, or provided by the parent node
The original contention window duration: $W$
The minimum acceptable duration for a sub-CW: $W_{th}$
Output:
A sub-contention window $W_N$ determined by its first ($S_N$) and last ($E_N$) contention slot
Phase1:
Extend = False
$i = 1$
while $i \leqslant 100$ and Extend == False do
    $W_N = \Psi(i, \lambda) \times W$
    if $W_N \leqslant W_{th}$ then
        Extend = True
    end if
    $i = i + 1$
end while
$i = i - 2$
Phase2:
if $c > \gamma$ then
    $c = c - \gamma + 1$
end if
if $i < 1$ then
    $S_N = 1$
    $E_N = W$
else if $c \geqslant i$ and $i > 1$ then
    $S_N = \lceil (1 - (1 - \lambda)^{c-1}) \times W \rceil + 2$
    $E_N = W$
else if $i == 1$ then
    $S_N = 1$
    $E_N = \lceil \lambda \times W \rceil + 1$
else if $i > 1$ then
    $W_N = \Psi(c, \lambda) \times W$
    $S_N = \lceil (1 - (1 - \lambda)^{c-1}) \times W \rceil + 2$
    $E_N = \lceil ((1 - (1 - \lambda)^{c-1}) \times W) + W_N \rceil + 1$
end if
return $S_N$ and $E_N$
```

$$\sum_{i=1}^{N} \Psi(i, \lambda) = \sum_{i=0}^{N-1} (\lambda \times (1 - \lambda)^{i}) = 1 - (1 - \lambda)^{N}, \tag{15}$$

the start (S_N) and end (E_N) of the Nth sub-CW can be obtained through

$$S_N = \lceil (1 - (1 - \lambda)^{N-1}) \times W \rceil + 2, \tag{16}$$

$$E_N = \lceil ((1 - (1 - \lambda)^{N-1}) \times W) + W_N \rceil + 1. \tag{17}$$

The aforementioned approach works well as long as W_N is not lower than 1. Generally, it is desired to avoid assigning a very short sub-CW to a high N value. Therefore, we utilize a threshold value (W_{th}) that limits the minimum assignable sub-CW size. Assume that $W_N \leqslant W_{th}$ for $N \geqslant N_{th}$. In this instance, for $N \geqslant N_{th} - 1$ the algorithm assigns a sub-CW that is not shorter than W_{th} and it is located at the end of the original contention window. In Algorithm 2, the value of N_{th} is found in Phase 1, and it is denoted as i.

Assume that node i broadcasts a cost packet. Also, the γ value computed by this node equals 2. Therefore, this indicates that $\Phi_1^1 = \varnothing$. In this instance, the portion of the contention window that could have been assigned to those child-parent link costs with $\lfloor c \rfloor = 1$ is wasted. To remedy this problem, Phase 2 of the Geowindow algorithm starts with $c = c - \gamma + 1$ when $c > \gamma$. It should be noted that condition $c > \gamma$ is necessary due to the possible differences in the link estimations of node i and its neighbors. For example, if node i has estimated link $l_{i,j}$ as $ETX(l_{i,j}) = 2.5$, node j may have estimated this link as $ETX(l_{j,i}) = 1.5$. These variations arise during the NDLE phase due to the issues such as beacon collision, multipath variations and noise variations.

Fig. 12 shows sample results obtained through applying the Geowindow algorithm. We can observe the followings from this figure: For a given contention window size, since increasing the λ value intensifies contention at lower cost values, the algorithm improves channel arbitration through increasing the assigned sub-CW size to lower cost values. This figure also shows that the minimum assigned sub-CW size is not shorter, but may be larger than the minimum allowable sub-CW size. Also, for a given contention window size, increasing the λ value reduces the minimum allocated sub-CW size due to the increase in the portion of the contention window devoted for channel arbitration with low cost values. The next observation is that for a given λ value, enlarging the contention window size increases the number of cost intervals for which a dedicated sub-CW is assigned. This specifically improves transmission prioritization.

4.4. Exploiting collision detection

In this section we try to investigate the potential benefits of utilizing partial packet recovery and collision detection for improving the accuracy of CTC.

Partial Packet Reception. In contrast to the NDLE phase in which the size of beacon packets is equal to the normal data packet size, the exchanged cost packets during the CTC phase are considerably smaller. The information that each node should embed in its cost packets includes node's address, its minimum cost to the sink, and its minimum number of hops to the sink. Therefore, since the reception of all the included information are required for cost computation and update, utilizing partially recovered packets during the CTC phase is useless and has not been investigated in this paper.

Cost Packet Retransmission. While the number of beacon packets that should be sent during the NDLE phase is predetermined, the CTC phase can employ packet retransmission for improving reception reliability. To this aim, we

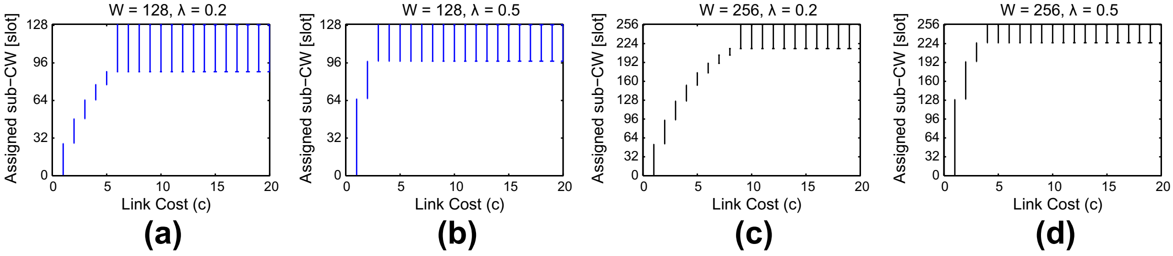

Fig. 12. The assigned sub-CW corresponding to various cost values. Each vertical bar determines the length and position of the assigned sub-CW. The minimum assignable sub-CW size (W_{th}) is 8 slots. The Geowindow algorithm does not resize the original contention window duration; instead, it performs collision avoidance through sub-CW assignment and prioritization.

propose a packet retransmission mechanism that works as follow: Whenever a node detects a collision, it immediately broadcasts a collision-indicative cost packet to ask its neighbors for cost packet retransmission. This cost packet includes all the fields that a normal cost packet includes, as well as an extra *collision flag* that implies collision detection at the sender. Notice that the path cost field of a sent collision-indicative packet may not represent a valid value if the sender has not yet found a path towards the sink. In this instance, the receivers cannot use this packet for cost improvement.

Assume node i sends a collision-indicative cost packet. When node j receives this packet, it evaluates some conditions to determine whether it is eligible to send a reply cost packet. Without these conditions, nodes may send unnecessary cost packets that has no effect on path cost update and may also cause further collisions. First, node j checks whether it has a path towards the sink. If this condition is not met, node j stops further evaluations because it cannot help to improve the cost of node i. If this condition is met, node j evaluates the received path-cost value. If node i does not have a path towards the sink, node j immediately broadcasts a cost packet. Otherwise, if the sender node has already found a path, node j should evaluate inequality $ETX(j) + ETX(l_{j,i}) < ETX(i)$ and broadcast a cost packet if this inequality holds. Here, $ETX(j)$ is the cost of node j to the sink, $ETX(l_{j,i})$ is the ETX of the link between node j and node i, and $ETX(i)$ is the received cost value (i.e., cost of node i). Therefore, node j sends a reply packet if its transmission may result cost improvement at node i. Although these conditions significantly reduce the number of reply packets, they cannot avoid an infinite round of broadcasting collision-indicative packets. Specifically, we observed that when network density is high, sending collision-indicative packets causes further collisions that finally result in an endless collision-report-collision loop. Accordingly, we limit the number of collision-indicative packets a node can send. It is worth mentioning that nodes do not use the Geowindow algorithm for sending collision-indicative beacon packets and their reply packets, rather, they use the original contention window.

4.5. Simulation settings and definitions

The simulation settings have been described in Section 3.5. In particular, Table 1 presents the simulation parameters, and Table 2 shows the networks used for the evaluations. In addition, the minimum assignable sub-CW size (W_{th}) is set to 8 slots. Each result value is the median of 20 simulation runs. Error bars represent upper and lower quartiles.

We implemented the Collection Tree Protocol of TinyOS [20] and evaluated its performance with the following channel access mechanisms:

- **CSMA**: The CSMA MAC protocol that employs a slotted contention window.
- **CSMA [Geowindow:Fixed λ]**: The CSMA protocol that utilizes the Geowindow algorithm. In this protocol all the nodes use the fixed λ value computed through the mathematical approach given in Section 4.2.
- **CSMA [Geowindow:Adaptive λ]**: The CSMA protocol that utilizes the Geowindow algorithm. In this protocol each node uses the adaptive approach given in Section 4.2 to compute its λ value.
- **CSMA [CD (*x*)]**: The CSMA protocol that utilizes the collision detection mechanism for cost packet retransmission. The value in the parenthesis indicates that each node is allowed to send at most x collision-indicative cost packets.
- **CSMA [Geowindow:Adaptive λ][CD (1)]**: The CSMA protocol that utilizes the Geowindow algorithm and cost packet retransmission.

In the presented results, for those protocols that utilize collision detection we have considered at most 4 collision-indicative packet transmissions. This is due to the much longer duration of the CTC phase that makes the comparison unfair when higher values are used.

4.6. Performance evaluations and discussions

Path Cost and CTC Duration. Since the final goal of the CTC phase is to provide each node with an efficient path towards the sink, we evaluate the efficiency of the various channel access mechanisms through measuring the average cost of the nodes towards the sink (Fig. 13). Furthermore, since CTC should also be periodically run during the data gathering phase, rapid and energy efficient CTC completion is desired. Therefore, the effects of various channel access mechanisms on

CTC duration and energy efficiency are also measured and shown in Fig. 14.

We can observe that CSMA cannot find the optimal paths, even when the contention window duration is large. In addition, its significant performance variations indicate the high influence of collisions. Therefore, the low accuracy of this protocol reduces network performance during the data gathering phase.

Our results show that utilizing collision detection for improving CTC accuracy comes at the cost of significantly higher duration and energy consumption. For example, for neighborhood size 30 in Fig. 13b, CSMA [CD (4)] presents about 47% improvement over CSMA; however, this improvement comes at the cost of 134% increase in duration (and energy consumption). Therefore, although we observed that the conditions given in Section 4.4 can significantly reduce the number of reply transmissions, the duration of these mechanisms is high, which makes them unsuitable for periodical cost update during the data gathering phase. In addition to this drawback, CSMA [CD (x)] mechanisms may not necessarily improve accuracy. For example, for neighborhood size 10 in Fig. 13b, the performance of the CSMA [CD (x)] mechanisms is lower than that of CSMA. Our analyses showed that, since the neighborhood size is low and contention window size is large, the number of hidden-node collisions is high and the collision detection mechanism can effectively detect these collisions. However, the transmission of collision-indicative cost packets and their reply packets have prevented some nodes from receiving the cost packets that could have resulted cost improvement. In addition, the number of collision-indicative packets was not large enough to compensate for the loss of these cost packets. Therefore, the other drawback of this approach is that it is hard to find an optimal number of cost retransmissions for a given network configuration.

In contrast with the CSMA [CD (x)] mechanisms, the Geowindow-based mechanisms can achieve high accuracy without sacrificing duration or energy efficiency. This is because the Geowindow-based mechanisms utilize collision avoidance instead of collision detection and packet retransmission. For example, for neighborhood size 50 in Fig. 13a, CSMA [Geowindow:Adaptive λ] reduces the cost by about 74% compare with the CSMA mechanism; while, it increases the duration and energy consumption by only 7%. The slightly higher duration of the Geowindow-based mechanisms is due to the employed transmission prioritization approach. In particular, since the Geowindow algorithm assigns sub-CWs that not only differ in length, but their positions also vary in time, some nodes are required to send their cost packet later than their neighbors. On the other hand, the transmission latency of a node also propagates to those nodes that their cost values depend on this node's cost value. Although the mathematical approach produces less accurate λ estimations compare with the adaptive approach, CSMA [Geowindow:Fixed λ] and CSMA [Geowindow:Adaptive λ] demonstrate similar efficiency. Through investigating this behavior we observed that those collisions affecting the accuracy of the CTC phase mostly happen on the nodes away from the network's margins. Therefore, since the mathematical approach can provide a good estimation of the λ value for these nodes, both mechanisms behave similarly. Consequently, when the network density is uniform and network parameters are known, the mathematical model can be used; otherwise, the adaptive approach should be implemented.

The presented results also show that combining the collision detection mechanism with the Geowindow algorithm does not present any considerable accuracy improvement. However, this approach suffers from high duration and energy consumption. Therefore, based on the given discussions, CSMA [Geowindow:Adaptive λ] provides the highest performance among the evaluated mechanisms.

Data Gathering Performance. In order to show the effects of non-optimal paths on the data gathering phase, we

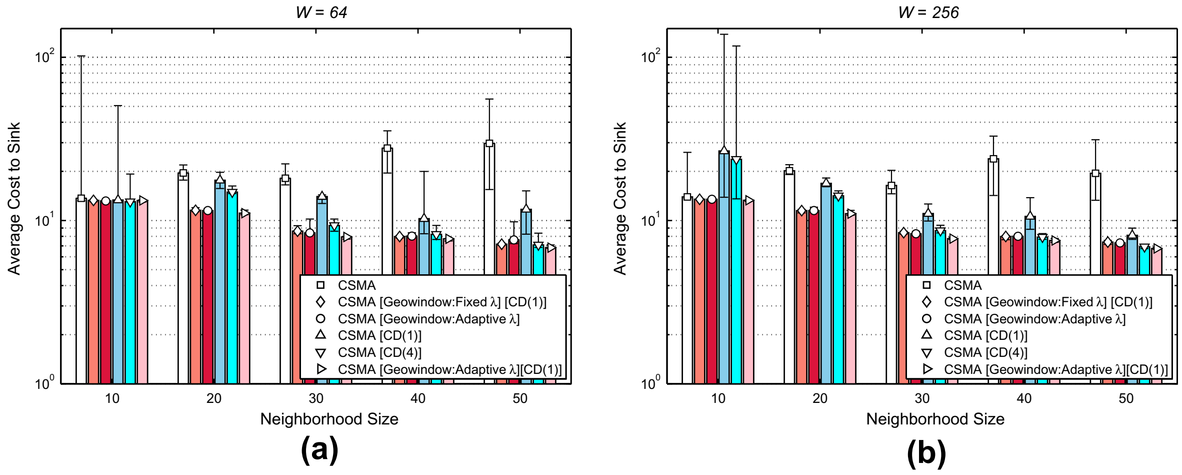

Fig. 13. Influence of the MAC mechanisms on CTC accuracy defined as the average cost of the nodes towards the sink. CSMA has the lowest performance and presents significant variations which are due to the effects of collisions. Although employing cost retransmission through collision detection can improve CTC accuracy, this approach significantly increases CTC duration and energy consumption, as shown in Fig. 14. The Geowindow-based mechanisms considerably improve CTC accuracy (up to 74%) without presenting long duration or high energy consumption (as Fig. 14 shows).

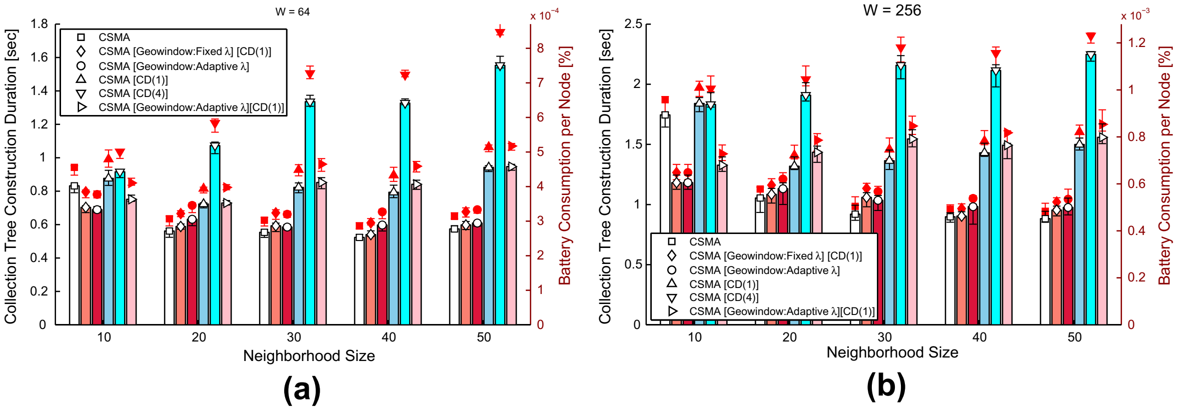

Fig. 14. Influence of the MAC mechanisms on CTC duration and energy consumption. The bars are corresponding to the left axis, which indicates duration. The markers above the bars indicate energy consumption, which is corresponding to the right axis. Utilizing cost retransmission causes significant increase in duration and energy consumption, which reduces the efficiency of the CSMA [CD (x)] mechanisms for periodical execution during the data gathering phase. The Geowindow-based mechanisms demonstrate short duration and low energy consumption, because they do not alter the original contention window size, and they reduce the number of cost broadcasts.

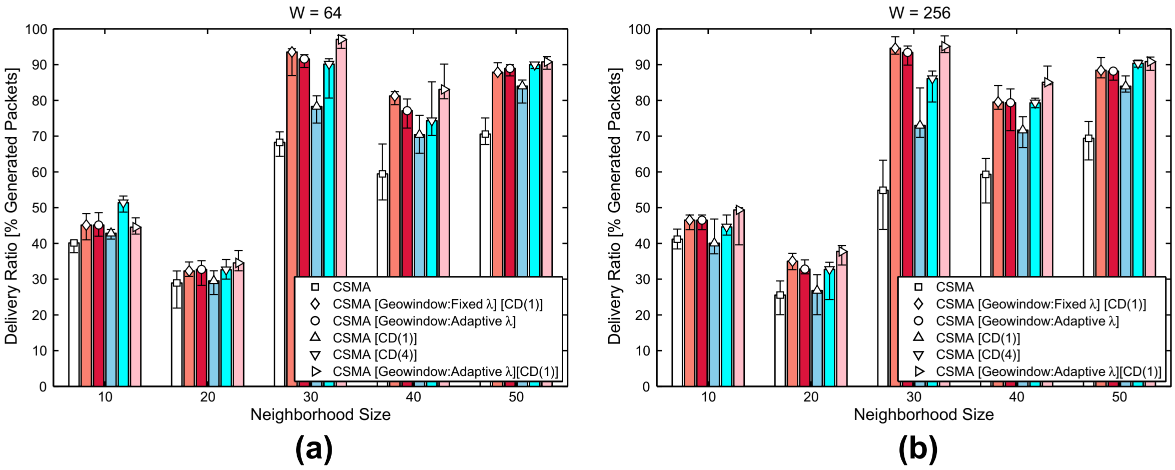

Fig. 15. Percentage of packet delivery at the sink node. Through comparison with Fig. 13, it can be observed that packet delivery performance depends on CTC accuracy. Here, CSMA:Geowindow [Adaptive λ] shows up to 70% improvement over the CSMA mechanism.

have considered a data gathering scenario in which each of the 20 nodes located at the farthest distance from the sink generate 100 packets. As Fig. 15 shows, packet reception performance highly depends on CTC accuracy, therefore, confirming the benefits of utilizing the Geowindow-based mechanisms. For example, CSMA:Geowindow [Adaptive λ] shows up to 70% improvement compare with CSMA. Additional studies also showed that as the network size enlarges, the Geowindow-based mechanisms present higher improvement than the CSMA mechanism. Further investigation showed that the Geowindow-based mechanisms provide higher path improvement as the number of hops towards the sink increases. Except data delivery percentage, the higher accuracy of these mechanisms also affects energy efficiency. For example, when a node sends up a packet that cannot be finally received by the sink node, the energy spent by that node is wasted. In addition, when a low-power MAC protocol such as B-MAC [51] is used at the MAC layer, the neighbors of the packet sender also spend energy for preamble reception and address evaluation.

5. Conclusions

NDLE and CTC phases play an important role in the functionality and performance of network protocols. With respect to the challenges of providing collision avoidance with CSMA and broadcast transmissions, this paper presented mechanisms that improve broadcast reliability during NDLE and CTC.

We proposed mechanisms that rely on the capture effect to perform collision detection and adjust the contention window size. Our results show that when the linear and exponential backoff schemes are combined, the resulting MAC mechanism can provide fast contention window adaptation. Performance evaluations also showed that partially recovered packets can be used for improving NDLE accuracy without any extra overhead.

Since pre-deployment parameter adjustment becomes harder as the network size enlarges, the proposed mechanisms are particularly useful for large-scale networks with random topology. Therefore, even when topology change is caused by node mobility, these mechanisms can provide adaptive collision avoidance based on collision intensity. With respect to the operation of higher-layer protocols, the proposed mechanisms are useful as long as a higher-layer protocol relies on neighborhood information or link qualities. For example, even for a geographic routing protocol, while neighbor discovery directly affects next-hop selection, link estimation can be used for avoiding the selection of low-quality links. It should be mentioned that the performance of the proposed mechanisms is independent of network size, and only depends on neighborhood density. Accordingly, this paper conducted performance evaluations considering low to high neighborhood densities.

In addition to the contention window adjustment mechanisms, this paper also presented a mathematical model which can be used for configuring contention window size based on network parameters. Therefore, this method can be used for a pre-deployment configuration that achieves a desired broadcasting success probability.

In order to improve broadcast reliability during the CTC phase, we proposed the Geowindow algorithm that manages channel access among those nodes that need to broadcast their newly computed cost value after a cost packet reception. We also proposed a medium access mechanism which improves broadcast reliability through collision detection and cost packet retransmission. We considered the Collection Tree Protocol of TinyOS and analyzed the effects of various channel access mechanisms. Our results showed that the Geowindow algorithm significantly improves nodes' path costs towards the sink, compare with the CSMA mechanism. The higher optimality of the tree constructed with the Geowindow mechanism directly affects the performance of packet forwarding during the data gathering phase. Besides, while achieving broadcast reliability through cost rebroadcasting increases CTC duration and energy consumption, the Geowindow algorithm does not present such inefficiencies. Therefore, this access mechanism is also suitable for periodical path updates during the data gathering phase. This is particularly important because not only link variations and node death, but strategies such as sink mobility also trigger CTC.

References

[1] A. Keshavarzian, E. Uysal-Biyikoglu, F. Herrmann, A. Manjeshwar, Energy-efficient link assessment in wireless sensor networks, IEEE INFOCOM' 04, vol. 3, IEEE, 2004, pp. 1751–1761.

[2] A. Meier, M. Weise, J. Beutel, L. Thiele, NoSE: Efficient initialization of wireless sensor networks, Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems – SenSys '08, vol. 285, ACM Press, 2008, p. 397.

[3] M. Radi, B. Dezfouli, K.A. Bakar, S.A. Razak, M.A. Nematbakhsh, Interference-aware multipath routing protocol for QoS improvement in event-driven wireless sensor networks, Tsinghua Sci. Technol. 16 (2011) 475–490.

[4] M. Radi, B. Dezfouli, K.A. Bakar, M. Lee, Multipath routing in wireless sensor networks: survey and research challenges., Sensors (Basel, Switzerland) 12 (2012) 650–685.

[5] K. Seada, M. Zuniga, A. Helmy, B. Krishnamachari, Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems – SenSys '04, ACM Press, 2004, p. 108.

[6] A. Warrier, M. Aia, M. Sichitiu, Z-MAC: a hybrid mac for wireless sensor networks, IEEE/ACM Trans. Network. 16 (2008) 511–524.

[7] A. Bachir, M. Dohler, T. Watteyne, K.K. Leung, MAC essentials for wireless sensor networks, IEEE Commun. Surveys Tutor. 12 (2010) 222–248.

[8] B. Dezfouli, M. Radi, M.A. Nematbakhsh, S.A. Razak, A medium access control protocol with adaptive parent selection mechanism for large-scale sensor networks, in: International Conference on Advanced Information Networking and Applications – WINA "11, IEEE, Biopolis, Singapore, 2011, pp. 402–408.

[9] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, P. Levis, Collection tree protocol, in: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems – SenSys '09, ACM Press, 2009, p. 1.

[10] O. Durmaz Incel, A. Ghosh, B. Krishnamachari, K. Chintalapudi, Fast data collection in tree-based wireless sensor networks, IEEE Trans. Mobile Comput. 11 (2012) 86–99.

[11] T. Schoellhammer, B. Greenstein, D. Estrin, Hyper: a routing protocol to support mobile users of sensor networks, Technical Report, Center for Embedded Network Sensing, UC Los Angeles, Los Angeles, 2006.

[12] I. Rhee, A. Warrier, DRAND: distributed randomized TDMA scheduling for wireless ad hoc networks, IEEE Trans. Mobile Comput. 8 (2009) 1384–1396.

[13] D. Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, Wireless Networks 11 (2005) 419–434.

[14] M.Z. Zamalloa, B. Krishnamachari, An analysis of unreliability and asymmetry in low-power wireless links, ACM Trans. Sensor Networks 3 (2007) 63–81.

[15] K. Srinivasan, P. Dutta, A. Tavakoli, P. Levis, An empirical study of low-power wireless, ACM Trans. Sensor Networks 6 (2010) 1–49.

[16] N. Baccour, A. Koubâa, L. Mottola, M.A. Zúñiga, H. Youssef, C.A. Boano, M. Alves, Radio link quality estimation in wireless sensor networks, ACM Trans. Sensor Networks 8 (2012) 1–33.

[17] C. Gomez, A. Boix, J. Paradells, Impact of LQI-based routing metrics on the performance of a one-to-one routing protocol for IEEE 802.15.4 multihop networks, EURASIP J. Wireless Commun. Network. (2010) 205.

[18] Chipcon CC2420, 2.4 GHz IEEE 802.15.4/ ZigBee-ready RF Transceiver, http://www.ti.com/lit/gpn/cc2420, 2013.

[19] A. Cerpa, J.L. Wong, M. Potkonjak, D. Estrin, Temporal properties of low power wireless links, in: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing – MobiHoc '05, January, ACM Press, 2005, p. 414.

[20] A.W. Rodrigo Fonseca, Omprakash. Gnawali, Kyle. Jamieson, Sukun. Kim, Philip. Levis, The Collection Tree Protocol (CTP), www.tinyos.net/tinyos-2.x/doc/html/tep123.html, 2007.

[21] A. Woo, T. Tong, D. Culler, Taming the underlying challenges of reliable multihop routing in sensor networks, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems – SenSys '03, ACM Press, 2003, pp. 14–27.

[22] TinyOS, The MultiHopLQI protocol, www.tinyos.net/tinyos-1.x/tos/lib/MultiHopLQI/, 2012.

[23] TinyOS 2.x, http://www.tinyos.net/tinyos-2.x/, 2013.

[24] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, D. Culler, Tinyos: an operating system for sensor networks, Ambient Intelligence, vol. 35, Springer Verlag, 2005, pp. 115–148.

[25] G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function, IEEE J. Selected Areas Commun. 18 (2000) 535–547.

[26] W. Ye, J. Heidemann, D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Trans. Network. 12 (2004) 493–506.

[27] V. Rajendran, K. Obraczka, J.J. Garcia-Luna-Aceves, Energy-efficient collision-free medium access control for wireless sensor networks, in: Proceedings of the first International Conference on Embedded Networked Sensor Systems – SenSys '03, ACM Press, 2003, p. 181.

[28] X. Yang, J. Liu, F. Zhao, N.H. Vaidya, A vehicle-to-vehicle communication protocol for cooperative collision warning, in: The First Annual International Conference on Mobile and Ubiquitous

Systems: Networking and Services, MOBIQUITOUS 2004, IEEE, 2004, pp. 114–123.

[29] D. Jiang, R. Sengupta, D. Chrysler, Design and analysis of highway safety communication protocol in 5.9 GHZ dedicated short range communication spectrum, in: The 57th IEEE Semiannual Vehicular Technology Conference, VTC 2003-Spring, vol. 4, IEEE, 2003, pp. 2451–2455.

[30] B. Krishnamachari, C. Raghavendra, An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks, Proceedings of the 18th International Parallel and Distributed Processing Symposium, vol. 00, IEEE, 2004, pp. 224–231.

[31] N. Balon, J. Guo, Increasing broadcast reliability in vehicular ad hoc networks, in: Proceedings of the 3rd international workshop on Vehicular ad hoc networks – VANET '06, ACM Press, 2006, p. 104.

[32] K. Jamieson, H. Balakrishnan, Y.C. Tay, Sift: a MAC protocol for event-driven wireless sensor networks, in: Third European Workshop, EWSN 2006, pp. 260–275.

[33] C.A. Boano, T. Voigt, N. Tsiftes, L. Mottola, R. Kay, A. Marco, Making Sensornet MAC protocols robust against interference, in: Proceedings of the 7th European conference on Wireless Sensor Networks, EWSN'10, Springer, Berlin, Heidelberg, 2010, pp. 272–288.

[34] R. Kumar, A. Paul, U. Ramachandran, D. Kotz, On improving wireless broadcast reliability of sensor networks using erasure codes, in: Mobile Ad-hoc and Sensor Networks, 2006, pp. 155–170.

[35] W. Lou, J. Wu, Toward broadcast reliability in mobile ad hoc networks with double coverage, IEEE Trans. Mobile Comput. 6 (2007) 148–163.

[36] P.T. Eugster, R. Guerraoui, S.B. Handurukande, P. Kouznetsov, A.-M. Kermarrec, Lightweight probabilistic broadcast, ACM Trans. Comput. Syst. 21 (2003) 341–374.

[37] R. Fonseca, O. Gnawali, K. Jamieson, P. Levis, Four bit wireless link estimation, in: Proceedings of the Sixth Workshop on Hot Topics in Networks, HotNets '07.

[38] K.-H. Kim, K.G. Shin, On accurate measurement of link quality in multi-hop wireless mesh networks, in: Proceedings of the 12th Annual International Conference on Mobile Computing and Networking – MobiCom '06, ACM Press, 2006, p. 38.

[39] D. Lal, A. Manjeshwar, F. Herrmann, E. Uysal-Biyikoglu, A. Keshavarzian, Measurement and characterization of link quality metrics in energy constrained wireless sensor networks, GLOBECOM '03. IEEE Global Telecommunications Conference, vol. 1, IEEE, 2003, pp. 446–452.

[40] G.-s. Ahn, S.G. Hong, E. Miluzzo, A.T. Campbell, F. Cuomo, Funneling-MAC: a localized, sink-oriented MAC for boosting fidelity in sensor networks, in: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems – SenSys '06, ACM Press, 2006, p. 293.

[41] K. Whitehouse, A. Woo, F. Jiang, J. Polastre, D. Culler, Exploiting the capture effect for collision detection and recovery, in: The Second IEEE Workshop on Embedded Networked Sensors, EmNetS-II, IEEE, Sydney, Australia, 2005, pp. 45–52.

[42] D. Son, B. Krishnamachari, J. Heidemann, Experimental study of concurrent transmission in wireless sensor networks, in: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems – SenSys '06, ACM Press, 2006, p. 237.

[43] D. Son, B. Krishnamachari, J. Heidemann, Evaluating the Importance of Concurrent Packet Communication in Wireless Networks, Technical Report April, USC/ISI Technical Report, 2007.

[44] J.-H. Yun, S.-W. Seo, Novel collision detection scheme and its applications for IEEE 802.11 wireless LANs, Comput. Commun. 30 (2007) 1350–1366.

[45] J. Hauer, A. Willig, A. Wolisz, Mitigating the effects of RF interference through RSSI-based error recovery, in: Proceedings of the 7th European conference on Wireless Sensor Networks, EWSN'10, February, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 224–239.

[46] K. Jamieson, H. Balakrishnan, PPR: partial packet recovery for wireless networks, ACM SIGCOMM Comput. Commun. Rev. 37 (2007) 409.

[47] T.S. Rappaport, Wireless Communications: Principles and Practice, second ed., Prentice Hall, 2002.

[48] Chipcon CC1000, Single Chip Very Low Power RF Transceiver, www.ti.com/lit/ds/symlink/cc1000.pdf, 2013.

[49] OMNeT++, The OMNeT++ Network Simulation Framework, http://www.omnetpp.org, 2013.

[50] A. Iyer, C. Rosenberg, A. Karnik, What is the right model for wireless channel interference?, IEEE Trans Wireless Commun. 8 (2009) 2662–2671.

[51] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems – SenSys '04, ACM Press, 2004, p. 95.

Behnam Dezfouli holds B.Sc. in Hardware Engineering and M.Sc. in Software Engineering. He is currently a Ph.D. Candidate in Computer Science at the Faculty of Computing, Universiti Teknologi Malaysia. He was research assistant at the Institute for Infocomm Research (I2R), Singapore, during 2012. He has more than two years of experience as a lecturer, and more than six years as a network engineer. His research interests include medium access mechanisms, routing algorithms, cross-layer optimization techniques, mathematical modeling, and developing simulation tools for wireless networks.

Marjan Radi holds B.Sc. and M.Sc. degrees both in Software Engineering. She is currently pursuing her Ph.D. degree in Computer Science at the Faculty of Computing, Universiti Teknologi Malaysia. She was research assistant at the Institute for Infocomm Research (I2R), Singapore, during 2012. She has more than two years of experience as a lecturer, and more than three years as a software developer. Her research interests include routing algorithms, congestion control, quality of service support, and link quality estimation mechanisms in wireless networks.

Shukor Abd Razak received his Ph.D. in Computer Science from University of Plymouth, United Kingdom. He is currently a senior lecturer at Universiti Teknologi Malaysia. His research interests are on wireless ad hoc and sensor networks, mobile IPv6 networks, vehicular ad hoc network and network security.

Kamin Whitehouse received his Ph.D. in Computer Science at University of California at Berkeley, where he worked on wireless sensor networks with his advisor David Culler. He has been in the Computer Science Department at the University of Virginia since 2006. His current research is focused on intelligent buildings, an interdisciplinary research area that sits at the boundary between Computer Science and Building Science.

Kamalrulnizam Abu Bakar received his Ph.D. in Computer Science from Aston University, United Kingdom. He is currently an Associate Professor in Computer Science at Universiti Teknologi Malaysia, and member of the Pervasive Computing research group. His specialization includes mobile and wireless computing, information security and grid computing.

Tan Hwee-Pink is currently the Unit Head of the Sense and Sense-abilities National Initiative Program, Institute for Infocomm Research (I2R), Singapore. Before joining I2R, he was a Post-doctoral Researcher at EURANDOM, The Netherlands from December 2004 to June 2006, and a Research Fellow with CTVR, Trinity College Dublin, Ireland from July 2006 to March 2008. His research has mainly focused on the performance analysis of wireless networks, and his current research interests are in underwater networks, cognitive radio networks and wireless sensor networks powered by ambient energy harvesting.