Correlated Flooding in Low-Duty-Cycle Wireless Sensor Networks

Shuo Guo*, Song Min Kim*, Ting Zhu‡, Yu Gu†, and Tian He*
*University of Minnesota, Twin Cities, USA
†Singapore University of Technology and Design, Singapore
‡Binghamton University, USA
Email: {{sguo,ksong,tianhe}@cs.umn.edu, tzhu@binghamton.edu, jasongu@sutd.edu.sg}

***Abstract*—Flooding in low-duty-cycle wireless sensor networks is very costly due to asynchronous schedules of sensor nodes. To adapt existing flooding-tree-based designs for low-duty-cycle networks, we shall schedule nodes of common parents wake up simultaneously. Traditionally, energy optimality in a designated flooding-tree is achieved by selecting parents with the highest link quality. In this work, we demonstrate that surprisingly more energy can be saved by considering link correlation. Specifically, this work first experimentally verifies the existence of link correlation and mathematically proves that the energy consumption of broadcasting can be reduced by letting nodes with higher correlation receive packets simultaneously. A novel flooding scheme, named Correlated Flooding, is then designed so that nodes with high correlation are assigned to a common sender and their receptions of a broadcasting packet are only acknowledged by a single ACK. This unique feature effectively ameliorates the ACK implosion problem, saving energy on both data packets and ACKs. We evaluate Correlated Flooding with extensive simulations and a testbed implementation with 20 MICAz nodes. We show that Correlated Flooding saves more than 66% energy on ACKs and 15%∼50% energy on data packets for most network settings, while having similar performance on flooding delay and reliability.**

I. Introduction

Wireless sensor networks (WSNs) have been used in many long-term sustainable applications such as environment monitoring [1], [2], target tracking [3], [4] and infrastructure protection [5]. To ensure service continuity, a sensor network normally operates at a very-low-duty-cycle (e.g., 5% or less), in which a sensor node schedules itself to be active briefly and then stays dormant for a long time. While the lifespan of a network is greatly prolonged, such low-duty-cycle operation significantly reduces the performance of many network operations including flooding [6], [7], an important networking primitive for code dissemination, system configuration, and routing tree formation. Due to the loss of connectivity when sensor nodes are sleeping, the performance of flooding degrades significantly. It has been studied [6] that the flooding coverage ratio of a pure flooding process drops to less than 10% as the duty cycle of the network decreases to 5%, showing strong evidence of the demand for a tailored flooding design for low-duty-cycle networks.

This research was partially supported by the US National Science Foundation (NSF) grants CNS-0845994, CNS-0917097, and Singapore-MIT IDC Grant IDD61000102.

In order to improve the flooding coverage ratio in low-duty-cycle networks, several previous works [6]–[8] have been proposed to achieve broadcasting via unicasts, i.e., a sender transmits a broadcasting packet multiple times to reach its neighbors with different working schedules. This design makes a very inefficient use of energy and causes a very long flooding delay. Another possible solution is to build a flooding tree, such that nodes with common flooding senders tune their working schedules to wake up and receive broadcasting packets simultaneously. This approach significantly reduces the energy consumption compared with the solution based on unicasts since a single transmission can be heard by multiple receivers.

However, there are new challenging issues associated with the flooding-tree-based design. First, a sender has to ensure that a broadcasting packet reaches all its children to avoid the miss reception of this packet by any child and all its subtree nodes. Since wireless transmission is notoriously unreliable for low-power embedded devices, there is no guarantee that all receiving nodes would be able to successfully receive the broadcasting packet. It is also difficult for a sender to know the reception status of its receivers by overhearing their transmissions due to the low-duty-cycle operation. The ARQ (Automatic Repeat reQuest)-based mechanism is thus needed to guarantee reliability. While many works have utilized ARQ-based mechanism and provided solutions for reliable broadcasting [6], [7], [9], another problem arises as they require the senders to collect ACKs from all intended receivers, leading to the ACK implosion problem [10], [11] which introduces more energy consumption for both senders and receivers.

This paper proposes a novel flooding design, named Correlated Flooding, which solves aforementioned problems caused by both low-duty-cycle operations and ACK implosion. For energy efficiency, we adopt the tree-based approach where sensor nodes sharing common flooding senders add or update a common wake-up time unit in their working schedules so as to receive broadcasting packets simultaneously. The flooding tree is constructed to be more energy efficient than the traditional energy-optimal tree by considering not only the link quality, but also *link correlation*, the phenomenon we observed from extensive experiments that *the receptions of a broadcasting packet at different receiving nodes are not independent*. To address the ACK implosion problem, we again exploit link correlation such that the *ACK from one receiving node serves* *as the ACKs of other highly correlated receivers.* We also theoretically prove that by considering link correlation, our solution is more energy-efficient than the legacy solutions which assume link independence.

We summarize our contributions as follows:

1) We experimentally reveal the existence of link correlation, i.e., the reception results of a broadcasting packet at multiple receivers are not independent. The reception of the broadcasting packet at one node can indicate a high chance of the reception at other nodes.
2) We mathematically show the impact of link correlation on broadcasting/flooding. We show that the energy consumption can be reduced by allowing nodes with high correlations to wake up at the same time.
3) We propose Correlated Flooding, a novel distributed flooding design for low-duty-cycle WSNs in which highly correlated nodes wake up and receive broadcasting packets simultaneously. The receptions of the packet at multiple correlated nodes are acknowledged by one ACK to avoid ACK implosion. We show the energy efficiency of our design from both simulation and testbed implementation. As far as we know, this is the first work that explores link correlation both mathematically and experimentally in the flooding design for low-duty-cycle WSNs.

The rest of the paper is organized as follows: Section II presents the network model and assumptions. Section III introduces link correlation, followed by the main design in Section IV. Evaluation results from simulations and testbed experiments are shown in Sections V and VI. Section VII summarizes related work. Section VIII concludes the paper.

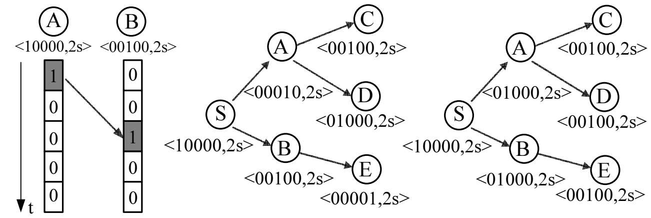
II. Model and Assumptions

This section introduces the network model and the assumptions related to our Correlated Flooding design.

A. Low-Duty-Cycle Network Model

Similar to the low-duty-cycle network model in [6], [12], we assume a WSN consisting of a number of sensor nodes is deployed in a given field. Each node is duty-cycled with two possible states, the active state and the dormant state, switched according to a working schedule. In the dormant state, a node turns all function modules off except a timer to wake itself up, and thus, it only receives packets when it is active. A node switches from the dormant state to the active state when (i) it is scheduled to be actively receiving, or (ii) one of its neighboring nodes is ready to receive its packets.

Suppose each node's working schedule is represented by $< \omega, \tau >$, where ω is a bitmap with each bit indicating dormant or active state. τ is the duration of each bit in ω. Fig.1(a) shows an example of this model where ω has 5 bits each of which (τ) is $2s$ long. Nodes A and B have the working schedules of $< 10000, 2s >$ and $< 00100, 2s >$, respectively. Take node A as an example, it is active during the first $2s$ and dormant during the next $8s$ in a period. Suppose A has a packet to send to B when A is active. Since a node can only receive packets



(a) Sleep Latency (b) Random Schedules (c) Updated Schedules

Fig. 1. Low-Duty-Cycle Network Model

when it is active (the corresponding time unit in the figure is shadowed), A has to wait $4s$ to start the transmission. Based on this low-duty-cycle model, Fig.1(b) and (c) show an example of flooding where node S wants to send a flooding packet to all the nodes in the network. We suppose all links are perfect. It is easy to find that, with a randomly set schedules in (b), $S \rightarrow A \rightarrow C$ is the path with the longest delay of 7 time units (14s). Also, since A and B have different working schedules, S has to transmit the packet twice to reach both of them. The same thing happens to A whose receivers C and D have different working schedules. As a result, the total number of transmissions for a single flooding packet is 5. If the working schedules are updated as shown in Fig.1(c), however, both the flooding delay and energy consumption can be significantly reduced. From the figure, the flooding delay is only 2 time units (4s), and the total number of transmissions is 3. We see from this example the benefits of the flooding-tree-based scheme in terms of both flooding delay and energy consumption if allowing nodes with a common flooding sender wake up and receive broadcasting packets simultaneously.

The example shown above represents ideal network conditions with perfect links. The real challenging issue is how to construct an energy-efficient reliable flooding tree for low-duty-cycle sensor networks with unreliable wireless links, which is the focus of the rest of this paper.

B. Assumptions

Suppose the source node has flooding packets to send throughout the network. The following assumptions are made for our Correlated Flooding design:

1) The network is locally synchronized so that a node knows when it can communicate with neighboring nodes given their working schedules. Local synchronization can be achieved by using a MAC-layer time stamping technique [13], which achieves an accuracy of 2.24μs with very low cost (exchanging a few bytes of packets every 15 minutes). The accuracy of 2.24μs is sufficient given that τ is mostly greater than 20,000μs.
2) Neighboring nodes are first discovered by rendezvous process [14]. Working schedules are then shared so that they know when to wake up and communicate with each other again. When a node changes its working schedule, it notifies all its neighboring nodes about the update to avoid permanent disconnection. New working schedules are only operated after they are updated to all possible neighbors.

3) We assume the existence of wireless communication failure. The probability of a successful transmission through a wireless communication link with no information about the reception status of other links is quantified by link qualities. We will later discuss how this probability changes with the information of other links (link correlation) in Section III. Link qualities can be measured using probe-based methods in [15], [16] or through low-cost piggybacking on regular data traffic. It is affected by many factors and changes over time. However, it has been empirically studied in [17]–[19] that the changing rate is slow. Therefore, the measurements of the link quality can be updated at a very low frequency (e.g., every half an hour).
4) We assume the ARQ-based mechanism is used in flooding process to improve reliability in low-duty-cycle networks. Collisions of ACKs can be partly resolved by either CSMA or some other techniques including the TDMA-based RMAC [20] or OFDM-based SMACK [21]. Given the length of active time unit τ, we assume that only a limited number of ACK packets (up to M) can be successfully transmitted.

III. Impact of Link Correlation on Broadcasting

One important feature that distinguishes Correlated Flooding from all previous works is that, in addition to link quality, which is considered when building up a flooding tree in previous works, we also explore link correlation, the phenomenon that the receptions of a broadcasting packet at different receiving nodes are not independent. By taking link correlation into account, we not only build a more energy-efficient flooding tree but also solve the ACK implosion problem, saving the energy consumption on both data packets and ACKs. It is thus important to first study the existence of link correlation, as well as its impact. In III-A we show the existence of link correlation using data from real experiments. In III-B we further study how we can explore link correlation in the flooding design to further reduce energy consumption.

A. Existence of Link Correlation

Link correlation has been experimentally studied in [22] that when a sender sends out a broadcasting packet to multiple receivers, the reception results at these receivers are not independent with each other. To verify this statement and further study its impact, we did experiments and deployed 42 sensor nodes. We placed a sender in the center of the topology and all the other nodes were randomly placed as receivers. The sender broadcasted 6000 packets identified by sequence numbers and receivers recorded the reception result. We first study the pairwise link correlation. For each pair of receivers, we counted the number of successful receptions at the node with higher link quality (denoted as N_H) when the node with lower link quality (denoted as N_L) had received the packet successfully, i.e., $Pr(N_H|N_L)$. We compare this number with the number of successful receptions at N_H regardless of the reception result at N_L, i.e., $Pr(N_H)$. We found that for about 82% of the receiver pairs, the former is greater than the latter, i.e., $Pr(N_H|N_L) > Pr(N_H)$. The distributions of $Pr(N_H)$ and $Pr(N_H|N_L)$ are shown in Fig.2 from which we clearly see that the conditional probability $Pr(N_H|N_L)$ is closer to 1 than $Pr(N_H)$, showing the existence of link correlation.

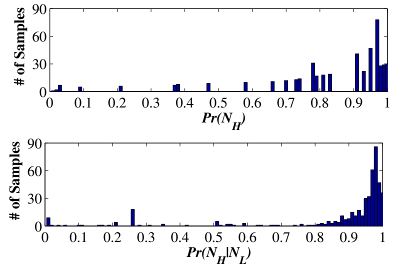


Fig. 2. Statistics of Receiving Probability

We further study the receiving probability given the condition that a number of nodes that have worse link qualities have received the packet successfully. Fig.3 shows a sampled result of this conditional probability $Pr(N_H|\{N_L\})$ (where, $\{N_L\}$ denotes an arbitrary combination of nodes whose link qualities are lower than that of N_H) compared with individual $Pr(N_H)$. It is clear to see that given the knowledge of the reception at those nodes with lower link qualities, the receiving probability of N_H is higher. Again this shows the evidence of link correlation.

B. Impact of Link Correlation on Broadcasting

Given the existence of link correlation, we mathematically show how it affects the energy efficiency in flooding. Fig.4 shows an example where a flooding tree is being built and node E is selecting its flooding senders from A and B. Given the link qualities, it is obvious that without the consideration of link correlation, E should select A since the link quality of AE (85%) is higher than that of BE (80%) while the link qualities of AC and BD are equal. In another word, the expected number of transmissions needed for delivering a flooding packet to both receivers (denoted by m) by A is smaller than that by B. If link correlation is considered, however, this conclusion no longer holds. We next derive the equation for m mathematically.

Let p_1 and p_2 ($p_1 \geq p_2$) denote the link qualities for the two receivers respectively. The corresponding packet loss probabilities are denoted as $q_1 = 1 - p_1$ and $q_2 = 1 - p_2$. Let q_{12} denote the probability that a broadcasting packet is not received by either receiver. For an arbitrary positive integer k, the number of transmissions a sender needs to deliver the packet to both receivers m will satisfy the following equation:

$$Pr(m > k) = q_1^k + q_2^k - q_{12}^k \quad (1)$$

Taking the difference yields

$$\begin{aligned} & Pr(m = k) = Pr(m > k-1) - Pr(m > k) \\ = \;& (q_1^{k-1} + q_2^{k-1} - q_{12}^{k-1}) - (q_1^k + q_2^k - q_{12}^k) \end{aligned} \quad (2)$$

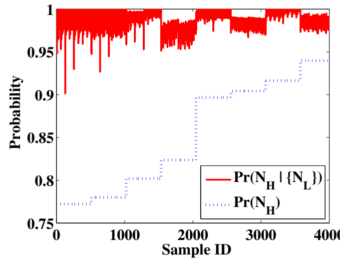


Fig. 3. Statistics of Receiving Probability

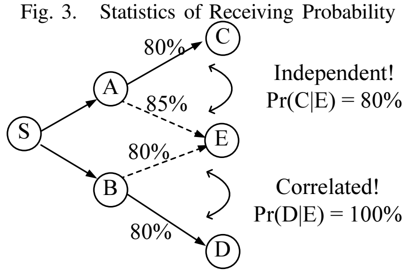


Fig. 4. Example of Correlation

and the expectation can be calculated as

$$E(m) = \sum_{k=1}^{+\infty} kPr(m = k) = \frac{1}{p_1} + \frac{1}{p_2} - \frac{1}{1 - q_{12}} \tag{3}$$

If two links are independent, q_{12} is simply the product of q_1 and q_2. Otherwise, q_{12} is the product of q_2 and $q_{1/2}$, where $q_{1/2}$ is the conditional probability that the node with better link quality fails to receive the packet given the failure of the node with a worse link. We prove in Appendix A that $q_{1/2} \geq q_1$, and thus q_{12} becomes greater when links are correlated. With greater q_{12}, it is easy to see that $E(m)$ is reduced.

For the network in Fig.4, if E chooses A as its flooding sender, $E(m)$ is 1.3955 based on Eq.3 given that AE and AC are independent. If E chooses B, however, since BE and BD are correlated, it is not difficult to calculate based on total probability equation that $q_{D/E}$ is 1. As a result, $E(m)$ is 1.25. Considering link correlation, E should choose B as its flooding sender instead of A, although AE has a higher link quality than BE. We conclude that for two-receivers case, broadcasting costs less energy for correlated receivers.

We next extend the conclusion into N receivers. Suppose N receivers are denoted as $R_1, ...R_N$, sorted in decreasing order of link quality. The number of transmissions in which the sender first delivers to the ith receiver successfully is denoted as m_i. Based on Eq.2 and 3:

$$\begin{aligned}
& E(m) = \sum_{k=1}^{+\infty} kPr(m = k) \\
= & \sum_{k=1}^{+\infty} k(Pr(m > k-1) - Pr(m > k)) \\
= & Pr(m > 0) - Pr(m > 1) + 2Pr(m > 1) - \\
& 2Pr(m > 2) + 3Pr(m > 3) - ... \\
= & \sum_{k=0}^{+\infty} Pr(m > k) = \sum_{k=0}^{+\infty} (1 - Pr(m \leq k)) \\
= & \sum_{k=0}^{+\infty} (1 - Pr(m_1 \leq k, m_2 \leq k, ..., m_N \leq k)) \\
= & \sum_{k=0}^{+\infty} [1 - Pr(m_N \leq k) Pr(m_{N-1} \leq k | m_N \leq k) \\
& \cdots Pr(m_1 \leq k | m_2 \leq k, ..., m_N \leq k)] \\
= & \sum_{k=0}^{+\infty} [1 - (1 - Pr(m_N > k)) \\
& \cdot (1 - Pr(m_{N-1} > k | m_N \leq k)) \\
& \cdots (1 - Pr(m_1 > k | m_2 \leq k, ..., m_N \leq k))]
\end{aligned} \tag{4}$$

In Eq.4, each conditional probability $Pr(m_i > k | m_{i+1} \leq k, ..., m_N \leq k)$ can be translated as: the probability node i has not received the packet in k transmissions given the condition that all nodes with worse link quality have already received the packet in k transmissions. If all the nodes are independent and have no correlation at all, the conditions in each term can be eliminated and the probability becomes $Pr(m_i > k)$. Recall in Fig.3 that the conditional probability of a node receiving a packet successfully given the knowledge of the reception at an arbitrary set of nodes with lower link qualities ($Pr(N_H | \{N_L\})$) is always higher than the marginal probability ($Pr(N_H)$). As a result, given the reception at all the nodes with lower link qualities, the probability that node i has not receive the packet in k transmissions ($Pr(m_i > k | m_{i+1} \leq k, ..., m_N \leq k)$) is lower than $Pr(m_i > k)$. Thus, $E(m)$ in Eq.4 has smaller value when nodes have correlations. By allowing nodes with correlation to wake up at the same time, the energy consumption on flooding can be further reduced.

IV. Main Design

The goal of Correlated Flooding is to exploit link correlation in the construction of an energy-efficient flooding tree, to save the energy consumption on both data packets and ACKs. First, the energy consumption on transmitting data packets can be reduced compared with the flooding tree that only considers link quality, as have already been shown in Section III. Second, by exploiting link correlation, a single ACK can acknowledge the reception at not only the node who sends this ACK, but also some other nodes that receive the broadcasting packet at the same time. This effectively ameliorates the ACK implosion problem and saves energy for both senders and receivers.

We divide Correlated Flooding into two parts, the sender side and the receiver side, and present them separately. In Section IV-A we present sender side design about how a sender collects the information of link correlation among all its receivers and utilizes this information to group the receivers. In Section IV-B we illustrate the idea of receiver side sender selection so that only one flooding sender is selected by each node. In Section IV-C, we summarize the design and discuss scheduling issues following the flooding tree construction.

A. Sender Side: Link Measurements and Group Division

On the sender side, each node divides all its possible flooding receivers into groups according to link correlation.

Nodes within the same group are highly correlated such that the reception of a broadcasting packet at the node with the worst link quality indicates a very high probability of the reception at every node in the group. This node with the worst link quality in the group is named as the *critical node* of the whole group, or for short, the *c-node*. Due to the limited number of ACKs can be successfully received by a broadcasting sender, only the *c-node*s send ACKs back to the sender. A sender stops broadcasting after it collects all ACKs from its *c-node*s. By doing this, senders have certain guaranteed coverage of their receivers without encountering the ACK implosion problem. Thus, it is very important for a sender to know the link correlation of its receivers and divide them into groups accordingly, such that the nodes within the same group have so high correlation that one ACK serves as an implicit ACK of the whole group.

1) Link Measurements: The information of link correlation is collected at the same time when link quality is measured or updated. In WSNs, every node periodically sends out a hello message at an adaptive time interval, the length of which is adjusted based on the link's stability. Every hello message is identified by the node ID and a packet sequence number. It is used for not only neighbor discovery, but also updating link qualities and link correlation. To ensure the network's loop-free property, a node considers another node as its possible flooding sender only when that node has smaller hop count (defined as the minimum number of hops to reach the source [6]). Each node maintains a reception record of hello messages from possible senders and periodically shares this information with them. In order to reduce the required memory space and mitigate the overhead of control messages, each node only keeps the information for the most recent hello messages, e.g., the latest 10 messages, and the record is represented in a bitmap format (e.g., [01100]), with "1" denoting a successful reception and "0" denoting a missed packet.

By receiving these bitmaps from flooding receivers, a sender easily collects information about link quality and link correlation. While the calculation of link quality is simply done by counting the number of "1"s divided by the length of the bitmap, the calculation of link correlation deserves more explanation. Suppose a sender S receives bitmaps from receivers A and B, and $S \rightarrow A$ has higher link quality than $S \rightarrow B$. The correlation of these two links can be represented by the conditional probability $Pr(A|B)$, which is calculated by counting the number of "1"s that appears in both A and B's bitmaps divided by the number of "1"s in B's bitmap. For example, if the bitmaps of A and B are [0111] and [0110], respectively, the correlation of A and B is calculated as 100%, since for all the "1" in B's bitmap, the corresponding bit in A is 1. The larger the conditional probability is, the higher the correlation is between the links. An alternative metric to represent how the two bitmaps are correlated is "Hamming Distance", defined as the number of positions that the corresponding bits are different. For example, the Hamming distance is 0 for [0111] and [0111], and 4 for [0111] and [1000]. The larger the Hamming distance is, the lower the correlation is. Both these two metrics can be used in the group division process as will be discussed later.

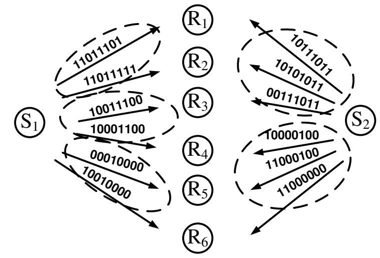


Fig. 5. Example of Group Division

From the above process we see that the measurement of both link quality and link correlation only requires exchanging a few hello messages containing bitmap information at a low frequency. It thus introduces very low overhead to the network.

2) Group Division: As the sender collects all bitmaps from its flooding receivers, it begins the clustering process. In Correlated Flooding, clustering is done by k-means method, which divides n nodes into k clusters with least intra-cluster distance. Initially k nodes are randomly chosen as cluster centers and each node is assigned to the nearest cluster center. Then the new cluster centers are recomputed and this process is repeated until the assignment result no longer changes.

As mentioned above, both the conditional probability and Hamming distance can be used as the metric of distance in k-means algorithm (For the conditional probability, use $1 - Pr(A|B)$ as the distance between A and B to ensure that a shorter distance indicates a higher correlation). In Correlated Flooding, we choose Hamming Distance as the distance metric because it avoids possible false indication of correlation caused by the limited bitmap length. For example, the conditional probability of bitmaps A ([11101]) and B ([00001]) is calculated as 100%. Since the conditional probability only cares about the bits in A at the positions where the corresponding bits in B is "1", the sample size is only 1 in this example. Since the sample size is so limited, one can hardly say that next time B receives the packet, A is 100% to receive the packet as well. If Hamming distance is chosen, on the other hand, the distance in this example is 4, which is a large number given that the length of bitmap is 5. In summary, a small Hamming distance is a sufficient (although not necessary) condition for nodes with high correlation and using Hamming distance as the metric in k-mean method is thus safer.

The k-mean method starts from a small k value (e.g., 2) and increments k by 1 every time. It stops until the intra-group distance is smaller than a certain threshold or k exceeds M, which is the maximum number of ACKs that can be supported to be received by the sender without collision. Fig.5 shows an example of group division based on the "Hamming Distance" where sender S_1 divides its six receivers into 3 groups and sender S_2 divides its six receivers into 2 groups. It is easy to see that there is strong correlation on the reception result for the nodes within the same group, and thus, one ACK from the group can easily serve as the ACK for the other nodes within the same group. With the minimized intra-group

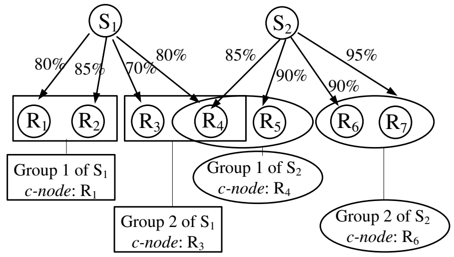


Fig. 6. Example of Flooding Sender Selection

distance, nodes within the same group have high correlation and the reception at all these nodes can be acknowledged by one ACK. The node with the worst link quality, i.e., the critical node (*c-node*) in the group is selected to be responsible to send this ACK. In the example in Fig.5, R_1, R_4 and R_5 are selected to be the *c-node* of the corresponding group for S_1, respectively. And the probability that the entire group has received a broadcasting packet given the successful reception at these *c-node*s can be easily calculated to be 100%.

B. Receiver Side: Sender Selection

After group division, each sender divides its potential receivers into multiple groups. For each receiver, since it normally has multiple senders, it also belongs to multiple groups: one for each of its senders. For example, as shown in Fig.6, since node R_4 is a common receiver of node S_1 and node S_2, it belongs to a group for S_1 as well as another group for S_2. However, if we keep node R_4 in both of those two groups, node S_1 and node S_2 would both try to broadcast the message to node R_4 and wasting energy unnecessarily. Therefore, in this section, we aim at selecting the optimal sender from which a node should be receiving the flooding message, so as to minimize the expected energy consumption for receiving ACK. The selection is based on not only link quality but also the group information to achieve energy efficiency. During the selection process, senders keep tracking of the group information and the *c-node*s are dynamically updated as receivers select or not select a sender. To better reveal the intuition on sender selection, let us first look at the example in Fig.6. Assuming node S_1 and node S_2 have divided all their potential receivers into two groups, with the *c-node* labeled below each group. The directional link quality from a sender to a receiver is also labeled next to the edge connecting them. For node R_4, it can either choose to receive the flooding message from node S_1 or from node S_2. If without considering the link correlations, clearly node R_4 should select node S_2 as its broadcasting sender as the link quality between node S_2 and R_4 is better than that between node S_1 and R_4. However, if we do consider the link correlation, the outcome of the sender selection would be different. As node R_4 has the worst link quality in group 1 for node S_2, it is the *c-node* in that group. The broadcasting from node S_2 would not stop if it has not received an ACK from node R_4 while the other node R_5 in group 1 may have received multiple copies of this message since it has better link quality from S_2. However, if node R_4 selects S_1 as its sender instead, since R_3 has a weaker link connection with S_1 and is the bottleneck of the group,

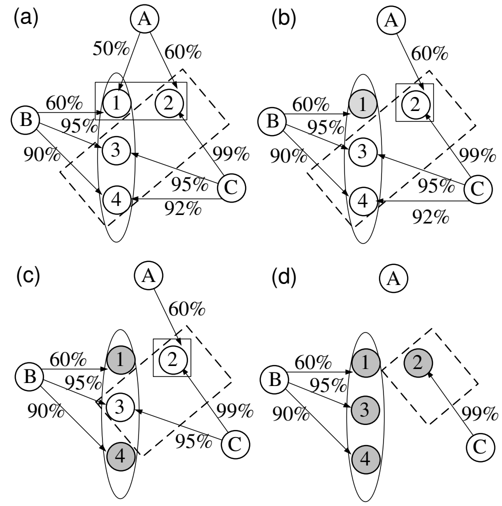


Fig. 7. Example of the Selection Process

the inclusion of R_4 introduces no additional transmission cost to node S_1. At the same time, as node R_4 leaves the group 1 of node S_2, the *c-node* in the group becomes node R_5, which has better link connection with node S_2 than node R_4. Consequently, after considering link correlations, R_4 should choose S_1 as its sender, although the link quality from S_1 to R_4 is lower.

From the above example, we can see that energy efficiency can be achieved when the link quality for *c-node* in each group is maximized. However, building such an energy-optimal flooding tree even without considering of link correlation is equivalent to find a Minimum Connected Dominating Set (MCDS), which is NP-hard [23]. Therefore in this section, we discuss a heuristic solution for the sender selection which is shown to be very effective in evaluation.

With the idea of selecting senders with high link qualities while increasing the link quality of *c-node*s, we propose a distributed sender selection process where flooding senders are selected based on the following two rules:

1) A node selects the sender with the highest priority if it is not a *c-node* in this sender's group. This is because by not selecting those senders that this node is the *c-node*, other nodes that have better link qualities for these senders will become the new *c-node*s and the link qualities of *c-node*s is increased. If there are more than one qualifying senders, a node selects the one with the best link quality as its flooding sender.
2) When a node is the *c-node* for all senders, it has to select the one with the best link quality as its flooding sender since there is no other way to improve the link quality of *c-node* any more.

Fig.7 shows an example of a selection process where nodes A, B and C are upper level nodes of nodes 1, 2, 3, and 4. A, B and C are also possible flooding senders of nodes 1, 2, 3, and 4. Suppose the initial group division result is shown in Fig.7(a) where 1 and 2 are in the same group of A, 1, 3, and 4 are in the same group of B, and 2, 3, and 4 are in the same group of C. For node 1, it is the *c-node* for both A and

B, and as a result, it selects B as its flooding sender since the corresponding link quality is higher, as shown in Fig.7(b) where the nodes that have selected their flooding sender are shadowed. With node 1's join to B, nodes 3 and 4 are never *c-node* for B. Since node 4 is the *c-node* for C, it then selects node B as its flooding sender although the link quality from C to 4 is higher, as shown in Fig.7(c). With node 4's leave from C, node 3 becomes the new *c-node*, and it clearly selects node B since it is not the *c-node* for B, and the corresponding link quality is higher. Then node 2 becomes the new *c-node* of C. Since it is the *c-node* for both C and A, it selects C as its flooding sender since the link quality is higher. The final result of the selection process is shown in Fig.7(d).

C. Design Summary and Discussion

From the sender side group division process in IV-A and the receiver side sender selection process in IV-B, an energy-efficient flooding tree, consisting of high-quality links with high correlation among siblings, can be constructed. With this flooding tree, each node assigns a common active time unit to all its flooding receivers, i.e., all its children on the tree, so that they can either add this time unit into their working schedules or change one of their original active time units into this common active time unit.

It is worth noting that the selection of the common active time unit by each sender does not affect the energy efficiency of the flooding tree, as long as the senders that can reach the same receiving node do not assign the same active time unit to their children. However, the selection of common active time units does affect the flooding delay. For the example shown in Fig.1(c), if node B picks the last bit as the common active time unit, node E's working schedule will become $< 00001, 2s >$, and the flooding delay to E will increase to 4 time units, which is $8s$. This is $4s$ longer than the delay using the schedule shown in the figure. We thus conclude that with a more carefully chosen scheduling scheme such as the streamline technique in [24], the flooding delay can be further reduced (although the energy efficiency remains the same). As we focus on the energy part by exploring link correlation in the construction of the energy-efficient flooding tree, we leave the scheduling problem as future work.

V. Large-Scale Simulations

In this section, we evaluate our Correlated Flooding design in simulations with various network settings. We will later show the evaluation result based on a testbed implementation in Section VI.

A. Simulation Setup

In simulation, up to 1000 nodes are randomly generated with different duty cycles and densities. In order to reflect the link quality and link correlation in the real world, the bitmaps of all the sender-receiver pairs are generated by sampling data from testbed experiments. For a single broadcasting packet, the reception results at multiple receiving nodes are generated by selecting the bits at the same position of different bitmaps, and this position is randomly chosen. For example, if two receivers' bitmaps are [000111] and [111000], respectively, the reception results can be either "0" and "1", or "1" and "0", reflecting the fact that in the real world, these two nodes never receive (or fail to receive) a broadcasting packet simultaneously. Unless otherwise explicitly specified, the default network size is 800, and the map size is $250m \times 250m$ with a communication range of $25m$. M is set to 5, which means in a single active time unit, up to 5 ACKs can be successfully received by a sender. Each data point in the simulation results is based on 10 network topologies with 1000 flooding packets sent for each topology.

B. Baseline

Since an energy-optimal flooding tree even without considering of link correlation is equivalent to finding a Minimum Connected Dominating Set (MCDS), which is NP-hard [23], we use a heuristic energy-optimal tree solution in [6] (denoted as "Energy Optimal Tree") as our baseline. In this solution, each node selects the node that has the best link quality among those upper-level nodes with less hop count.

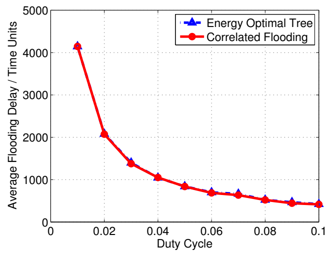
C. Performance Metrics

We use three performance metrics for evaluation: (i) the flooding delay, defined as the total time spent for flooding a packet from the source; (ii) the energy consumption, defined as the total number of transmissions for sending a single flooding packet to the entire network, including data packets and ACKs; and (iii) the coverage ratio, defined as the percentage of nodes that have received flooding packets reliably.

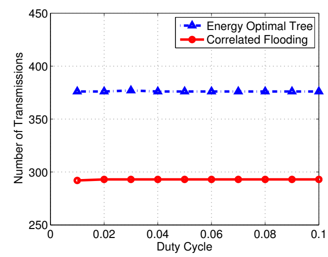
D. Simulation Results

1) Impact of Different Duty Cycles: Fig.8 shows the performance comparison for duty cycles from 1% to 10%. It can be seen from the figure that the flooding delays of the two designs are about the same. However, Energy Optimal Tree costs 34% more energy on data packets transmissions than Correlated Flooding, which means Correlated Flooding is more energy efficient while providing similar flooding delay. For flooding coverage, Energy Optimal Tree uses ACK for every node and it thus has 100% coverage ratio. For Correlated Flooding, since only one ACK is used for nodes in the same group of the same sender, there is still possibility that some of the nodes fail to receive the packet when the *c-node* of the same group has already received it. As shown in Fig.8(c) we can see that the coverage ratio of Correlated Flooding is very close to 1 (e.g., above 99.95%). This is very good performance since with the network size of 800, the average number of nodes that are not covered by a flooding process is less than 1. It again verifies the existence of link correlation.

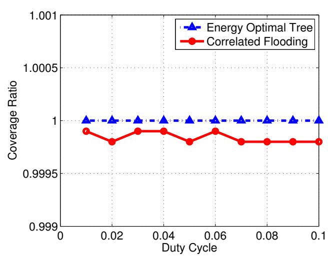
2) Impact of Different Network Sizes: Fig.9 compares the network performance of two designs under different network sizes from 200 to 1000. It shows that with a similar curve of the flooding delay, the energy consumption of the two designs are quite different. As the network size increases, both Correlated Flooding and Energy Optimal Tree transmit more data packets and ACKs, as expected. However, Correlated Flooding saves



(a) Flooding Delay

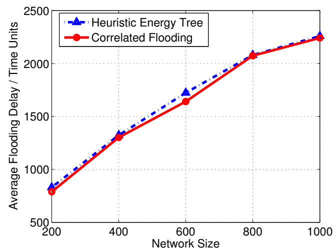


(b) Data Packets Sent

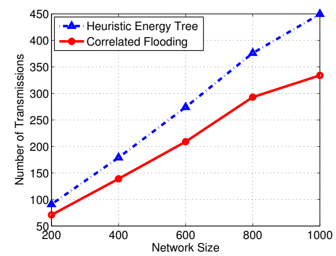


(c) Coverage Ratio

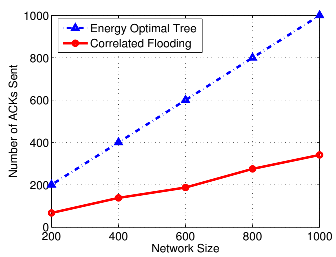
Fig. 8. Different Duty Cycles



(a) Flooding Delay

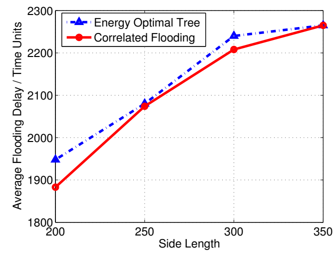


(b) Data Packets Sent

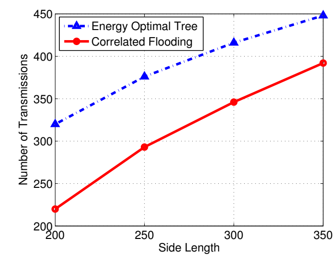


(c) ACKs Sent

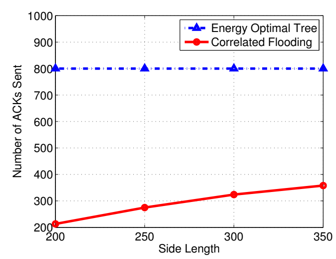
Fig. 9. Different Network Sizes



(a) Flooding Delay

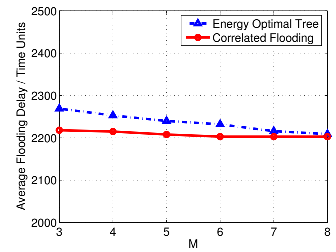


(b) Data Packets Sent

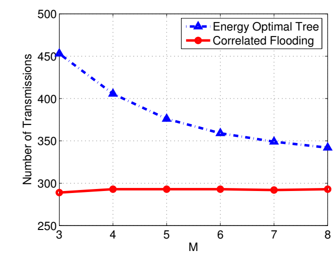


(c) ACKs Sent

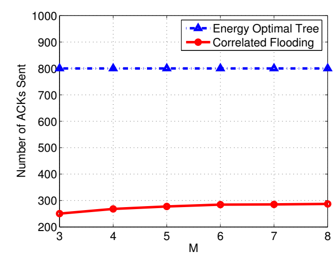
Fig. 10. Different Densities



(a) Flooding Delay



(b) Data Packets Sent



(c) ACKs Sent

Fig. 11. Different M values

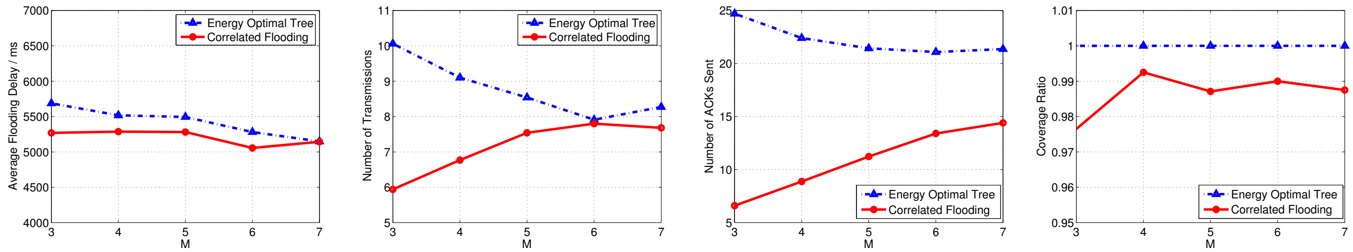
more than 20% of data packets and 60% of ACKs than Energy Optimal Tree. In addition, the curves of Correlated Flooding have smaller slope than Energy Optimal Tree, meaning that Correlated Flooding scales better and is more appropriate to be applied in large-scale networks.

3) Impact of Different Network Densities: Fig.10 shows the delay and energy performance of the two designs for networks with different network densities. Again, we see similar delay performance, and Correlated Flooding performs slightly better. In Fig.10(b), the number of data packets sent increases as the area of the map increases (and thus the density decreases). This is because as the network becomes sparser, the diameter of the network becomes larger and a sender has fewer children to receive broadcasting packets simultaneously. However, Correlated Flooding still has a 15%∼40% save on data packets compared with Energy Optimal Tree. In Fig.10(c), we see that Correlated Flooding sends more ACKs as the density decreases because there are fewer nodes in a group and the number of nodes whose reception status can be acknowledged by a single ACK is smaller. However, even for the sparsest case, we still saves 55% ACKs than Energy Optimal Tree.

4) Impact of Different M: We change M, the maximum number of ACKs that can be successfully received in one active time unit, from 3 to 8. As shown in Fig.11(a), the flooding delays of both designs decrease as M increases. This is because with a larger number of ACKs that can be received in a single active time unit, the flooding process stops earlier since senders receive all ACKs more quickly, although this may require longer duration of a single time unit to resolve collision, which may lead to even longer flooding delay. For the number of data packets sent, Energy Optimal Tree costs 16% $\sim$ 55% more energy than Correlated Flooding, and this number slightly decreases as M increases since the senders stop sending earlier. We also plot the number of ACKs sent for a single flooding packet in Fig.11(c). From this figure, we see that as M changes from 3 to 6, the number of ACKs packets sent in a flooding process increases by about 20%. This is because with a larger M value, receivers are divided into more groups and there are more *c-node*s that will send the ACKs. When M increases from 6 to 8, however, the numbers of ACKs sent are almost the same. This is because the group division process in IV-A stops as soon as there is small enough intra-group distance, and thus a larger M does not affect the number of groups divided by a sender any more. Again we see that Energy Optimal Tree transmits as many as 200% more ACKs than Correlated Flooding, showing the effectiveness of exploiting link correlation.

VI. Testbed Implementation

We implemented both Correlated Flooding and Energy Optimal Tree on a TinyOS/Mote platform consisting of 20 MICAz motes to further evaluate our design.



(a) Flooding Delay (b) Data Packets Sent (c) ACKs Sent (d) Coverage Ratio

Fig. 12. Performance Comparisons in Testbed Experiment

A. Experiment Setup

We deploy 20 MICAz motes randomly on an in-door testbed. The transmission powers of these motes are tuned down so that they form a 2-hop network. After deployment, the experiment is done in three phases: initialization, computation, and flooding. In initialization phase, each node takes turns to broadcast 100 packets, from which link qualities and correlations are calculated. In computation phase, flooding trees are constructed. For Correlated Flooding, the flooding tree is built based on the sender side group division and receiver side sender selection in Section IV. For Energy Optimal Tree, the flooding tree is built by simply selecting the parents with the best link qualities. In flooding phase, the nodes first turn themselves to operate at 2% duty cycle with 40ms unit time. Flooding is then done using the topology computed in the computation phase. To minimize the effect of the temporal link quality changes, nodes automatically switch between the two designs alternatively. Flooding delay is measured using the global time provided by the Flooding Time Synchronization Protocol (FTSP [13]).

B. Experiment Results

Fig.12 shows the experiment results of different performance metrics in this 20-node network. The flooding delay is plotted in Fig.12(a), from which we see that as M increases, the flooding delay of Energy Optimal Tree decreases. This is because with a larger M, more ACKs can be received simultaneously and the flooding process stops earlier. The delay of Correlated Flooding is not affected much by M and is about 3% shorter than Energy Optimal Tree. For the number of data packets sent in Fig.12(b), we see that Correlated Flooding is up to 40% less than Energy Optimal Tree. The reason that this number varies so much is that in this small network, the flooding trees built by the two design may be similar or exactly the same. For the number of ACKs sent in Fig.12(c), we see that Correlated Flooding saves 30%$\sim$70% ACKs compared with Energy Optimal Tree, showing the effectiveness of exploiting link correlation. The flooding coverage is plotted in Fig.12(d). We see that the coverage ratio increases by 1% as M increases from 3 to 4. A larger M increases the coverage ratio because with smaller groups, nodes within the group have higher correlation so that it is less likely that a node fails to receive a flooding packet given its *c-node*'s reception. We also see that the coverage ratio of Correlated Flooding is slightly lower than that in simulation. This is because testbed experiment experiences larger unexpected noise than simulation, and a single node's miss affects the coverage ratio a lot for this 20-node small network. However, with a 99% coverage ratio, the average number of nodes that fail to receive a flooding packet is 0.2, which is still very good flooding coverage.

VII. Related Work

Our contribution lies at the intersection of three areas: low-duty-cycle WSNs, broadcasting/flooding design and link correlation. Since our paper proposes the first flooding design exploring the characteristics of both low-duty-cycle network and link correlation, we summarize related works in individual field separately.

As essential operations for wireless networks, multicasting and flooding have been extensively studied in the literature [25]–[36]. Due to space constraints, we here focus only on those designs for low-duty-cycle WSNs. RBS [37] proposes a broadcasting design where transmissions are scheduled based on the information of receivers notified from control messages. However, sending control messages itself consumes a lot of energy. Wang et al. [38] proposes a centralized solution for multi-hop broadcasting by modeling the problem as a shortest path problem with a much simplified assumption of perfect link, which does not hold for wireless transmissions. Guo et al. proposes opportunistic flooding [6] and achieves delay and energy efficiency by sending opportunistically early packets while avoiding sending late packets. ADB [7] proposes a MAC protocol for multihop broadcasting by changing the progress information of each broadcast. Jiao [39], [40] proposes scheduling algorithms for multihop broadcasting without considering unreliable links. Lai [8] proposes a broadcasting solution that saves energy by letting a node decide how long to wait for more receivers to wake up, but requires a relatively high duty cycle. All these works do not consider link correlation, and broadcasts are done mostly by unicasts, which consumes a significant amount of energy on transmissions.

The phenomenon of link correlation is first experimentally studied in [22] where Zhu et al. proposes a flooding design to reduce energy consumption on transmission by using implicit ACK inferred from link correlation. Later, Srinivasan [41] explores metrics that capture to what degree packet reception on different links is correlated. However, neither of them investigates flooding in low-duty-cycle wireless sensor networks.

VIII. Conclusion

In this paper we propose Correlated Flooding, an energy-efficient flooding design for low-duty-cycle WSNs that solves the problem caused by both low-duty-cycle operation and ACK implosion. Different from previous works that achieve broadcasting via costly unicasts, we adopt the flooding-tree-based solution and adapt it for low-duty-cycle operations by letting nodes of common parents wake up simultaneously. The flooding tree is constructed to be more energy efficient than the traditional energy-optimal trees by considering both link quality and link correlation, the phenomenon that the receptions of a broadcasting packet at different receiving nodes are not independent. More importantly, this is the first work that both experimentally and mathematically studies the impact of link correlation on the efficiency of flooding. It is also the first work that exploits link correlation in the flooding design to deal with the ACK implosion problem. Correlated Flooding is evaluated in various simulations and a testbed experiment. Results indicate that our design saves more than 66% energy on ACKs and 15% $\sim$ 50% energy on data packets for most network settings, while having similar flooding delay and reliability.

Appendix

We prove that given two correlated links with link qualities p_1 and p_2 ($p_1 \geq p_2$), $q_{1/2}$ is greater than q_1. In III-A we showed that when links are correlated, the reception of a broadcasting packet for a node with worse link indicates a higher probability of the reception for the node with better link, i.e., $p_{1/2} > p_1$.Based on total probability equation:

$$p_1 = p_2 p_{1/2} + q_2(1 - q_{1/2})$$

$$q_{1/2} = \frac{p_2 p_{1/2} + q_2 - p_1}{q_2} > \frac{p_2 p_1 + q_2 - p_1}{q_2} = \frac{q_2(1 - p_1)}{q_2} = q_1$$

As a result, $q_{1/2} > q_1$, Q.E.D.

References

[1] R. Szewczyk, A. Mainwaring, J. Anderson, and D. Culler, "An Analysis of a Large Scale Habit Monitoring Application," in *SenSys'04*, 2004.

[2] X. Liu, J. Cao, S. Lai, C. Yang, H. Wu, and Y. Xu, "Energy Efficient Clustering for WSN-based Structural Health Monitoring," in *INFOCOM'11*, 2011.

[3] Q. Zhang, G. Sobelman, and T. He, "Gradient-Driven Target Acquisition in Mobile Wireless Sensor Networks," *Mobile Ad-hoc and Sensor Networks*, 2006.

[4] Y. Zhang, L. Zhang, and X. Shan, "Robust Distributed Localization with Data Inference for Wireless Sensor Networks," in *ICC'08*, 2008.

[5] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A Wireless Sensor Network for Structural Monitoring," in *SenSys'04*, 2004.

[6] S. Guo, Y. Gu, B. Jiang, and T. He, "Opportunistic Flooding in Low-Duty-Cycle Wireless Sensor Networks with Unreliable Links ," in *MobiCom'09*, 2009.

[7] 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 *SenSys'09*, 2009.

[8] S. Lai and B. Ravindran, "On Multihop Broadcast over Adaptively Duty-Cycled Wireless Sensor Networks," in *DCOSS'10*, 2010.

[9] W. Lou and J. Wu, "Double-Covered Broadcast (DCB): A Simple Reliable Broadcast Algorithm in MANETs," in *INFOCOM'04*, 2004.

[10] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," in *MobiCom'99*, 1999.

[11] M. Impett, M. S. Corson, and V. Park, "A receiver-oriented approach to reliable broadcast ad hoc networks," in *WCNC'00*, 2000.

[12] Y. Gu and T. He, "Data Forwarding in Extremely Low Duty-Cycle Sensor Networks with Unreliable Communication Links," in *SenSys '07*, 2007.

[13] G. S. M. Maroti, B. Kusy and A. Ledeczi, "The Flooding Time Synchronization Protocol," in *SenSys'04*, 2004.

[14] P. Dutta and D. Culler, "Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications," in *SenSys'08*, 2008.

[15] D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A High Throughput Path Metric for Multi-Hop Wireless Routing," in *MobiCom'03*, 2003.

[16] A. Woo, T. Tong, and D. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," in *SenSys'03*, 2003.

[17] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, and J. A. Stankovic, "ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks," in *SenSys'06*, 2006.

[18] S. Lin, G. Zhou, K. Whitehouse, Y. Wu, J. A. Stankovic, and T. He, "Towards Stable Network Performance in Wireless Sensor Networks," in *SenSys'06*, 2006.

[19] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "Understanding the Causes of Packet Delivery Success and Failure in Dense Wireless Sensor Networks," in *RTSS'09*, 2009.

[20] W. Si and C. Li, "RMAC: A Reliable Multicast MAC Protocol for Wireless Ad Hoc Networks," in *ICPP'04*, 2004.

[21] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, "SMACK: a SMart ACKnowledgment scheme for broadcast messages in wireless networks," in *SIGCOMM Comput. Commun. Rev.*, 2009.

[22] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang, "Exploring Link Correlation for Efficient Flooding in Wireless Sensor Networks," in *NSDI'10*, 2010.

[23] B. Das and V. Bharghavan, "Routing in Ad-Hoc Networks Using Minimum Connected Dominating Sets," in *ICC '97*, 1997.

[24] Q. Cao, T. F. Abdelzaher, T. He, and J. A. Stankovic, "Towards Optimal Sleep Scheduling in Sensor Networks for Rare Event Detection," in *IPSN'05*, 2005.

[25] Q. Huang, C. Lu, and G.-C. Roman, "Spatiotemporal Multicast in Sensor Networks," in *SenSys'03*, 2003.

[26] S.-J. Park, R. Vedantham, R. Sivakumar, and I. F. Akyildiz, "GARUDA: Achieving Effective Reliability for Downstream Communication in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, 2008.

[27] M. J. Miller, C. Sengul, and I. Gupta, "Exploring the Energy-Latency Trade-Off for Broadcasts in Energy-Saving Sensor Networks," in *ICDCS'05*, 2005.

[28] P. Kyasanur, R. R. Choudhury, and I. Gupta, "Smart Gossip: An Adaptive Gossip-based Broadcasting Service for Sensor Networks," in *MASS'06*, 2006.

[29] F. Stann, J. Heidemann, R. Shroff, and M. Z. Murtaza, "RBP: Robust Broadcast Propagation in Wireless Networks," in *SenSys '06*, 2006.

[30] J. W. Hui and D. Culler, "The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale," in *SenSys'04*, 2004.

[31] T. W. Cho, M. Rabinovich, K. K. Ramakrishnan, D. Srivastava, and Y. Zhang, "Enabling Content Dissemination Using Efficient and Scalable Multicast," in *INFOCOM'09*, 2009.

[32] D. Zappala, "Alternate Path Routing for Multicast," in *IEEE/ACM Transactions on Networking*, 2004.

[33] K. R. W. Lou and Y. Zhang, "Multi-user broadcast authentication in wireless sensor networks," in *Encyclopedia on Ad Hoc and Ubiquitous Computing*, 2009.

[34] L. Li, G. Xing, Q. Han, and L. Sun, "Adaptive Voice Stream Multicast over Low-power Wireless Networks," in *RTSS'10*, 2010.

[35] J. Wu and Y. Wang, "A Non-Replication Multicasting Scheme in Delay Tolerant Networks," in *MASS'10*, 2010.

[36] W. Gao and G. Cao, "User-Centric Data Dissemination in Disruption Tolerant Networks," in *INFOCOM'11*, 2011.

[37] F. Wang and J. Liu, "RBS: A Reliable Broadcast Service for Large-Scale Low Duty-Cycled Wireless Sensor Networks," in *ICC'08*, 2008.

[38] Feng Wang and Jianchuan Liu, "Duty-cycle-aware broadcast in wireless sensor networks," in *INFOCOM'09*, 2009.

[39] 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 *ICDCS'10*, 2010.

[40] X. Jiao, W. Lou, X. Wang, J. Ma, J. Cao, and X. Zhou, "Interference-Aware Gossiping Scheduling in Uncoordinated Duty-Cycled Multi-Hop Wireless Networks," in *WASA'10*, 2010.

[41] K. Srinivasan, M. Jain, J. I. Choi, T. Azim, E. S. Kim, P. Levis, and B. Krishnamachari, "The kappa factor: inferring protocol performance using inter-link reception correlation," in *MOBICOM'10*, 2010.