Supporting Energy Efficient Broadcast with Unreliable Links for Wireless Sensor Networks

Chih-Min Chao[1] and Chao-Ying Huang
Department of Computer Science and Engineering
National Taiwan Ocean University, Taiwan

[1] E-mail: cmchao@ntou.edu.tw, Tel: +886-2-24622192 ext 6651, Fax: +886-2-24623249

Abstract

Broadcast is an essential network operation. Designing an energy efficient broadcast becomes an important issue in a wireless sensor network (WSN) because sensor nodes are battery-powered. In a low-duty-cycled WSN with unreliable links, the broadcast problem is challenging because 1) nodes may not be active simultaneously and 2) a transmission may fail to reach all the active nodes. In this paper, we define the least energy consumption broadcast problem, prove it to be NP-hard, and propose a Least-Wake-up-First broadcast scheme (LWF). By reducing the number of forwarders and the number of transmissions, LWF greatly reduces energy consumption while achieving broadcasting. Simulation results verify that LWF improves energy efficiency in duty-cycled WSNs.

Keywords: Wireless Sensor Network, Energy Efficient, Duty Cycle, Broadcast.

I. INTRODUCTION

Wireless sensor networks (WSNs) have attracted extensively attention recently. A WSN consists of a lot of wireless sensor nodes which are capable of collecting, processing, and storing environmental information. The data collected from sensor nodes are reported to one or multiple sink nodes. In general, sensor nodes are deployed in an ad hoc way, operate in a distributed manner, and coordinate with each other to accomplish a common task [22]. Typical applications for WSNs include environment monitoring, mobile object tracking, data collection, military surveillance supporting, emergency response, and navigation etc.

Sensor nodes are usually battery-powered and are not rechargeable. Therefore, how to reduce energy consumption is a critical issue in WSNs [6], [8]. A sensor node has different power consumption in transmit, receive, and idle mode. The main reason of inefficient energy usage is that sensor nodes spent too much time in idle mode. In the literature, many protocols have been proposed to reduce idle listening in order to increase energy efficiency. A common feature of these solutions is using low duty cycle wherein sensor nodes stay in sleep mode for most of the time [2], [5], [7], [11]. A node applies low duty cycle can reduce its idle listening duration and hence can reduce its power consumption.

Broadcast is an important operation in wireless networks and has been widely used in routing discovery, data collection, and information dissemination [4], [15], [16]. In a WSN, supporting broadcast in an energy-efficient way is essential because of nodes' limited energy resource. An intuitive broadcast mechanism is flooding: a node rebroadcasts every newly received broadcast packet. Such an intuitive mechanism may suffer from the broadcast storm problem which produces a lot of transmission collisions, redundant broadcasts, and energy waste. Furthermore, for a low-duty-cycled WSN, nodes may not be active at the same time and thus a node's transmission may not reach all its one-hop neighbors. That is, in a low-duty-cycled WSN, adopting the flooding mechanism is not enough to achieve a broadcast. A simple way to support broadcast in a low duty-cycled network is to enable a node to unicast to each of its neighbors separately. Such a scheme produces excessive energy consumption. Most existing broadcast schemes for low-duty-cycled WSN reduce the number of transmissions by waiting for a slot in which multiple/all neighbors are active simultaneously [17], [27], [28]. A flaw of these schemes is the long transmission delays. Another issue of existing broadcast solution is that, when calculating the energy consumption, they all omit the energy consumption when nodes are in the idle mode.

What makes the broadcast in a WSN more difficult is that wireless links are error-prone. The successful transmission rate differs for different wireless links. Existing broadcast mechanisms that consider the link quality issue often use multiple unicasts to overcome it [18], [19], [20], [13], [14], [25]. However, as mentioned earlier, such multiple-unicast mechanisms are energy-inefficient. In this paper, considering the practical situation in which links have different transmission qualities, we propose a distributed, least energy power consumption broadcast protocol for low-duty-cycled WSNs. By giving a higher priority to a node with less awake slots and reducing the number of forwarders, the proposed Least-Wake-up-First broadcast protocol (LWF) supports broadcast in an energy efficient way. Because positive acknowledges may produce the ACK implosion problem [3], in this paper, a predefined reception threshold is known to all the nodes. Each node in the network should receive a broadcast message with a probability no less than this threshold. This threshold is application-dependent: A higher value can be set if successfully receiving a broadcast message is a concern. To avoid unlimited delay, another constraint, one-hop delay bound is defined. This delay bound is set as a time constraint for a node to receive a broadcast message. This constraint is also application-dependent.

The rest of this paper is organized as follows. Literature reviews are provided in Section II. System model and problem definition are described in Section III. In Section IV, the proposed broadcast protocol is presented in detail. Performance analysis of LWF is provided in Section V. Simulation results are given in Section VI. Conclusions of this paper are in Section VII.

II. RELATED WORK

In this section, we review existing broadcast solutions for wireless networks based on the following three categories:

- Class I: Nodes without using duty cycling [21], [23], [24].
- Class II: Nodes use duty cycling [18], [19], [26], [27], [28].
- Class III: Nodes use duty cycling with unreliable links [20], [13], [14], [25].

A typical solution in Class I is flooding: Nodes rebroadcast each received unduplicated broadcast packet. This mechanism is simple but suffers from the broadcast storm problem [24]. To reduce redundant rebroadcasts, two kinds of rebroadcast methods with neighbor information, Self-Pruning (SP) and Dominant-Pruning (DP), have been proposed [21]. In SP, a sender node u piggybacks the neighbor list $N(u)$ in the packet to be transmitted. Another node v who receives the packet checks whether the set $N(v) - N(u) - \{u\}$ is empty or not. An empty set implies that all of v's neighbor nodes have received the packet and no rebroadcast is needed. Otherwise, node v rebroadcasts the packet. Nodes running DP require the information of their two-hop neighbors. A sender/forwarder will select some forwarders from its one-hop neighbors to help rebroadcast to its two-hop neighbors. Suppose that a node v receives a packet from node u and v is selected by u as a forwarder. Besides rebroadcasting the broadcast packet, node v selects its own forwarder from the set $N(v)-N(u)$ which excludes nodes that have received the broadcast packet from u. The forwarders selected by node v cover the nodes belonging to $N(N(v))-N(u)-N(v)$. Both SP and DP perform better than the flooding scheme while nodes running DP use the least number of rebroadcasts. The partial dominant pruning algorithm (PDP), an improvement of DP, is a broadcast scheme that can effectively reduce the number of forwarders [21], [23]. The basic idea of PDP is to avoid redundant forwarders by removing the common neighbors of two senders u and v during forwarder selection. That is, PDP further trims each sender v's two-hop neighbors, which in term reduces the number of forwarders of v.

Many researchers model the multihop broadcast problem as a shortest-path problem by using the time coverage graph and propose centralized and distributed algorithms belonging in Class II [26], [27], [28]. The main idea of these protocols is to find the minimum cost transmission time for each forwarder where the cost includes the number of transmission and transmission latency. A flaw of these solutions is that they find the broadcast schedule by exhausted searching which suffers from high complexity. Another class

II solution, the Hybrid Cast scheme, is proposed to reduce the number of unicasts to support broadcast [18], [19]. This scheme operates in an duty-cycled network with perfect links. The essence of the scheme is to defer a node's broadcast by one or more time slots to reach more nodes. However, the forwarder selection is quite intuitive and thus cannot find the optimal forwarders. Duc et cl. proposed broadcast mechanisms belonging to Class II to minimize the total number of transmissions [9], [10]. They proposed solutions constructed an efficient broadcast tree/backbone to reduce broadcast redundancy. A limit of these solutions is that the transmission links are considered reliable.

The opportunistic flooding protocol, denoted as OF hereafter, is a class III protocol that tries to reduce broadcast transmission delay and redundancy [13], [14]. In OF, an energy-optimal tree is constructed and a node is responsible for transmitting packets to its children nodes in the tree. A node u may also transmit a packet to its neighbor node v which is not a child node of u in the tree when v can receive the packet from u earlier than v's parent node in the tree. The OF protocol suffers from the hidden terminal problem. The overhead of the energy-optimal tree construction/reconstruction is also an issue of OF, especially when there are mobile nodes in the network. Moreover, OF achieves broadcast by a number of unicasts, which is energy-inefficient. In Deterministic Back-off Assignment and Overhearing (DBAO) [20], each sensor node requests broadcast packets from the neighbor that has the best connecting link quality. Specifically, a sensor node forms a cluster consisting of its neighbors with good link quality. A neighbor v is in node u's cluster if the link quality between u and v is larger than a predefined threshold. A node u assigns a deterministic back-off time for each of its neighbors in the cluster wherein a node v with a better link to u will be assigned a shorter back-off time. Node u will send an ACK packet as a request for receiving further broadcast packets and u's cluster members contend to broadcast to u. Nodes running DBAO suffer from extensive energy consumption for ACK packet receptions. Moreover, DBAO inefficiently achieves broadcast by a number of unicasts. The Asynchronous Duty-Cycle Broadcast protocol (ADB) [25] is also an asynchronous-duty-cycled broadcast protocol considering unreliable links. A downside of ADB is its inefficiency since the broadcast is supported by multiple unicasts.

III. SYSTEM MODEL AND PROBLEM DEFINITION

In this section, we first describe the system model and then define the problem to be solved in this paper.

TABLE I
VARIABLE LIST

Notation	Description
$N(u)$	The set of one-hop neighbors of node u.
$N(N(u))$	The set of two-hop neighbors of node u.
$q_{u,v}$	The successful transmission probability (link quality) between nodes u and v.
$k_{u,v}$	The required number of transmission from u to v to satisfy the p_T constraint.
$p_{u,v}$	The probability that node v successfully received a broadcast message from node u after $k_{u,v}$ transmissions.
p_T	The packet reception threshold.
$D_{lb,u}$	The lower bound of time span for all u's neighbors to receive the broadcast message.
$H_{k_{u,v}}$	The time needed for a node v to receive the broadcast message from u, after $k_{u,v}$ transmissions.
D_H	The one-hop broadcast delay bound.
$FL(u)$	The set of forwarders selected from node u.
C_u	The set of one-hop neighbors that u is responsible for forwarding a broadcast message.
D_{t_i}	The set of forwarders that will broadcast at time t_i.
$BS(u)$	A broadcast schedule of node u.
V_{tx,t_i}	The set of nodes in time t_i that are in transmission mode.
V_{rx,t_i}	The set of nodes in time t_i that are in receiving mode.
V_{idle,t_i}	The set of nodes in time t_i that are in and idle mode
P_{tx}	The power consumption for nodes in transmission mode.
P_{rx}	The power consumption for nodes in receiving mode.
P_{idle}	The power consumption for nodes in idle mode.
A_{t_i}	The set of nodes that are active at time t_i.
$R_{t_i}(v)$	The set of nodes that receive from node v at time t_i.

A. System Model

We consider a synchronous WSN, denoted as $G(V, E)$ where V is the set of randomly deployed sensor nodes with $|V| = n$ and E is the set of links connecting sensor nodes. Time is divided into a series of slots. Each node labels continuous slots with continuous numbers and determines its active/dormant schedule individually. A node u's active/dormant schedule is denoted as *ActiveSchedule(u)*. A node u's one-hop neighbors is denoted as $N(u)$. At the network initialization phase, a node keeps awake to broadcast its active/dormant schedule and the IDs of itself and its one-hop neighbors. During such message exchanges, a node is able to estimate the link quality to any of its neighbors. After the network initialization phase, a node is aware of the IDs and active/dormant schedules of its one-hop neighbors and the IDs of its two-hop neighbors. A node running LWF is required to know the IDs of its two-hop neighbors; however, the active/dormant schedule of any two-hop neighbor is not needed. Note that LWF is a broadcast protocol which can be built on any synchronous MAC protocol for low-duty-cycled WSNs.

B. Problem Definition

Before formally define the problem to be solved, some notations need to be described. Some important notations being used are listed in Tab I. The successful transmission probability (link quality) between nodes u and v is denoted as $q_{u,v}$. The probability that node v successfully receives a broadcast message from node u after $k_{u,v}$ transmissions, denoted as $p_{u,v}$, can be calculated by

$$p_{u,v} = 1 - (1 - q_{u,v})^{k_{u,v}}. \tag{1}$$

The packet reception threshold, denoted as p_T, is defined as the lower bound of the reception probability for a broadcast message to be considered received. That is, if a broadcast message is received by all the nodes, we have $p_{u,v} \geq p_T, \forall u, v \in V$. Given a sender u and a receiver v, the required number of transmissions $k_{u,v}$ to satisfy the p_T constraint can be calculated as follows:

$$\begin{aligned}
& p_{u,v} \geq p_T \\
& \to p_T \leq 1 - (1 - q_{u,v})^{k_{u,v}} \\
& \to (1 - p_T) \leq (1 - q_{u,v})^{k_{u,v}} \\
& \to \log(1 - p_T) \leq k_{u,v} \cdot \log(1 - q_{u,v}) \\
& \to k_{u,v} \geq \log(1 - p_T) / \log(1 - q_{u,v})
\end{aligned}$$

$$\to k_{u,v} = \lceil \frac{\log(1 - p_T)}{\log(1 - q_{u,v})} \rceil. \tag{2}$$

The lower bound of time period for all the neighbors of node u to receive a broadcast message from u, denoted as $D_{lb,u}$, is defined as the longest time span for u's neighbors to receive the message. That is,

$$D_{lb,u} = Max(H_{k_{u,v}} \mid v \in N(u), p_{u,v} \geq p_T). \tag{3}$$

where $H_{k_{u,v}}$ is the time needed for a neighbor node v of u to receive a broadcast message from u, after $k_{u,v}$ transmissions.

To limit the transmission delay, a constraint of one-hop broadcast delay, denoted as D_H is defined. A forwarder has to transmit a broadcast packet within D_H after the packet is arrived. The value of D_H is no less than the maximal $D_{lb,u}$, $\forall v \in V$, which can be obtained in the network initialization phase. To broadcast to nodes outside its one-hop range, the source/forwarder will select several forwarders from its one-hop neighbors. The set of forwarders selected from node u is denoted as $FL(u)$. For each forwarder u, the set of its one-hop neighbors that u is responsible for forwarding a broadcast message is denoted as C_u. The set of forwarders that will broadcast at time t_i is denoted as D_{t_i}. A broadcast schedule of node u, denoted as $BS(u)$, consists of a set of ordered pairs (t_i, D_{t_i}) which means that nodes belonging to D_{t_i} will broadcast at time t_i. Let V_{tx,t_i}, V_{rx,t_i}, and V_{idle,t_i} be the set of nodes, in time t_i, that are in transmission, receiving, and idle mode, respectively. Denote P_{tx}, P_{rx}, and P_{idle} as the power consumption for nodes in transmission, receiving, and idle mode, respectively. Let A_{t_i} be the set of nodes that are active at time t_i and $R_{t_i}(v)$ be the set of nodes that receive from node v at time t_i. The problem to be solved in this paper can be formally defined as follows.

Definition 1. The least energy consumption broadcast (LECB) problem with unreliable links.

Given an h-hop network G with different link quality $q_{u,v}$, $\forall l_{u,v} \in E$, find a broadcast schedule of the source u starting from time t_0,

$$BS(u) = (t_i, D_{t_i}) \mid 0 \leq i \leq m, (t_0 < t_1 < ... < t_m \leq B)$$

where $B \geq h \cdot D_H$ such that all nodes in the network receive the broadcast message and the cost of the broadcast schedule

$$C(BS) = \sum_{t=0}^{m} \qquad (P_{tx} \cdot \mid V_{tx,t_i} \mid + P_{rx} \cdot \mid V_{rx,t_i} \mid + P_{idle} \cdot \mid V_{idle,t_i} \mid)$$

is minimized with the constraint $p_x > p_T, \forall x \in V - u$, where $V_{tx,t_i} = \{x \mid x \in D_{t_i}\}$, $V_{rx,t_i} = \{x \mid x \in R_{t_i}(y), \forall y \in D_{t_i}\}$, and $\mid V_{idle,t_i} \mid = A_{t_i} - V_{tx,t_i} - V_{rx,t_i}$

Theorem 1. *The LECB problem is NP-hard.*

Proof. The Minimum Set Cover problem is proved to be NP-hard [12]. To prove this theorem, we demonstrate that the minimum set cover problem is a special case of the proposed LECB problem. Consider the special case of the LECB problem: In a network with k channels, a sender u and all its neighbors have the same duty cycle while the value of $q_{u,v}$ is either 0 or 1, $\forall v \in N(u)$ and $P_{idle} = 0$. In such a case, the LECB problem is in fact a minimum set cover problem, which proves that the our proposed problem is NP-hard.

□

IV. THE PROPOSED SOLUTION

To solve the LECB problem with unreliable links, the following two tasks must be handled: 1) " How a node selects a set of forwarders? " and 2) " How a source/forwarder node broadcasts to its one-hop neighbors? " In LWF, the mechanism to handle these two tasks is called the forwarder selection scheme and the one-hop broadcast scheme, respectively. In the following, we describe these two schemes separately.

A. Forwarder Selection

The forwarder selection scheme in LWF is based on the PDP broadcast mechanism [23]. Suppose that a node v receives a broadcast packet from node u and needs to rebroadcast it. In PDP, node v first finds the area U_v that the selected forwarders should be responsible for and then finds the next-hop forwarder from C_v = $N(v)$ - $N(u)$. Specifically, the area U_v is given by

$$U_v = \begin{cases} N(N(v)), & \text{if } v \text{ is the source} \\ N(C_v) - N(u) - N(v) - N(N(u) \cap N(v)), & \text{otherwise.} \end{cases} \tag{4}$$

With U_v, node v adds a node in C_v that covers the most uncovered nodes in U_v into its next-hop forwarder set $FL(v)$. This process is repeated until all the nodes within U_v are covered by nodes in $FL(v)$. The forwarder set of v is attached to the broadcast packet sent by v such that v's neighbors know whether they are selected as forwarders or not. Using the forwarder set selection scheme of PDP, it is possible that multiple forwarders may cover the same node. For any given node w yet to receive a broadcast packet, one forwarder is enough to achieve the broadcast task. The other forwarders that also cover node w are redundant. What is worse is that, if two or more forwarders are hidden to each other, these redundant forwarders may produce many transmission collisions due to the hidden terminal problem. To alleviate this redundant forwarder problem, LWF uses link quality information to uniquely assign a forwarder for each node not receiving a broadcast packet yet. When a node w is covered by multiple forwarders, the forwarder selection scheme of LWF selects the forwarder u that has the best link quality to w to be responsible for transmitting a broadcast message to node w. Similar to PDP, in LWF, the node assignment for each forwarder is also attached to the broadcast packets.

Fig. 1 is an example of the forwarder selection scheme of LWF where the numbers aside the links are the associated link qualities. The source node a selects nodes b and c as the forwarders while node f is

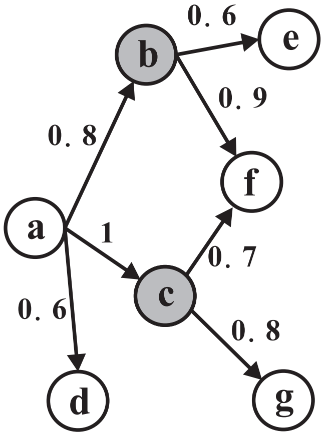


Fig. 1. Example of the forwarder selection

covered by both nodes b and c. Having better link quality, node b will be responsible for broadcasting to node f.

B. One-Hop Scheduling

In LWF, to build the one-hop broadcast schedule for a sender u, node u will first consider the one-hop neighbor i that has the least awake slots in the next D_H slots. The idea behind this mechanism is that the limited awake slots of node v are very likely to be scheduled for u to transmit such that node v can receive the packet. Considering node v first, node u has a high possibility to achieve broadcast with less number of transmissions. When node u schedules the transmissions for a particular node v, the active slot of v that has the most active nodes will be scheduled first. This enhances transmission efficiency because the number of recipients for a transmission is maximized. Another strategy in the one-hop scheduling is that u may ask some of its one-hop neighbors with better link qualities to broadcast to its isolated non-forwarder neighbors. An isolated node of u is one that shares no common active time slots with any one-hop neighbors of u. Such a strategy helps reduce the overall number of transmissions and thus the overall power consumption.

The pseudo code of the proposed one-hop scheduling mechanism is shown in Algorithm 1. The inputs of the algorithm include the source/forwarder u, the set of one-hop neighbors that u is responsible for (C_u), u's active/dormant schedule ($ActiveSchedule(u)$), link quality for any link from u to all u's neighbors ($q_{u,y}$, $v \in N(u)$), and the packet reception threshold (p_T). In lines 1 to 4, node u calculates $k_{u,v}$ and $D_{lb,u}$ based on equ. (2) and (3), respectively. The priority of each one-hop neighbor of u, based on the number

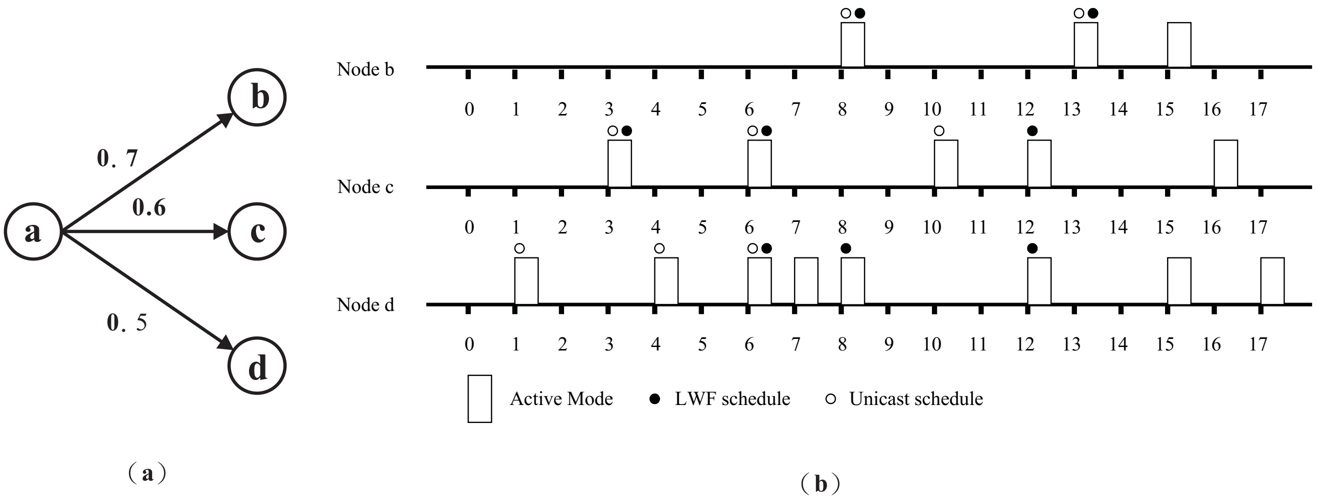


Fig. 2. An example of one-hop broadcast scheme of LWF

of awake slots, is determined in lines 5 to 7. A variable $isolated(v)$ is associated with each node v in C_u to indicate whether there is another node z in C_u such that nodes v and z are active in some time slot within D_H. The value of $isolated(v)$ is set to one if no node in C_u shares the same active time slot with v. The sequence of nodes to be considered by node u is built in line 8. The slot priority is built in line 12. In lines 13 to 33, the broadcast schedule *BS(u)* is built. For each node v being scheduled, a total of $k_{u,v}$ transmissions should be scheduled. The slots in which the most nodes are active are selected first, as shown in lines 13 to 25. If the number of such slots is less than $k_{u,v}$, the slots which no two nodes being active concurrently are selected, as shown in lines 21 to 25. The isolated nodes of u are checked for better forwarders in lines 26 to 33.

Here we use an example to demonstrate the operation of the one-hop broadcast scheme of LWF. The network topology and the link qualities can be found in Fig. 2(a) while each node's active/dormant schedule is shown in Fig. 2(b). We assume that p_T is 0.8 in this example. The source node a will first calculate the needed number of transmissions for each neighbor. In this example, $k_{a,b}, k_{a,c}$, and $k_{a,d}$ is 2, 3, and 3, respectively. Next, node a finds the value of $D_{lb,a}$. Since the value of $H^{a,b}$, $H^{a,c}$, and $H^{a,d}$ is 13, 10, and 6 respectively, we have $D_{lb,a} = \max\{H^{a,b}, H^{a,c}, H^{a,d}\} = 13$. Then, node a calculates the slot priority $PrioritySlot(a)$ (= 6, 8, 12, 1, 3, 4, 7, 10, 13). Among node a's one-hop neighbors, node b has the least awake slots within the next 13 slots so node b will be scheduled first. Based on $PrioritySlot(a)$, time slot 8 is first selected since it is the awake slot for node b with the

Algorithm 1: LWF one-hop broadcast scheme

```
input : $the\ sender\ u$, $C_u$, $ActiveSchedule(v), \forall v \in N(u)$, $q_{u,v}, \forall v \in N(u)$, $p_T$
output: $BS(u)$
for $v \in C_u$ do
    $k_{u,v} = \lceil \frac{\log(1-p_T)}{(1-q_{u,v})} \rceil$
end
$D_{lb,u} = \text{Max}(H_{k_{u,v}} \mid v \in N(u), p_{u,v} \geq p_T)$
for $v \in C_u$ do
    $\frac{1}{Priority(v)} \leftarrow$ number of awake slots in $ActiveSchedule(v)$ within $D_H$
end
Build $PendingList(u)$ which consists of $u$'s one-hop neighbors, sorted by their priorities in descending order.
for $v \in C_u$ do
    $isolated(v) \leftarrow 1$
end
Build $PrioritySlot(u)$ which consists of slots within $D_{lb,u}$ with two or more concurrent awake neighbors, sorted by the number of concurrent awake nodes in descending order.
for $v \in PendingList(u)$ do
    for $t_i \in ActiveSchedule(v)$ & $t_i < D_H$ do
        if $t_i \in PrioritySlot(u)$ & $k_{u,v} > 0$ then
            $BS(u) \leftarrow BS(u) \cup (t_i, \{u\})$
            $k_{u,v} \leftarrow k_{u,v} - 1$
            $isolated(v) \leftarrow 0$
        end
    end
    for $t_i \in ActiveSchedule(v)$ & $(t_i, \{u\}) \notin BS(u)$ & $k_{u,v} > 0$ do
        $BS(u) \leftarrow BS(u) \cup (t_i, \{u\})$
        $k_{u,v} \leftarrow k_{u,v} - 1$
    end
end
for $v \notin C_u$ & $v \notin FL(x)$ & $isolated(v) = 1$ do
    for $k \in C_u$ & $k \neq v$ do
        if $q_{k,v} > q_{k,u}$ then
            Add $v$ into $C_k$
            Delete $(t_i, \{u\})$ where $t_i \in ActiveSchedule(v)$ from $BS(u)$
        end
    end
end
```

highest priority. Time slot 13 will also be selected because it is the only candidate slot left. Node c will be scheduled next. Similarly, time slots 6 and 12 will be selected. The last scheduled transmission for node c is slot 3 which is c's highest priority active time slot. No more transmission will be scheduled for node d since node d can receive three transmissions from earlier scheduled transmissions. The final broadcast schedule of node a is thus $BS(A) = (3, \{A\}), (6, \{A\}), (8, \{A\}), (12, \{A\}), (13, \{A\})$. The transmission schedule for the earliest-slot-first unicast scheme is also shown in the figure for comparison purposes. In this unicast scheme, the sender broadcast to each neighbor in the earliest possible slots. The scheduled transmissions for node b will be at slots 8 and 13. For nodes c and d, the scheduled transmissions will be at time slots 3, 6, and 10 and 1, 4, and 6, respectively. The final broadcast schedule is $BS(A) = (1, \{A\}), (3, \{A\}), (4, \{A\}), (6, \{A\}), (8, \{A\}), (10, \{A\}), (13, \{A\})$. We can see that this unicast broadcast scheme uses two more transmissions when compared to LWF.

When receiving a broadcast packet from a node u, a node v running LWF will check if its ID is in $FL(u)$. If so, node v will execute both the forwarder selection and one-hop broadcast schemes. For each scheduled slot, to reduce transmission collisions, a sender applies the carrier sense and the random backoff schemes before its transmission.

V. PERFORMANCE ANALYSIS

In this section, given the one-hop delay bound D_H, we analyze the upper bound of the expected number of transmissions for a sender to broadcast to its one-hop neighbors. Here we consider the one-hop environment because analyzing a complete multihop environment is very difficult, if not impossible. The results of more complete and practical simulation results in a more practical multihop network will be reported in Section VI

A sender uses the maximum number of transmissions when all the neighbors are isolated. That is, each transmission reaches only one neighbor. The maximum number of transmissions k_u^{max} for sender u, from equ. (2), is given by

$$k_u^{max} = \sum_{v \in N(u)} \frac{\log(1 - p_T)}{\log(1 - q_{u,v})} \tag{5}$$

Since several neighbors of a sender may be active at the same time slot, the expected number of transmissions is usually less than k_u^{max}. The expected number of transmissions for a sender to broadcast to its neighbors can be reduced by x for each slot in which $x + 1$ nodes are active. There are many node combination in a slot for a given time span. We analyze the probability of multiple active nodes at a slot under the assumption that nodes have the same duty cycle of $\frac{1}{D_H}$. When sender u has two neighbors, the probability of both nodes being active at a particular time slot in a time span of D_H time slots is equal to

$$C_2^2 \cdot C_1^{D_H} \cdot \frac{1}{D_H} \cdot \frac{1}{D_H}$$

TABLE II

THE ANALYSIS AND SIMULATION p_{n,k,D_H} WITH DIFFEREN k VALUES VALUES (THE VALUES INSIDE THE PARENTHESES ARE SIMULATION RESULTS)

	k = 2	k = 3	k = 4	k = 5
n = 6	0.6108(0.6083)	0.0428(0.0432)	0.00169(0.00189)	0.00003562(0.00003)

TABLE III

THE ANALYSIS AND SIMULATION p_{n,k,D_H} WITH DIFFEREN n VALUES (THE VALUES INSIDE THE PARENTHESES ARE SIMULATION RESULTS)

	n = 3	n = 4	n = 5	n = 6
k = 2	0.1425(0.144)	0.2707(0.268)	0.4286(0.427)	0.6108(0.605)

When u has three neighbors, the probability that two neighbors are active and the other one is inactive at a particular slot in a time span of D_H time slots is equal to

$$C_2^3 \cdot C_1^{D_H} \cdot \frac{1}{D_H} \cdot \frac{1}{D_H} \cdot \frac{(D_H - 1)}{D_H}$$

In general, for a sender with n neighbors, the probability that k neighbors ($n \geq k \geq 2$) are active at a particular time slot in a time span D_H, denoted as p_{n,k,D_H}, is given by

$$p_{n,k,D_H} = C_k^n \cdot C_1^{D_H} \cdot \frac{1}{D_H} \cdot \frac{1}{D_H}^k \cdot (\frac{D_H - 1}{D_H})^{n-k}$$

The expected number of transmissions for a sender to broadcast to its n one-hop neighbors in a time span D_H, denoted as $E(k_u, n, D_H)$, is given by

$$E(k_u, n, D_H) = k_u^{max} - D_H \cdot \sum_{i=2}^{n} (i-1) \cdot p_{n,i,D_H} \tag{6}$$

To verify the correctness of our analysis, we have implemented a simulator with n and k in the range of [3,6] and [2,5], respectively. The comparison of analysis and simulation results of $p_{n,k}$ with $D_H = 20$ can be found in Table II and III. The simulation results are the average of 10000 simulation runs. We found that the results obtained from analysis and simulation are close, which verify that our analysis is trustworthy.

VI. SIMULATION RESULTS

We have implemented a simulator using C++ to evaluate the performance of the proposed LWF protocol. The PDP and OF protocols were also implemented for comparison purposes. We do not implement DBAO and ADB because we focus on a synchronous duty-cycled environment without acknowledgements.

TABLE IV
THE PARAMETERS SETTING OF SIMULATION

Network size	200 × 200
Number of nodes	600-900
Transmission range	10m
Duty cycle	5%-20%
Simulation runs	10
Power consumption for transmission	47 Watt
Power consumption for receiving	53 Watt
Power consumption for idle	0.054 Watt
p_T	0.8

Because nodes running PDP do not use duty cycling, we applied the first-awake-first-serve scheme as the one-hop broadcast mechanism for PDP in our simulation. For OF, we follow the original parameter settings [13], [14]. In our simulations, 600 to 900 nodes were uniformly deployed in an area of 200 m by 200 m. The transmission range of a sensor node is 10 m while the interference range is also 10 m. The one-hop delay bound is set to 50 s. A node's duty cycle is varied from 5% to 20%. The energy consumption model follows the MICAz platform [1], where the power consumption for transmission, receive, and idle mode is 17.4mA, 19.7mA, and 20 μA, respectively. The simulation parameters are also shown in Table V. Each point in the figures is the average of 10 simulation runs.

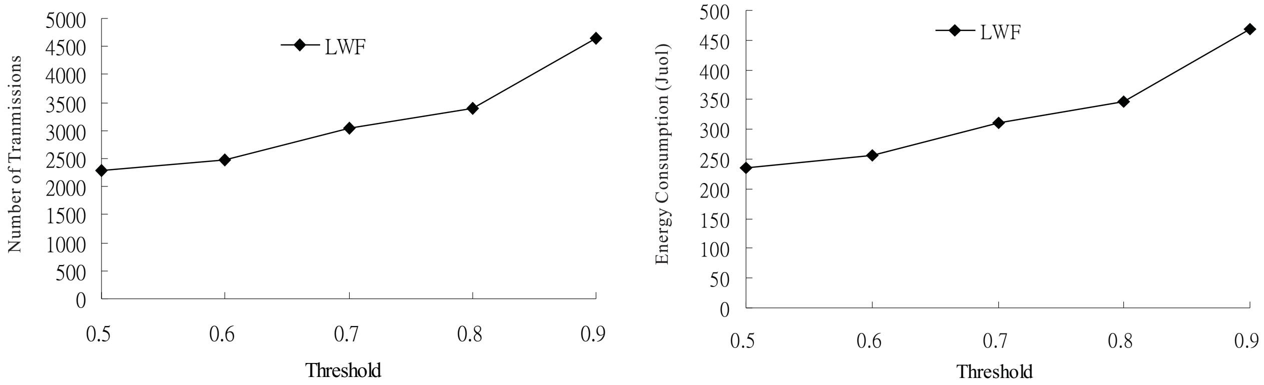
To evaluate performance, we use the following two metrics.

- 1: Number of Transmissions: The total number of transmissions to accomplish a broadcast.
- 2: Energy Consumption: The total energy consumption to accomplish a broadcast.

In the following, we made observations from three aspects.

A) Impact of p_T: In this experiment, we varied p_T to observe its effect on LWF. A total of 700 nodes were uniformly distributed in the network while each node has the same duty cycle of 5%. The results can be found in Fig. 3. The number of transmissions and energy consumption increases significantly when p_T is increased from 0.8 to 0.9. When p_T is adding by 0.1 from 0.5 to 0.9, the increased number of transmissions also increases 18%, 52%, 73%, and 155%, respectively. Similarly, the energy consumption increases 18%, 51%, 71%, and 150% when p_T is adding by 0.1 from 0.5 to 0.9, respectively. This reveals that the overhead to increase p_T increase significantly when p_T is increased from 0.8 to 0.9. To enable an energy-efficient broadcast, we set p_T to 0.8 in the following simulations.

B) Impact of duty cycle: Next, we varied the value of duty cycle from 5% to 20% in a network with 700



(a)

(b)

Fig. 3. Impact on P_T in terms of (a) the number of transmissions and (b) energy consumption.

nodes to observe the influence. The results are shown in Fig. 4. As shown in Fig. 4(a), LWF outperforms PDP and OF significantly. For all three protocols, the number of transmissions decreases as the duty cycle increases. This is reasonable since a transmission can reach more nodes when a larger duty cycle is applied. The improvement of LWF becomes larger at higher duty cycles because there are more concurrent awake neighbors at higher duty cycles and LWF can find a better schedule to accomplish a broadcast. For example, when duty cycle is 5%, the number of transmissions generated by LWF is 29% and 23% lower than that of PDP and OF, respectively. When duty cycle is 20%, the number of transmissions generated by LWF is 55% and 50% lower than that of PDP and OF, respectively. Similar trends can be found in Fig. 4(b). This verifies the benefits of the strategies of LWF (giving a higher priority to a node with less number of awake slots and reducing the number of forwarders). It should be noted that as the number of transmission getting lower, energy consumption is getting lower too. As a result, to achieve energy efficiency, broadcast protocols should be designed with less number of transmissions.

C) Impact of number of nodes: We varied the number of nodes from 600 to 900 in this experiment while each node has a duty cycle of 5%. Fig. 5 shows the comparison results. We can see that LWF still has the best performance. As expected, the number of transmissions to accomplish a broadcast increases in proportional to the number of nodes as shown in Fig. 5(a). LWF uses less number of transmissions because one transmission in LWF can reach more neighbors. The energy consumption also increases in proportional to the number of nodes as shown in Fig. 5(b). This is reasonable since nodes experience more transmissions, receptions, and idle listenings.

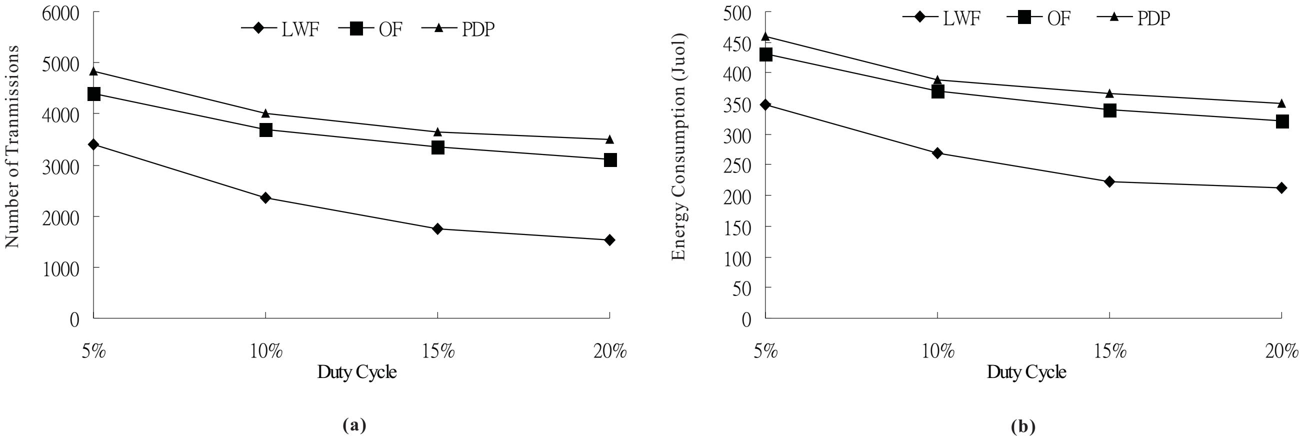


Fig. 4. Impact on duty cycle in terms of (a) the number of transmissions and (b) energy consumption.

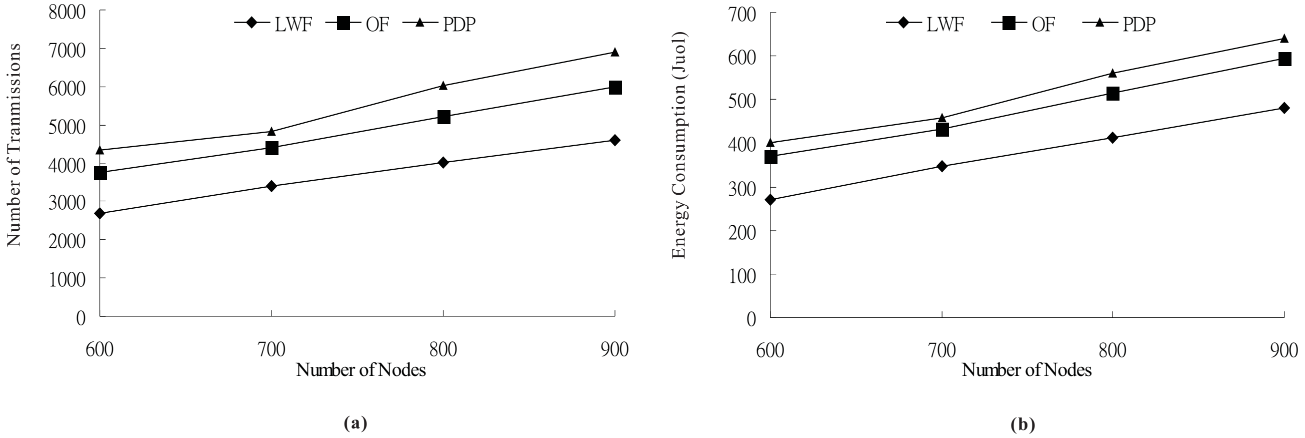
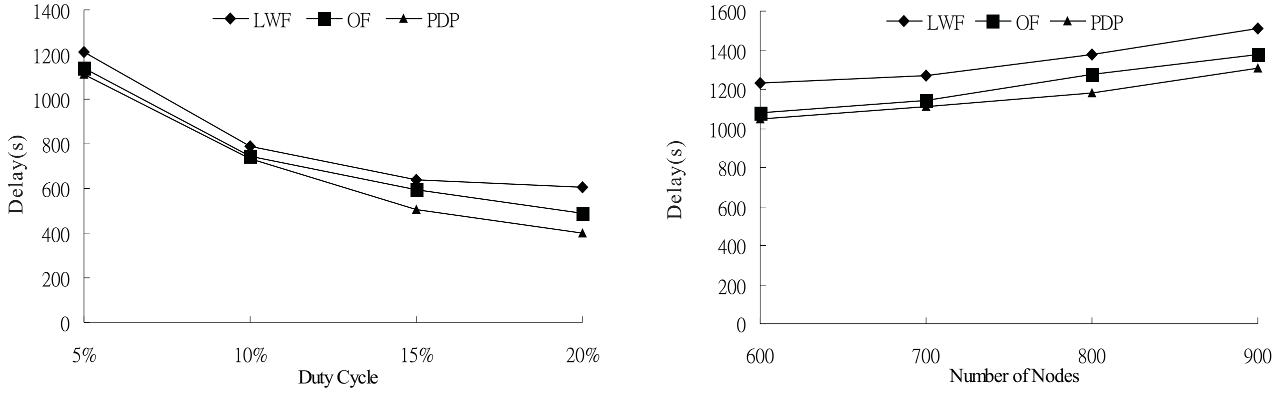


Fig. 5. Impact on number of nodes in terms of (a) the number of transmissions and (b) energy consumption.

D) Impact on delay: We have also observed the broadcast delay (defined as the time needed to accomplish a broadcast) generated by LWF, PDP, and OF protocols. The number of nodes is varied from 600 to 900 while the value of duty cycle is varied from 5% to 20%. As shown in Fig. 6(a), the PDP protocol produces the least delay which is followed by OF and LWF. It is reasonable because of the first-awake-first-serve broadcast mechanism of PDP. OF performs the second best because it enables a node u to transmit to a node v as long as v can receive the broadcast packet earlier. Aiming to minimize the energy consumption, LWF generates the longest delay as expected. In fact, it is a tradeoff between energy efficiency and delay in a wireless network. When compared with PDP and OF, when there are 600 nodes in the network, LWF conserves 32% and 26% of energy, respectively. On the other hand, LWF increases 15% and 12% of broadcast delay, respectively. Similar results can be found in Fig. 6(b). When the duty cycle is 5%, LWF conserves 24% and 20% of energy when compared to PDF and OF,



(a) (b)

Fig. 6. Impact on delay with different (a) number of nodes and (b) duty cycles

respectively. On the other hand, LWF increases 8% and 6% of broadcast delay when compared with PDP and OF, respectively. We consider using LWF is still beneficial when energy consumption is a concern.

VII. CONCLUSIONS

In this paper, we handle the energy-efficient broadcast problem. We have defined the least energy consumption broadcast problem in a network with unreliable links and have proposed the LWF scheme to solve it. LWF reduces the number of forwarders. More importantly, instead of using unicast to support broadcast as suggested by existing solutions, LWF achieves broadcast in an efficient way. LWF increases the number of receivers for each transmission such that the total number of transmissions is reduced. We also analyze the lower bound of the expected number of transmissions for a sender to broadcast to its one-hop neighbors. Simulation results verify that LWF improves existing opportunistic flooding protocol OF significantly.

ACKNOWLEDGEMENTS

This research was sponsored by Ministry of Science and Technology, R. O. C., under grants NSC 102-2221-E-019-017-MY3 and MOST 105-2221-E-019-050-MY3

REFERENCES

[1] Data Sheet for Chipcon CC2420 2.4GHz IEEE 802.15.4/Zigbee RF Transceiver. In *http://focus.ti.com/lit/ds/symlink/cc2420.pdf*.

[2] A. Ahmad, N. Javaid, S. H. Ahmed, S. H. Bouk, M. Ilahi, and D. Kim. COME: cost optimisation with multi-chaining for energy efficient communication in wireless sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 20(3):186–198, 2016.

[3] M. Calderon, M. Sedano, A. Azcorra, and C. Alonso. Active network support for multicast applications. *IEEE Network*, (3):46–52, 1998.

[4] C.-M. Chao, D.-J. Huang, and Y.-R. Peng. Energy-efficient broadcast in multihop cognitive radio networks. *Computer Communications*, 72:130 – 140, 2015.

[5] C.-M. Chao and Y.-W. Lee. A quorum-based energy saving mac protocol design for wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 59(2), 2010.

[6] C.-M. Chao and Y.-W. Lee. Load-aware energy-efficient medium access control for wireless sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 10(1), 2012.

[7] C.-M. Chao, L.-F. Lien, and C.-Y. Hsu. Rendezvous enhancement in arbitrary-duty-cycled wireless sensor networks. *IEEE Transactions on Wireless Communications*, 12(8), 2013.

[8] C.-M. Chao and M.-W. Lu. Energy-efficient transmissions for bursty traffic in underwater sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 13(1), 2013.

[9] T. L. Duc, D. T. Le, V. V. Zalyubovskiy, D. S. Kim, and H. Choo. Level-based approach for minimum-transmission broadcast in duty-cycled wireless sensor networks. *Pervasive and Mobile Computing*, 27:116–132, 2015.

[10] T. L. Duc, D. T. Le, V. V. Zalyubovskiy, D. S. Kim, and H. Choo. Towards broadcast redundancy minimization in duty-cycled wireless sensor networks. *International Journal of Communication Systems, DOI: 10.1002/dac.3108*, 2016.

[11] P. Dutta and D. Culler. Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications. *In Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 71–84, 2008.

[12] M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. In *New York: Freeman*, 1979.

[13] S. Guo, Y. Gu, B. Jiang, and T. He. Opportunistic Flooding in Low duty-cycle Wireless Sensor Networks With Unreliable Links. In *ACM MobiCom*, 2009.

[14] S. Guo, L. He, Y. Gu, B. Jiang, and T. He. Opportunistic Flooding in Low-Duty-Cycle Wireless Sensor Networks with Unreliable Links. *IEEE Transactions on Computers*, 60(11):2787–2802, 2014.

[15] A. H. Ho, Y. H. Ho, K. A. Hua, R. Villafane, and H.-C. Chao. An Efficient Broadcast Technique for Vehicular Networks. *Journal of Information Processing Systems*, 7(2):221–240, 2011.

[16] Y. H. Ho, M. C. Chen, and H.-C. Chao. Congestion avoidance routing for manets. *Int. J. Ad Hoc Ubiquitous Comput.*, 16(1):26–41, June 2014.

[17] Y.-C. Kuo and C.-N. Chen. CRT-MAC: A Power-Saving Multicast Protocol in the Asynchronous Ad hoc Networks. In *IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing*, pages 332–337, 2008.

[18] S. Lai and B. Ravindran. Efficient Opportunistic Broadcast over Duty-cycled Wireless Sensor Networks. In *IEEE INFOCOM*, 2010.

[19] S. Lai and B. Ravindran. On Multihop Broadcast over Adaptively Duty cycled Wireless Sensor Networks. In *Distributed Computing in Sensor Systems*, pages 158–171, 2010.

[20] Z. Li and M. Li. Approaching Efficient Flooding Protocol Design in Low-Duty-Cycle Wireless Sensor Networks. In *Wireless Algorithms, Systems, and Applications*, pages 292–301, 2011.

[21] H. Lim and C. Kim. Multicast Tree Construction And Flooding In Wireless Ad Hoc Networks. In *International Workshop Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 61–8, 2000.

[22] C.-H. Lin, C.-T. King, and T.-Y. Chen. Constrained multiple deployment problem in wireless sensor networks with guaranteed lifetimes. *Wireless Networks*, 17(2):385–396, 2011.

[23] W. Lou and J. Wu. On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks. 1(2):111–122, 2002.

[24] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu. The Boadcast Sorm Poblem In A Mobile Ad Hoc Network. In *ACM MobiCom*, pages 151–62, 1999.

[25] Y. Sun, O. Gurewitz, L. T. S. Du, and D. Johnson. ADB: An Efficient Multihop Broadcast Protocol based on Asynchronous Duty cycling in Wireless Sensor Networks. In *ACM SenSys*, 2009.

[26] F. Wang and J. Liu. RBS: A Reliable Broadcast Service for Large-scale Low Duty-cycled Wireless Sensor Networks. In *IEEE ICC*, 2008.

[27] F. Wang and J. Liu. On Multihop Broadcast over Adaptively Duty cycled Wireless Sensor Networks. In *IEEE INFOCOM*, 2009.

[28] F. Wang and J. Liu. On Reliable Broadcast in Low Duty-Cycle Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 11(5):767–79, 2012.