

Towards minimum-delay and energy-efficient flooding in low-duty-cycle wireless sensor networks



Long Cheng^{a,*}, Jianwei Niu^b, Chengwen Luo^c, Lei Shu^{a,d}, Linghe Kong^e, Zhiwei Zhao^f, Yu Gu^g

^a College of Engineering, Nanjing Agricultural University, China

^b Beihang University, Beijing, China

^c Shenzhen University, China

^d School of Engineering, University of Lincoln, UK

^e Shanghai Jiao Tong University, China

^f University of Electric Science and Technology of China, China

^g Watson Health Cloud, IBM Watson Health, USA

ARTICLE INFO

Article history:

Received 15 June 2017

Revised 30 November 2017

Accepted 15 January 2018

Available online 3 February 2018

Keywords:

Wireless sensor networks

Low-duty-cycle

Flooding

Minimum-delay

ABSTRACT

Wireless sensor networks (WSNs) play a very important role in realizing Internet of Things (IoT). In many WSN applications, flooding is a fundamental network service for remote network configuration, diagnosis or disseminating code updates. Despite a plethora of research on flooding problem in the literature, there has been very limited research on flooding tree construction in asynchronous low-duty-cycle WSNs. In this paper, we focus our investigation on minimum-delay and energy-efficient flooding tree construction considering the duty-cycle operation and unreliable wireless links. We show the existence of the latency-energy trade-off in flooding. We formulate the problem as a undetermined-delay-constrained minimum spanning tree (UDC-MST) problem, where the delay constraint is known a *posteriori*. Due to the NP-completeness of the UDC-MST problem, we design a distributed Minimum-Delay Energy-efficient flooding Tree (MDET) algorithm to construct an energy optimal tree with flooding delay bounding. Through extensive simulations, we demonstrate that MDET achieves a comparable delivery latency with the minimum-delay flooding, and incurs only 10% more transmission cost than the lower bound, which yields a good balance between flooding delay and energy efficiency.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Wireless sensor networks (WSNs) are important elements for realizing the Internet of Things (IoT), which are composed of tiny wireless sensing devices equipped with data processing and communication capabilities [1]. WSNs offer several advantages over traditional wired industrial monitoring and control systems including extended network coverage, easy and fast installation, resilience against single node failure and cost effective maintenance. On the contrary, traditional wired sensing and automation systems normally require expensive communication cables to be installed and regularly maintained [2]. In many WSN applications, e.g., factory automation, industrial process monitoring and control, and plant monitoring, flooding is a fundamental network service for remote network configuration, diagnosis or disseminating code up-

dates. The development of effective flooding protocol is hence a key research topic in this area. During flooding (or network wide broadcasting), messages from a root node are disseminated to the whole network via multi-hop communication. Since sensor nodes are usually energy constrained for WSN sustainable monitoring and surveillance applications, they normally operate at a very-low-duty-cycle (e.g., 1% or less) to ensure the service continuity [3].

Existing flooding protocols [4] utilize the broadcast nature of radio transmission to improve the delivery ratio and reduce transmission redundancy, i.e., a single transmission can be heard by multiple neighbors within the sender's radio range. However, in an asynchronous low-duty-cycle WSN, neighboring nodes do not always wake up at the same time. Flooding is essentially achieved through a number of unicasts [3,5], and thus more transmissions are required to ensure the flooding coverage than conventional wireless networks.

On the other hand, sensor nodes are subject to radio frequency interference. For example in harsh industrial environments, highly caustic or corrosive environments, high humidity levels, vibrations,

* Corresponding author.

E-mail address: chengl@vt.edu (L. Cheng).

dirt and dust, or other conditions challenge network performance [6]. As a result, wireless links can be highly unreliable. Considering the unreliable wireless links especially for low-power embedded devices, to forward a packet reliably, it is likely multiple re-transmissions are needed for an individual receiver to successfully receive a packet. In addition, flooding in low-duty-cycle WSNs suffers from a long sleep latency problem, where the sleep latency refers to the time that a sender spends on waiting for the receiver to wake up. Since each node only stays in active state for a very short period in each working cycle, a sender needs to wait for a long time until the receiver wakes up again and the interval between consecutive retransmissions is very large. Such an operation, poses new challenges for flooding protocol design on energy efficiency and latency.

Tree-based topology has been considered as an effective way to achieve efficient flooding in WSNs [3]. In low-duty-cycle WSNs, tree-based flooding tree aligns nodes' active slots for sending and receiving, which reduces idle-listening time. Compared with the asynchronous flooding without a tree structure, it avoids the sender sending probes for a long period that exceeds the sleeping period of the receiver. In addition, tree-based topology facilitates the reliable flooding, e.g., a parent node takes charge of the forwarding task to its children nodes. Such a flooding tree is usually constructed after the initial node deployment and re-constructed locally when the topology changes. A tree structure is time efficient if each node receives a flooding packet with minimum delay. However, it may not be energy optimal in terms of the total transmission cost for flooding a packet (i.e., there exists a latency-energy trade-off). In this paper, we study the problem of routing tree construction for minimum delay and energy efficient flooding in asynchronous low-duty-cycle WSNs with unreliable wireless links. The contributions from this work are summarized as follows:

- We show the existence of flooding latency-energy trade-off in asynchronous low-duty-cycle WSNs. Then, we formulate the minimum-delay and energy-efficient flooding problem as a undetermined-delay-constrained minimum spanning tree (UDC-MST) problem in low-duty-cycle WSNs, which is proven to be NP-complete.
- We present a distributed Minimum-Delay Energy-efficient flooding Tree (MDET) algorithm to construct an energy optimal tree with flooding delay bounding. The main idea is to first construct an ETX (Expected Transmission Count)-based shortest path tree, and then locally improve the energy optimality with delay constraint, by allowing a node to select its parent node with the best link quality while ensuring the network's loop-free property.
- To demonstrate the efficacy of MDET, we compare its performance with four different flooding tree construction methods. Evaluation results show that MDET achieves a comparable delivery latency with the minimum-delay flooding, only incurs 10% more transmission cost than the lower bound, and significantly outperforms the other baseline flooding tree structures.

The outline of the paper is as follows. Section 2 surveys the related work. Section 3 presents the network model. Section 4 elaborates the design of MDET protocol in detail. Section 5 provides the simulation results. Finally, conclusions are drawn in Section 6.

2. Related work

WSNs play a very important role in realizing IoT [7]. Typically, a WSN is composed of a large number of sensor nodes to measure physical phenomena. It provides valuable information to enable a wide range of applications, including smart battlefield, healthcare, environment and habitat monitoring, home automation, and traffic

control, fault diagnosis and prediction, and process control in industrial environments. In recent years, many research efforts have studied different enabling technologies that facilitate WSNs for real-world applications. TI's SensorTag provides a solution for the quick and efficient deployment of industrial sensor arrays that can be used for monitoring industrial equipment [8]. Rockwell have applied wireless sensors across the factory to synchronize production, link machines to smartphones to remotely monitor manufacturing processes, and to smart electricity grids to reduce energy costs [9]. Anastasi et al. [10] presented an adaptive staggered sleep protocol in WSNs. The proposed scheme dynamically adjusts the wakeup/sleep activities of sensor nodes based on the traffic pattern and the operating conditions experienced by nodes, achieving both low power consumption and delivery latency.

Since collecting data at a base station is a common requirement of WSN applications, many data gathering trees have been proposed in the literature [11–14]. Different from the data collection, flooding is another fundamental network service in WSNs, such as code update [15–18], remote network configuration and query [19], which has been extensively studied in the literature [20–22]. Construction of energy-efficient flooding and multicast trees in conventional wireless networks have been extensively investigated [4,23], which mainly takes advantage of the broadcast nature of wireless communication to improve the energy efficiency. Shen et al. [24] designed of a network-coding based multipacket flooding protocol that provides efficient and reliable message dissemination service for WSNs with unreliable and correlated links.

Recently, flooding in low-duty-cycle WSNs with unreliable links has attracted much attention in the wireless sensor network research community [3,5,25–29]. According to the radio duty-cycling model, flooding in low-duty-cycle WSNs can be generally classified into two categories: synchronous or asynchronous flooding. In synchronous flooding, for reducing energy consumption and sleep latency, a flooding tree synchronize nodes that have the same parent to wake up simultaneously to receive broadcast packets, by utilizing the wireless broadcast advantage. In asynchronous flooding, nodes set their own sleep/wakeup schedules independently.

In [3], Guo et al. proposed an energy optimal tree-based opportunistic flooding for asynchronous low-duty-cycle WSNs. Based on the delay distribution along the flooding tree, it makes a probabilistic forwarding decision at each sender, so that a packet always travels along a statistically minimum delay and energy efficient routing path. ADB [30] is another for asynchronous flooding protocol in duty-cycled WSNs. By introducing the transmission task delegation, ADB is able to avoid transmissions over poor links, thus reducing energy cost during flooding over unreliable wireless links. Lai et al. proposed an asynchronous and multihop broadcasting protocol in [31], which reduces redundant transmission via delivery deferring and online forwarder selection. The authors in [26] extended the dynamic switch-based forwarding [32] to the flooding scenario. Xu et al. [33,34] provided an adaptive control on the tradeoff between delay and energy efficiency for broadcasting in low-duty-cycle WSNs, by allowing receivers to defer their wake-up time slots to opportunistically overhear the broadcasting messages sent by their neighbors. Chen et al. [35] investigated the minimum active time slot augmentation for delay-bounded multicast problem in duty-cycled WSNs, which can be applied for the broadcasting communication scenario.

Recent advance in physical-layer concurrent transmission allows multiple senders transmit the same packet simultaneously and the constructively synchronized transmissions can be decoded at individual receivers. Glossy [22] exploits concurrent transmissions over interference for reliable flooding in WSNs, which removes unnecessary channel contention and improves the flooding performance. Cao et al. [36] presented a distributed concurrent broadcast layer for Low-Power-Listening (LPL) flooding in asyn-

chronous duty cycle WSNs. The key idea is to utilize the concurrent transmission (*i.e.*, capture effect) to improve flooding efficiency and remove the influence of packet contentions/collisions in asynchronous duty cycle networks. Du et al. [37,38] presented the Pando, a contention-free data dissemination protocol based on constructive interference and channel diversity in WSNs. Pando achieves 100% reliability and significantly reduces the dissemination latency by encoding data using Fountain codes and disseminating the encoded packets along the fast and parallel pipelines. However, such concurrent transmission based flooding can not be directly applied to low-duty-cycle WSNs.

Existing works on tree-based flooding in low-duty-cycle WSNs assume that a flooding tree has already been built, leaving the efficient flooding tree construction unaddressed. In [39], the authors present a centralized heuristic algorithm to solve the delay-constrained minimum spanning tree (DC-MST) problem. Different from previous works, this work is designed for flooding in asynchronous low-duty-cycle WSNs, where the delay constraint is determined in an a posteriori manner. In addition, MDET is a localized solution, which is able to achieve a good scalability in large scale WSNs.

3. Network model

3.1. Low-duty-cycle model

We consider a connected stationary multi-hop WSN with a single sink node. To efficiently flood packets over the whole network, we construct a flooding tree after the node deployment. In the initialization phase, we assume all nodes are awake during the tree construction. After the initialization phase, each node operates with a low-duty-cycle (*e.g.*, 1% or less) setting, periodically turning on the radio to transmit or receive packets according to a wakeup schedule. Our design is applicable to diverse duty-cycle schedules, and we do not assume any specific sleep/wakeup schedule. Nodes share wakeup schedules with one-hop neighboring nodes periodically, so that a sender always knows the rendezvous time for exchanging data with an intended receiver. A sleeping node switches to the active state when it is scheduled to switch to the active state, or it has some packets to transmit to a receiver that is active at that time slot.

A working cycle T is equally divided into fixed length slots, called time slots. However, due to the low-duty-cycle operation, the length of an active time slot τ is usually very limited. We assume a round-trip packet transmission can be completed in τ , including data and ACK transmissions. In each working cycle, for simplicity, we suppose there is one active time slot to receive packet. Our solution can be easily extended to scenarios where a node has multiple active time slots in one working cycle. Since it is very rare for multiple receivers waking up in the same time slot in low-duty-cycle WSNs, flooding is achieved through a number of unicasts, and requires multiple retransmissions because of the lossy links.

An example is shown in Fig. 1, where nodes A, B and C are scheduled to wake up to receive data packets at time slots $\{t_a, t_b, t_c\}$. When node S floods a packet to its child nodes, since A, B, and C wake up at different time slots, flooding in such a network is essentially achieved via multiple unicasts.

3.2. Unreliable wireless links

We assume the link quality is unreliable and measurable through MAC layer data loss measurement [41]. Each node v_i maintains a neighbor table $N(v_i)$ which stores the PRRs and wakeup schedules of all one hop neighbors. The local clocks of neighbors

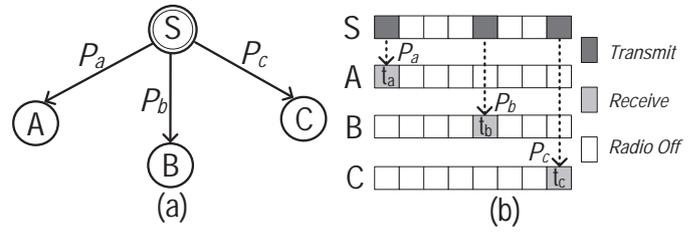


Fig. 1. Flooding is essentially achieved through a number of unicasts in asynchronous low-duty-cycle WSNs.

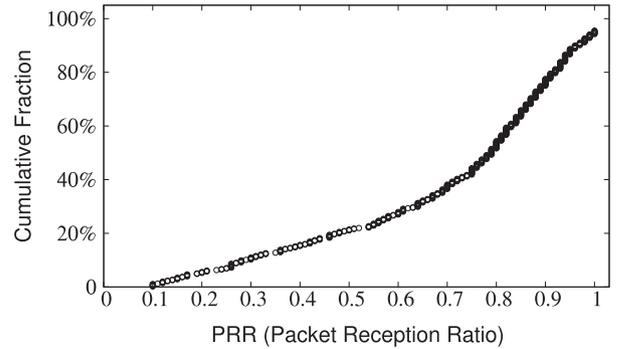


Fig. 2. Link qualities collected from the Indriya testbed [40], each node broadcasts 1000 packets consecutively, PRR is calculated as the ratio of successful transmissions over the total transmissions.

are synchronized and a sender is aware of the wakeup time of receivers [3]. There has been a number of existing research efforts on measuring wireless link quality in an efficient and accurate manner, such as using probe-based methods or through low-cost piggybacking on regular data traffic. Although link qualities change over time, empirical studies [42,43] demonstrated that the changing rate is relatively low. Therefore, the link quality will be updated in a low frequency.

Fig. 2 shows the wireless link qualities collected from the Indriya testbed [40], which is composed of 139 TelosB [44] sensor nodes. As shown in the figure, wireless communication links are normally unreliable, more than 60% links with PRR lower than 0.8.

4. Problem statement

4.1. Flooding cost and flooding delay

Given a connected network $G = (V, E)$, where V is a set of nodes and E is a set of time-dependent links. Each edge $e \in E$ is associated with a weight w_e , *e.g.*, the expected transmission count (ETX) of each hop. The link quality (PRR) threshold θ is used to determine whether a wireless link is good or not. Only if the PRR between two nodes is above θ , we consider there exists an edge between these two nodes. In other words, any link in E should have a better PRR than θ . Let $FloodingCost(\Gamma)$ denote the expected total transmission cost of flooding given a spanning tree Γ ($\Gamma \subseteq E$). Since flooding process is realized by a number of separate unicasts, we have

$$FloodingCost(\Gamma) = \sum_{e \in \Gamma} w_e \quad (1)$$

Let $HopDelay(e)$ denote the expected single-hop delivery delay on an edge e during flooding. For simplicity, assume that each node only wakes up once in one working cycle. Given the link quality $Pr(e)$ on edge e , which denotes the probability of a successful

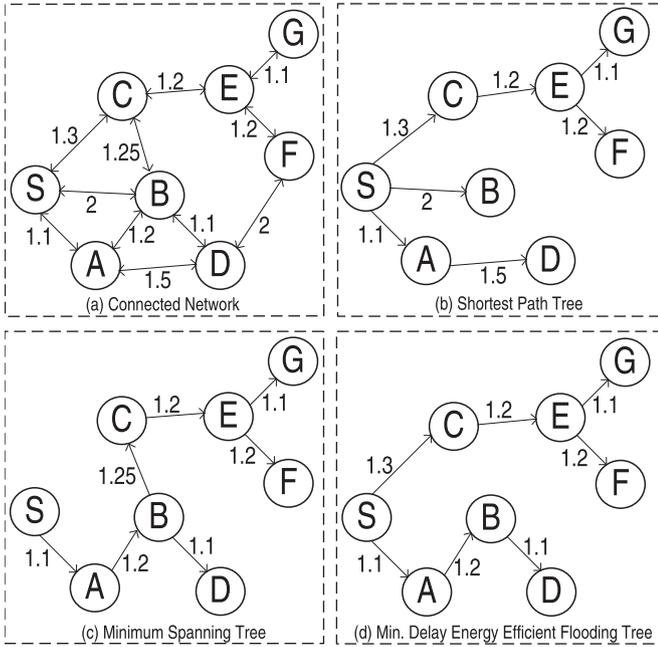


Fig. 3. Example of different flooding tree structures.

transmission, we have the expected single-hop delivery delay.

$$\begin{aligned}
 \text{HopDelay}(e) &= \text{Prr}(e) \cdot \text{SleepDelay}(e) + \\
 &\overline{\text{Prr}(e)}\text{Prr}(e) \cdot (\text{SleepDelay}(e) + T) + \dots + \\
 &\overline{\text{Prr}(e)}^k \text{Prr}(e) \cdot (\text{SleepDelay}(e) + kT) \\
 &\stackrel{k \rightarrow \infty}{=} \text{SleepDelay}(e) + (\text{ETX}(e) - 1)T \\
 &= \frac{T}{2} + (w_e - 1)T
 \end{aligned} \quad (2)$$

where $\overline{\text{Prr}(e)} = 1 - \text{Prr}(e)$, i.e., the probability of transmission failure, and $\text{ETX}(e) = 1/\text{Prr}(e)$. $\text{SleepDelay}(e)$ is the sleep latency of link e in one working cycle, and its expectation is $\frac{T}{2}$ in asynchronous low-duty-cycle WSNs. Similarly, for a node with multiple active slots in one working cycle, given the wakeup schedules of the sender and receiver on edge e , we can calculate the expected transmission cost [45].

Consequently, we have the expected end-to-end delay of a node v_i along a path $P(s, v_i)$, which is the set of tree links connecting from v_i to the sink node s .

$$\text{PathDelay}(s, v_i) = \sum_{e \in P(s, v_i)} \text{HopDelay}(e) \quad (3)$$

Flooding delay is defined as the time from the sink node flooding a packet to the last node in a network receiving the packet, we have

$$\text{FloodingDelay}(\Gamma) = \max_{v_i \in V} \{\text{PathDelay}(s, v_i)\} \quad (4)$$

4.2. Latency-energy trade-off

From Eqs. (2) and (3), an ETX-based shortest path tree (SPT) can minimize the flooding delay since every node receives a flooding packet with minimum delay. On the contrary, the energy optimality is achieved by constructing a minimum spanning tree (MST), where the total transmission cost is minimized. Fig. 3 shows an example of different flooding tree structures. Fig. 3(a) is the original connected network, where the edge weight represents the link ETX, denoted as ETX . In Fig. 3(b), the path distance from the sink node s to any other node in the network is the shortest ETX-based

path distance. In Fig. 3(c), the tree topology achieves the minimum sum of edge weights. Thus, Fig. 3 (b) and (c) show the corresponding SPT and MST, respectively. Fig. 3(d) illustrates an example of the minimum delay energy efficient flooding tree, since it provides the same flooding delay as the SPT tree in Fig. 3(b), while improves the flooding energy efficiency compared with the SPT tree. We will discuss how to construct such a tree in Section 5.

We observe that the two objectives (i.e., minimum flooding delay and minimum transmission cost) usually are contradictory, since the criteria of optimality for the two objectives are different. For example, the SPT in Fig. 3(b) is not energy optimal, while the flooding delay of the MST in Fig. 3(c) is much larger than that of the SPT.

4.3. Problem formulation

Our design objective is to construct such a flooding tree that achieves a low flooding delay and high energy efficiency. It can be described as a multiobjective optimization problem: given a network $G = (V, E)$, finding a spanning tree Γ with minimum flooding delay and transmission cost.

$$\begin{aligned}
 \min \max_{v_i \in V} \{\text{PathDelay}(s, v_i)\} \\
 \min \text{FloodingCost}(\Gamma)
 \end{aligned} \quad (5)$$

Due to the existence of flooding latency-energy trade-off and the contradictory objectives, we transform the above multiobjective optimization problem into a single objective optimization problem with constraint. The minimum-delay energy-efficient flooding tree construction problem is defined as follows: we first construct an ETX-based SPT, where the accumulated edge weights (i.e., flooding delay) from the sink node to each node is minimized. We get the minimum of expected flooding delay $\min \max_{v_i \in V} \{\text{PathDelay}(s, v_i)\}$, denoted as Δ . Then, we construct a spanning tree Γ that satisfies:

$$\begin{aligned}
 \min \sum_{e \in \Gamma} w_e \\
 \text{FloodingDelay}(\Gamma) \leq \Delta
 \end{aligned} \quad (6)$$

Theorem 1. *The minimum-delay and energy-efficient flooding tree construction problem is NP-complete unless all edge weights are equal.*

Proof. Consider a simplified version of the minimum-delay energy-efficient tree construction problem, where the delay constraint is fixed. It is equivalent to the delay-constrained minimum spanning tree (DC-MST) problem, which is proven NP-complete unless all edge weights are equal [39]. Therefore, the original minimum-delay energy-efficient tree construction problem, i.e., undetermined-delay-constrained minimum spanning tree (UDC-MST), is NP-complete. \square

5. Protocol design

Since the minimum-delay energy-efficient tree construction problem is NP-complete, in this section, we propose a distributed flooding tree protocol, called MEDT, to construct an energy optimal tree with flooding delay bounding. The MEDT protocol consists of two phases: 1) constructing an ETX-based shortest path tree; and 2) locally adjusting the tree structure to improve the energy optimality with delay constraint. Our idea is to locally adjust the tree structure to improve the energy optimality of a flooding tree while maintaining a minimum flooding delay. As shown in Fig. 3(d), given a SPT in Fig. 3(b), the total number of transmissions can be reduced without increasing the flooding delay if node B selects A as its parent node, then node D chooses B as its parent node.

5.1. Construction of minimum delay flooding tree

The process of building a minimum delay flooding tree starts from the sink, and propagates throughout the network iteratively. The sink node broadcasts a TREECONSTRUCT message to construct an ETX-based shortest path tree. The accumulated path ETX (denoted as $pETX$) and a limited number (k -hop) of the most recent forwarder IDs are piggybacked onto the TREECONSTRUCT message. Each node gets the $pETX$ value recursively starting from the sink node. In the tree construction phase, all nodes need to rebroadcast the TREECONSTRUCT. It suffers the inefficiency problem of blind flooding, such as redundant transmission and high probability of collision. In order to reduce the control message overhead, we introduce a biased backoff mechanism to propagate the TREECONSTRUCT.

When a node v_i receives a TREECONSTRUCT with a smaller $pETX$ value, it stores the sender ID, $pETX$, and upstream k -hop forwarder list. Then it delays the rebroadcasting of this TREECONSTRUCT, where the aim is to amplify the differences of message traversing delays along different paths.

Let $t_{backoff}$ denote the backoff delay at the current forwarding node v_j , who receives a TREECONSTRUCT from v_i . $t_{backoff}$ is calculated as defined in Eq. (7). When the backoff timer expires, the TREECONSTRUCT message is rebroadcasted.

$$t_{backoff} = \left(\frac{\sum_{t \in P(s, v_i)} w_t}{HopCount} - 1 \right) * t_{max} + random(0, \tau) \quad (7)$$

where $\frac{\sum_{t \in P(s, v_i)} w_t}{HopCount}$ is the average $lETX$ along the path connecting from v_j to the sink node via v_i . t_{max} is a constant representing the maximum delay time that a forwarder will wait for rebroadcasting, and τ is a parameter that is much smaller than t_{max} .

Each node only rebroadcasts the TREECONSTRUCT if the new $pETX$ value is smaller. Otherwise, the received message is dropped. From Eq. (7), TREECONSTRUCT travels faster along the path with a lower $pETX$ value, so that earlier received TREECONSTRUCT can suppress the rebroadcasting of later received TREECONSTRUCT messages, consequently reducing unnecessary control message overhead. Finally, a minimum-delay flooding tree is constructed. Each node maintains the child set $C(v_i)$, and its parent node $f(v_i)$. Algorithm 1 describes how a node v_i handles a received TREECONSTRUCT message from node v_k .

Taking Fig. 4(a) for example, node A defers $[0.1t_{max}, 0.1t_{max} + \tau]$, and nodes B and C will defer $[t_{max}, t_{max} + \tau]$, $[0.3t_{max}, 0.3t_{max} + \tau]$, respectively. $\{t_0, t_1, \dots, t_7\}$ are sorted in the chronological order of nodes' sending time of TREECONSTRUCT. In this case, A will rebroadcast the TREECONSTRUCT earlier, where $pETX$ is 1.1. Once nodes B and D receive this message, they record A's $pETX$ in the neighbor table. Since B's current $pETX$ is 2 from the sink node S, it won't rebroadcast the TREECONSTRUCT received from A, where the $pETX$ value is 2.3. For node D, it only forwards the TREECONSTRUCT that is sent from A. In an ideal case, each node needs to forward the TREECONSTRUCT once, i.e., the message complexity of the flooding tree construction is $\Omega(|V|)$.

5.2. Improving the energy optimality

A straightforward design for the UDC-MST problem is to disseminate the global minimum flooding delay Δ over the network. Then, a minimum spanning tree is constructed without violating the delay constraint. Once an ETX-based shortest path tree is constructed, each node is aware of its neighboring nodes' $pETX$ values. This means, each node knows the local maximum path delay. However, in order to get the network-wide maximum path delay (i.e., the flooding delay) distributedly, each node with a local maximum path delay needs to flood its $pETX$ value over the network.

Algorithm 1: How v_i handles a TREECONSTRUCT received from v_k .

```

1 Procedure: void recvTreeConstruct (Packet *p)
2 if  $pETX(v_k) > \widehat{pETX}(v_i)$  then
3    $\widehat{pETX}(v_i) \leftarrow pETX(v_k)$ ;
4   //keep track of the local maximum pETX among
   //neighboring nodes;
5 end
6 if  $lETX(v_i, v_k) < lETX(v_i, f(v_i))$  and
    $pETX(v_k) + lETX(v_i, v_k) < \widehat{pETX}(v_i)$  then
7   sendJoinRequest(p);
8   //locally adjust tree structure to improve the energy
   //optimality;
9 end
10 if  $pETX(v_i) > pETX(v_k) + lETX(v_i, v_k)$  then
11    $pETX(v_i) \leftarrow pETX(v_k) + lETX(v_i, v_k)$ ;
12   //update  $f(v_i) \leftarrow v_k$ , and upstream k-hop forwarder
   //list;
13   Calculate Backoff(p);
14   //schedule a timer whose value is  $t_{backoff}$ ,
15   //then call forwardTreeConstruct(p) when the timer
   //expires;
16 else
17   Drop(p);
18 end

```

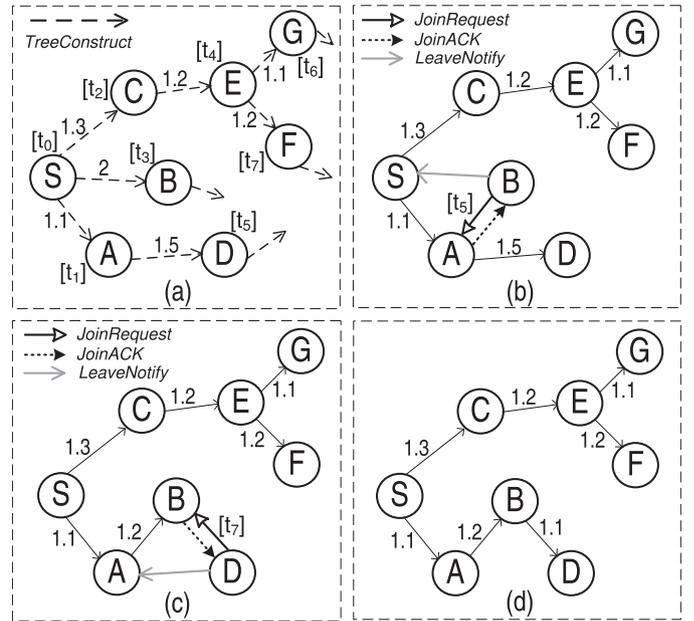


Fig. 4. Example of tree construction in MDET.

Together with the reconstruction of the minimum spanning tree after obtaining the delay constraint, this method introduces too much message overhead and time complexity, which is undesirable for the resource-constrained WSNs. Therefore, we propose to improve the energy optimality through local knowledge in a distributed manner.

5.2.1. Locally adjusting tree structure

During the TREECONSTRUCT propagation phase, for any node v_i , it keeps track of the $pETX$ information for each neighboring node.

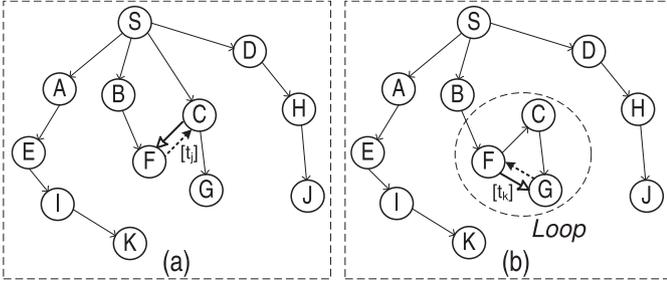


Fig. 5. Example of loop avoidance and detection.

Once collecting all $pETX$ from one-hop neighbors, v_i knows the local maximum $pETX$ in the neighborhood, denoted as $\widehat{pETX}(v_i)$.

$$\widehat{pETX}(v_i) = \max_{\forall v_j \in N(v_i)} pETX(v_j) \quad (8)$$

When v_i realizes that its own $pETX$ is smaller than $\widehat{pETX}(v_i)$, it considers a neighboring node v_k ($v_k \neq f(v_i)$ and $v_k \notin C(v_i)$) as its potential parent candidate if meeting the following conditions.

$$\begin{aligned} IETX(v_i, v_k) < IETX(v_i, f(v_i)) \\ pETX(v_k) + IETX(v_i, v_k) < \widehat{pETX}(v_i) \end{aligned} \quad (9)$$

From In Eq. (9), switching to a potential parent candidate should improve the energy efficiency while not increasing the flooding delay. Among all the potential parent candidates, v_i selects the neighbor v_k which has the highest PRR to it. Then, it sends a JOINREQUEST to v_k to request to join v_k 's subtree.

When v_k receives a JOINREQUEST from v_i , it checks whether keeping v_i as a child node violates the loop-free property or not, which will be introduced in detail in the next subsection. If yes, v_k replies a JOINDENY to v_i . Otherwise, v_k replies a JOINACK to v_i confirming that v_i is now a child of v_k , where the JOINACK includes v_k 's upstream k -hop forwarder list. Then, v_i sends a LEAVENOTIFY to its original parent $f(v_i)$. Once the previous parent receives a LEAVENOTIFY, it removes v_i from the child set. The LEAVENOTIFY also includes the updated $pETX$ value of the sender, so that neighbors that overhear this message can update their neighbor tables regarding to the change of $pETX$.

We also introduce a random backoff delay before sending the JOINREQUEST message to avoid multiple nodes sending the switching requests at the same time. Due to the wireless broadcast transmission nature, if a node v_i overhears the JOINREQUEST and the following JOINACK transmissions, it re-assesses the switching benefit to make sure its flooding delay is less than the delay constraint $\widehat{pETX}(v_i)$ after a potential switching.

Fig. 4(b) and (c) show examples of locally adjusting tree structure. Nodes A, C, D and S are direct neighbors of B. As shown in Fig. 4(b), at time t_5 , B collects all neighbor's $pETX$ information. This triggers B to select the best parent candidate A and send a JOINREQUEST to it. Suppose B receives a JOINACK from A, it broadcasts a LEAVENOTIFY. Then S removes B from its child set. In the meantime, A, C and D update B's $pETX$ value in their neighbor tables. Fig. 4(c) shows the control message exchange when D switches to B at time t_7 , after D overhearing a TREECONSTRUCT from F. The constructed MDET tree is shown in Fig. 4(d).

5.2.2. Loop avoidance

The flooding tree structure is a kind of directed acyclic graph (DAG) with $|V|$ vertices rooted at the sink node. In the tree structure adjusting phase, it may result in a routing loop when a node switches to a neighbor with a larger hopcount. As shown in Fig. 5, suppose node C switches to F at time t_j after collecting the local maximum $pETX$. Then, node K broadcasts a TREECONSTRUCT at time

t_k , which triggers node F to improve the energy efficiency. If F finds that G is the best parent candidate, it will send a JOINREQUEST to G. In this case, it creates a routing loop if G accepts F's switching request.

Let us take Fig. 5(a) for example to show the loop avoidance design. When F returns a JOINACK to C, F's upstream k -hop (e.g., $k=2$) forwarder list is piggybacked on the JOINACK. C updates its upstream k -hop forwarder list, then F will notify the change of route to its descendants within k hops. In this example, G will know its ancestor nodes C, F and B. When F attempts to join G's subtree, a potential routing loop can be avoided if G finds that F is in the upstream forwarder list. If so, G replies a JOINDENY to F, thus avoiding the routing loop.

Note that k can be just a very small integer, since the probability that a node switches to a neighbor with 2 more hopcount is quite low. The loop avoidance approach only requires extra control messages when a node switches to a neighbor with a larger hopcount. However, we limit the dissemination of k -hop forwarder list within k nodes each time, thus bounding the control message overhead.

5.2.3. Loop detection

In addition to the loop avoidance mechanism, we also introduce the loop detection in MEDT at the end of tree construction phrase. An upstream node sends a loop detection message to downstream nodes, and each node caches the data identifier of a received message. A routing loop is detected if a node receives a duplicate message. In this case, the node sends a control message back to the sender to break the loop, and switches to its original parent node. For example in Fig. 5(b), assume nodes C, F and G forms a routing loop. When F receives a duplicate message from G, it will send a notification to G. Consequently, F sends a JOINREQUEST to B for tree structure recovery.

5.3. Discussion

5.3.1. Fault tolerance in tree construction

Since the wireless links are unreliable, we consider the possibilities of packets loss of control messages in the process of tree construction. If a node fails to receive a TREECONSTRUCT message, it is likely to receive another TREECONSTRUCT message at a later time since all nodes in the network forward the message. If a node realizes it is an isolated node (without receiving any TREECONSTRUCT MESSAGE), it will broadcast JOINREQUEST messages to join any neighboring node. The impact of transmission failure of JOINREQUEST messages can be mitigated by introducing the retransmission mechanism. However, the loss of control packets may result in sub-optimal flooding tree structure. On the other hand, link qualities may change over time. Therefore, when link qualities are updated, the flooding tree structure will be reconstructed. In this case, the network potentially recovers from transmission failures of control messages.

5.3.2. Synchronization overhead and link dynamics

Our design assumes that local clocks of neighboring nodes are synchronized, where given their duty-cycle working schedules a sender knows when it can communicate with an intended receiver. In this way, it improve the energy efficiency by avoiding a sender sending probes for a long period to wait for the intended receiver to wake up at the cost of extra local time synchronization overhead. On the other hand, many low overhead local time synchronization methods have been proposed, which can be achieved by piggybacking the MAC layer timestamp on routing beacons [46].

Link qualities may change over time, and thus the link quality updating is normally needed. While empirical studies have shown that the changing rate is slow [42], if the link qualities of nodes

change largely, the flooding tree should be reconstructed accordingly to avoid the performance degradation. The incurred overhead for synchronization and tree reconstruction is expected to be amortized over a reasonably long period of network operation with improved energy efficiency.

6. Evaluation

In this section, we present performance evaluation results. We implement MDET over NS-2 [47] simulator, and compare the performance with four different tree construction methods.

- *Centralized minimum spanning tree (Cen-MST)*: the tree is constructed based on Prim's algorithm [48].
- *Hop-based shortest path tree (HOP-SPT)*: each node chooses a neighbor as its parent node with the minimum number of hops to reach the sink.
- *ETX-based shortest path tree (ETX-SPT)* [11]: each node chooses a neighbor as its parent node with the minimum $pETX$ value to reach the sink.
- *Heuristic Energy Optimal Tree (HEOT)* [3]: each node selects a neighbor which has the best link quality among those neighboring nodes with less hopcounts towards the sink as its parent node.

6.1. Simulation setup and performance metrics

In the implementation of our simulation, we carry out both random topology and grid topology tests in a $200m \times 200m$ square area. The node density is defined as the number of nodes deployed in the field. 100 random topologies for each node density setting ranging from 100 to 300 in a $200m \times 200m$ field are generated using the *setdest* tool in NS-2. In the grid topology, 10×10 to 20×20 nodes are uniformly placed in the field, forming a two dimensional grid. The sink node is positioned at bottom left (0m,0m) and the node transmission range is set as 40m.

For the unreliable wireless link model, we use the Nakagami fading model defined as (10) to describe the power x of a received signal and derive the PRR which is related with the distance between two nodes.

$$f(x, m, \Omega) = \frac{m^m x^{m-1}}{\Gamma(m)\Omega^m} \exp\left(-\frac{mx}{\Omega}\right) \quad (10)$$

where Γ is the Gamma function, m denotes the Nakagami fading parameter and Ω is the average received power. We set $m = 1$ in our simulation. Assuming *TwoRayGround* signal propagation, Ω can be expressed in (11) as a function of d , the distance between the sender and receiver.

$$\Omega(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^n L} \quad (11)$$

where P_t is the transmission power, G_t and G_r are the antenna gains, h_t and h_r are the antenna heights, L is the loss factor, and n is the path-loss exponent. We set $G_t = G_r = 1$, $h_t = h_r = 1.5m$, $L = 1$, and $n = 4$ in our simulation. We assume a packet is received successfully if the received signal power is greater than the receiving power threshold. Then by using (10) and (11), we can derive the PRR at a certain distance d [49].

The MAC protocol used in the simulation is the IEEE 802.15.4 implementation in NS-2. All the results have been averaged over the number of different random topologies, and the related standard deviations are provided as error bars.

We choose three main evaluation metrics.

- *Expected Flooding Delay*: the expected delay taken to flood a packet to all nodes in the network. Considering that sleep latency dominates communication delay, this metric is calculated

as the expected working cycles (T) needed to complete flooding a packet.

- *Expected Average Delivery Delay*: the average expected delay for each node receiving the flooded packet since the sink node injects a flooding packet in the network.
- *Expected Transmission Cost*: the expected total number of data transmissions to flood a packet to the entire network. This metric reflects the energy efficiency of a flooding protocol.

6.2. Simulation results

6.2.1. Overview

Fig. 6 shows an overview of the performance comparison. We fix the node density to 200 and threshold θ to 0.3, respectively. The results are depicted as cumulative distribution functions calculated based on the results from 100 runs with different random topologies.

The expected flooding delay and average delivery delay per node are illustrated in Fig. 6(a) and (b), respectively. MDET shows a very close performance to the HOP-SPT and HEOT. The flooding delay of MDET is slightly increased compared with ETX-SPT. The reason is that when upstream nodes adjust the tree structure, downstream nodes are not aware of the potential increase of $pETX$. Consequently, the $pETX$ of downstream nodes may exceed the local maximum when switching to a neighbor with a larger $pETX$. Cen-MST neglects the hop distance from the sink node, it performs poorly in terms of the flooding delay. In Fig. 6(b), it shows the similar results as in Fig. 6(a), MDET achieves a comparable delivery delay with HOP-SPT, ETX-SPT and HEOT.

Energy efficiency is shown in Fig. 6(c) as the expected transmission cost. Since the Cen-MST is constructed in a centralized manner, it provides the lower bound of the transmission cost. HEOT remarkably improves the energy efficiency compared with HOP-SPT, ETX-SPT and HEOT. It clearly shows that, MDET achieves a very good energy efficiency. It only introduces 10% more transmission cost than the lower bound, and outperforms the other three baseline tree structures.

6.2.2. Impact of node density

In this test, we evaluate what the impact of the node density is on the performance of different tree structures, by varying the number of nodes from 100 to 300 for random topologies, from 10×10 to 20×20 for grid topologies, and θ is set to 0.3.

Fig. 7(a) and (b) plot the flooding delay and average delivery delay per node under different node densities in random topology, respectively. The flooding delay and average delivery delay of Cen-MST increase linearly with node density increased. While for other tree structures, the results do not vary obviously, which show that the flooding delay is only related with the network diameter, having very little relevance to the network density. Fig. 8 shows the same trend as observed in Fig. 7. As the node density increases, results of Cen-MST increase correspondingly. For the other tree structures, the flooding delay fluctuates with changing node densities. From Figs. 7(c) and 8(c), it can be seen that the energy cost is directly related to the network density. MDET achieves a higher energy efficiency under different network densities.

6.2.3. Impact of threshold θ

In this test, we study the impact of link quality threshold θ used to build a tree structure, where θ is varied from 0.1 to 0.5, and the node density is set to 200 in random topology.

Fig. 9(a) and (b) depict the flooding delay and average delivery delay under different θ values. Cen-MST is less influenced by the changes of θ , since it always chooses routes with the best link qualities. For the other schemes, the expected delays are decreasing but at a slow rate.

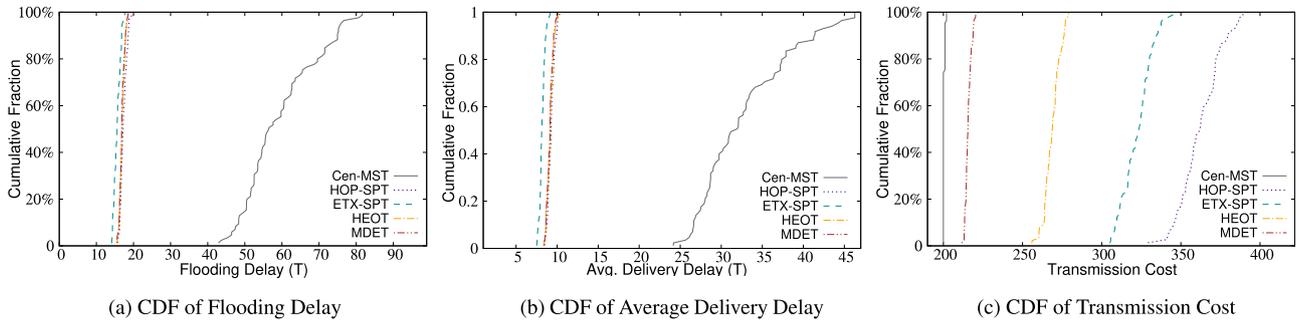


Fig. 6. Performance comparison in randomly deployed 200-node networks with $\theta = 0.3$.

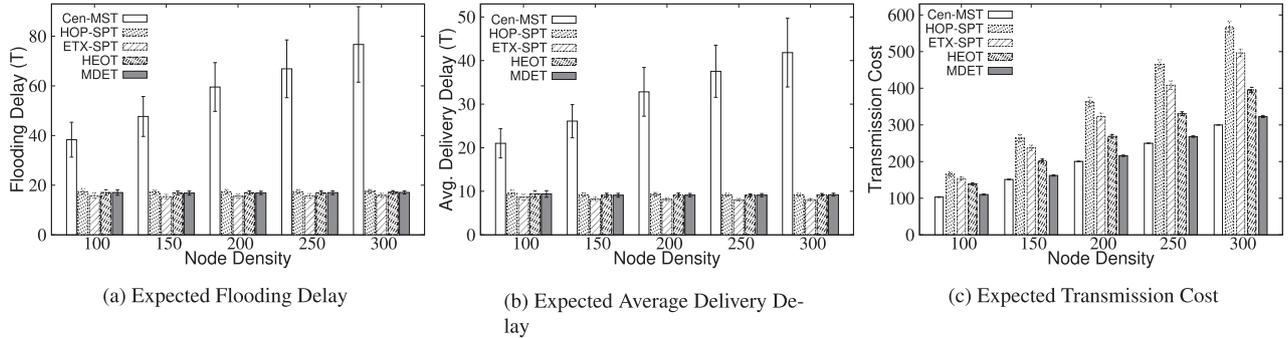


Fig. 7. Impact of node density in random topology.

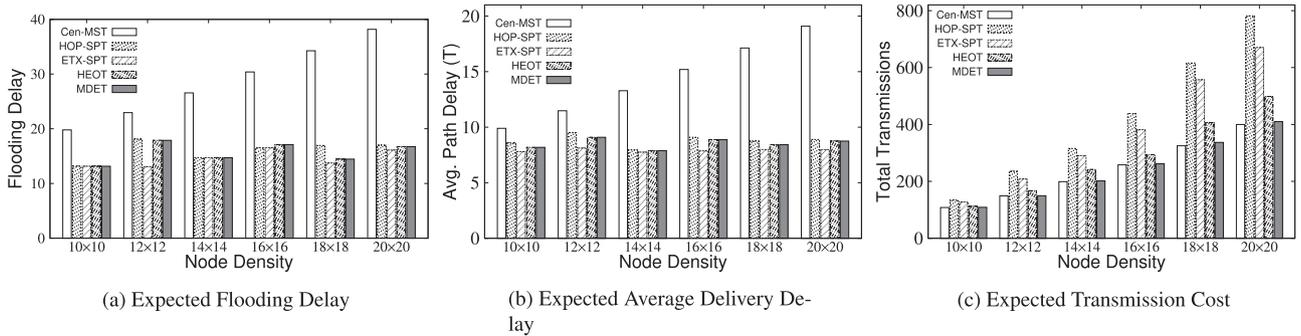


Fig. 8. Impact of node density in grid topology.

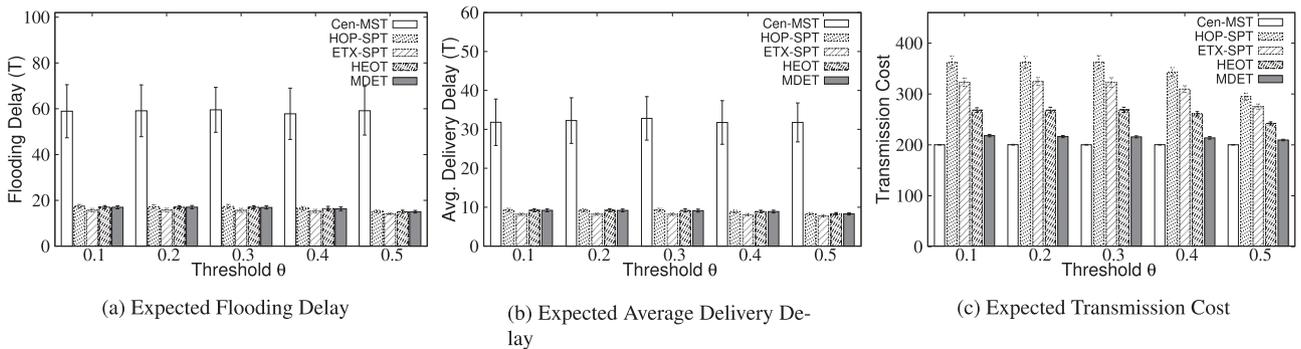


Fig. 9. Impact of threshold θ .

Fig. 9(c) plots the performance comparison on the total transmission cost. It is interesting that, the transmission cost of Cen-MST and MDET remain constant, while the results of the other schemes decrease when θ is increased. The intuition is that the average link quality increases as the threshold θ increasing, thus improving the energy efficiency. This observation also indicates that

MDET always involves good links in a tree, while HOP-SPT, ETX-SPT and HEOT may choose poor links to construct a tree.

6.3. Insights

From the simulation results, MDET shows a very good balance between flooding delay and energy efficiency. However, this im-

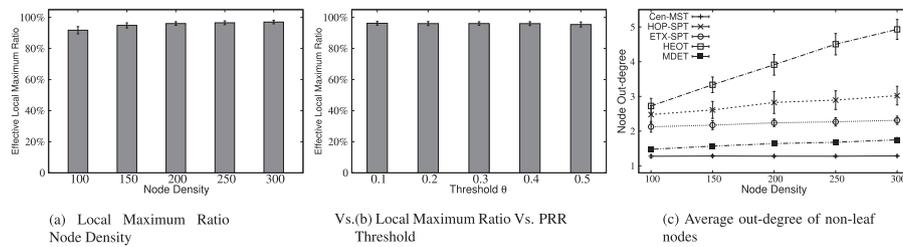


Fig. 10. System Insights.

provement comes at the cost of certain number of control message overhead. Like most optimization methods performed at the time of deployment, the design philosophy is that the high initial overhead will be eventually compensated by improved throughput and energy efficiency [50]. Therefore, the overhead incurred at the tree construction phase in MDET is expected to be amortized over a long-term of network operation.

MDET only uses the local knowledge to construct the flooding tree. To show the effectiveness of MDET, we calculate the percentage that a node considers itself having the local maximum $pETX$ (defined as the effective local maximum ratio). As shown in Fig. 10(a) and (b), we observe that only less than 5% nodes can not improve the energy optimality due to the lack of global maximum $pETX$. Besides, we also compare the average non-leaf-node out-degree (which reflects the number of child nodes that a parent node has) for different tree structures, as shown in Fig. 10(c), we find that MDET's non-leaf-node out-degree is quite low and its variance is insignificant. While the HEOT has the largest non-leaf-node out-degree. This indicates that MDET also holds a good load balancing property in flooding packets.

7. Conclusion

In this work, we investigate the flooding tree construction problem to minimize the flooding cost and flooding delay in asynchronous low-duty-cycle WSNs with unreliable links. Existing works on tree-based flooding in low-duty-cycle WSNs assume that a flooding tree has already been built, leaving the efficient flooding tree construction unaddressed. We present a distributed Minimum-Delay Energy-efficient flooding Tree (MDET) algorithm to construct an energy optimal tree with flooding delay bounding. We evaluate MDET with extensive simulations, and results demonstrate the efficiency of the proposed MDET.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grant Nos. U1713212, 61300174, 61572060, 61672349, 61602319, 61772060, CERNET Innovation Project (NGII20151004, NGII20160316), International and Hong Kong, Macao, and Taiwan collaborative innovation platforms, major international cooperation projects of colleges in Guangdong Province (Grant No. 2015KJGJH2026) and by Maoming Engineering Research Center of Industrial Internet of Things (No. 517018).

References

- [1] F. Li, J. Luo, G. Shi, Y. He, Art: adaptive frequency-temporal co-existing of zig-bee and wifi, *IEEE Trans. Mob. Comput.* 16 (3) (2017) 662–674.
- [2] L.D. Xu, W. He, S. Li, Internet of things in industries: a survey, *IEEE Trans. Ind. Inf.* 10 (4) (2014) 2233–2243.
- [3] S. Guo, Y. Gu, B. Jiang, T. He, Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links, in: *Proc. ACM MobiCom '09*, 2009.
- [4] L. Cheng, S.K. Das, J. Cao, C. Chen, J. Ma, Distributed minimum transmission multicast routing protocol for wireless sensor networks, in: *Proc. ICPP '10*, 2010.
- [5] Z. Li, M. Li, J. Liu, S. Tang, Understanding the flooding in low-duty-cycle wireless sensor networks, in: *Proc. ICPP '11*, 2011.
- [6] G. Anastasi, M. Conti, M.D. Francesco, A comprehensive analysis of the mac unreliability problem in IEEE 802.15.4 wireless sensor networks, *IEEE Trans. Ind. Inf.* 7 (1) (2011) 52–65, doi:10.1109/TII.2010.2085440.
- [7] J.Q. Li, F.R. Yu, G. Deng, C. Luo, Z. Ming, Q. Yan, Industrial internet: a survey on the enabling technologies, applications, and challenges, *IEEE Commun. Surv. Tutorials* 19 (3) (2017) 1504–1526.
- [8] Industrial sensing, <https://www.arrow.com/en/research-and-events/articles/industrial-sensing>. [Online; accessed 20-Feb-2017].
- [9] Rockwell leading way in next industrial revolution, <http://www.greatmanufacturingstories.com/news.html>. [Online; accessed 20-Feb-2017].
- [10] G. Anastasi, M. Conti, M.D. Francesco, Extending the lifetime of wireless sensor networks through adaptive sleep, *IEEE Trans. Ind. Inf.* 5 (3) (2009) 351–365, doi:10.1109/TII.2009.2025863.
- [11] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, P. Levis, Collection tree protocol, in: *Proc. ACM SenSys '09*, 2009.
- [12] M. Khan, G. Pandurangan, V.S. Anil Kumar, Distributed algorithms for constructing approximate minimum spanning trees in wireless sensor networks, *IEEE Trans. Parallel Distrib. Syst.* 20 (1) (2009) 124–139.
- [13] J. Li, A. Deshpande, S. Khuller, On computing compression trees for data collection in wireless sensor networks, in: *Proc. IEEE INFOCOM '10*, 2010.
- [14] L. Cheng, Y. Gu, J. Niu, T. Zhu, C. Liu, Q. Zhang, T. He, Taming collisions for delay reduction in low-duty-cycle wireless sensor networks, in: *Proc. INFOCOM '16*, 2016, pp. 1–9.
- [15] F. Stann, J. Heidemann, R. Shroff, M.Z. Murtaza, Rbp: robust broadcast propagation in wireless networks, in: *Proc. SenSys '06*, 2006, pp. 85–98.
- [16] L. Huang, S. Setia, Cord: energy-efficient reliable bulk data dissemination in sensor networks, in: *Proc. INFOCOM '08*, 2008, pp. 574–582.
- [17] J.W. Hui, D. Culler, The dynamic behavior of a data dissemination protocol for network programming at scale, in: *Proc. SenSys '04*, 2004, pp. 81–94.
- [18] Z. Zhao, W. Dong, J. Bu, T. Gu, C. Chen, X. Xu, Exploiting link correlation for core-based dissemination in wireless sensor networks, *SECON'14*, 2014.
- [19] L. Cheng, J. Niu, M.D. Francesco, C. Luo, S.K. Das, Y. Gu, Seamless streaming data delivery in cluster-based wireless sensor networks with mobile elements, *IEEE Syst. J.* (2015).
- [20] T. Zhu, Z. Zhong, T. He, Z.-L. Zhang, Exploring link correlation for efficient flooding in wireless sensor networks, *NSDI'10*, 2010.
- [21] J. Lu, K. Whitehouse, Flash flooding: exploiting the capture effect for rapid flooding in wireless sensor networks, in: *Proc. INFOCOM'09*, 2009, pp. 2491–2499.
- [22] F. Ferrari, M. Zimmerling, L. Thiele, O. Saukh, Efficient network flooding and time synchronization with glossy, in: *Proc. IPSN'11*, 2011, pp. 73–84.
- [23] J.E. Wieselthier, G.D. Nguyen, A. Ephremides, Energy-efficient broadcast and multicast trees in wireless networks, *Mob. Netw. Appl.* 7 (6) (2002) 481–492.
- [24] X. Shen, Y. Chen, Y. Zhang, J. Zhang, Q. Ge, G. Dai, T. He, Oppcode: correlated opportunistic coding for energy-efficient flooding in wireless sensor networks, *IEEE Trans. Ind. Inf.* 11 (6) (2015) 1631–1642, doi:10.1109/TII.2015.2436336.
- [25] G. Shuo, S.M. Kim, T. Zhu, Y. Gu, T. He, Correlated flooding in low-duty-cycle wireless sensor networks, in: *Proc. IEEE ICNP '11*.
- [26] W. Feng, L. Jiangchuan, On reliable broadcast in low duty-cycle wireless sensor networks, *IEEE Trans. Mob. Comput.* 11 (5) (2012) 767–779.
- [27] L. Cheng, Y. Gu, T. He, J. Niu, Dynamic switching-based reliable flooding in low-duty-cycle wireless sensor networks, in: *Proc. INFOCOM '13*, 2013, pp. 1393–1401.
- [28] J. Niu, L. Cheng, Y. Gu, J. Jun, Q. Zhang, Minimum-delay and energy-efficient flooding tree in asynchronous low-duty-cycle wireless sensor networks, in: *Proc. WCNC '13*, 2013.
- [29] L. Cheng, J. Niu, Y. Gu, C. Luo, T. He, Achieving efficient reliable flooding in low-duty-cycle wireless sensor networks, *IEEE/ACM Trans. Networking* 24 (6) (2016) 3676–3689.
- [30] Y. Sun, O. Gurewitz, S. Du, L. Tang, D.B. Johnson, Adb: an efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks, in: *Proc. SenSys '09*, 2009, pp. 43–56.
- [31] S. Lai, B. Ravindran, On multihop broadcast over adaptively duty-cycled wireless sensor networks, in: *Proc. DCOSS'10*, 2010, pp. 158–171.
- [32] Y. Gu, T. He, Dynamic switching-based data forwarding for low-duty-cycle wireless sensor networks, *IEEE Trans. Mob. Comput.* 10 (12) (2011) 1741–1754.

- [33] L. Xu, G. Chen, J. Cao, S. Lin, H. Dai, X. Wu, F. Wu, Optimizing energy efficiency for minimum latency broadcast in low-duty-cycle sensor networks, *ACM Trans. Sens. Netw.* 11 (4) (2015) 57:1–57:31.
- [34] L. Xu, G. Yang, L. Wang, J. Xu, B. Wang, Opportunistic broadcasting for low-power sensor networks with adaptive performance requirements, *Wirel. Netw.* (2017).
- [35] Q. Chen, H. Gao, S. Cheng, X. Fang, Z. Cai, J. Li, Centralized and distributed delay-bounded scheduling algorithms for multicast in duty-cycled wireless sensor networks, *IEEE/ACM Trans. Networking* PP (99) (2017) 1–14.
- [36] Z. Cao, D. Liu, J. Wang, X. Zheng, Chase: taming concurrent broadcast for flooding in asynchronous duty cycle networks, *IEEE/ACM Trans. Networking* 25 (2) (2017) 2872–2885.
- [37] W. Du, J.C. Liando, H. Zhang, M. Li, When pipelines meet fountain: Fast data dissemination in wireless sensor networks, in: *SenSys '15*, 2015, pp. 365–378.
- [38] W. Du, J.C. Liando, H. Zhang, M. Li, Pando: fountain-enabled fast data dissemination with constructive interference, *IEEE/ACM Trans. Networking* 25 (2) (2017) 820–833.
- [39] H. Salama, D. Reeves, Y. Viniotis, The delay-constrained minimum spanning tree problem, in: *Proc. IEEE ISCC '97*, 1997.
- [40] M. Doddavenkatappa, M.C. Chan, A.L. Ananda, Indriya: a low-cost, 3d wireless sensor network testbed, *TRIDENTCOM '11*, 2011.
- [41] R. Fonseca, O. Gnawali, K. Jamieson, P. Levis, Four-bit wireless link estimation, in: *Proc. HotNets VI*, 2007.
- [42] K. Srinivasan, P. Dutta, A. Tavakoli, P. Levis, Understanding the causes of packet delivery success and failure in dense wireless sensor networks, *RTSS'09*, 2009.
- [43] L. Tang, K.C. Wang, Y. Huang, F. Gu, Channel characterization and link quality assessment of IEEE 802.15.4-compliant radio for factory environments, *IEEE Trans. Ind. Inf.* 3 (2) (2007) 99–110, doi:10.1109/TII.2007.898414.
- [44] Telosb mote platform, <http://www.memsic.com/products/wireless-sensor-networks>. [Online; accessed 20-Feb-2017].
- [45] L. Cheng, J. Niu, Y. Gu, T. He, Q. Zhang, Energy-efficient statistical delay guarantee for duty-cycled wireless sensor networks, in: *Proc. SECON'15*, 2015, pp. 46–54.
- [46] Z. Cao, Y. He, Q. Ma, Y. Liu, Lazy forwarding in low-duty-cycle wireless sensor network, *IEEE/ACM Trans. Networking* 23 (3) (2015) 922–930.
- [47] Network simulator, <http://www.isi.edu/nsnam/ns/>. [Online; accessed 20-Feb-2017].
- [48] Prim's algorithm, (http://en.wikipedia.org/wiki/Prim's_algorithm). [Online; accessed 20-Feb-2017].
- [49] L. Cheng, J. Cao, C. Chen, J. Ma, S. Das, Exploiting geographic opportunistic routing for soft qos provisioning in wireless sensor networks, in: *Proc. IEEE MASS '10*, 2010.
- [50] I. Rhee, A. Warrier, M. Aia, J. Min, M.L. Sichitiu, Z-MAC: a hybrid mac for wireless sensor networks, *IEEE/ACM Trans. Netw.* 16 (3) (2008) 511–524.



Long Cheng is a visiting scholar at College of Engineering, Nanjing Agricultural University, China. He received his first Ph.D. degree in Computer Science from the State Key Lab of Network and Switching Technology, Beijing University of Posts and Telecommunications, China in 2012. His research interests include network security and forensics, wireless sensor networks, cyber-physical systems, mobile and pervasive computing.



Jianwei Niu received his M.S. and Ph.D. degrees in 1998 and 2002 in computer science from Beijing University of Aeronautics and Astronautics (BUAA, now renamed as Beihang University). He was a visiting scholar at School of Computer Science, Carnegie Mellon University, USA from Jan 2010 to Feb 2011. He is a professor in the School of Computer Science and Engineering, BUAA. He is an IEEE member and has published more than 100 referred papers and filed more than 20 patents. He has been on various chairs and TPC members for many international conferences. He served as the Program Chair of IEEE SEC 2008. He received New Century Excellent Researcher Award from Ministry of Education of China in 2009. His current research interests include mobile and pervasive computing.



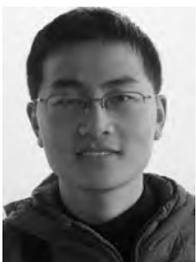
Chengwen Luo received the PhD degree from the School of Computing, National University of Singapore, Singapore. He is currently an assistant professor in the College of Computer Science and Software Engineering, Shenzhen University (SZU), China. Before joining SZU, he was a postdoctoral researcher in CSE, The University of New South Wales, Australia. He is the author and co-author of several research papers in top venues of mobile computing and WSN such as ACM SenSys, ACM/IEEE IPSN, etc. His research interests include mobile and pervasive computing, indoor localization, wireless sensor networks, and security aspects of Internet of Things.



Lei Shu is currently a professor with the Nanjing Agricultural University, Nanjing, China. He received the B.Sc. Degree in computer science from South Central University for Nationalities, Hubei Sheng, China, in 2002, the M.Sc. degree in computer engineering from Kyung Hee University, Seoul, Korea, in 2005, and the Ph.D. degree from the Digital Enterprise Research Institute, from the National University of Ireland, Galway, Ireland, in 2010. Until March 2012, he was a Specially Assigned Researcher with the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, Japan.



Linghe Kong is currently a research professor with the Department of Computer Science and Engineering at Shanghai Jiao Tong University. From 2013 to 2015, he was a Postdoctoral Fellow at McGill University and a Postdoctoral Researcher at Singapore University of Technology and Design. He received his Ph.D. degree in Computer Science from Shanghai Jiao Tong University 2012, Master degree in Telecommunication from TELECOM SudParis 2007, and B. Eng. degree in Automation from Xidian University 2005. His research interests include wireless communications, sensor networks, mobile computing, and RFID.



Zhiwei Zhao received his PhD degree in computer science from Zhejiang University in 2015. He is currently an associate professor at the College of Computer Science and Engineering in University of Electronic Science and Technology of China (UESTC). His research interests include on wireless computing, heterogeneous wireless networks, protocol design and network coding. He is a member of IEEE.



Yu Gu is currently a research scientist at IBM Research, Austin. He was an assistant professor at the Singapore University of Technology and Design between 2010–2014. He received the Ph.D. degree in the Department of Computer Science and Engineering at the University of Minnesota, 2010. He is the author and co-author of over 100 papers in premier journals and conferences. He has received several prestigious awards from the University of Minnesota.