Understanding the Flooding in Low-Duty-Cycle Wireless Sensor Networks

Zhenjiang Li†‡, Mo Li†, Junliang Liu‡, and Shaojie Tang§
†School of Computer Engineering, Nanyang Technological University, Singapore
‡Department of Computer Science and Engineering, Hong Kong University of Science and Technology
§Department of Computer Science, Illinois Institute of Technology
Email: lzjiang@cse.ust.hk, limo@ntu.edu.sg, junll@cse.ust.hk, stang7@iit.edu

***Abstract*—In low-duty-cycle networks, sensors stay dormant most of time to save their energy and wake up based on their needs. Such a technique, while prolonging the network lifetime, sets excessive challenges for efficient flooding within the network. Tailored for obtaining short delay in low-duty-cycle networks, recently proposed flooding protocols have achieved some initial success. Many fundamental problems of flooding in low-duty-cycle networks, however, are still not well understood. In this paper, we thoroughly investigate how the flooding behaviors are fundamentally affected from theory to practice in a low-duty-cycle sensor network. We study how practical factors like duty cycle length and link loss affect the flooding delay. We mathematically quantify the performance deterioration caused by those factors and present initial learning in achieving efficient flooding against them. Our theoretical analysis brings us not only an in-depth understanding of several fundamental trade-offs in low-duty-cycle sensor networks, but also insights on the design of flooding protocols that can approach excellent performance.**

***Keywords*-Flooding, Low-duty-cycle, wireless sensor networks.**

I. Introduction

Wireless Sensor Network (WSNs) are usually deployed over wide fields composed of a number of sensors communicating with each other [1]. Flooding serves as a fundamental operation for information exchange in such large-scale distributed systems. During the flooding process, information is disseminated from a source to the whole network [2]. Such information normally includes sensory data [3], user queries over the network [4], a variety of control messages for network configuration [5], localization [6], [7], diagnosis [8], [9] and so on. Thus, the flooding process severely affects the efficiency of wireless sensor networks. A well-designed flooding scheme pursues low dissemination delay and it is critical to a wireless sensor network.

In order to achieve smaller flooding delay, there have been a number of flooding protocols proposed for wired and wireless networks. Unfortunately, recent literatures reveal that traditional flooding protocols suffer extremely poor performance when they are used in low-duty-cycle WSNs [10][11]. In low-duty-cycle networks, sensors stay dormant most of time to save energy and wake up in need basis. Sensors repeat this on/off working paradigm such that the energy consumption at each sensor is reduced and the entire lifetime of the network can be prolonged. Such a technique, on the other hand, sets excessive challenges for efficient flooding in the network. Due to the low-duty-cycle property, a sender node can only transmit when the receiver node is active, introducing inevitable delay during data delivery. Once one packet transmission fails, the sender must wait for a long time until the receiver wakes up again, and then launch the retransmission. Such a delay is unique in low-duty-cycle WSNs and it is referred to as **sleep latency** [12]. On the other hand, since sensors are not always active in low-duty-cycle networks, a sensor has to transmit a same packet multiple times so that the packet can be received by all its neighbors if those neighbors are not awake in the same time period. Such a phenomenon brings a fundamental difference in designing flooding protocols for low-duty-cycle WSNs. Basically, flooding in low-duty-cycle WSNs is achieved through a number of unicasts. A packet is delivered through a series of unicasts from the sender to multiple neighboring receivers [11][13][14][15].

Tailored for obtaining short delay in low-duty-cycle networks, recently proposed flooding protocols have achieved some initial success [10][11]. Many fundamental problems, however, are still not well understood, especially from the theoretical perspective. When a packet is flooded over a sensor network, naturally, we are interested in how fast flooding can achieve at most. Answering such a question lays the performance foundation of flooding in low-duty-cycle WSNs and it is still open to the community. Furthermore, practical factors like duty cycle length and imperfect link quality directly impact the flooding performance in real systems. How much does each of them affect the practical performance? Which one contributes most to the performance degradation? Is it always beneficial to set an extremely low-duty-cycle in the network? So far as we know, an instrument for answering all those important questions is still missing.

0190-3918/11 $26.00 © 2011 IEEE
DOI 10.1109/ICPP.2011.56

In this work, we theoretically study the low-duty-cycle sensor network and thoroughly analyze the packet flooding process. We first derive the lower bound of the flooding delay that can be achieved in theory. We find that the duty cycle period is a major factor that increases the flooding delay. Due to the intrinsic blocking effect in low-duty-cycle sensor networks, flooding a number of packets cannot be fully pipelined. Nevertheless, such blocking effect is limited and the multiple packets flooding can be pipelined in a certain degree. We further generalize our analysis with practical constraints like imperfect link qualities. We observe that the transmission loss over links significantly magnifies the delay caused by the duty cycle length. Compared with existing protocol layer literatures, this is the first theoretical work that aims to understand the flooding behavior and its fundamental impact in low-duty-cycle sensor networks. We find that both 1) adjusting the duty cycle length to optimize the networking gain by trading off the flooding delay and the network lifetime and 2) a cross-layer design by combining the duty cycle configuration with the opportunistic forwarding, are critical but still missing in existing low-duty-cycle networks. In the end, we conduct extensive simulations driven by a 298-node real system trace to validate our theoretical analysis.

The rest of this paper is organized as follows: we present the preliminary of this paper in Section III. In Section IV, we theoretically analyze the flooding delay with regard to the duty cycle length and the link loss. Based on extensive trace-driven simulations, we valid our analysis in Section V. In the end, we conclude this paper in Section VI.

II. Related Works

As an important networking service, data flooding and forwarding has been extensively studied in low-duty-cycle sensor networks. [16] introduces a multi-parent forwarding technique and proposes a heuristic algorithm so that the fast delivery of time sensitive actuation commands can be achieved. [14] designs a dynamic switch-based packet forwarding scheme that optimizes data delivery ratio, end-to-end delay and energy consumption. In [17], authors utilize the capture effect to achieve fast flooding by ensuring that each node receives the flooding packet from at least one of its neighbors, and introducing new techniques to either recover from or prevent too many concurrent transmissions. Lai et al. in [18] present hybrid-cast, which is a broadcast protocol tailored to shorten the broadcast delay. Moreover, [19] offers a probability-based broadcast protocol. In [10], Wang et al. design a duty-cycle-aware flooding scheme tailored for the network with reliable links. Guo et al. propose the optimal energy tree-based and opportunistic flooding protocols in [11]. Opportunistic flooding makes the probabilistic forwarding decision at each sender based on the delay distribution along an optimal energy tree. In [20], we propose a flooding protocol utilizing both deterministic back-off and overhearing two components to approximate the optimal flooding performance in practice. So far as we know, all existing works focus on providing protocol-level solutions. None of them focuses on fundamentally understanding the flooding problem in low-duty-cycle WSNs.

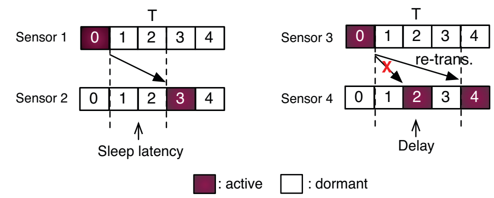


Fig. 1. Illustration of working schedule

There are also some works targeting at adapting the duty cycle length in sensor networks. In [21], authors introduce novel solutions for bounding sink-to-node communications in energy-harvesting sensor networks. They present an optimal solution for the sink-to-one case and its distributed implementation. In [22], authors propose DutyCon, a control theory-based dynamic duty cycle control approach. Based on feedback control theory, DutyCon features a queuing delay adaptation scheme that adapts the duty cycle of each node to unpredictable packet rates, as well as a novel energy balancing approach that extends the network lifetime by dynamically adjusting the delay requirement allocated to each hop. However, these works mainly study how to meet the flooding delay constraint. An instruction to configure the duty cycle length such that the flooding delay and the system lifetime can be well balanced is still missing.

For the opportunistic forwarding technique, [23] uses short-term estimation of wireless links to accurately identify short-term stable periods of transmission on bursty links. In [24], authors present Collective Flooding (CF), which exploits the link correlation to achieve flooding reliability using the concept of collective ACKs. Kim et al. in [25] conduct a comparison of opportunistic and deterministic forwarding in mobile wireless networks. Through our study, we find that existing opportunistic forwarding schemes are designed separately from the duty cycle length optimization. As we will see that both the duty cycle length and the link loss together dominate the flooding performance, a cross-layer design will be a promising way to improve the flooding service in low-duty-cycle WSNs.

III. Preliminary

In this section, we describe the basic network model and introduce the assumptions used in this paper.

A. System Model

As we mentioned before, a sensor in duty-cycle sensor networks alternates between two states: the active state and the dormant state. In the active state, the sensor opens its radio to transmit or receive data packets. In the dormant state, the sensor disables all its function modules expect a timer to wake itself up. The timer wakes up a sensor only when 1) the sensor should switch to be active based on its working schedule or 2) the sensor has a packet to send to a neighbor which is active at that moment. By such a means, a sensor can become active to transmit a packet at any time; nevertheless, it can receive a packet only when it is active. As shown in Fig. 1, sensor 1 wakes up at time slot 0, receives a packet and forwards it. However, sensor 1 cannot send the packet out immediately and needs to wait until sensor 2 wakes up at time slot 3.

The **working schedule** defines the active-dormant pattern of a sensor. Normally, the working schedule of a sensor is periodic [11]. We denote the duration of the cycle period as T. In one period, the duration T is divided into multiple time slots with equal length. Each sensor randomly selects several slots in which it stays to be active. The remaining slots serve for the dormant state. The number of active slots over the number of dormant slots defines the **duty ratio**. **Low-duty-cycle** means that the duty ratio is extreme small (e.g. $\leq 5\%$). In a low-duty-cycle sensor network, each sensor repeats the T-time working schedule in its lifetime. Without loss of generality, we will conduct a *normalized* duty-cycle-based analysis in Section IV. Under such a duty cycle, a sensor randomly picks up one active time slot in one period and repeats this selected working pattern. The duty ratio of each sensor is simply $1/T$.

During the flooding process, a source node generates packets to flood. All other (nominal) sensors in the network act as packet receivers and forwarders. Later in our discussions, we suppose the network contains N sensors and one source. A unique ID numbered from 1 to N is assigned to each sensor and the source node has ID 0.

B. Assumptions

In this paper, we make following assumptions in our network model:

- **Slotted time model**: the time axis is divided into time slots with equal length. The duration of each time slot is appropriate for the transmission of one packet.
- **Local synchronization**: the system works in a locally synchronized mode. With local synchronization, a sender knows when it shall wake up to transmit a packet to each of its neighbors according to their working schedules. There have been many low-cost local synchronization mechanisms proposed to this end [26][27].
- **Radio model**: the radio equipped in each sensor is semi-duplex, i.e., a sensor can either transmit or receives a packet at any given time slot, but not both.
- **Unreliable links**: a transmission may fail and a retransmission is needed to compensate this failure. The delay increases accordingly. In Fig. 1, sensor 3 fails its transmission to sensor 4 at time slot 2. It needs to wait one more slot before sensor 4 wakes up again.
- **Unicast**: in low-duty-cycle WSNs, it is easy for us show that it is rare for multiple neighboring sensors waking up at the same time period. As a result, to flood one packet, a sensor needs to transmit the same packet to each of its neighbors one by one. Thus the flooding is achieved via a number of unicasts [11].

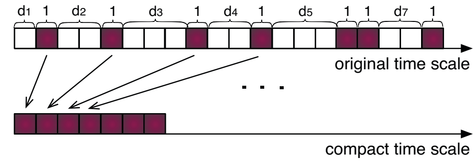


Fig. 2. Original time scale & Compact time scale

C. Problem Statement

Flooding normally involves one or more packets and we denote the total number of packets as M. The source sequentially injects M packets into the network. Suppose packet q is the last packet received by all sensors in the network, where $q \in [0, M-1]$. In this paper, we will study the delay performance to receive all these M packets.

Due to the sleep latency in the low-duty-cycle network, a packet may be queued at the source side waiting for prior packets delivered before its own transmission, i.e. the FCFS policy. We denote the number of packets injected into the network before packet q as K_q. On the other hand, after packet q enters the network, it needs to be relayed multiple hops until it reaches all sensors in the network. At each intermediate relay node, packet q follows the FCFS policy as well. For the last copy of packet q received by the network, we denote the total number of waitings for prior packets experienced at all intermediate relay nodes during its multi-hop dissemination as W_q. Based on K_q and W_q, we can define *Flooding Waiting Limit* (FWL) as $\min_{p\in[0,M-1]}\{K_p + W_p\}$ that represents the minimum number of waitings (imposed by the FCFS policy) needed for the last copy of packets q to be received during the flooding. To examine how fast flooding can achieve at most in low-duty-cycle networks, we are interested in *Flooding Delay Limit* (FDL) as follows:

$$FDL = \sum_{h=1}^{FWL} (d_h + 1), \tag{1}$$

where d_h is the queueing delay experienced for the h-th. waiting and 1 stands for the transmission delay of packet q. Due to the low-duty-cycle property, there may exist idle time slots, during which no transmissions occur, in the entire flooding process. To theoretically examine FDL in Eq. (1), we use a corresponding *compact time scale* analysis. As shown in Fig. 2, the time slots of actual transmissions in the original time scale are sequentially mapped to the compact time scale while all idle time slots are excluded. As we will show in the next section, by utilizing the concept of compact time scale, we simplify the mathematical analysis and obtain the achievable FWL. To avoid confusion, we use c and t to denote the time slot indices in the *compact* time scale and the *original* time scale, respectively.

IV. Theoretical Performance Analysis

In this section, we thoroughly study the delay performance of flooding in low-duty-cycle networks. Before the formal analysis, we introduce following notations first:

- t is the index of time slots over the original time scale, $t = 0,1,2,....$
- c is the index of time slots over the compact time scale, $c = 0,1,2,....$
- p is the index of packets, $p = 0,1,2,....$
- N is the number of nominal sensors in the network *excluding* the source.
- $\mathbb{X}_p^{(c)}(n)$ is an $n \times 1$ vector indicating the possession of packet p at each **node** (i.e. either a source or a nominal sensor) in the network at the beginning of time slot c.
- $x_p^{(c)}(i)$ indicates the element of $\mathbb{X}_p^{(c)}(n)$ in row i, $i = 0,1,...,n-1$, where $x_p^{(c)}(i) \in \{0,1\}$.
- $\mathbb{S}_p^{(c)}(n)$ is an $n \times n$ matrix indicating how packet p is disseminated among n nodes in time slot c.
- $s_p^{(c)}(i,j)$ indicates the element of $\mathbb{S}_p^{(c)}(n)$ in row i and column j, $i = 0,1,...,n-1$, $j = 0,1,...,n-1$, where $s_p^{(c)}(i,j) \in \{0,1\}$.

A. Flooding Delay Limit in Low-duty-cycle Networks

As aforementioned, we figure out FWL first. To this end, we conduct the compact time scale analysis to obtain its achievable value. Then, we further derive the corresponding flooding delay limit, FDL. We start the discussion from the case that a single packet is flooded in the network.

In the single packet flooding case, the source node has a packet ready to send at the beginning of time slot 0 (i.e. $c = 0$). Without loss of generality, we denote the index of this packet as p. Before packet p reaching all N sensors, sensors with this packet cooperatively deliver it to all other sensors without packet p yet. We use a $(1+N) \times 1$ column vector $\mathbb{X}_p^{(c)}(1+N)$ to represent all the nodes (including the source) that have already possessed packet p at the beginning of time slot c, $c = 0,1,....$ If node i, $i = 0,1,...,N$, has already had packet p, $x_p^{(c)}(i)$ is 1; otherwise, it is 0. At time slot c, if node j transmits packet p to node i, $s_p^{(c)}(i,j)$ is 1; otherwise, it is 0. Therefore, the evolution of packet p's dissemination in the network can be captured by the following equation:

$$\mathbb{X}_p^{(c+1)}(1+N) \quad = \quad \mathbb{X}_p^{(c)}(1+N) + \mathbb{S}_p^{(c)}(1+N) \times \mathbb{I}, \quad (2)$$

where $\mathbb{I}$ is a $(1+N) \times 1$ unit vector and $c = 0,1,....$ Note that Eq. (2) is universal to describe the evolution processes of a single packet dissemination in existing flooding protocols [10][11]. Different flooding protocols lead to different packet dissemination strategies, i.e. different $\mathbb{S}_p^{(c)}(1+N)$s at each time slot c; thus, the time consumed by various schemes for flooding a packet to all N sensors can be different. For our problem, we are willing to answer the question as mentioned before: how fast can flooding achieve at most. We can rephrase this question as figuring out the limit of FWL bounded from below as follows:

$$FWL \triangleq \min_c \{X_p^{(c)}(1+N) = 1+N\}, \quad (3)$$

where $X_p^{(c)}(1+N)$ indicates the number of 1s in $\mathbb{X}_p^{(c)}(1+N)$.

We find that the packet evolution pattern of Eq. (2) produces a *Galton-Watson process* and the *supercritical process* serves as a classic technique that can be applied to derive the flooding delay limit theoretically [28]. To be more precise, the sequence $\{X_p^{(c)}(1+N)\}_{c=0,1,...}$ forms a Galton-Watson process, where $X_p^{(0)}(1+N) = 1$ and $1 < E[X_p^{(1)}(1+N)] \leq 2$ (i.e., due to the unreliable wireless links).

Definition 1: **Supercritical process**: if $\{X_p^{(c)}(1+N)\}_c$ is a Galton-Watson process, $\{X_p^{(c)}(1+N)/\mu^c\}_c$ forms a supercritical process, where $\mu \triangleq E[X_p^{(1)}(1+N)]$. In addition, $1 < \mu \leq 2$.

Lemma 1: If $1 < \mu < \infty$, $\{X_p^{(c)}(1+N)/\mu^c\}_c$ *almost surely* converges to a random variable X such that $E[X] = 1$ and $Var[X] = \frac{\sigma^2}{\mu^2-\mu}$, where $\sigma \triangleq Var[X_p^{(1)}(1+N)]$.

The proof of Lemma 1 is given by Theorem 2.2.1 in [28]. With Lemma 1, we are ready to derive FWL needed for a single packet flooding in the following lemma.

Lemma 2: **In a WSN with one source and N sensors, when N is large, the average flooding waitings for any packet p, $E[FWL]$, equals $\lceil \frac{log_2(1+N)}{log_2(\mu)} \rceil$.**

Proof: According to Lemma 1 and Eq. (3), when N is large, we have

$$\lim_{N\to\infty} \frac{X^{(FWL)}}{\mu^{FWL} \times X} = \lim_{N\to\infty} \frac{1+N}{\mu^{FWL} \times X} \to 1$$
$$\Rightarrow \quad FWL = \frac{log_2((1+N)/X)}{log_2(\mu)}. \qquad \text{a.s.}$$

Since FWL is an integer, the following equation holds

$$FWL = \lceil \frac{log_2(1+N)log_2(X)}{log_2(\mu)} \rceil \quad (4)$$
$$\Rightarrow \quad E[FWL] = \lceil \frac{log_2(1+N)}{log_2(\mu)} \rceil. \quad (5)$$

■

As μ changes from 1 to 2, Eq. (5) shows that FWL is not upper bounded since the wireless links can be unlimited lossy.

On the other hand, from Lemma 1, we know that both the mean (i.e. $E[X] = 1$) and the variance (i.e. $\frac{\sigma^2}{(\mu^2-\mu)}$) of X are small constants. According to Chebyshev's Inequality, we can derive that the probability of X being much larger than its average value is extremely small. Mathematically, for any $\alpha > 1$, we have

$$Pr\{X > \alpha \times E[X]\} = Pr\{X > \alpha \times 1\} < \frac{\sigma^2}{(\alpha-1)^2(\mu^2-\mu)}.$$

It means that $log_2(\frac{1+N}{X}) = log_2(1+N) - log_2 X$ can be well approximated by $log_2(N+1)$ with high probability. Therefore, according to Eq. (4) and $1 < \mu \leq 2$, we get

$$\begin{aligned} FWL &= \lceil \frac{log_2((1+N)/X)}{log_2(\mu)} \rceil \geq \lceil log_2((1+N)/X) \rceil \\ &= \lceil log_2(1+N) \rceil. \qquad \text{w.h.p.} \end{aligned} \tag{6}$$

At the first glance, the flooding waitings limit given in Eq. (6) seems trivially achievable by a binary tree. However, as suggested in the next subsection, the binary tree is not efficient enough to achieve such a minimum value. The importance of Lemma 2 is as follows. The conclusion made in Lemma 2 can be used to obtain FWL that is achievable in a general case, based on which we can further derive the fundamental delay limit (i.e. FDL) for multiple packets flooding.

An intuitive guess about FWL in the general case is whether it is simply linearly increased with the total number of packets flooded. Through our study, we find that the flooding waitings (i.e. time slots in the compact time scale) needed are beyond this intuition. Consider the problem complexity, we start our further analysis with making two simplifying assumptions:

- **I**: the radio equipped in each sensor is full-duplex.
- **II**: $N = 2^n$ for some positive integer n, where $n \geq 0$.

Assumptions **I** and **II** will be relaxed by the end of this subsection as we proceed. We will further examine how much does link loss widen the gap between the fundamental limit and the performance in reality. In this subsection, however, we for a moment ignore the impact of link loss by assuming wireless links are reliable. We will relax this assumption and conduct a detailed discussion about link loss in next subsection. To ease the presentation, we name a network with the reliable communication link as an *ideal* network in this subsection.

1) Flooding delay limit in ideal networks: In order to achieve a short flooding delay when multiple packets are flooded, sensors in the network should "*well cooperatively*" disseminate these packets. The major challenges are as follows: after the system runs for a while, each nominal sensor in the network usually has more than one packet in its local buffer. Probably, some of them have already been retrieved by the whole network. In other words, these packets do not need to be transmitted any more. We call this type of packets "expired". It is not easy for a sensor to decide whether a packet is expired or not. Even if a sensor is aware which packets are not expired, since it can send out one packet in each time slot at most, it is also difficult to determine which non-expired packet should be transmitted at first.

Algorithm 1 Matrix-based multi-packet flooding algorithm

Initialization: $c = p = 0$.

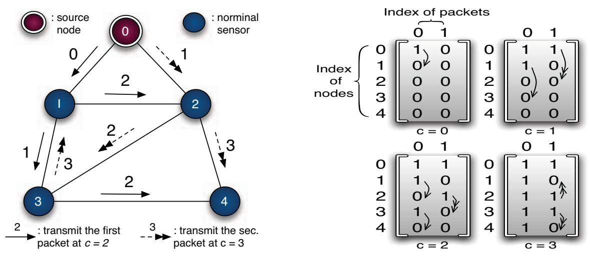
while flooding does not terminate **do**
  if $p < M$ **then**
    $x_p^{(c)}(0) = 1, x_p^{(c)}(i) = NIL$, $i = 1,2,...,N$.
  end if
  for $i = 0$ to $N-1$ **do**
    if $f(i,c) \neq NIL$ **then**
      Node i transmits packet $f(i,c)$ to the node $(2^{c \ mod \ n} + i) \ mod \ N$.
      /*If $(2^{c \ mod \ n} + i) \ mod \ N$ is 0, the packet is delivered to node N.*/
    end if
  end for
  $c = c+1, p = p+1$.
end while

Against above issues, we introduce the concept of *expired time* for each packet. Based on FWL derived in Eq. (6), we define the expired time for every packet (e.g. p) as $K_p + \lceil log_2(N+1) \rceil$ over the compact time scale. If the index p of a packet is smaller than $K_p + \lceil log_2(N+1) \rceil$, packet p is treated as "expired". Later, we will show that the defined expired time is long enough for each packet under assumptions **I** and **II** in ideal networks. According to the expired time, a sensor can categorize all its received packets into expired and non-expired two groups. In addition, among all non-expired packets, we propose to transmit the most recently received non-expired packet first and use $f(i,c)$ to represent such a packet, needed to be transmitted, at sensor i for time slot c. Note that if there is no such a packet, $f(i,c)$ returns NIL. As we will see, this simple strategy works very effectively. How to achieve the flooding waiting limitation over compact time scale is given in Algorithm 1.

In Fig. 3, we give a simple example to illustrate Algorithm 1. Fig. 3 shows that the flooding waitings experienced by each packet in Algorithm 1 have achieved the limit derived in Eq. (6). Since packets are pushed into to the network one after one immediately, by intuition, FWL of flooding multiple packets has been achieved. Thus, we can further derive the flooding delay limit based on the obtained FWL. This statement turns out to be true and it is formally introduced in Lemma 3.

Lemma 3: **Under assumptions I and II in an ideal low-duty-cycle network with one source and N sensors, if M packets flooded by the source, Flooding Delay Limit equals $M + \lceil log_2(N+1) \rceil - 1$.**

Readers can easily verify the conclusion of Lemma 3. Lemma 3 indicates that with assumptions **I** and **II**, flooding



(a) Topological illustration (b) Matrix-based illustration

Fig. 3. Example of Algorithm 1

multiple packets can be well pipelined in ideal low-duty-cycle networks. However, as we will show later, they cannot be fully pipelined in the general situation. Now, we are ready to relax assumption **I**.

*2) Relaxing assumption **I**:* With assumption **I**, a sensor can either transmit or receive a packet in one time slot only. After relaxing this assumption, we review the example in Fig. 3 again. We can observe that there exist two types of time slots.

- In the first type of time slots, each node conducts *only* one of three following actions: transmitting a packet, receiving a packet or keeping idle. Time slots 0, 1 and 3 all belong to this category.
- All remaining time slots belong to the second type. In this type of time slots, there must be some nodes both transmitting and receiving packets in the same time slot, i.e. time slot 2 in Fig. 3(b).

Without the full-duplex ratio assumption, transmitting and receiving packets cannot be conducted simultaneously any more. Hence, we need to modify Algorithm 1 accordingly for the second type of time slots. We use $c^\star$ to denote the second type of time slots. In addition, we set the duration of each $c^\star$ to be twice as long as the length of each c and evenly divide each $c^\star$ into two parts: $c_l^\star$ and $c_r^\star$. Actions of nodes in the first type of time slots keep the same as in Algorithm 1. On the other hand, for each time slot $c^\star$, we modify Algorithm 1 as follows: in the first half of time slot $c^\star$, i.e. $c_l^\star$, half of N sensors who possess packet $p(=c_l^\star - c)$ transmit this packet to another half of the nodes without it yet; in the second half of time slot $c^\star$, i.e. $c_r^\star$, the remaining nodes who do not transmit at time slot $c_l^\star$ send packets just as what they are supposed to do at the originally corresponding time slot c in Algorithm 1 shown in Fig. 4. Based on Lemma 3 and the way how Algorithm 1 is modified, we can quantify the multi-packet flooding delay limit after assumption **I** is relaxed.

Theorem 1: **After assumption I has been relaxed in an ideal low-duty-cycle network with one source and N sensors, if the total number of packets generated by the source is M, the average overall multi-packet flooding delay limit is**

TABLE I
WAITINGS OF PACKETS IN THE NETWORK

$M < m$		$M \geq m$	
p	W_p	p	W_p
0	m	m	$m+(m-1)$
1	$m+1$	$m+1$	$m+(m-1)$
...	...	...	...
$M-1$	$m+(M-1)$	$M-1$	$m+(m-1)$

$$E[FDL] = \begin{cases} T(\frac{1}{2}m + M - 1) & \textbf{if } M < m \\ T(m + \frac{1}{2}M - 1) & \textbf{if } M \geq m, \end{cases}$$

where $m = \lceil log_2(1+N) \rceil$**.**

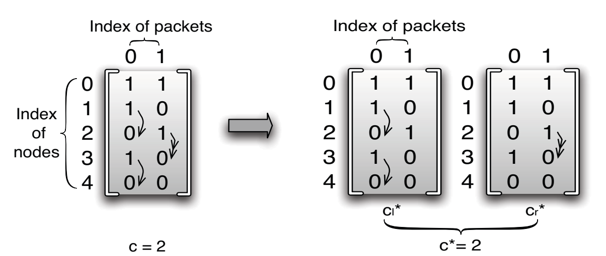
Proof: We prove the case when $M < m$ first. The case $M \geq m$ can be similarly proved. For each packet p in this case, its W_p can be tabulated in Table I according to Lemma 3. Since packets are generated by the source sequentially, at the beginning of time slot c, packet $p(=c)$ is ready at the source side. From the Table I, we can observe an interesting pattern: after packet p is pushed into the network at time slot $c(=p)$, it reaches the whole network within $p + \lceil log_2(1+N) \rceil + p$ time slots. By definition, FWL can be expressed as follows:

$$\begin{aligned} FWL &= \min_{p \in [0, M-1]} \{K_p + W_p\} \\ &= (M-1) + m + (M-1) \\ &= m + 2M - 2. \end{aligned}$$

According to the flooding delay policy proposed in Algorithm 1, i.e. $f(i,c)$, the distribution of d_h in Eq. (1) follows: $P(d_h = k) = \frac{1}{T}, k = 0, 1, ..., T-1$. People can verify that such a distribution does not hold for an arbitrary flooding policy. Thus, we have $E[FDL|FWL] = \frac{1}{2} \times T \times FWL$ and $FDL \leq T \times FWL$. There is only a factor 2 difference between the average value and the maximum value of FDL. As a result, we mainly focus on $E[FDL]$ in this theorem and we have

$$E[FDL] = T(\frac{1}{2}m + M - 1).$$

■



(a) Flooding in Algorithm 1 (b) Flooding in the modified algorithm

Fig. 4. Algorithm modification for the second type of time slots

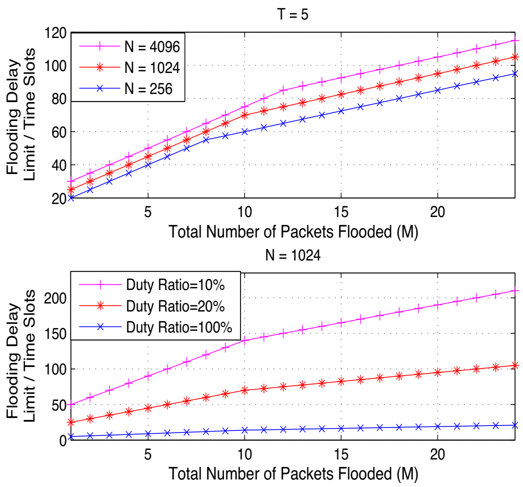


Fig. 5. FDL in Theorem 1

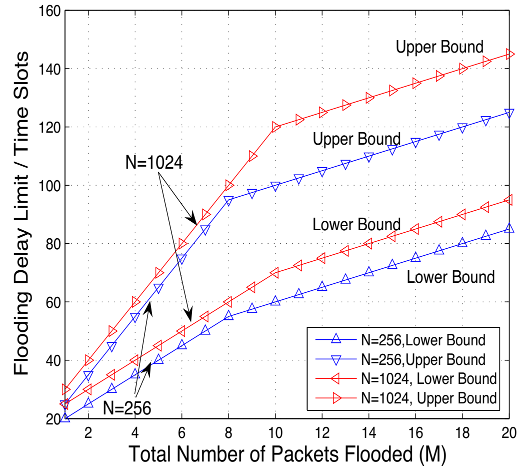


Fig. 6. FDL in Theorem 2

The overall flooding delay limit derived in Theorem 1 can be illustrated in Fig. 5. Different from our intuitive guess, the flooding delay limit is not a strictly linear function in terms of M. As M increases in Fig. 5, a knee point exists in each curve of the delay limit. The knee points actually deliver some positive news for us: when M is small, the flooding delay of each packet linearly increases with the total number of packets pushed into the network before. With more and more packets injected into the network, however, the flooding delay of each late coming packet does not further increase, i.e. the flooding delay of each packet is solely impacted by a certain number (i.e. $\lceil log_2(1+N)\rceil - 1$) of packets immediately before it. As we will see soon, we can obtain the similar conclusion after assumption **II** is relaxed.

Another important information delivered from Fig. 5 is that the duty cycle length tends to dominate the flooding delay limit as duty ratio becomes smaller in low-duty-cycle sensor networks. Similar observations have been reported in recent literatures [10][11] by experiments as well.

*3) Relaxing assumption **II**:* After this assumption is relaxed, it is extraordinarily difficult to derive the close form of the flooding delay limit for an arbitrary N in low-duty-cycle networks. Instead of obtaining the close form of the delay limit, we provide tight lower and upper bounds for it.

Theorem 2: **In an ideal network with one source and arbitrary N sensors, if the total number of packets generated by the source is M, the average overall flooding delay limit is within**

$$E[FDL] \in \begin{cases} T(\frac{1}{2}m+M-1) \sim T(m+\frac{3}{2}M-\frac{3}{2}) & \textbf{if } M<m \\ T(m+\frac{1}{2}M-1) \sim T(2m+\frac{1}{2}M-1) & \textbf{if } M\geq m. \end{cases}$$

where $m=\lceil log_2(1+N)\rceil$.

We omit the proof detail of Theorem 2 due to the page limitation. Though we do not obtain the exact mathematical expression for the delay limit, from Fig. 6, we can infer that the flooding delay limit in this case shares the similar behavior as Fig. 5. As a result, we have the following corollary:

Corollary 1: **The duty cycle length is a major factor that increases the flooding delay. Due to the intrinsic blocking effect in low-duty-cycle sensor networks, flooding a number of packets cannot be fully pipelined. Nevertheless, such blocking effect is limited within a certain number (i.e. $\lceil log_2(1+N)\rceil - 1$) of packets and flooding multiple packets can be pipelined in a certain degree.**

The limited blocking effect demonstrates that sensor networks might provide an efficient flooding service in principle. However, the low-duty-cycle technique, used to prolong the lifetime of the network, dominates the flooding delay limit in theory. As we will see soon, such a negative impact from the low-duty-cycle nature will be further amplified by link loss significantly in practice. Therefore, a guidance or a policy for configuring the duty cycle length is necessary to trade off the system lifetime and the flooding delay performance in the low-duty-cycle network design; nevertheless, so far as we know, such a work is still missing.

B. Impact of Link Loss

So far, we have discussed the impact of duty cycle on the flooding delay performance. In this subsection, we investigate the influence of the link quality on the delay performance. In real networks, link loss may occur from time to time. Once the transmission fails, the sender needs to wait for a sleep latency and launch a retransmission in the next active time slot of the receiver. Therefore, we quantify the link quality by introducing a variable k. A k-class link is defined as follows: with high probability, a packet can be transmitted successfully via k transmission(s), where $k=1,2,\ldots$. If $k=1$, the link

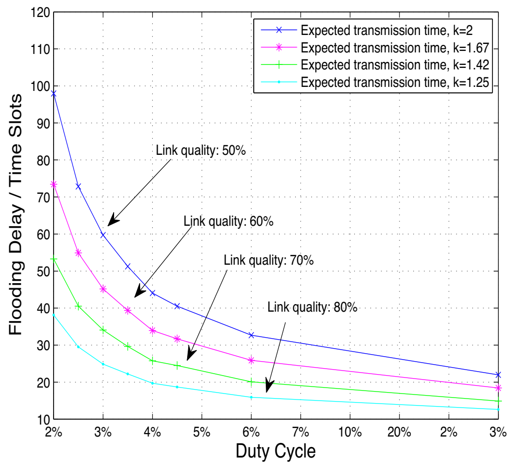


Fig. 7. Impact of link loss

quality can be viewed as perfect. In this subsection, we first discuss a homogeneous case, in which all the links share the same quality, i.e. they are all k-class links for some k. Then, we extend the discussion on the heterogenous case by the simulation. In a homogeneous-link-quality network with N nodes, the evolution pattern of any packet p's dissemination can be predicted by the following inequality:

$$X_p^{(t+1)} \leq X_p^{(t)} + \min\{X_p^{(\lceil \max\{0,t-kT\}\rceil)}, N - X_p^{(\lceil \max\{0,t-kT\}\rceil)}\}.$$

When $t \geq kT$, the evolution pattern can be approximated by

$$X_p^{(t+1)} \leq X_p^{(t)} + X_p^{(t-kT)}. \quad (7)$$

The eigenfunction of Eq. (7) can be easily written as

$$X^{kT+1} = X^{kT} + 1. \quad (8)$$

After obtaining the largest positive eigenvalue of Eq. (8), we can predict the flooding delay for given k and T as shown in Fig. 7 (the value of duty cycle in the figure is obtained from $1/T$).

More importantly, Fig. 7 essentially delivers some negative information. The flooding delay becomes very large in low-duty-cycle networks if there is high link loss. It is possible that the average time consumed on the flooding of each single packet is larger than the packet generation rate at the source side, i.e. early sent packets may significantly block the transmissions of late coming packets. As a result, the conclusion "the blocking effect is limited within a certain number of packets" found in ideal networks by Theorems 1 and 2 will not hold any more if there exists high link loss. In Section V, we indeed observe this phenomenon in simulations. The duty cycle length has already dominated the flooding delay (this increase is indispensable due to the energy saving propose), while link loss significantly magnifies such a negative impact. Therefore, in order to design an efficient

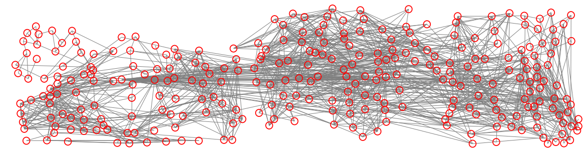


Fig. 8. Topology from GreenOrbs

flooding protocol, the link loss will be another critical issue to tackle and the opportunistic forwarding technique serves as a promising way to this end [11], since the opportunistic forwarding technique can grab more chances in the packet transmission to largely compensate the negative effect caused by link loss.

V. Analysis Validation

In this section, we conduct extensive trace-driven simulations to validate our theoretical analysis presented in Section IV.

A. Flooding Protocols in the Experiment

In this section, we examine the delay performances of three flooding schemes to validate our previous theoretical analysis, i.e., a theoretically optimal scheme with global optimization, a practical scheme with maximum possible local optimization, as well as a most advanced scheme proposed in existing literatures.

The first scheme has an optimal flooding delay performance by using oracle information, denoted as OPT. In OPT, each sensor (e.g. s) can always receive a packet from the neighbor who has the best link quality to s. In addition,we assume that there is no collision occurring in OPT. Clearly, sensors in OPT have the globally optimal flooding delay performance.

Due to the hidden terminal and the random working schedule, however, it is difficult to guarantee that each sensor receives a packet from the best neighbor in practice. In real implementations, each sensor maintains a subset of its neighbors in which those neighbors can hear each other. As a result, the carrier sense can be used to prevent them from sending packets at the same time. In [20], we propose using both Deterministic Back-off Assignment and Overhearing mechanisms to approach the optimal flooding performance once the subset of neighbors is given for each sensor. The scheme in [20] is denoted as DBAO and the performance of DBAO can be used to approximate the optimal flooding performance in practice.

The third flooding scheme used in the experiment is Opportunistic Flooding (OF), the most recent low-duty-cycle flooding scheme proposed in [11]. So far as we know, OF is the best practical flooding scheme known to the community working in low-duty-cycle wireless sensor networks. OF makes the probabilistic forwarding decision at each sender side based on the delay distribution along an optimal energy tree.

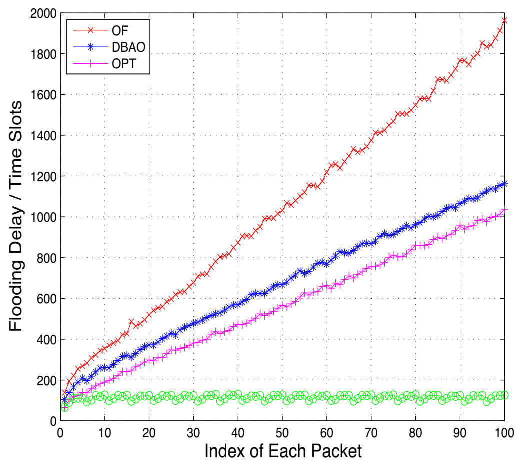


Fig. 9. Delay v.s. Packet index

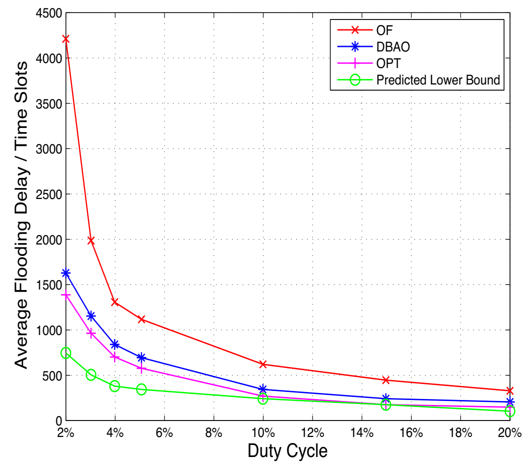


Fig. 10. Flooding delay v.s. Duty cycle

B. Simulation Settings

The network topology trace is obtained from a large-scale sensor network system, GreenOrbs [1] as shown in Fig. 8. In GreenOrbs, a number of sensors are deployed in the forest for environment monitoring. The system has kept working for more than half a year. The trace comprises 298 sensors and the link quality between each pair of neighbors is calculated based on six-month RSSI measurement results. By default, M and the duty cycle in three flooding schemes are set as 100 and 5% respectively. The flooding delay is based on 99% delivery ratio instead of 100% to eliminate the sensors which have extraordinarily low connectivity in the network. The *flooding delay* measured in this section is the average time consumed by each packet from the time it has been pushed into the network until it reaches 99% sensors in the network.

C. Validations

1) Network blocking effect: In subsection IV-A, we have pointed out that the flooding delay of each packet consists of the time elapsed for its actual transmission and the time consumption due to the blocking effect, i.e. the waiting delay, when multiple packets are sequentially transmitted. Due to the intrinsic blocking effect in low-duty-cycle sensor networks, flooding a number of packets cannot be fully pipelined. We measure the flooding delay of each packet with the three protocols and depict in Fig. 9. We further separate the transmission delay of each packet from the total delay and depict it in Fig. 9, i.e. the total delay is composed of the transmission delay and queueing (blocking) delay. From the statistics, we find that the actual packet transmission consumes almost the same in all three protocols. As more and more packets are flooded, however, the network blocking effect gradually dominates the total flooding delay. We further examine the total flooding delay of packets when different numbers of packets are flooded within the network. Observations from Fig. 9 matches the theoretical analysis in subsection IV-A.

2) With different duty cycle lengths: As shown in Fig. 10, the flooding delay performance deteriorates significantly when the duty cycle becomes small (e.g. 5%) in all three protocols. Such an observation matches our previous analysis in Section IV. The results reflect the negative side in setting the duty cycle extremely low in WSNs. While the system lifetime linearly increases (will be shown soon) as the duty cycle becomes small, the delay performance drops exponentially at the same time. As a result, the total energy benefit obtained with low-duty-cycle networks decreases exponentially. The performance gap between DBAO and OPT is mainly separated by the capability of the collision (e.g. hidden terminal) avoidance. Through our study, we find that this gap is difficult to be further reduced unless the hidden terminal can be well avoided in wireless networks. In addition, prediction inequality derived in subsection IV-B still serves as a valid lower bound of the flooding delay for real systems.

We further explore the impact of the duty cycle on the transmission failures. The importance to examine the transmission failures is related to the energy consumption. The receiver-side energy consumption is determined by its working schedule and the energy consumption for successful packet transmissions is the same in different systems. Thus, the energy consumed by both transmission failures and the duty cycle operation are mainly related to the energy consumption in the network. We illustrate the packet losses of three flooding schemes in Fig. 11 under the same network setting. We can observe that the total number of packet loss keeps almost the same in each of these systems when the duty cycle ratio changes. It implies that the energy consumption of each sensor is approximately

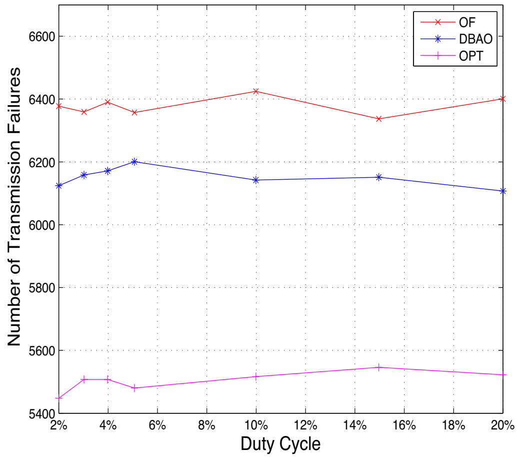


Fig. 11. Packet loss v.s. Duty cycle

linearly proportional to the duty cycle ratio. As a matter of fact, Fig. 10 and Fig. 11 together show that the overall benefit obtained in low-duty-cycle networks decreases exponentially as the duty cycle ratio decreases. In other words, it is **NOT** always beneficial to set the duty cycle extremely low in sensor networks.

VI. Conclusion and Future Work

In this paper, we thoroughly study the flooding problem in low-duty-cycle WSNs. We conduct theoretical studies on this problem and find the major factors that contribute to the flooding delay. We generalize our analysis with practical constraints, like the duty cycle length and the imperfect link quality. Our theoretical analysis in this paper brings us not only an in-depth understanding of several fundamental trade-offs in low-duty-cycle wireless sensor networks, but also insights on the design of flooding protocols that can approach excellent performance.

Based on our study, there are two research directions in our future work. Due to the joint impact of sleep latency and link loss, setting duty cycle length to be extremely low has shown not always beneficial. In the future, we will figure out how to configure the duty cycle length such that the obtained networking gains can be maximized. On the other hand, since the transmission loss over links significantly magnifies the delay caused by the duty cycle length, we will also explore how to utilize the opportunistic forwarding technique combined with the optimization of the duty cycle length to conduct a cross-layer design.

Acknowledgement

This work is supported by SUG COE_SUGRSS_20Aug 2010_1314 of Nanyang Technological University of Singapore.

References

[1] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X. Li, and G. Dai, "Canopy closure estimates with greenorbs: Sustainable sensing in the forest," in *ACM SenSys*, 2009.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, 2002.

[3] Z. Zhong, T. Zhu, D. Wang, and T. He, "Tracking with unreliable node sequences," in *IEEE Infocom*, 2009.

[4] M. Li, Y. Liu, J. Wang, and Z. Yang, "Sensor network navigation without locations," in *IEEE Infocom*, 2009.

[5] W. Dong, Y. Liu, X. Wu, L. Gu, and C. Chen, "Elon: enabling efficient and long-term reprogramming for wireless sensor networks," in *Proceedings of ACM SIGMETRICS*, 2010.

[6] Z. Yang and Y. Liu, "Quality of trilateration: Confidence-based iterative localization," *IEEE Transactions on parallel and distributed systems (TPDS)*, 2010.

[7] J. Lian, Y. Liu, K. Naik, and L. Chen, "Virtual surrounding face geocasting in wireless ad hoc and sensor networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 1, pp. 200–211, 2009.

[8] K. Liu, Q. Ma, X. Zhao, and Y. Liu, "Self-diagnosis for large scale wireless sensor networks," in *Proceedings of IEEE INFOCOM*, 2011.

[9] X. Miao, K. Liu, Y. He, Y. Liu, and D. Papadias, "Agnostic diagnosis: Discovering silent failures in wireless sensor networks," in *Proceedings of IEEE INFOCOM*, 2011.

[10] F. Wang and J. Liu, "Duty-cycle-aware broadcast in wireless sensor networks," in *IEEE Infocom*, 2009.

[11] 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.

[12] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *IEEE Infocom*, 2002.

[13] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay efficient sleep scheduling in wireless sensor networks," in *IEEE Infocom*, 2005.

[14] Y. Gu and T. He, "Data forwarding in extremely low duty-cycle sensor networks with unreliable communication links," in *ACM, Sensys*, 2007.

[15] J. Zhu, S. Chen, B. Bensaou, and K. L. Huang, "Tradeoff between lifetime and rate allocation in wireless sensor networks: A cross layer approach," in *IEEE Infocom*, 2007.

[16] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup scheduling in wireless sensor networks," in *ACM Mobihoc*, 2006.

[17] J. Lu and K. Whitehouse, "Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks," in *IEEE INFOCOM*, 2009.

[18] S. Lai and B. Ravindran, "Efficient Opportunistic Broadcasting over Duty-Cycled Wireless Sensor Networks," in *IEEE INFOCOM*, 2010.

[19] M. Miller, C. Sengul, and I. Gupta, "Exploring the energy-latency tradeoff for broadcasts in energy-saving sensor networks," in *IEEE ICDCS*, 2005.

[20] Z. Li and M. Li, "Flooding in Low-duty-cycle Wireless Sensor Networks," in *WASA*, 2011.

[21] Y. Gu and T. He, "Bounding Communication Delay in Energy Harvesting Sensor Networks," in *IEEE ICDCS*, 2010.

[22] X. Wang, X. Wang, G. Xing, and Y. Yao, "Dynamic duty cycle control for end-to-end delay guarantees in wireless sensor networks," in *IEEE IWQoS*, 2010.

[23] M. Alizai, O. Landsiedel, J. Link, S. Gotz, and K. Wehrle, "Bursty traffic over bursty links," in *Proceedings of ACM Sensys*, 2009.

[24] T. Zhu, Z. Zhong, T. He, and Z. Zhang, "Exploring link correlation for efficient flooding in wireless sensor networks," in *Proceedings of USENIX NSDI*, 2010.

[25] J. Kim and S. Bohacek, "A comparison of opportunistic and deterministic forwarding in mobile multihop wireless networks," in *ACM MobiSys*, 2007.

[26] J. Koo, R. K. Panta, S. Bagchi, and L. Montestruque, "A tale of two synchronizing clocks," in *ACM SenSys*, 2009.

[27] C. Lenzen, P. Sommer, and R. Wattenhofer, "Optimal clock synchronization in networks," in *ACM SenSys*, 2009.

[28] G. Sankaranarayanan, *Branching Processes and Its Estimation Theory*.