Received May 31, 2019, accepted July 6, 2019, date of publication July 9, 2019, date of current version July 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2927627

A Fast and Efficient Broadcast Protocol With a Mobile Sink Node in Asynchronous Wireless Sensor Networks

HUI CHENG[1], LEI TAO[2,3], AND XINMING ZHANG[2,3], (Senior Member, IEEE)
[1]School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China
[2]National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China
[3]School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China

Corresponding author: Xinming Zhang (xinming@ustc.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFC0804402, and in part by the National Natural Science Foundation of China under Grant 61672485 and Grant 61379130, in part by the Open Research Fund of National Mobile Communications Research Laboratory, Southeast University, under Grant 2019D13, in part by the Anhui Initiative in Quantum Information Technologies under Grant AHY150300, in part by the Natural Science Research Projects of Higher Education in Anhui Province under Grant KJ2018B04, and in part by the Scientific Research Foundation of Education Department of Anhui Province under Grant KJ2018A0431.

ABSTRACT We investigate a broadcast scenario that a mobile sink disseminating broadcast packets while moving in asynchronous wireless sensor networks (WSNs), to prolong the network lifetime and reduce the broadcast delay. Such a scenario may yield multiple logical sink nodes in WSNs and the moving speed and trajectory of the mobile sink have the significant impact on the speed and efficiency of the broadcast. Thus, the conventional broadcast protocols may not fit for the mobile sink scenario. We propose a fast and efficient broadcast (FEB) protocol with a mobile sink in asynchronous WSNs. We analyze the moving pattern as well as the travel speed of the mobile sink. We jointly consider the broadcast process, the node location, and the neighbor coverage information within two-hops and then propose an efficient broadcast scheme together with a heuristic moving strategy for the mobile sink. The proposed broadcast scheme needs to exchange node coverage information before the transmission begins, thus to avoid redundant packet transmissions, and the proposed moving pattern strategy decreases the network broadcast delay by moving the mobile sink to monitoring areas, where fewer nodes are covered by the broadcast process. The simulation results show that our proposed protocol significantly reduces the broadcast delay and energy consumption.

INDEX TERMS Asynchronous wireless sensor networks, broadcast protocol, mobile sink, path planning.

I. INTRODUCTION

Broadcast is a fundamental service in wireless sensor networks (WSNs), in which sensor nodes deliver messages across the whole network, and it serves a wide range of higher-level operations such as networking configuration, data collection, and event observation. The sensor nodes in WSNs are usually powered by energy-limited batteries, which are hard or costly to replace and, thus, energy efficiency is always a primary concern in WSNs. Efficient sleep/wake scheduling schemes [1]–[3] have been proposed for WSNs in order to save energy and prolong the network lifetime. With sleep/wake scheduling, sensor nodes adopt duty-cycled schedules to wake up (turn on the radio) to transmit packets, and go to sleep (turn off the radio) when there is no packet to forward. Broadcast in a duty-cycled WSN is different from that in a convectional wireless network without sleep/wake scheduling, a sensor node need to utilize several unicasts to transmit a broadcast packet to each specific neighboring node to ensure all neighbors can receive the broadcast packet, since neighbors may not be awaking at the same time, especially in low-duty-cycled WSNs. In asynchronous WSNs, nodes adopt independent sleep/wake schedules and can achieve better scalability compared to a synchronous WSN without the burden of periodical synchronization.

Existing broadcast protocols in asynchronous WSNs can be classified into two categories. The first category is backbone based protocols such as tree based broadcast protocols with local cooperation mechanisms. [6], [7] can be classified into this category. Each node obtains a parent node except the root of the tree, e.g., the sink node, which is also the source of broadcast packets. Nodes broadcast packets to their dedicated children nodes to avoid redundant transmissions. And another category is broadcast status and local

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

information based protocols, which was first proposed by Sun *et al.* [8] and further improved in [9], [10]. Forwarders nodes which receive broadcast packets coordinate with each other based on broadcast progress information and delegate the uncovered neighbors to the covered ones with better link qualities.[1]

However, existing sturdies have not exploited the benefit of sink mobility technique [5] on broadcast in asynchronous WSNs. Sink mobility enables the sink node to move around in the area of monitoring and to the vicinity of sensor nodes, which have been proven an energy-efficient method in data dissemination [22], [23] to mitigate the hot-spot problem [4]. Similarly, broadcast can benefit from sink mobility since a mobile sink can generate many sub-sink nodes, which means the sensor nodes that directly receive broadcast packet from the mobile sink, in the way of moving. These sub-sink nodes help to re-broadcast their received packets from the mobile sink to the whole network, and thus, can accelerate the broadcast process to a great extent.

Nevertheless, sink mobility incurs several challenges for the design of fast and efficient broadcast protocols in asynchronous WSNs. The first challenge is to improve the energy efficiency of broadcast. A mobile sink can generate information-isolated forwarders, which represent nodes that do not know about their state information in terms of the broadcast progress. Thus, an uncovered node can be delegated twice independently by two forwarders. In the worst case, both forwarders broadcast packets to this uncovered node, resulting decreased broadcast efficiency. Another challenge in mobile broadcasting is how to promote the speed of broadcast. On the one hand, if a mobile sink travels too slowly, it can not accelerate the broadcast process. On the other, if the speed of a mobile is too fast, nodes may miss the chances to receive the broadcast packet directly from the mobile sink itself due to their low duty-cycles. Thus, the traveling route and speed of the mobile sink should be carefully designed.

In order to achieve fast and efficient broadcast while reducing broadcast delay and prolonging the network lifetime in a low-duty-cycle WSN with the mobile sink, we propose a fast and efficient broadcast protocol in this paper. The main contributions of this paper include:

- We propose a fast and efficient broadcast (FEB) protocol for mobile sink aided asynchronous WSNs. We investigate the proper traveling speed and moving patterns of the mobile sink. The lower and upper bound of the moving speed is theoretically analyzed to guarantee that the mobility can accelerate the broadcast process.
- To mitigate the redundant transmission problem incurred by the mobile sink, we optimize the multi-hop status-based FEB protocol so that the proposed protocol can be applied to the mobile scenario. We add the local broadcast state information to the beacon packet as a footer to avoid redundant transmissions.
- We devise several mobility patterns for the mobile sink, including a simple straight-line mobility pattern and a dynamic mobility pattern. In the dynamic mobility pattern, in order to decrease broadcast delay, the mobile sink chooses one next moving target sensor node at each stage based on the broadcast process information through information exchange with the current receiver. Then the mobile node prefers to move to monitoring areas with less uncovered sensor nodes. The possibility of redundant transmissions and collisions can be reduced with the proposed dynamic mobility pattern.

The rest of this paper is organized as follows. In Section II, we introduce the related work of this paper. In Section III, we describe a network model with underlying assumptions. In Section IV, we investigate the traveling speed and patterns of a mobile sink and then design the detailed FEB protocol in asynchronous duty-unaware WSNs. The simulation results are shown in Section V. We conclude this paper in Section VI.

II. RELATED WORK

A number of studies have been conducted on how to design broadcast protocols with single static sink. One category is status-based broadcast protocols achieved through multiple unicasts. Since a sensor node independently sleeps or wakes up according to its own schedule in asynchronous low-duty-cycle WSNs, neighboring nodes do not always wake up at the same time to receive broadcast packets and, thus, the broadcast nature of wireless channel cannot be fully utilized. Sun *et al.* [8] proposed a broadcast protocol integrated with the MAC layer named the asynchronous duty-cycle broadcasting (ADB). A node usually can be covered by more than one neighbor nodes. Then the ADB chooses a proper sender among the potential senders with a better link quality to avoid transmissions through bad quality links to reduce the possibility of collisions and redundancy. The ADB achieves this by delegating a communal uncovered neighbor node to the potential sender with a better link-quality in a triangle topology. However, the ADB does not operate well in sparse networks because it can only solve a multi-sender issue in triangle topology. Jang *et al.* [9] proposed an efficient multi-hop broadcast protocol for asynchronous (EMBA) WSNs to improve the performance of the ADB. With two-hop information, the EMBA adopts forwarder's guidance and overhearing of broadcast messages and ACKs. These two techniques help to solve the collision problem of the ADB in quadrangle topologies and further reduce the number of redundant transmissions. These protocols use unicast to replace broadcast for multi-hop broadcast in the entire network. However, the unicast approach generates too many transmissions and the characteristics of wireless channel have not been utilized at all. As a result, it causes a lack of efficiency and wastes the energy especially in large-scale networks or in delivering large chunks of data to the entire network such as code update, because the transmission cost dominates the total energy consumption in these application scenarios. Zhang *et al.* [10] proposed a coverage

[1] An uncovered/covered node means a sensor node that haven't/have received the broadcast packet from other sensor nodes.

efficient-based broadcast (CEB) protocol for asynchronous WSNs. CEB focuses on the coverage efficiency of a broadcast process. Considering the coverage area and coverage order of a node, a novel transmission/retransmission strategy is proposed. In dynamic delegation-based efficient broadcast protocol (DDEB) [13] for asynchronousWSNs, an uncovered node is delegated to a senderwith the best link quality. DDEB adapts to the dynamic change of broadcast process and therefore can avoid ineffcient transmissions and conicts in quadrilateral topologies.

Another category of broadcast protocols achieved by unicast is backbone-based protocols. Several asynchronous duty-cycling broadcast protocols have been proposed that adopted global guidance to reduce the number of collisions and redundancy, such as a minimum spanning tree [7], an energy-optimal tree [11], and a minimum-delay yet energy-efficient flooding tree [12]. Note in these scenarios with single sink, the sink is regarded as the root of the tree. Niu *et al.* [7] formulated an undetermined delay-constrained minimum spanning tree (UDC-MST) problem where the delay is known as a posteriori to construct a minimum-delay and energy-efficient flooding tree. They designed a distributed heuristic algorithm to solve the UDC-MST problem. Guo *et al.* [11] constructed an energy-optimal tree, where they probabilistically forward the broadcast along the tree by utilizing links with best link qualities, to reduce the energy consumption. The opportunistic packet forwarding via links outside the tree structure can also help to reduce broadcast delay according to the delay distribution of nodes in the next-hop. Cheng *et al.* [12] proposed a dynamic switching flooding framework where nodes make switching decisions when encountering transmission failures, on the basis of a pre-constructed, dynamically adjusted flooding tree. Backbone-based protocols are based on the global guidance, which is built at the beginning with static spatial information. However, it is not practical and can be versatile in practice due to the sleep/wake schedules and energy exhaustion for nodes. Moreover, the link quality can rapidly fluctuates [14] over time. Though there are several schemes to conduct local repair on the global guidance tree such as probabilistic forwarding, the construction and maintenance of the guidance tree are also costly.

In order to overcome the disadvantages generated through pure unicast, several protocols are devised that utilize the broadcast nature of wireless channel. Guo *et al.* [15] proposed a flooding-tree-based correlated flooding scheme. Highly correlated nodes under the same parent are assigned to a common sender and the reception of the broadcast packet is acknowledged with a single ACK sent by the node with the worst link-quality in the correlated nodes. This scheme ameliorates the ACK implosion problem and reduces the number of transmissions on both broadcast packets and ACKs. However, this approach needs to simultaneously schedule the wake up of nodes under the common parents. Lai *et al.* [16] presented a hybrid-cast scheme. It achieves a less number of transmissions via delivery deferring and online forwarder selection, however, since it deals with all the nodes in the same way, the nodes located in key positions may be deferred for transmission. Thus, a decrease in the number of transmissions yields longer broadcast delay. Xu *et al.* [17] transformed a broadcasting problem into a latency-optimal group Steiner tree problem on a spatio-temporal relationship graph. Then they designed a broadcast algorithm, whose key idea is to postpone the wake-up slots of the early wake-up nodes to overhear the broadcast packets with their late wake-up neighbor nodes. Wu *et al.* [18] proposed a delay-aware energy-optimized flooding (DEF) algorithm. A flooding tree is globally adjusted to maximize the energy efficiency while satisfying a delay constraint. In [12], Cheng *et al.* proposed a dynamic switching-based reliable flooding (DSRF) scheme by adjusting the flooding tree dynamically based on successful packet reception ratio. A parent node can delegate a forwarding task to a sibling node of a child node if this delegation can save energy.

A number of researches attempted to divide the time interval into multiple equal-length slots to investigate the broadcast problem [12], [17], [18]. Xu *et al.* and Cheng *et al.* [12], [17] proposed algorithms to change the active slots for nodes to promote the efficiency of the broadcast. Cheng *et al.* [12] proposed to make dynamic switching decisions making when encountering a transmission failure, and dynamically adjusts a flooding tree structure for energy saving and delay reduction. Xu *et al.* [17] found that letting some early wake-up nodes postpone their wake-up slots to overhear broadcasting messages from its neighbors can help reduce the number of transmissions. The problem of postponing wake-up slots is proven to be NP-hard and an approximation algorithm was proposed to solve the problem. Wu *et al.* [18] globally adjusted a constructed flooding tree, to improve the energy efficiency while guaranteeing the delay constraint, and proposed a delay-aware energy-optimized flooding algorithm.

The authors in [19], [20] introduced sink mobility to the scenario of data collection in a WSN. In our previous work [21], we investigated how to leverage sink mobility together with network coding technique to benefit broadcast in a duty-cycled WSN. In [21], Yan *et al.* proposed a network coding based analytical model and a dynamic sink trajectory planning algorithm, which can significantly improve the speed and efficiency of flooding with a schedule tree. Different from the aforementioned studies, we investigate the principles of a fast and efficient broadcast protocol in an asynchronous WSN with the help of a mere mobile sink.

III. MODELS AND ASSUMPTIONS

Before presenting our fast and efficient broadcast protocol with a mobile sink in detail, we first describe the system model adopted and underlying assumptions of this paper.

A. NETWORK MODEL

Suppose there are N sensor nodes in a network. Our design is based on duty-unaware asynchronous WSNs, in which each sensor node adopts an independent duty-cycle schedule and sleeps or wakes up according to its own schedule. Each node does not know the duty-cycle schedules of other nodes. $G(t) = (V, E(t))$ denotes the network at time t, where V is the set of sensor nodes at time t with a size of N and $E(t)$ is the set of edges of the network at time t. Edge $E_{(i,j)}$ represents that node i and node j are active and can communicate with each other at time t.

The wireless links are unreliable and can rapidly fluctuate in some applications [14]. The transitional region phenomenon, caused by a lossy nature of wireless channel, was studied by the previous empirical work [24]. A transmission over a lossy link may not always be successful, using the empirical results in [24], the packet reception ratio (PRR) at distance d is expressed as:

$$PRR(d) = (1 - \frac{1}{2}\exp^{-\frac{\gamma(d)}{2}\frac{1}{0.64}})^{8f}, \quad (1)$$

where $\gamma(d)$ is the signal-to-noise ratio (SNR) at distance d and f is the frame length (in bytes). An estimated transmission count (ETX) [25] is used to estimate the link quality between two sensor nodes and we assign their link qualities from 0 to 1 (thus, ETX from 1 to infinity) using a two-way ground reflection radio channel model. $lq(S, R)$ denotes the link quality between node S and node R. Due to the network dynamic caused by varying link qualities, only one-hop information is directly used in this paper. We refer to a node as the one-hop neighbor of node S if it can communicate with S directly. Each node maintains a one-hop-neighbor table $N(S)$, which contains the IDs of one-hop neighbor nodes, the link quality to a specific node and the coverage status of neighbor nodes. $N_c(S)$ and $N_u(S)$ denote the set of neighbors of node S, covered and uncovered, respectively. $N(S) = N_c(S) + N_u(S)$. Moreover, the neighbor node information of a node is proactively updated during the broadcast packet transmission process.

We utilize the modified RI-MAC (Receiver-Initiated MAC) [26], [27] as our underlying MAC protocol. When a sensor node changes its state into an active state, it sends a beacon to its neighbor nodes indicating it is awake now and can receive a data packet. If there exists a sender node staying awake waiting for forwarding the broadcast packets to its neighbor nodes when they wake up, it transmits the data packet to its neighbor nodes right after it receives a beacon packet, otherwise, the beacon-sender changes into a dormant state again.

B. SINK MOBILITY MODEL

The mobility patterns of the mobile sink can be classified into two categories: a predictable mobility pattern and a dynamic mobility pattern. Predictable mobility indicates that a mobile sink moves along a predetermined path like a straight-line or in a circle within the monitoring area, which we call straight-line mobility and circular mobility, respectively. Predictable mobility is easy to configure, however, has its own limitations since it can not adapt to the dynamic changes of network conditions, including link qualities changes and broadcast status changes. Dynamic mobility means a mobile sink can dynamically adjust its moving speed and trajectory on the way of traveling. With the overhead of information exchanging with nearby sensor nodes on the way of moving in terms of the real-time broadcast status and link information, the mobile sink is able to make better decisions on the planning of moving path. We will study the influence of both mobility patterns on the performance of broadcast protocols in this paper. Note that in order to successfully deliver a broadcast packet from the mobile sink to a sub-sink node, the mobile sink needs to first wait for a neighbor node to wake up and then send the broadcast packet to this node. The mobile sink keeps moving and stops when it receives a beacon request packet from a sensor node to cover itself.

IV. DESIGN OF THE PROPOSED FEB PROTOCOL

In our paper, we consider how to reduce the number of transmissions, which compromises the main part of energy consumption [18], and broadcast latency for broadcasting a single packet with the help of the mobile sink. The broadcast latency/delay is defined as the time duration from a broadcast packet is generated from a sink node until it is delivered to all of the sensor nodes in the network.

A. AN ANALYTICAL MODEL FOR THE MOVEMENT OF THE MOBILE SINK

When a mobile sink node moves in the area of monitoring while broadcasting packets, it generates a number of sub-sinks. The sub-sink nodes are generated along the trajectory of the sink node and spread their received broadcast packets to sensor nodes located at different directions in the network to decrease the broadcast latency. Fig. 1 demonstrates two cases where the mobile sink travels along a straight-line and a circular trajectory, respectively. The solid arrows in the network indicate the possible propagation directions of the broadcast packets when the mobile sink moves along its trajectory. The broadcast based on these poly-directional sub-sink nodes can cover the network more quickly than a single-source static sink. We next analyze the feasible moving speed of the mobile sink. The moving speed of the mobile sink should meet the following two constraints with various moving strategies:

- Expected lower bound: The time duration that the mobile sink moves along its route should be shorter than the broadcast delay of a broadcast process completed by free dissemination with only a static sink node in the network. If the broadcast has already been completed before the mobile sink node finishes moving, it represents that the mobile sink can bring no merit to the broadcast process. The broadcast delay is the time duration that a packet is forwarded from a sink node to the last node in the network, within a maximum hop count mxh. The mxh value can be obtained in the experiment with a randomly generated uniform topology. We assume that each node waits for a duty-cycle for the next hop to wake up and the links are regarded to

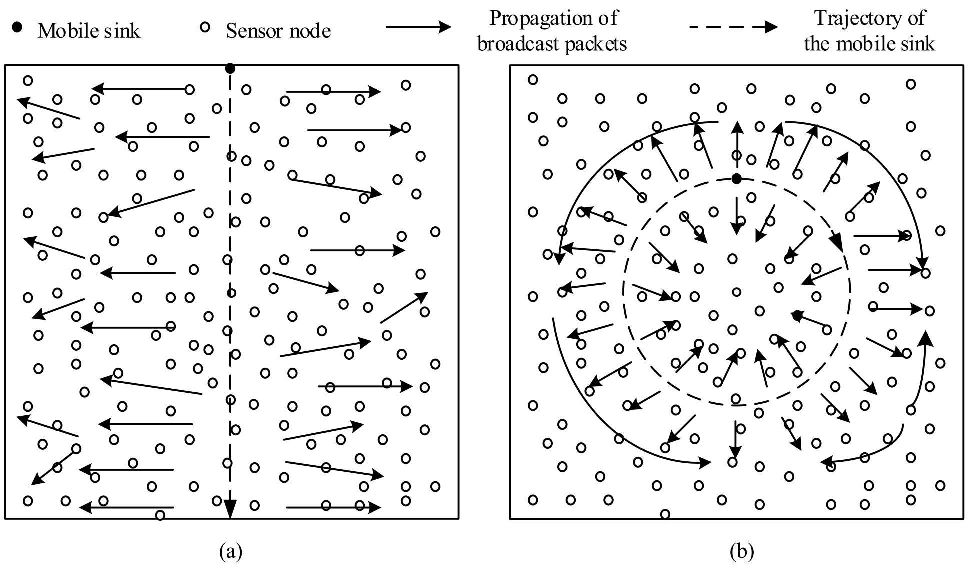


FIGURE 1. **An example for a mobile sink to move around in the monitoring area. (a) A straight-line trajectory. (b) A circular trajectory.**

be perfect in the path to the last node. Then the expected free dissemination broadcast delay (ED_{fd}) is expressed as follows:

$$s/v < ED_{fd}, \tag{2}$$

where s denotes the length of the total path. v is the speed of the mobile sink.

$$ED_{fd} = (t_{off} + \frac{L_{data} + L_{ACK}}{b_{data}}) \cdot mxh, \tag{3}$$

where L_{data} and L_{ACK} represent the size of broadcast packet and ACK packet, respectively. t_{on} and t_{off} denote the active time and the dormant time of a duty-cycle, respectively. b_{data} is the data rate. The expected lower bound of the traveling speed v_{el} of the mobile sink is expressed as:

$$v_{el} > \frac{s}{(t_{off} + \frac{L_{data}+L_{ACK}}{b_{data}}) \cdot mxh}. \tag{4}$$

- Expected upper bound: The time duration that the mobile sink moves along its route should be larger than the time duration that at least one sensor node wakes up and becomes a new sub-sink node. Otherwise, the arrangement of the mobile sink is useless and there is no difference compared with a network with only one static sink. Moreover, the delay and energy consumption cannot be reduced effectively, because although the nodes adopt their own sleep scheduling, the total time duration of a duty-cycle is the same. Thus, as long as we guarantee that the travel duration of the mobile sink is longer than a duty-cycle, there is at least one other node wakes up and the mobile sink covers itself.

$$s/v > t_{on} + t_{off}. \tag{5}$$

The expected upper bound of the traveling speed v_{eu} of the mobile sink node is expressed as:

$$v_{eu} < \frac{s}{t_{on} + t_{off}}. \tag{6}$$

Substitute constraints (2) to (6), we can derive that the moving speed of the mobile sink node should meet the following formula:

$$\frac{s}{(t_{off} + \frac{L_{data}+L_{ACK}}{b_{data}}) \cdot mxh} < v < \frac{s}{t_{on} + t_{off}}. \tag{7}$$

We will adjust the speed parameter within the derived range in the simulation experiments with given topologies to observe its effect on broadcast performance.

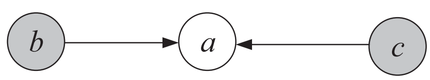


FIGURE 2. **An example of redundant transmissions in the ADB protocol.**

B. REDUCTION OF REDUNDANT TRANSMISSIONS

In the previous subsection, we introduced the straight-line moving strategy for the mobile sink node to accelerate the broadcast process. The default broadcast scheme for the mobile sink is the ADB protocol. Thus, we call the straight-line moving strategy together with the broadcast scheme the SM-ADB protocol. The SM-ADB is equivalent to extending the ADB protocol to a simple mobile scenario. However, this induces some problems for the broadcast protocol. As discussed in the previous sections, redundant transmission could happen with status based broadcast protocols [8], [9] since sensors have local and incomplete information about the real-time broadcast status of their neighbors. Take a simple example in Fig. 2, where the white node a is the uncovered node, the grey node b and node c are the covered nodes both of whom attempt to cover node a without knowing the existence of each other. Whether or not node b and node c try to cover node a within the same time interval, there must be a redundant packet transmission to node a. If both nodes b and c try to transmit a broadcast packet simultaneously, a collision will occur. If nodes b and c try to transmit a broadcast packet at different instances, a redundant packet transmission will occur. In either case, the efficiency of broadcast is decreased. The basic reason is that there is no way to pass the information that node a should be covered only once by those potential senders who do not know the existence of each other. In the original ADB protocol, such cases do not occur frequently [9] because with only one static sink, the broadcast process proceeds in the same direction. However, with an additional mobile sink as a broadcast source and sub-sink nodes that the mobile sink generates along the moving path, the broadcast process could proceed from different directions without knowing each other, causing a large amount of possible redundant transmissions. To deal with this problem, we need to improve the current status based broadcast protocols. We use the following mechanisms to reduce the probability of redundant transmission.

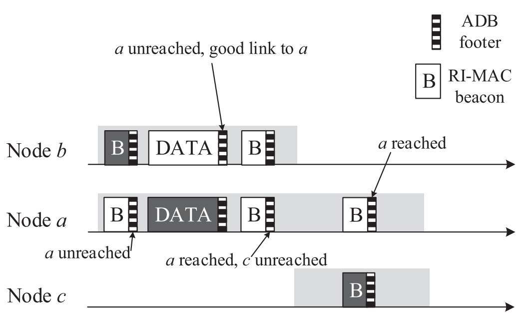


FIGURE 3. The interaction details of the example in Fig. 2.

1) FOOTER ADD-ON

Firstly, we add information about self-coverage state to the RI-MAC beacon packet as a footer to avoid redundant transmissions. Fig. 3 shows the interaction details of the example in Fig. 2. Node a wakes up and then sends a beacon to request for packets, which contains a footer stating that it is unreached by the broadcast packet. The first footer does not influence the interaction. When node a interacts with node c, however, it sends a beacon containing the information that node a has been reached, then instead of another redundant transmission process, node c directly updates the state of node a and goes to sleep. Node a also goes to sleep after a short period of time without listening from any other node.

The footer add-on allows us to make little change in the current protocol while improving the efficiency of the broadcast. On the other hand, interfering too much with the MAC layer is not good for the robustness of our proposed method. Alternatively, we can make the integrated footers which carry broadcast information to be stand-alone network layer packets instead of MAC layer packets. This causes performance degradation, although it promotes the universality of the proposed protocol.

2) OVERHEARING OF BROADCAST PACKETS

In addition, we adopt overhearing of ACK packets to avoid redundant transmissions. Remind that the mobile sink creates many sub-sink nodes, namely, forwarders, on the way of traveling. An uncovered node may receive the same broadcast packet multiple times from the forwarders from different directions without proper cooperation among them. A similar approach which has been adopted in ADB [8] with RI-MAC is used in design of the FEB protocol: after successfully receiving a broadcast packet, the node sends a message indicating that it has been delegated to potential forwarders to avoid redundant transmissions.

3) OVERHEARING OF ACKNOWLEDGEMENT PACKETS

However, collisions cannot be completely eliminated. If a collision happens, the receiver cannot receive the broadcast packet and then it goes to sleep again, which causes bad effects on both the broadcast delay and energy consumption. In this case, we adopt overhearing of ACKs and information exchange to reduce and avoid the number of collisions and redundant transmissions. When the node in active state overhears an ACK packet destined to a certain node, then it moves the nodes specified in the packet to the covered set. Moreover, nodes can exchange the neighbor coverage status information in the neighbor table when transmitting the beacon, data packet and ACK to further avoid collisions and redundancies. When a transmission fails, the sender retransmits the packet after a duty-cycle period if the receiver is not obligated to other senders.

4) ELIMINATION OF OBLIGATED SENDERS

The footer contains information about the one-hop neighbor nodes of the forwarder, however, for a node which receives or overhears a message with the footer, it can not only obtain its one-hop neighbor information, but also the two-hop neighbor information. In ADB, the node only utilizes one-hop neighbor status and link quality information. In EMBA, the node utilizes one-hop status information and one-and-two-hop link quality information. In our proposed mechanism, we use both the one-hop and two-hop status and link quality information. For one-hop neighbor nodes, the node records whether a neighbor node is either reached, delegated (to the obligated sender), or obligated. For two-hop neighbor nodes, the node only records whether a neighbor node is reached or unreached. When a node v receives or overhears a data or ACK message from forwarder w, for node u which is a one-hop neighbor node of node w and a two-hop neighbor node of node v, if forwarder w marks node u as reached, node v also marks it as reached.

When node v receives a beacon from an uncovered neighbor node u, if it marks node u as delegated, it does not forward a data message to node u. If node v marks node u as obligated, it checks whether there is a better potential sender according to the neighbor information within two hop. More specifically, node v checks the statuses and link qualities of the neighbor nodes of node u. If there is a neighbor node t which is marked as reached and the link quality $lq(t, u)$ is better than $lq(v, u)$, node v does not forward a data message even it marks node u as obligated, because node t must also mark node u as obligated. This mechanism is called elimination of obligated senders with neighbor information within two hops. With this mechanism, collisions can be greatly reduced and a potential sender with better link quality is more likely to be selected as an eventual forwarder.

C. PATH PLANNING FOR MOBILE SINK

The predetermined moving path of the mobile sink helps to disseminate the broadcast packet in a configuration-light way. However, predictable mobility does not necessarily plan the best moving path for the mobile node for broadcasting. Due to the inhomogeneity of the broadcast, the forward speed of the broadcast process could vary in different directions. If the path is predetermined, in the worst case where the moving path lies in the direction that the broadcast process advances the fastest, the speed of the mobile sink is lower than the broadcast speed of the static sink and, thus, the mobile sink can not bring merit to the broadcast process. To prevent such situation from occurring, the planning of the moving path for the mobile sink is essential so that it can dynamically adapt to the real-time broadcast process of the network.

We propose a stepwise method to plan the path of the mobile sink, which follows the positions of a series of sensor nodes. Starting from the location of the static sink, we call it the current node, at each step the two-hop neighbor broadcast states (maintained by the EMBA protocol) are analyzed: if all neighbors of current node have been reached, we stop moving because the mobile sink can not catch up with the initiative broadcast process. Else, we select a neighbor that has not been reached yet and add its position into the end of the path under planning as the next moving target. If there exist multiple neighbors that have not been reached, we compare the number of two-hop unreached nodes that are neighbors of these unreached nodes, which is similar to the concept of growth spaces in [29], and select the one with the most unreached neighbors as the next moving target. The underlying insight is that we tend to select a path in the direction that less nodes have been covered, in which way the delay of broadcast can be further reduced by the mobile sink and redundant transmissions can be reduced as well.

Take a situation in Fig. 4 as an example of the proposed path planning scheme, a mobile sink moves from node u to node v and transmits broadcast packet to node v. Note that the grey and white nodes in the figure represent the nodes that have been covered and have not been covered, respectively. Upon arrived at node u, the mobile sink has to determine the next moving target. The candidate positions are the neighbors of node u, i.e., node a, b and c. Since node u has more than one uncovered neighbors, the mobile sink compares the

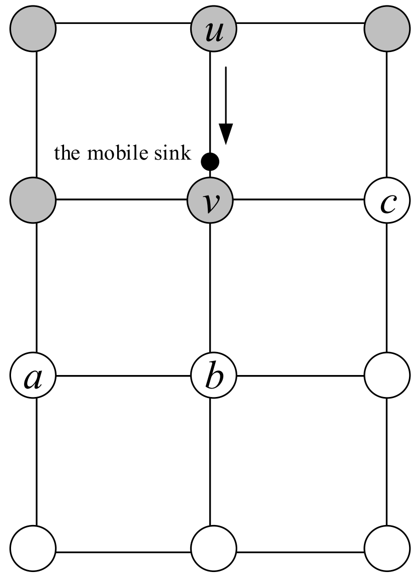


FIGURE 4. How the mobile sink selects its next moving position at a certain step.

two-hop neighbors. The growth spaces of node a is 2, including node b and the node below a. Similarly, the growth spaces of nodes b and c is 3 and 1, respectively. Thus, the mobile sink determines to select node b as its next moving target, which matches the intuitive judge of selection: node b is in the middle of multiple sets of uncovered nodes and is more possible to improve the speed of broadcast.

To guarantee that each node in the planned path can be covered by the mobile sink, there is an implicit velocity constraint to the mobile sink. The time interval that the mobile sink travels within the transmission radius of the each receiver node in the planned path should not be less than the duty cycle period of this node. Otherwise, the receiver node may not recognize the existence of the mobile sink node and, thus, can not receive broadcast packets from the mobile sender. The speed of the mobile sink should satisfy:

$$v < \frac{2R}{t_{on} + t_{off}}, \tag{8}$$

where R is the transmission range of sensor nodes. We formally describe the details of the path planning algorithm in Algorithm 1 and the communication part of the FEB protocol in Algorithm 2.

V. PERFORMANCE EVALUATION

A. SIMULATION SETTINGS

In this section we compare the performance of the proposed broadcast protocol FEB with those of the benchmarks in terms of broadcast delay and energy consumption (number of transmissions) with varying parameter settings link varying speeds, network scales and sleep intervals. The benchmarks are chosen as follows. First, we will evaluate two settings of the ADB protocol: NM-ADB and SM-ADB. NM-ADB indicates the default ADB protocol with a static sink, e.g.,

Algorithm 1 The Path Planning Part of the FEB Protocol

P: the planned path;
N_i^k: the set of k-hop neighbors of node i;
$S[i]$: the broadcast state of node i is set to 1 if node i is reached, otherwise 0;
$current = static\_sink$;
$P = \emptyset$;
// thread 1
$all\_reached = true$;
$n\_max\_unreached = MIN\_NUM$;
for each node i in $N^1_{current}$ **do**
  if $S[i] == 0$ **then**
    $all\_reached = false$;
    for each node j in $N_i^1 \bigcap N^2_{current}$ **do**
      if $S[j] == 0$ **then**
        $num\_unreached + +$;
      end if
    end for
    if $num\_unreached > n\_max\_unreached$ **then**
      $n\_max\_unreached = num\_unreached$;
      $id = j$;
    end if
  end if
  if $all\_reached = true$ **then**
    stop moving.
  end if
  $P = P \bigcup id$;
end for
// thread 2
when the mobile sink moves to a new target point p in P, update $current = p$, run thread 1.

Algorithm 2 The Communication Part of the FEB Protocol

Upon receiving the footer $F(s, S, L)$, where s, S and L represent the self-coverage state, the coverage states of neighbors and the link qualities to neighbors, respectively.
n_r and n_s represent the identifier of the receiver node and the sender node.
if current node is a receiver **then**
  for each common uncovered neighbor i of myself and the sender n_s **do**
    if the link quality from myself to i is better than that from n_s to i **then**
      obligate node i;
    else
      delegate node i to node n_s;
    end if
  end for
else
  if current node is a sender **then**
    if F belongs to a wake-up beacon **then**
      if $s == reached$ **then**
        $S[n_r] = reached$;
        return sleep state.
      end if
    else
      if F belongs to an ACK beacon **then**
        $S[n_r] = reached$;
        update the coverage states of neighbors according to S;
      end if
    end if
  end if
end if

none-mobility (NM) setting is adopted. SM-ADB is a modified ADB protocol with the help of a mobile sink that adopts simple straight-line mobility pattern. Then, we also choose a mobile broadcast protocol NCFMS in [21] that adopt network coding (NC) and mobile sink to help accelerate flooding delay. To make fair comparison, we use a modified version of this protocol called FMS that does not adopt NC technique. The topology is randomly generated and the nodes are assumed to be uniformly distributed in the square, each node randomly occupies an identical position in the monitoring area. In the simulation, the mobile sink starts from the position of the static sink and then travels along the middle-line of the monitoring area as shown in Fig. 1(a). We perform simulation for the protocols using the Castalia-OMNET++ Tools, where we set up detailed models in asynchronous duty-cycled WSNs such as physical layer transmission model, energy model, and sleep scheduling model. By default, the simulator adopts the T-MAC [34] as an underlying MAC protocol. To make a fair comparison with the ADB, we use the RI-MAC [27] which is the default MAC protocol of the ADB. Note that our proposed protocol is independent of the MAC layer and can work with either sender-based or receiver-based MAC protocols with minor modifications. The parameters of sensor nodes are taken from the CC2420 chipset datasheet [30]. The transmission range is set to 30m. The data rate is set to 250kbps. Other paraments follow the CC2420 datasheet if not specified. The packet size and ACK size are set to 100Bytes and 7Bytes, respectively. We vary the number of sensor nodes from 150 to 600, the speed of the mobile sink ranges from 10m/s to 30m/s, and the sleep interval ranges from 1s to 5s.

We use the following performance metrics to evaluate the performance of the broadcast protocols:

- Broadcast latency: The time duration from the beginning of broadcast until the time that the last node in the network receives the broadcast packet. The waiting delay induced by asynchronous scheduling and the delay due to retransmissions after collisions are the major factors of the broadcast delay.
- Number of transmissions: The number of transmissions required to broadcast a packet to all the nodes in the network. Both of the successful and unsuccessful transmissions are taken into consideration. The number of

TABLE 1. Experimental settings.

Experimental Settings	Value
Topology size	200m×200m
Frame length F	100Bytes
Transmission range	30m
Channel data rate b_{data}	250kbps
Transmitting power P_{tx}	52.2mW
Receiving power P_{rx}	59.1mW
Waiting power P_w	59.1mW
Sleeping power P_{sl}	1.278mW
Sleep interval t_{off}	1-5s
Number of nodes	150-600
Speed of the mobile sink	10-30m/s

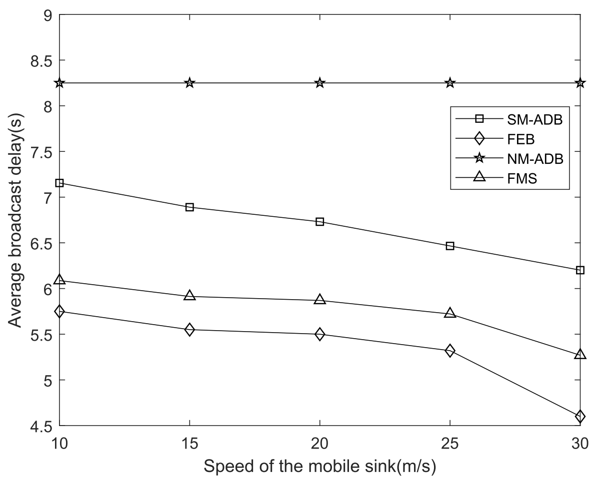


FIGURE 5. Average broadcast delay vs. mobile sink speed.

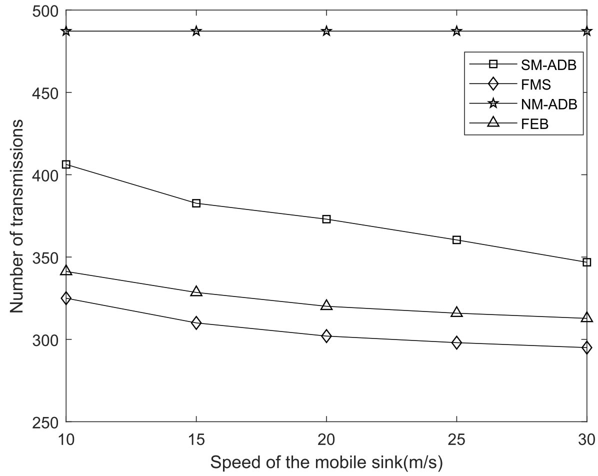


FIGURE 6. Average number of transmissions vs. mobile sink speed.

transmission of packet's control overhead is also considered.

- Path length of the mobile sink: The length of the path that the mobile sink travels from the position of the static-sink to where the mobile sink stops.

All the results are obtained from 10 different network topologies. In each network topology, we obtain the results of 100 broadcast processes and take the average values.

B. EFFECT OF THE SPEED OF THE MOBILE SINK NODE

Fig. 5 and Fig. 6 show the effect of the moving speed of the mobile sink node on the broadcast delay and the broadcast efficiency in terms of the number of transmissions to accomplish the broadcast task, respectively. The number of nodes in the network is set to 150 and the sleep interval of nodes is set to 2s. When we set the *mxh* value to 6 hops, we can derive that the upper bound and the lower bound of the speed of the mobile sink node are respectively 33.4m/s and 4.76m/s. In the experiment, we vary the speed of the mobile sink node from 10m/s to 30m/s with at a step size of 5m/s. As the mobile sink node moves, multiple sub-sink nodes appear in different positions of the network to help to disseminate the broadcast packet. The impact of the sub-sink nodes on broadcast is just like multi-sinks generating broadcast packets at different times in the network, resulting in the reduction of the broadcast delay. The NM-ADB works with a static sink as a broadcast source and performs the ADB protocol to accomplish the broadcast task, thus its broadcast delay and the number of transmissions are not influenced by the speed of the mobile sink. We can regard the NM-ADB as a special case of the SM-ADB where the speed of the mobile sink is set to zero. The delay performance of the NM-ADB is the worst compared with others because it has only one static sink node in the network, the broadcasting direction is relatively simple. The SM-ADB and FEB perform well in terms of broadcast delay. When the mobile sink moves at low speeds, the broadcast latency becomes shorter because though more sub-sink nodes are generated in the network, the sub-sink nodes are densely distributed on both sides of the path of the mobile sink node, and thus, the broadcast latency becomes shorter. As the speed of the mobile sink node increases and becomes larger than that of free dissemination, more sub-sink nodes are generated in the network in a uniform way and they initiate broadcast at different directions in the network, which helps to reduce the broadcast latency in the bottleneck area, resulting in reducing the broadcast delay. The FEB outperforms the SM-ADB because with path planning the FEB tends to choose the direction where less nodes are covered, at the same speed with that of the SM-ADB. FEB outperforms FMS in flooding delay since FMS adopt a tree-based broadcast scheme, non-tree-based ADB has better delay performance than tree-based broadcast schemes. Compared with the benchmark protocols, the proposed FEB protocol can reduce the broadcast delay by approximately 10% to 25%.

On the indicator of the number of transmissions, in line with our analysis, the proposed FEB performs the best and the SM-ADB performs the worst. With a mobile sink to help to broadcast, the SM-ADB significantly reduces the broadcast delay. However, the SM-ADB yields a large amount of redundant broadcast packets at the same time. The original ADB protocol cannot handle well when the broadcast process comes from multiple directions to a certain uncovered node, which is the case incurred by the mobility of the SM-ADB protocol. With an increase in the speed of the mobile sink, more uncovered nodes receive broadcast packets from different directions without timely transferring their own states to those potential senders, resulting in more redundant transmissions in the network. The proposed FEB protocol handles the redundant broadcast problem well by encapsulating the self-state information into the interaction packet before the data packet transmission starts to avoid unnecessary broadcast packets. The path planning part of the proposed FEB protocol also helps to reduce the redundant broadcast by choosing less-covered paths in case that the mobile sink has not timely received and updated the information of the covered nodes. The number of transmissions of FEB decreases with an increase in the speed of the mobile sink because more separate sub-sinks allows more adequate thus better broadcast delegation decisions. FMS yields less transmission number than FEB since tree-based protocol does not need ADB's delegation control packets.

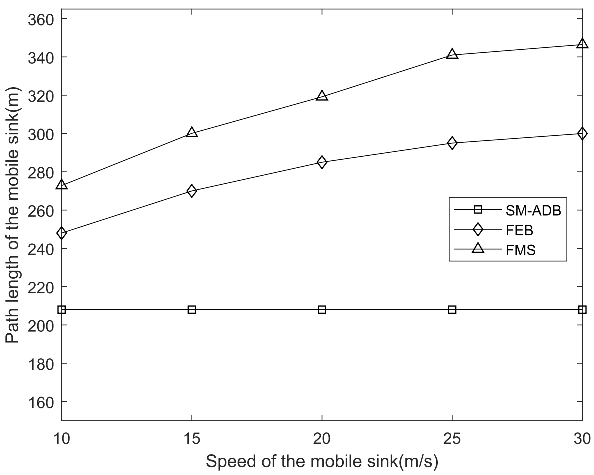


FIGURE 7. Average path length vs. mobile sink speed.

The traveling path length of SM-ADB is fixed in Fig. 7 with different speeds of the mobile sink because the mobile sink always move along the middle-line of the monitoring area and does not change when the number of the node is fixed. The path length of the proposed FEB protocol is a bit larger than that of the SM-ADB and increase slightly with an increase of the mobile sink speed probably because with a higher traveling speed, the mobile sink tends to select father neighboring node as next moving target which has more uncovered neighbors in order to promote delay efficiency, and it causes the planned path longer. Also, it can be observed that FEB has shorter traveling paths than FMS with varying speed of the mobile sink.

C. EFFECT OF NUMBER OF NODES

Fig. 8 and Fig. 9 show the effect of the number of nodes on the average broadcast delay and the average number of transmissions, respectively. The sleep interval of nodes is set to 2s and the speed of the mobile sink is set to 10m/s. As the number of nodes in the network increases, the number of transmissions and the broadcast delay for all the protocols increase because more nodes need to be covered. Since all the protocols achieve broadcast through pure unicasts, the most important factor that influences the delay efficiency and energy efficiency is the trace of the broadcast process, which is simultaneously determined by the broadcast scheme and the path of the mobile sink, if there exists the mobile sink. Note that we do not consider the influence of collisions on the broadcast due to the extreme low-duty-cycle adopted by sensor nodes. The FEB achieves delay efficiency through the carefully designed path with a help of the mobile sink and also achieves energy efficiency by improving the traditional ADB protocol to reduce the number of redundant broadcast packet transmissions. Thus, compared with the NM-ADB, the proposed FEB protocol can reduce the broadcast delay and the number of transmissions by approximately 30% and 20%, respectively.

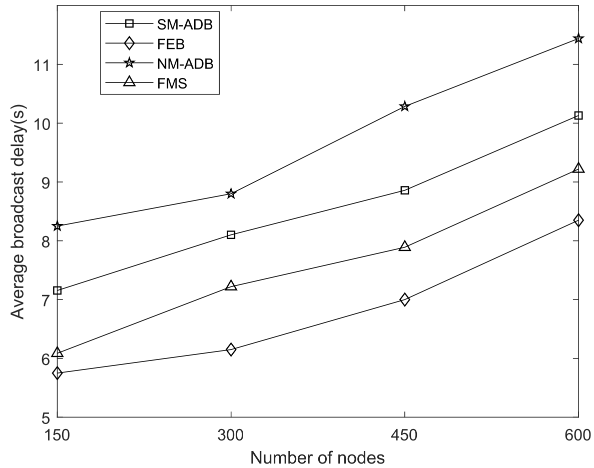


FIGURE 8. Average broadcast delay vs. the number of nodes.

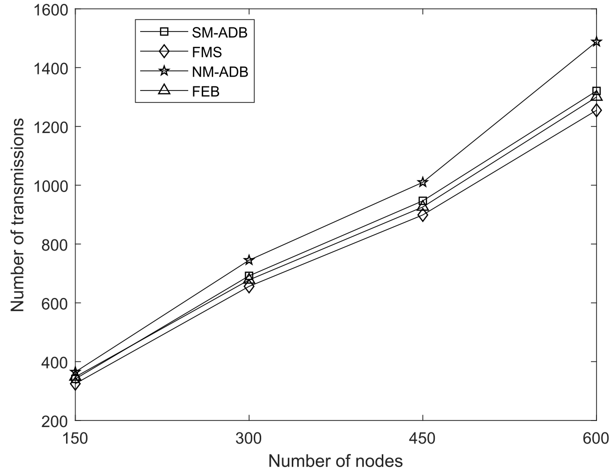


FIGURE 9. Average number of transmissions vs. the number of nodes.

Fig. 10 shows the effect of the number of nodes on the path length. In all protocols, the traveling path length increases with the number of the nodes and the path length of the

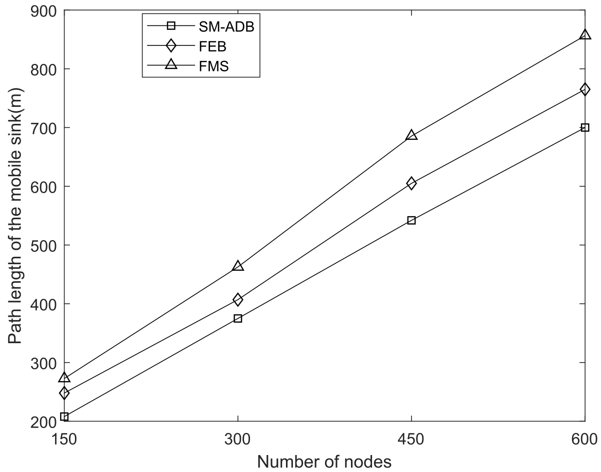


FIGURE 10. Average path length vs. the number of nodes.

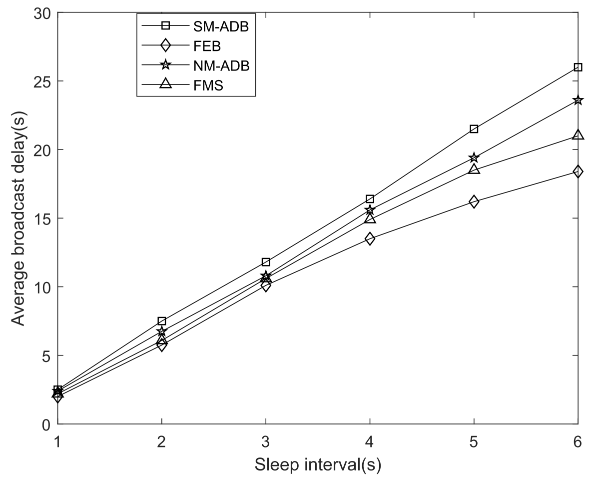


FIGURE 11. Average broadcast delay vs. sleep intervals.

proposed FEB protocol is longer than that of the SM-ADB. The FEB decrease the broadcast delay and number of transmissions at the cost of the longer traveling path length. In all evaluation metrics, the proposed FEB have better performance than FMS without network coding.

D. EFFECT OF DUTY CYCLE LENGTH

Fig. 11 shows the effect of sleep interval on the broadcast delay. The number of nodes is set to 150 and the speed of the mobile sink is 10m/s. The impact of sleep interval on the number of transmissions is not shown in the figure because sleep interval does not influence the energy efficiency without considering collisions in the paper. As the sleep interval increases, the average time to wait for a node to wake up increases and, thus, the broadcast delay of all four broadcast protocols increases. The proposed FEB protocol and the NM-ADB protocol yield the smallest and the largest number of transmissions, respectively. The SM-ADB yields a smaller number of transmissions than the NM-ADB with a smaller slop. And with an increase in the sleep interval, the slop of the FEB line becomes flat compared with the SM-ADB and FMS.

VI. CONCLUSION

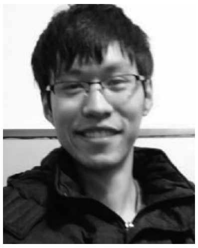
In this paper, we proposed a fast and efficient broadcast (FEB) protocol in duty-unaware asynchronous WSNs with a mobile sink node. This protocol aims to reduce the energy consumption and broadcast delay. As the mobile sink node moves along the predefined routine, nodes receiving the broadcast packet from the mobile sink node become sub-sinks to accelerate the broadcast process. Taking the broadcast process, the node location and the neighbor coverage information within two hops into consideration, we proposed an efficient broadcast scheme and two moving patterns for the mobile sink. Simulation results show that the proposed FEB protocol can significantly improve the broadcast performance in term of the broadcast delay and energy efficiency.

REFERENCES

[1] X. Wang, X. Wang, G. Xing, and Y. Yao, "Dynamic duty cycle control for end-to-end delay guarantees in wireless sensor networks," in *Proc. IEEE IWQoS*, Jun. 2010, pp. 1–9.
[2] R. Cohen and B. Kapchits, "An optimal wake-up scheduling algorithm for minimizing energy consumption while limiting maximum delay in a mesh sensor network," *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, pp. 570–581, Apr. 2009.
[3] X. Jiao, W. Lou, J. Ma, J. Cao, X. Wang, and X. Zhou, "Duty-cycle-aware minimum latency broadcast scheduling in multi-hop wireless networks," in *Proc. IEEE ICDCS*, Jun. 2010, pp. 754–763.
[4] M. Medidi and Y. Zhou, "Extending lifetime with differential duty cycles in wireless sensor networks," in *Proc. IEEE GLOBECOM*, Nov. 2007, pp. 1033–1037.
[5] X. Liu, H. Zhao, X. Yang, and X. Li, "SinkTrail: A proactive data reporting protocol for wireless sensor networks," *IEEE Trans. Comput.*, vol. 62, no. 1, pp. 151–162, Jan. 2013.
[6] X. M. Zhang, J. P. Jin, and F. Yan, "The broadcast based on optimal transmission cost tree in duty-unaware wireless sensor networks," in *Proc. IEEE WCNC*, Apr. 2013, pp. 2357–2362.
[7] J. Niu, L. Cheng, Y. Gu, J. Jun, and Q. Zhang, "Minimum-delay and energy-efficient flooding tree in asynchronous low-duty-cycle wireless sensor networks," in *Proc. WCNC*, Apr. 2013, pp. 1261–1266.
[8] Y. Sun, O. Gurewitz, S. Du, L. Tang, and D. B. Johnson, "ADB: An efficient multihop broadcast protocol based on asynchronous duty-cycling in wireless sensor networks," in *Proc. ACM SenSys*, 2009, pp. 43–56.
[9] I. Jang, S. Yang, H. Yoon, and D. Kim, "EMBA: An efficient multihop broadcast protocol for asynchronous duty-cycled wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1640–1650, Apr. 2013.
[10] X. Zhang, F. Yan, C. Li, and Q. Ding, "Coverage efficiency-based broadcast protocol for asynchronous wireless sensor networks," *IEEE Wireless Commun. Lett.*, vol. 5, no. 1, pp. 76–79, Feb. 2016.
[11] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links," in *Proc. ACM MobiCom*, 2009, pp. 133–144.
[12] L. Cheng, J. Niu, Y. Gu, C. Luo, and T. He, "Achieving efficient reliable flooding in low-duty-cycle wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3676–3689, Dec. 2016.
[13] X. Zhang, X. Jia, and F. Yan, "Dynamic delegation-based efficient broadcast protocol for asynchronous wireless sensor networks," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1195–1198, Jun. 2016.
[14] C. E. Koksal and H. Balakrishnan, "Quality-aware routing metrics for time-varying wireless mesh networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 11, pp. 1984–1994, Nov. 2006.
[15] S. Guo, S. M. Kim, T. Zhu, Y. Gu, and T. He, "Correlated flooding in low-duty-cycle wireless sensor networks," in *Proc. ICNP*, Oct. 2011, pp. 383–392.
[16] S. Lai and B. Ravindran, "On multihop broadcast over adaptively duty-cycled wireless sensor networks," in *Proc. DCOSS*, 2010, pp. 158–171.
[17] L. Xu, J. Cao, S. Lin, H. Dai, X. Wu, and G. Chen, "Energy-efficient broadcast scheduling with minimum latency for low-duty-cycle wireless sensor networks," in *Proc. IEEE MASS*, Oct. 2013, pp. 163–167.

[18] S. Wu, J. Niu, W. Chou, and M. Guizani, "Delay-aware energy optimization for flooding in duty-cycled wireless sensor networks," *IEEE Trans. Commun.*, vol. 15, no. 12, pp. 8449–8462, Dec. 2016.
[19] W. Liang, W. Xu, X. Ren, X. Jia, and X. Lin, "Maintaining large-scale rechargeable sensor networks perpetually via multiple mobile charging vehicles," *ACM Trans. Sensor Netw.*, vol. 12, no. 2, 2016, Art. no. 14.
[20] S. Guo, Y. Yang, and C. Wang, "DaGCM: A concurrent data uploading framework for mobile data gathering in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 3, pp. 610–626, Mar. 2016.
[21] F. Yan, X. Zhang, L. Tao, and H. Zhang, "Network coding-based flooding with a mobile sink in low-duty-cycle wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 8, pp. 1857–1869, Aug. 2019.
[22] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, "TTDD: Two-tier data dissemination in large-scale wireless sensor networks," *Wireless Netw.*, vol. 11, nos. 1–2, pp. 161–175, 2005.
[23] E. B. Hamida and G. Chelius, "A line-based data dissemination protocol for wireless sensor networks with mobile sink," in *Proc. IEEE ICC*, May 2008, pp. 2201–2205.
[24] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *Proc. IEEE SECON*, Oct. 2004, pp. 517–526.
[25] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Netw.*, vol. 11, no. 4, pp. 419–434, 2005.
[26] X. Fafoutis, A. Di Mauro, M. D. Vithanage, and N. Dragoni, "Receiver-initiated medium access control protocols for wireless sensor networks," *Comput. Netw.*, vol. 76, pp. 55–74, Jan. 2015.
[27] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in *Proc. ACM SenSys*, 2008, pp. 1–14.
[28] S. Upadhyayula and S. K. S. Gupta, "Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (DAC) in wireless sensor networks," *Ad Hoc Netw.*, vol. 5, no. 5, pp. 626–648, 2007.
[29] H. Dai and R. Han, "A node-centric load balancing algorithm for wireless sensor networks," in *Proc. IEEE GLOBECOM*, vol. 1, Dec. 2003, pp. 548–552.
[30] Texas Instrument. *CC2420 Data Sheet*. Accessed: Jul. 1, 2018. [Online]. Available: http://www.ti.com/product/cc2420
[31] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proc. USENIX NSDI*, 2004, pp. 1–15.
[32] F. Wang and J. Liu, "On reliable broadcast in low duty-cycle wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 767–779, May 2012.
[33] P. Kyasanur, R. R. Choudhury, and I. Gupta, "Smart gossip: An adaptive gossip-based broadcasting service for sensor networks," in *Proc. IEEE MASS*, Oct. 2006, pp. 91–100.
[34] T. van Dam and K. Langendoen, "An adaptive energy-efficient MAC protocol for wireless sensor networks," in *Proc. ACM ICENSS*, 2003, pp. 171–180.

HUI CHENG received the B.S. degree from Anhui Normal University, in 2003, and the master's degree from the University of Science and Technology of China, in 2012. He joined the School of Computer and Information Engineering, Chuzhou University, in 2003. His research interest includes wireless networks.



LEI TAO received the B.S. and Ph.D. degrees in computer science and technology from the University of Science and Technology of China, Hefei, China, in 2014 and 2019, respectively. His research interest includes wireless networks.

XINMING ZHANG (M'08–SM'18) received the B.E. and M.E. degrees in electrical engineering from the China University of Mining and Technology, Xuzhou, China, in 1985 and 1988, respectively, and the Ph.D. degree in computer science and technology from the University of Science and Technology of China, Hefei, China, in 2001. From 2005 to 2006, he was a Visiting Professor with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, South Korea. Since 2002, he has been with the faculty of the University of Science and Technology of China, where he is currently a Professor with the School of Computer Science and Technology. He is also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China. He has published more than 100 papers in wireless ad hoc and sensor networks. His research interest includes wireless networks. He won the Second Prize of Science and Technology Award of Anhui Province of China in Natural Sciences, in 2017.