Cooperative Scheduling for Adaptive Duty Cycling in Asynchronous Sensor Networks

Jianhui Zhang1*, Zhi Li1, Feng Xia2, Shaojie Tang3, Xingfa Shen1 and Bei Zhao1,4

1College of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, Zhejiang 310018, China
2School of Software, Dalian University of Technology, Dalian, Liaoning 116620, China
3College of Science and Technology, Temple University, Philadelphia, PA19122, USA
4College of Biomedical Engineering and Instrument Science, Zhejiang University, Hangzhou, Zhejiang 310058, China
*Corresponding author: jhzhang@ieee.org

To support the sustainable operation of wireless sensor networks using limited energy, duty cycling is a promising solution. However, it is a challenge to guarantee each node communicating with its neighbors under duty cycle when the network is asynchronous. The challenge becomes bigger when nodes’ duty cycles are required to be adjusted separately according to their demands to save energy and achieve high channel utilization. Existing low power listening- and contention-based protocols are not energy-efficient and cannot ensure high channel utility. Additionally, synchronization-based media access control (MAC) protocols suffer from extra energy consumption and low synchronization precision. This paper proposes a localized and on-demand (LOD) duty cycling scheme based on a specifically designed semi-quorum system. LOD can adjust duty cycle of each node adaptively according to its demand so as to avoid channel contention, consequently achieving high channel utilization. This allows the fairness for channel access within asynchronous sensor networks. Extensive experiments are conducted on a real test-bed of 100 TelosB nodes to evaluate the performance of LOD. As compared with B-MAC, LOD substantially reduces contention for channel access and the energy consumption, thus improving the network throughput significantly.

Keywords: duty cycling; semi-quorum system; medium access control; wireless sensor networks

Received 29 December 2013; revised 14 June 2014
Handling editor: Zhangbing Zhou

1. INTRODUCTION

Limited energy and channel resources are two main constraints in wireless sensor networks (WSNs). Duty cycling is one of the key ways to save energy and increase channel utilization [1]. In the duty cycled WSNs, one necessary task is to manage the cooperation among communicating nodes efficiently. Each sensor node’s time is easy to shift because of the imprecise clock. Although there are some existing ways, such as time division multiple access (TDMA)-based protocols, to cooperate the communication within networks [2, 3], they require precise synchronization, which suffers from extra energy cost. The synchronization precision is by no means guaranteed when the network scale is large, such as GreenOrbs [4], because of the various uncontrollable factors in the natural environment [3]. Carrier sense multiple access (CSMA)-based protocol can avoid energy consumption on synchronization, but it costs much channel resources and energy on the channel contention. This results in low channel utilization. It becomes worse or even infeasible when the duty cycle is extremely low.

Furthermore, sensor nodes must adjust their own duty cycles according to their communication loads in many applications. A proper duty cycle enables each node to finish its communication load sufficiently and can be as low as...
possible to save energy. It is a challenging problem how to adjust each node’s duty cycle adaptively according to its load especially when the whole network is asynchronous and the communication loads among nodes are different.

This paper designs a localized and on-demand (LOD) scheme for duty cycle adjustment based on a specifically designed semi-quorum system (SQS), denoted by $\hat{Q}$, based on the quorum system (QS) [5]. LOD can adjust each node’s duty cycle adaptively according to its load and schedule the active time of every node without adopting any synchronization and channel contention protocol. The key properties of LOD owe to SQS, inheriting the advantages of some QSs with the non-empty intersection and rotation closure properties. QS was applied to establish channel control in dynamic spectrum access networks [6], to save power [7], to maximize throughput in limited information multiparty media access control (MAC) with QS [8] and to schedule duty cycling [9].

(i) We design the new QS, SQS, which satisfies the non-empty intersection and the rotation closure properties. SQS has high flexible and can obtain lower duty cycle and higher channel utilization and hence save more energy than the QSs under same load.

(ii) Based on SQS, the new protocol, LOD, is designed to adjust the duty cycle of each node adaptively according to its demand. Thus, the fairness of channel access and channel utilization are increased compared with existing contention-based MAC protocols.

(iii) LOD guarantees that each pair of neighboring nodes have proper common active time to communicate with each other. The rotation closure property guarantees that any pair of neighboring nodes can have rendezvous active time to communicate with each other without adopting any synchronization protocol within asynchronous networks. However, SQS is more flexible and can ensure each node to be more energy efficient and to have higher channel utilization than QS.

Each node requires its own duty cycle different from others because of its location and task to undertake in its network. So the fixed duty cycle cannot achieve high channel utilization. In many applications, such as canopy closure estimation [4], the duty cycle is simply fixed beforehand or independent among nodes as discussed in Section 1. Compared with the traditional duty cycling protocols, which adjust only the ratio of the active time to the period, LOD can adjust not only the ratio but also the slot active mode. LOD can assign each node with a set of active slots. When the ratio is same among nodes, they can wake up at different slots to increase the channel utilization and energy efficiency.

LOD can achieve higher channel utilization and fairness for channel access than previous MAC protocols. Energy efficiency under LOD is correspondingly higher than those under other protocols. The detailed contributions of this paper are listed as follows:

<table>
<thead>
<tr>
<th>Sym.</th>
<th>Description</th>
<th>Sym.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>Period</td>
<td>$</td>
<td>\ast</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Time slot</td>
<td>$P$</td>
<td>Power</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Clock shift</td>
<td>$\chi$</td>
<td>Color number</td>
</tr>
<tr>
<td>$E$</td>
<td>Energy</td>
<td>$M$</td>
<td>Interference model</td>
</tr>
<tr>
<td>$D$</td>
<td>Demand</td>
<td>$R$</td>
<td>SQS row</td>
</tr>
<tr>
<td>$Q$</td>
<td>QS</td>
<td>$L$</td>
<td>SQS column</td>
</tr>
<tr>
<td>$\hat{Q}$</td>
<td>SQS</td>
<td>$\Gamma$</td>
<td>Rotation operation</td>
</tr>
<tr>
<td>$Q$</td>
<td>Quorum</td>
<td>$r$</td>
<td>$z$ of selected BSQ</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Neighborhood</td>
<td>$I$</td>
<td>$z$ of selected NSQ</td>
</tr>
<tr>
<td>SQS</td>
<td>Semi-quorum system</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NSQ</td>
<td>Normal semi-quorum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSQ</td>
<td>Basic semi-quorum</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QS</td>
<td>Quorum system</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQ</td>
<td>Semi-quorum</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(ii) LOD guarantees that each pair of neighboring nodes have proper common active time to communicate with each other without synchronization. Therefore, energy and time are saved in comparison with the contention- and synchronization-based MAC protocols.

(iv) A quorum selection method is proposed to allocate quorums for each node so the worst case of channel utilization is bounded and the conflict of active time is decreased or avoided within networks.

(v) This paper sets up a real test-bed consisting of 100 TelosB nodes, and evaluates the performance of LOD compared with B-MAC.

Most of symbol, notation, abbreviation and their meanings are given in Table 1.

The organization of this paper is as follows. Section 2 presents the network model and the problem studied in this paper. In Section 3, we introduce the QS technology and present the designing and properties of the new system SQS. Section 4 presents the designing of our scheme LOD and suggests some improvement for it. Meanwhile, the performance of LOD is presented when the certain demand is implemented in asynchronous networks by Section 5. In Section 6, we implement our protocol on the real test-bed with 100 TelosB sensor nodes and analyze the performance of LOD with the experimental results. Section 7 discusses the related works on duty cycling and MAC designing in recent years. The paper is concluded in Section 8.

2. NETWORK MODEL AND PROBLEM STATEMENT

Suppose that the network contains some sensor nodes, $v_i$, $i = 1, 2, \ldots$. Nodes have the same receiving power $P_r$ and the listening power $P_l$, and different transmission power $P_t$ with
every two subsets intersect with non-empty $M$. Each period $T$ contains $|T|$ equivalent slots: $\tau_1, \ldots, \tau_T$. Every node may wake up in a subset of slots to communicate with its neighbors. The subset must contain sufficient common active slots with its neighbors’ so that the communication demand among them can be finished. Furthermore, each node’s clock is not precise and may result in an unpredictable clock shift $\delta$. The subset of slots has a shift $\delta$ accordingly. The challenging problem studied in this paper is how each node wakes up in proper time so that it has sufficient common time with its arbitrary neighbor even when the clock shift exists. Meanwhile, nodes must wake up in as less time as possible to save energy. This paper proposes the new system SQS and designs the new duty cycling scheme LOD so as to save energy and improve the channel utilization.

3. SEMI-QUORUM SYSTEM

Before designing SQS, the technology QS [10], denoted by $Q$, is firstly introduced.

3.1. Quorum system

Definition 3.1. Given a period $T$ (a set of slots) to be a universal set, a QS $Q \subseteq 2^T$ is a set of subsets of $T$ such that every two subsets intersect with non-empty, i.e. $Q_i \cap Q_j \neq \emptyset$ where $Q_i, Q_j \in Q$ are called quorums.

Denote the quorum allocated to node $v_i$ by $Q_i$, $Q_i \subset Q$. Figure 1 illustrates a grid QS, denoted by $Q_g$, containing two quorums: $Q_1$ and $Q_2$. Each of them composes of a column and a row. For example, $Q_1$ composes of the time slots with the black shade in Fig. 1. There are some kinds of QSSs, and only a few of them, such as the grid, torus and cyclic QS, have the important property: the rotation closure [11].

Definition 3.2. The rotation of the quorum $Q$ is defined by $\Gamma(Q, \delta) = \{(\tau_j + \delta) \mod |T| | \tau_j \in Q\}$, where $\delta$ is a non-negative value and $Q \in Q$.

Definition 3.3. A QS $Q$ satisfies the rotation closure property in $T$ if any two quorums $Q_i$ and $Q_j$, $Q_i, Q_j \in Q$, satisfy the condition: $\forall i \in \{0, \ldots, |T| - 1\}$: $Q_i \cap \Gamma(Q_j, \delta) \neq \emptyset$.

3.2. SQS design

This section designs the new QS and semi-QS (SQS). The reason to design the SQS is that it can achieve higher energy efficiency than the previous. SQS contains two kinds of semi-quorums: basic semi-quorum (BSQ) and normal semi-quorum (NSQ). It is defined as follows.

Definition 3.4. Given a period $T$ to be a universal set, an SQS $\hat{Q}$ is a set of subsets of the universal set $T$ and composed of BSQ $Q_b$, and NSQ $Q_n$. BSQ and NSQ have non-empty intersection, i.e. $Q_b \cap Q_n \neq \emptyset$, while there is no intersection among BSQs/NSQs.

BSQ and NSQ are two relative concepts. Let $\hat{Q}$’s size is $R \times L$. Choose the row in $\hat{Q}$ as BSQ and the column in $\hat{Q}$ as NSQ. Without loss of generality, let $Q_i$ and $Q_j$ be the $i$th row and $j$th column of $\hat{Q}$, respectively, where $1 \leq i \leq R$ and $1 \leq j \leq L$. Thus, we have $i$th BSQ: $Q_i = \{\tau_k, k = (i - 1)L + l, l = 1, \ldots, L\}$, and $j$th NSQ: $Q_j = \{\tau_k, k = j + (m - 1)L, m = 1, \ldots, R\}$. For example, if SQ $Q_1$ is a BSQ, then $Q_2$ and $Q_3$ are NSQs in Fig. 2. Vice versa. Figure 2 shows the SQS constructed based on the grid QS. Note that SQS can also be designed based on other QSSs, such as torus and cyclic QS [11], only if they satisfy the rotation closure property.

This paper takes the grid QS as the basic technology to construct the SQS, and analyzes its properties. The grid-QS based SQS contains $|T|$ slots distributed in a rectangle with $R \times L$ grids. In the grid-QS-based SQS, a BSQ is a set of a full row slots while a NSQ is a set of a full column slots. For example, the SQS contains three semi-quorums: $Q_1, Q_2$ and $Q_3$, where $Q_1$ is a BSQ and $Q_2$ and $Q_3$ are two NSQs in Fig. 2. $Q_1 = \{6, 7, 8, 9, 10\}$; $Q_2 = \{2, 7, 12, 17, 22\}$ and $Q_3 = \{4, 9, 14, 19, 24\}$. SQS has the properties including the non-empty intersection and the rotation closure as well. Furthermore, this paper designs the transmutation operation for SQS.

Non-empty intersection. SQS inherits the non-empty intersection property from the previous QSs as Lemma 3.1.
According to the method of constructing SQS in this subsection, any two different NSQ/BSQ cannot occupy the same column/row. Lemma 3.2 thus follows.

**Lemma 3.1.** There is at least one rendezvous slot between each pair of NSQ and BSQ in SQS.

**Lemma 3.2.** The intersection among NSQs/BSQs is empty in SQS.

**Rotation closure.** This important property will be beneficial to design SQS-based protocols in asynchronous networks. With Lemma 3.2, the rotation closure property of SQS is different from the previous QSs.

**Lemma 3.3.** In SQS, any pair of NSQ and BSQ satisfies the rotation closure property.

**Proof.** Let \( \hat{Q} \) be an SQS. \( \hat{Q} \) totally contains \(|T|\) elements, which are labeled by some consecutive integers with the row-first manner, such as the example in Fig. 2. \( \hat{Q} \)’s size is \( R \times L \). Choose an arbitrary BSQ \( Q_i \) and NSQ \( Q_j \), where \( Q_i, Q_j \in \hat{Q} \).

Without loss of generality, let \( Q_i \) and \( Q_j \) be the \( r \)th row and \( j \)th column of \( \hat{Q} \), respectively, where \( 1 \leq i \leq R \) and \( 1 \leq j \leq L \). Thus, \( Q_i = \{ \tau_k, k = (i-1)L + l, l = 1, \ldots, L \} \) and \( Q_j = \{ \tau_k, k = j + (m-1)L, m = 1, \ldots, R \} \). There always exist \( l \) and \( m \), equaling to \( j \) and \( i \), respectively, so that \((i-1)L + l = j + (m-1)L \). The above statement proves that \( Q_i \cap Q_j \neq \emptyset \).

The rotation of \( Q_i \) and \( Q_j \) are, respectively, \( \Gamma(Q_i, \delta) \) and \( \Gamma(Q_j, \delta) \), where \( \delta \) is a time value. \( \Gamma(Q_i, \delta) = \{ \tau_k + \delta \mod |T|, k = (i-1)L + l, l = 1, \ldots, L \} \). When \( \delta \mod |T| = 0 \), \( \Gamma(Q_i, \delta) = Q_i \) and thus \( \Gamma(Q_i, \delta) \cap \Gamma(Q_j, \delta) \neq \emptyset \). When \( \delta \mod |T| \neq 0 \), there are two cases: \( \delta \mod L = 0 \) and \( \delta \mod L = 0 \).

In each period, the energy consumption of a single node \( v \) can be calculated by the following equation.

\[
E_T = P_1 t(D_v) + P_2 t(D_v) + P_3 t_v^r
\]

where \( t_v^r \) is the total listening time of node \( v \) in each period. \( t(D_v) \) denotes the time to finish \( D_v \).

The energy consumption of the three SQSs in Fig. 4 can be calculated out by Equation (1). Suppose that the current consumption of each sensor node, such as TelosB and MICA2 [12], under receiving, listening and transmitting mode with power \(-5 \text{dBm}\) are 18.8, 18.8 and 14 mA, respectively. Suppose that node \( v_1 \) has three neighbors \( v_2, v_3 \) and \( v_4 \). \( v_1 \) selects two BSQs \( Q_1 \) and \( Q_2 \), \( v_2, v_3 \) and \( v_4 \) select the NSQs \( Q_3, Q_4 \) and \( Q_5 \), respectively. In the common slots between \( Q_1, Q_2 \) and \( Q_4, Q_5 \), such as slots 1, 2, 3, 13, 14 and 15, \( v_1 \) communicates with \( v_2, v_3 \) and \( v_4 \). In other slots, each node turns its radio into the states: listening or sleeping. Without loss of generality, we suppose the four nodes spend one half common time to receive and other half common time to transmit. In Fig. 4, the energy consumption of the three SQSs is illustrated in Table 2.

Define the network life as the duration from the network start moment to the first node using up its energy. In Table 2, the total energy consumption in the SQS of Fig. 4a is minimal but the network lifetime is \( \min\{1/17, 1/13, 1/15, 1/17\} \).
TABLE 2. Energy consumption of the three SQSs in Fig. 4 and the unit is $|\tau|/V$.

<table>
<thead>
<tr>
<th>Node</th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
<th>Total energy consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 4a</td>
<td>211.2</td>
<td>70.4</td>
<td>70.4</td>
<td>70.4</td>
<td>422.4</td>
</tr>
<tr>
<td>Figure 4b</td>
<td>436.8</td>
<td>32.8</td>
<td>32.8</td>
<td>32.8</td>
<td>535.2</td>
</tr>
<tr>
<td>Figure 4c</td>
<td>98.4</td>
<td>145.6</td>
<td>145.6</td>
<td>145.6</td>
<td>535.2</td>
</tr>
</tbody>
</table>

one $\tau V$. In the SQS of Fig. 4b, the energy consumption of $v_2$, $v_3$ and $v_4$ is minimized but the network lifetime is $\min\{\frac{1}{436.8}, \frac{1}{145.6}, \frac{1}{145.6}, \frac{1}{145.6}\} = \frac{1}{436.8}$. In the SQS of Fig. 4c, the energy consumption of all nodes is not minimal as well. But the network lifetime is $\min\{\frac{1}{436.8}, \frac{1}{32.8}, \frac{1}{32.8}, \frac{1}{32.8}\} = \frac{1}{436.8}$, and maximal among all SQSs in Fig. 4. Thus the energy minimization does not necessarily result in maximizing the network life. Although the SQSs in Fig. 4b and c consumes the same energy totally, the later has better energy consumption balance. The example explores the hint that the network life can be prolonged by balancing the energy consumption among nodes.

4. LOCALIZED AND ON-DEMAND

This section designs the new method LOD to let each node locally decide its duty cycle by cooperating with its neighbors when nodes are asynchronous.

4.1. Designing LOD

LOD composes of three steps: period assignment, tree construction and duty cycle designing.

Step I. Assign each node with a period by the vertex coloring. By the distributive coloring algorithm [13], each node is assigned a color. The minimal number of colors to color all nodes is denoted by $\chi$, which depends on the interference model $M$. Assigned a period $T_k$ to $v_i$ with color $k$, $k = 1, \ldots, \chi$.

Step II. Each node adjusts its duty cycle according to its demand. Each node $v_i$ collects its neighbor $v_j$’s ID, $v_j \in \theta(v_i)$. $v_i$ inquires its neighbors’ demands. The demand composes of transmitting or receiving data. Denote the data transmitted from $v_i$ to $v_j$ by $D_{ij}$.

(i) $v_i$ determines the new size of SQS by calculating the number of NSQs;
(ii) $v_i$ sends the number of NSQs and the size of SQS to its neighbors.

Note that each node is assigned a period in Step I. In the period, the node assigns itself some BSQs and some NSQs to its neighbors. Thus each node can wake up in its own and neighbors’ periods. The below context explains the detailed implementation of the above three steps and determines the values of the variables in these steps.

Demand time. In Step II, each node’s active time of depends on not only its demand but also other factors, such as the data rate. The data rate is calculated by averaging the data delivering over a duration so that variety of factors affecting the time to implement can be included. Recall that the demand consists of two types: data transmitting and receiving. $v_i$ thus adjusts its active time to be at least $t(D_{ij} + D_{ji})$. $v_i$ may have several neighbors, its active time should be the sum of the time to implement all the demands between it and its neighbors. Like lots of previous works [14–16], nodes can exchange their demands by some beacons at the initial phase of the network life or the beginning of each period under LOD.

Duty cycling. Step II wakes up each node in certain amount of quorums so that the demand can be implemented. The energy consumption in each period can be represented by two parts: $E_i$ consumed by $v_i$ and $E_j$ consumed by $v_j$’s neighbors $v_j$, $v_j \in \theta(v_i)$. The energy consumption $E_i$ is described in Equation (2).

$$E_i = P_i \sum_{v_j \in \theta(v_i)} t(D_{ij}) + P_r \sum_{v_j \in \theta(v_i)} t(D_{ji}) + P_l t_j^l$$  \hspace{1cm} (2)

where $t_j^l$ is $v_j$’s listening time. The energy consumption $E_j$ of the neighbor $v_j$ is described in Equation (3).

$$E_j = P_i t(D_{ji}) + P_r t(D_{ij}) + P_l t_j^l$$  \hspace{1cm} (3)

Suppose that $v_i$ selects $r_i$ BSQs and $v_j$ selects $l_j$ NSQs. $v_i$ should have sufficient common slots with its neighbors to implement the demands among them. The number of common slot should be at least $(1/|\tau|)(D_{ij} + D_{ji})$, where $|\tau|$ is the size of slot $\tau$, such as one second or millisecond. To save energy, there should be $r_i \times l_j = (1/|\tau|)(D_{ij} + D_{ji})$. On the other hand, each node should spend energy on the listening at those uncommon slots. For example, when a node selects the quorum $Q_3$ in Fig. 4a, it should listen at the slots 13 and 19. Therefore, the node $v_i$ and its neighbor $v_j$ have the listening time as described, respectively, in Equations (4) and (5).

$$t_j^l = |\tau| \left( r_i L_i - \sum_{v_j \in \theta(v_i)} r_j l_j \right)$$  \hspace{1cm} (4)

$$t_j^l = |\tau| l_j (R_i - r_j)$$  \hspace{1cm} (5)
where neighbors have different demand and thus select their own row amount \( r_j \), \( v_i \) has to select sufficient number of BSQs to meet the all demands of its neighbors so \( r_i = \max_{v_j \in \Theta(v_i)} r_j \).

In Step II, each node \( v_i \) selects BSQs for its own and NSQs for its neighbors so that the overall energy consumption can be reduced. The overall energy consumption of \( v_i \) and its neighbor can be obtained by adding two sides of Equations (2) and (3), respectively. Note that the time to implement the demand \( t(D_{ij} + D_{ji}) \) should be satisfied. Equations (2) and (3) indicate that the only way to change the energy consumption is to adjust the listening time. It thus can minimize the energy consumption in Equations (2) and (3) to save the overall energy. Since the listening power is same for all nodes, to minimize the overall energy consumption is equivalent to minimizing the listening time.

\[
\min \ t_i + \sum_{v_j \in \Theta(v_i)} t_j
\]

The solution to the above equation can be easily obtained by follows. According to Equations (4) and (5), we can have the below expression.

\[
t_i + \sum_{v_j \in \Theta(v_i)} t_j = |r| \left( \sum_{v_j \in \Theta(v_i)} r_j \right) - 2 \sum_{v_j \in \Theta(v_i)} t(D_{ij} + D_{ji})/r_j
\]

where the second equality is because \( r_i \times l_j = (1/|r|)t(D_{ij} + D_{ji}) \). The first inequality is because \( r_i = \max_{v_j \in \Theta(v_i)} r_j \).

The second inequality is because the Cauchy inequality and \( |r| L_i R_i = |T| \). Thus, the minimal energy consumption in (6) can be obtained when the condition \( r_i = \sqrt{|\Theta(v_i)| R_i t(D_{ij} + D_{ji})}/(|r| L_i) \) is satisfied.

We can check the correctness of the condition. In the example of Fig. 4a, \( R_i = 4 \), \( L_i = 6 \), \( |\Theta(v_i)| = 3 \) and the slot amount to implement is \( t(D_{ij} + D_{ji})/r_j = 2 \) so \( r_i = \sqrt{3 \times 4 / 2} = 2 \). By Step II, we can adjust nodes’ duty cycled so that the overall energy consumption is minimized. Table 2 has already proved that the energy consumption in the

![FIGURE 5. The way to select NSQs for neighbors affect energy consumption. (a) Method one and (b) method two.](http://comjnl.oxfordjournals.org/)

example of Fig. 4a is minimal among all the example in Fig. 4. Note that we can induce that \( r_i/|\Theta(v_i)| l_j = R_i / l_i \) from the condition. In the equality, the ratio between the numbers of selected BSQs and NSQs is proportional to the ratio of the row amount to the column amount of \( v_i \)’s SQS. The equality hints an explicit way to find the optimal solution to (6).

The remaining question is how to select NSQs for each neighbor. The key metric for NSQs selection is still energy saving. Recall that the condition enabling the first inequality correct in Equation (7) is that \( r_i = \max_{v_j \in \Theta(v_i)} r_j \). When all neighbors need the same time to implement their demands, the first inequality in Equation (7) becomes equality and thus the listening time in Equation (7) can be minimal. However, neighbors may need different time to implement their demands so the way to select NSQs affects the energy consumption. Figure 5 gives an example in which node \( v_i \) has two neighbors \( v_j \) and \( v_k \) with different demands. Suppose that \( v_j \) needs two slots to implement its demand while \( v_k \) needs four slots. Under the method one in Fig. 5a, \( v_j \) selects the NSQ \( Q_5 \) and \( v_k \) selects the NSQ \( Q_6 \). \( v_x \) needs only the slots, 26 and 32 to listen. \( v_i \) has to select four BSQs: \( Q_1 \), \( Q_2 \), \( Q_3 \) and \( Q_4 \). Under the method two in Fig. 5b, \( v_j \) selects the NSQ \( Q_5 \) and \( v_k \) selects the NSQs \( Q_6 \) and \( Q_7 \). \( v_x \) has to wake up to listen at the slots: 14, 15, 20, 21, 26, 27, 32 and 33 \( v_i \) need only select two BSQs: \( Q_1 \) and \( Q_2 \). \( v_x \) need not wake up to listen at the slots: 3, 9, 13, 15, 16, 17, 18, 19, 21, 22, 23 and 24. Compared with the method one, \( v_k \) needs 6 slots more to listen and \( v_x \) saves 12 slots to listen. Obviously, the method two saves energy much more than the method one. Actually, it can save more energy to minimize the number of BSQs selected by \( v_j \) according to the condition to satisfy the first inequality in Equation (7). So the rule to select NSQs for each neighbor follows: (i) suppose that the neighbor \( v_j \) needs minimal time to implement its demand among all neighbors. Select one NSQ for \( v_j \). Denote the number of its rows in the NSQ to implement its demand by \( r_j \). (ii) Select \( \lceil r_j/|\Theta(v_i)| \rceil \) NSQs for other neighbor \( v_k \) where \( t_j \) is the time that \( v_k \) needs.

Recall that the network life is defined as the time up to the first node using up its energy. The energy consumption balance is thus very important. Step III is to implement the transmutation operation so that the energy consumption can be balanced between each node and its neighbors. After each node selects its BSQ and the NSQs for its neighbors, the variables \( r \)
and $l$ are determined in Equations (4) and (5). Recall that the active time $t(D_{ij} + D_{ji})$ should be satisfied to implement the demand. So only the listening time can be adjusted. Usually, node $v_i$ consumes energy much faster than its neighbors $\theta(v_i)$ since it has often more than one neighbors. We expect all nodes consume energy in almost the same speed. The easy way is to choose the neighbor $v_j$ from $\theta(v_i)$, whose energy consumption is middle in $\theta(v_i)$, and then to reduce the energy consumption difference between $v_i$ and $v_k$. According to Equations (4) and (5), the row of the SQS should be adjusted by the following equation:

$$L_i(\text{new}) = \max \left\{ L_i(\text{old}) \right\} - \left\lfloor \frac{E_i - E_k}{2P_l|\tau|} \right\rfloor \sum_{v_j \in \theta(v_i)} l_j$$

(8)

where $L_i(\text{old})$ and $L_i(\text{new})$ are, respectively, the column amount in the SQS before and after the transmutation operation. By Equation (8), $v_i$ adjusts the column amount $L_i(\text{old})$ and obtains $L_i(\text{new})$ so that its energy consumption can be close to the node $v_i$'s, whose energy consumption is middle among $v_i$'s neighbors.

### 4.2. Dispersing quorums for asynchronization

As stated in Lemma 3.3, the rotation closure property of SQS guarantees the non-empty intersection between BSQs and NSQs when the network is asynchronous. When a node’s work load is not too high to overcrowd the SQs in its period, the quorum dispersing can avoid or decrease the overlap among SQs caused by the clock shift. For example, the $Q_3$ overlaps with $Q_4$ on the slot 2 because of the clock shift in Fig. 6. Both of $Q_3$ and $Q_4$ are active at the slot 2 so the confliction happens.

The clocks of the sensor nodes, such as TelosB and MicaZ, are often imprecise but the clock shift is limited [17]. Elson et al. tested the Berkeley sensor node [18] based on TinyOS and obtained the distribution of the receivers’ clock shifts, which obey the Gaussian distribution with zero expectation and the variance 11.1 $\mu$s under confidence 99.8% [17]. Knowing the clock shift limitation, the overlap confliction among SQs can be avoided or decreased by properly dispersing the SQs. Set the variable $\delta$ to be three times of the variance, i.e. $\delta = 33.3 \mu$s. We disperse the quorums including BSQs and NSQs apart from each other with the interval $\delta$ as shown in Fig. 7. These quorums then do not overlap with each other with high probability. Since NSQs are assigned to different neighbors, there should be $f(\delta)$ NSQs interval between the adjacent NSQs selected by different neighbors as shown in Fig. 7. $f(\delta)$ is determined by Equation (9).

$$f(\delta) = \begin{cases} \left\lfloor \frac{\delta}{\tau R} \right\rfloor, & \text{if } \frac{\delta}{\tau R} < R_i - r_i \\ \left\lfloor \frac{\delta}{\tau R} \right\rfloor + 1, & \text{otherwise}. \end{cases}$$

(9)

### 4.3. Energy efficiency improvement

This block provides two methods to improve the energy efficiency for each node. They are the quorum tessellating and shrink. In the quorum tessellating, those active slots are tessellated into new SQSs as show in Fig. 8. Recall that each node has to be active at each slot in its quorum even when it is listening. By the quorum tessellating, much energy can be saved. In Fig. 8, the slot 30 in the original SQS is tessellated into a new refined SQS. The properties of the non-empty intersection and rotation closure are still hold after the quorum tessellating. After one generation of QS tessellating, the saved time not to wake up in the example of Fig. 8 can be easily calculated as follows. The size of a BSQ in the original SQS is $R$, and then the BSQ can save energy $R(L - 1/L)$ by the quorum tessellating.

In the method of the quorum shrink, each node need wake up at part slots of its quorum. The real sensor nodes, such as TelosB and MicaZ, have limited clock shift as 11.1 $\mu$s under confidence 99.8% [17]. So, it need only wake up around the intersected slot. For example, the slot size is 11.1 $\mu$s in Fig. 9, and $v_i$ selects the quorum $Q_1$. $v_i$ can have high probability to connect with other node selecting the quorum $Q_2$ when it wakes up only at slots 6–8. This method can greatly save energy and increase the channel utilization.
5. PROPERTIES OF LOD

In the previous sections, SQS enables LOD to have two properties: the non-empty intersection and the rotation closure. LOD can establish connection among nodes even when they are not synchronized. It can adjust each node’s duty cycle so that its demand can be implemented and energy can be saved and the energy consumption can be balanced in the whole network. This section analyzes another properties of LOD: the non-empty intersection and the rotation closure.

The maximal SQS load. The maximal load that each node can afford of is affected by the interference models $\mathcal{M}$, the SQS structure and duty cycling. In each period, the time to implement node $v$’s demand is $\sum_{v \in \theta(v)} (D_{ij} + D_{ji})$, which is determined by the data rate. Section 4.2 indicates that there are $f(\delta)$ NSQs between adjacent NSQs to overcome the negative affection of asynchronization. Since there are $|\theta|$ neighbors and each neighbor selects $\ell_i$ NSQs, $\sum_{v \in \theta(v)} (L_{ij} + f(\delta)) \leq L_{ij}$. There are at most $(L_{ij} - |\theta(v)|f(\delta))R_i\tau$ time to implement demands of all neighbors in each period. Recall that it needs $\chi$ colors to color all nodes under the interference model $\mathcal{M}$. Each color is assigned a period so there are totally $\chi$ periods. That means each node’s period can wake up in every $\chi$ periods. Therefore, the maximal demand that each node can afford of per period is $\sum_{v \in \theta(v)} (D_{ij} + D_{ji}) \leq (\rho/\chi)(L_{ij} - |\theta(v)|f(\delta))R_i\tau$, where $\rho$ is the data rate. When the network works under (ultra) low duty cycle, i.e. $R_i\tau < 1$, $f(\delta) = 0$. So $\sum_{v \in \theta(v)} (D_{ij} + D_{ji}) \leq \rho T/\chi$. It means the asynchronization would have no effect on the maximal demand that each node can implement when the network works under (ultra) low duty cycle.

Channel utilization. Channel utilization is a popular metric to measure MAC protocols’ performance [16]. Rhee et al. [19] gave an equation to calculate the channel utilizations under B-MAC, PTDMA and Z-MAC as Equation (10).

$$\mathcal{U} = \frac{t_c}{t_c + t_l + t_s}$$

where $t_c$ is the communication time including transmitting and receiving data. $t_l$ and $t_s$ are the listening and sleeping time, respectively. Under the LOD, the channel is utilized in the common active slots between BSQs and NSQs, which includes the time to transmit and receive data. LOD wakes nodes up in some additional slots so that the rotation closure property can be satisfied. The utilization under LOD can be described by the followed equation:

$$\mathcal{U} = \frac{l_i r_j}{\chi |T|}$$ (11)

where $l_i$ and $r_j$ are selected sufficient to implement the node $v_i$’s demand. On one hand, the channel utilization can be adjusted by changing the cardinality $T$ of the period. On other hand, it is also related to the demand in each period. SQS has higher channel utilization than the Grid QS. For example, nodes should be active at total 16 slots in Fig. 1 to have two common active slots. Meanwhile, nodes need only be active at total 13 slots in Fig. 2.

6. EXPERIMENTAL RESULTS

This section evaluates the performance of LOD and compares it with B-MAC on a real test-bed, which consists of 100 TelosB sensor nodes as shown in Fig. 10. Each node runs TinyOS. We compare the performance of LOD with B-MAC on the network throughput, Packet Reception Ratio (PRR) and the energy consumption.

6.1. Experimental setup

The experiment takes the data aggregation as the application example so a minimum spanning tree (MST) is constructed based on all nodes by the existing method: breadth-first search. The experiments compose of two phases. In the first phase, LOD is implemented. In the second phase, the MST is constructed and rooted at the sink. The data aggregation is implemented under LOD. In the tree, all leaf nodes are not assigned periods and share the periods with their parent nodes. Meanwhile, the protocol B-MAC is also implemented based on the MST. The reason to adopt B-MAC is among the first random access MAC protocols to adopt the low power listening (LPL) technique, which is a feasible way to deal with the asynchronous communication among nodes. Since B-MAC
has no ability of duty cycling, the duty cycle is set as 20% in the experiment for B-MAC.

Under LOD, each $Q$ contains 100 time slots, i.e. $|T| = 100$. Each slot is, respectively, set as 50 ms, 1, 2 and 5 s. Each node samples data in every 100, 200, 300, 500, 800 ms, 1, 1.5 and 2 s, which is called as the data generation period in Figs 11 and 12.

6.2. Performance comparison

The evaluation of LOD and B-MAC on the network capacity and PRR illustrate their performance on channel utilization and fairness indirectly. Energy consumption is also measured.

Throughput. Figure 11 shows the network throughput under LOD and B-MAC, respectively. Each node generates data at different rates. Since all nodes must compete for channel access in B-MAC, much channel resources is wasted when delivering data packets. Thus B-MAC achieves a lower network capacity than LOD, as shown in Fig. 11. When the sampling period is small, such as 100, 200 and 300 ms, the throughput under B-MAC is much lower than that under LOD. Because packets are transmitted more frequently under smaller sampling period, B-MAC observes a greater degree of channel contention. Although the throughput under LOD is higher than B-MAC, it is actually quite low because the maximal data rate of TelsoB node can reach 250 kbps. One of main reasons is that asynchronous clocks make quorums overlap with each other and each node’s clock shift is arbitrary and unknown in LOD. Thus connotative channel contention cannot be avoided. When the size of slot is smaller, the possibility that quorums overlap is higher. Therefore, the throughput under both B-MAC and LOD is close to each other as shown in Fig. 11a.

PRR. The PRR reflects the channel utility in the network. The PRRs under both B-MAC and LOD are shown in Fig. 12. The PRR under B-MAC increases with the data generation period. But PRR under LOD keeps >80% under almost all data generation periods. The slots size has much effect on the throughput under B-MAC instead of that under LOD. When the size of slot is smaller, such as, 50 ms, PRR under LOD decreases much, and is lower than that under B-MAC when data generation is higher than 900 ms as shown in Fig. 12a. The reasons are similar to those given for the network throughput.

Energy consumption. The experimental results on energy consumption are shown in Fig. 13 when the slot size sets to 1 s. In this figure, the energy consumption is the average value per node per second. The energy consumption under LOD keeps ~6 mA/s per nodes in different data generation periods, while under B-MAC varies from more than 20 to 8 mA/s.
It is obvious that energy consumption under B-MAC is greatly affected by the data generation periods because shorter data generation periods cause more contention for channel under B-MAC. The energy consumption in Fig. 13 indicates that LOD generation periods cause more contention for channel under B-MAC. Energy consumption comparison between LOD and B-MAC.

FIGURE 13. Energy consumption comparison between LOD and B-MAC.

Since the clock is easy to shift in WSNs, some works considered network operation under asynchronous clock. One of most famous techniques to establish communication among nodes under asynchronization is the LPL technique [16]. Under this technique, some preambles are first sent before the data so that nodes can communicate with each other while the clock shift exists. The LPL technique cannot work well when the clock shift is larger than the overall length of all the preambles. Sun et al. [23] designed an asynchronous duty-cycled broadcast, under which, nodes may be active for very long time when they need broadcast their data to a large number of neighbors, which awake up in different time.

There are also many other works studying the network performance under duty cycled networks, such as latency [24–26], opportunistic data aggregation [27] and reliable data delivery [28, 29]. Nath and Gibbons [30] analyzed the performance of geographic routing over duty-cycled nodes and presented a sleeping scheduling algorithm that can be tuned to achieve an expected routing latency and coverage. After these works, researchers found that the network traffic or the demand has great effect on the duty cycle. Lee et al. [31] proposed a traffic-adaptive MAC: AMAC, to adjust the duty cycle of each node according to its traffic but it did not consider the asynchronization among nodes and the feasibility of the traffic-based duty cycle adjustment since WSNs often affords of different kinds of tasks and network traffic is variable.

7.2. MAC protocol

In WSNs, MAC is one main class of protocols considering the node sleep/wakeup or duty cycle adjustment. Their goals are to save energy, to improve the network throughput and/or to shorten the transmission delay. They can be classified into two classes: synchronization based and asynchronization based. Synchronization-based protocols include S-MAC [32], T-MAC [33], U-MAC [34], P-MAC [35] and H-MAC [36]. Asynchronization-based ones include D-MAC [37], B-MAC [16], Wise-MAC [38], SyncWUF [39] and ACDA [40]. Paper [40] offered the classification and summarization of the above protocols.

Some protocols are designed to combine the advantages of TDMA and CSMA. Rhee et al. [19] proposed a hybrid MAC protocol, called Z-MAC. In Z-MAC, a node always performs carrier-sensing before it transmits during a slot. Thus Z-MAC consumes much energy on the carrier-sensing. Z-MAC also requires local synchronization among senders in two-hop neighborhoods. S-MAC [32] and T-MAC [33] are also a hybrid of CSMA and TDMA and employ RTS/CTS mechanism to solve synchronization failure. Since these protocols use RTS/CTS, their overhead is quite high [19]. B-MAC [16] is the default MAC in the operating system of Mica2 and adopts LPL to solve the asynchronization. Since LPL consumes much energy, X-MAC reduces the energy consumption and latency by employing short preamble and embedding address information of the target in the preamble [41]. Thus the non-target receivers can quickly go back to sleep and the energy is saved. LPL-based preamble transmission may occupy the medium for much longer than actual data transmission. So [42] designed an asynchronous duty cycle MAC: RI-MAC. In RI-MAC, the energy would be wasted especially when the traffic load is low and the interference would be increased since each node should broadcast a beacon periodically no matter the sender has data to transmit or not.

7. RELATED WORK

7.1. Duty cycle

In WSNs, operation at a certain duty cycle can bridge the gap between the limited energy and longing network life [20]. Gu and He [1] designed a data forwarding technique to optimize the data delivery ration, end-to-end delay or energy consumption in low-duty cycle sensor networks under a synchronized mode. Guo et al. [21] designed an Opportunistic Flooding scheme for low-duty cycle networks with unreliable wireless links and predetermined wording schedules when the network is locally synchronized. Wang and Liu [22] provided a benchmark for assessing diverse duty cycle-aware broadcast strategies and extended it to distributed implementation. It translated the broadcast problem into a graph equivalence in order to seek a balance between efficiency and latency with coverage guarantees.

Since the clock is easy to shift in WSNs, some works considered network operation under asynchronous clock. One of most famous techniques to establish communication among nodes under asynchronization is the LPL technique [16]. Under this technique, some preambles are first sent before the data so that nodes can communicate with each other while the clock shift exists. The LPL technique cannot work well when the clock shift is larger than the overall length of all the preambles. Sun et al. [23] designed an asynchronous duty-cycled broadcast, under which, nodes may be active for very long time when they need broadcast their data to a large number of neighbors, which awake up in different time.

There are also many other works studying the network performance under duty cycled networks, such as latency [24–26], opportunistic data aggregation [27] and reliable data delivery [28, 29]. Nath and Gibbons [30] analyzed the performance of geographic routing over duty-cycled nodes and presented a sleeping scheduling algorithm that can be tuned to achieve an expected routing latency and coverage. After these works, researchers found that the network traffic or the demand has great effect on the duty cycle. Lee et al. [31] proposed a traffic-adaptive MAC: AMAC, to adjust the duty cycle of each node according to its traffic but it did not consider the asynchronization among nodes and the feasibility of the traffic-based duty cycle adjustment since WSNs often affords of different kinds of tasks and network traffic is variable.

7.2. MAC protocol

In WSNs, MAC is one main class of protocols considering the node sleep/wakeup or duty cycle adjustment. Their goals are to save energy, to improve the network throughput and/or to shorten the transmission delay. They can be classified into two classes: synchronization based and asynchronization based. Synchronization-based protocols include S-MAC [32], T-MAC [33], U-MAC [34], P-MAC [35] and H-MAC [36]. Asynchronization-based ones include D-MAC [37], B-MAC [16], Wise-MAC [38], SyncWUF [39] and ACDA [40]. Paper [40] offered the classification and summarization of the above protocols.

Some protocols are designed to combine the advantages of TDMA and CSMA. Rhee et al. [19] proposed a hybrid MAC protocol, called Z-MAC. In Z-MAC, a node always performs carrier-sensing before it transmits during a slot. Thus Z-MAC consumes much energy on the carrier-sensing. Z-MAC also requires local synchronization among senders in two-hop neighborhoods. S-MAC [32] and T-MAC [33] are also a hybrid of CSMA and TDMA and employ RTS/CTS mechanism to solve synchronization failure. Since these protocols use RTS/CTS, their overhead is quite high [19]. B-MAC [16] is the default MAC in the operating system of Mica2 and adopts LPL to solve the asynchronization. Since LPL consumes much energy, X-MAC reduces the energy consumption and latency by employing short preamble and embedding address information of the target in the preamble [41]. Thus the non-target receivers can quickly go back to sleep and the energy is saved. LPL-based preamble transmission may occupy the medium for much longer than actual data transmission. So [42] designed an asynchronous duty cycle MAC: RI-MAC. In RI-MAC, the energy would be wasted especially when the traffic load is low and the interference would be increased since each node should broadcast a beacon periodically no matter the sender has data to transmit or not.
MAC protocols are also designed to reduce energy consumption, such as S-MAC [32] and T-MAC [33]. Zheng et al. [14] considered LPL approaches, such as Wise-MAC and B-MAC, are limited to duty cycles of 1–2% and designed a new MAC protocol called scheduled channel polling to ensure that duty cycles of 0.1% and below are possible. It dynamically adjusts duty cycles in the face of busy networks and streaming traffic to reduce the latency.

Shah et al. [43] presented a MAC protocol, C-MAC to achieve high-throughput bulk communication for data intensive sensing applications, but it did not consider duty cycle. Kim et al. [44] proposed a lightweight channel hopping mechanism, thus avoiding redundant channel assignment by not allocating fixed channels to the nodes. Synchronization was also implemented by initial time assignment by not allocating fixed channels to the nodes. Channel hopping mechanism, thus avoiding redundant channel


MAC mainly allocates channel resources for these conflicting neighbors. Some of them considered the communication among nodes when their clocks were asynchronous. However, they still need synchronization protocols to cooperate the communication time, or add extra preamble to overcome the clock shift, such as LPL. This paper argues the new method LOD to ensure the communication among nodes without using any synchronization protocols in case of imprecise clock.

8. CONCLUSION
This paper designed the SQS based on QS. Based on SQS, we proposed the localized scheme, LOD, to adjust nodes’ duty cycle adaptively according to its demand such that they can fairly use the channel. LOD combines the advantages of both TDMA and CSMA and does not need extra synchronization protocol. When the network is asynchronous, LOD can still guarantee that any pair of neighboring nodes has sufficient common active time to communicate with each other. LOD can increase the channel utilization by adjusting each node’s duty cycle based on its demand. We established the WSNs tested with 100 TelosB nodes and evaluated the performance of LOD. Compared with B-MAC, LOD significantly improves the performance such as network throughput and PRR. It is still a challenging problem to design energy-efficient protocols to establish communication in the duty-cycled networks when the clock is imprecise, especially when the duty cycle is ultra-low.

FUNDING
This work is supported by the Zhejiang Province Natural Science Foundation under Grants No.LY14F020042, the National Natural Science Foundation of China (NSFC) under Grants No.61473109 and 61003298.

REFERENCES


Asynchronous Duty Cycling


