Sleep Scheduling for Critical Event Monitoring in Wireless Sensor Networks

Peng Guo, Tao Jiang, *Senior Member*, *IEEE*, Qian Zhang, *Fellow*, *IEEE*, and Kui Zhang

Abstract—In this paper, we focus on critical event monitoring in wireless sensor networks (WSNs), where only a small number of packets need to be transmitted most of the time. When a critical event occurs, an alarm message should be broadcast to the entire network as soon as possible. To prolong the network lifetime, some sleep scheduling methods are always employed in WSNs, resulting in significant broadcasting delay, especially in large scale WSNs. In this paper, we propose a novel sleep scheduling method to reduce the delay of alarm broadcasting from any sensor node in WSNs. Specifically, we design two determined traffic paths for the transmission of alarm message, and *level-by-level offset* based wake-up pattern according to the paths, respectively. When a critical event occurs, an alarm is quickly transmitted along one of the traffic paths to a center node, and then it is immediately broadcast by the center node along another path without collision. Therefore, two of the big contributions are that the broadcasting delay is independent of the density of nodes and its energy consumption is ultra low. Exactly, the upper bound of the broadcasting delay is only $3D + 2L$, where D is the maximum hop of nodes to the center node, L is the length of sleeping duty cycle, and the unit is the size of time slot. Extensive simulations are conducted to evaluate these notable performances of the proposed method compared with existing works.

Index Terms—Wireless Sensor Network (WSN), critical event monitoring, sleep scheduling, broadcasting delay, multichannels.

1 Introduction

In mission-critical applications, such as battlefield reconnaissance, fire detection in forests, and gas monitoring in coal mines, wireless sensor networks (WSNs) are deployed in a wide range of areas, with a large number of sensor nodes detecting and reporting some information of urgencies to the end-users. As there may be no communication infrastructure, users are usually equipped with communicating devices to communicate with sensor nodes. When a critical event (e.g., gas leak or fire) occurs in the monitoring area and is detected by a sensor node, an alarm needs to be broadcast to the other nodes as soon as possible, which is shown in Fig. 1 as an example. Then, sensor nodes can warn users nearby to flee or take some response to the event.

As sensor nodes for event monitoring are expected to work for a long time without recharging their batteries, sleep scheduling method is always used during the monitoring process. Obviously, sleep scheduling could cause transmission delay because sender nodes should wait until receiver nodes are active and ready to receive the message. The delay could be significant as the network scale increases. Therefore, a delay-efficient sleep scheduling method needs to be designed to ensure low broadcasting delay from any node in the WSN.

- P. Guo, T. Jiang, and K. Zhang are with the Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, B-307, 1037 Luoyu Road, Wuhan 430074, P.R. China. E-mail: guopeng@mail.hust.edu.cn, Tao.Jiang@ieee.org, truelykyle@gmail.com.
- Q. Zhang is with the Hong Kong University of Science and Technology, Tower 10, 3B, Clear Water Bay, Kowloon, Hong Kong. E-mail: qianzh@cse.ust.hk.

Manuscript received 25 Mar. 2010; revised 15 July 2010; accepted 15 Apr. 2010; published online 25 May 2011.
Recommended for acceptance by A. Kshemkalyani.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2010-03-0177.
Digital Object Identifier no. 10.1109/TPDS.2011.165.

Recently, many sleep schedules for event monitoring have been designed [1], [2], [3], [4]. However, most of them focus on minimizing the energy consumption. Actually, in the critical event monitoring, only a small number of packets need to be transmitted during most of the time. When a critical event is detected, the alarm packet should be broadcast to the entire network as soon as possible. Therefore, broadcasting delay is an important issue for the application of the critical event monitoring.

To minimize the broadcasting delay, it is needed to minimize the time wasted for waiting during the broadcasting. The ideal scenario is the destination nodes wake up immediately when the source nodes obtain the broadcasting packets. Here, the broadcasting delay is definitely minimum. Based on this idea, a *level-by-level offset schedule* was proposed in [5]. As shown in Fig. 2, the packet can be delivered from node a to node c via node b with minimum delay. Hence, it is possible to achieve low transmission delay with the *level-by-level offset schedule* in multi-hop WSNs [6], [7], [8], [9].

However, it is still a challenge for us to apply the *level-by-level offset* to alarm broadcasting in the critical event monitoring. First, the order of nodes' wake-up should conform to the traffic direction. If the traffic flow is in the reverse direction (as show in Fig. 2), the delay in each hop will be as large as the length of the whole duty cycle. Second, the *level-by-level offset* employed by the packet broadcasting could cause a serious collision. Finally, the transmission failure due to some unreliable wireless links may cause the retransmission during the next duty cycle, which also results in large delay equaling the whole duty cycle.

In this paper, we propose a novel sleep scheduling method, which is still based on the *level-by-level offset schedule*, to achieve low broadcasting delay in a large scale WSN. As the alarm message may be originated by any possible node, we set two phases for the alarm broadcasting

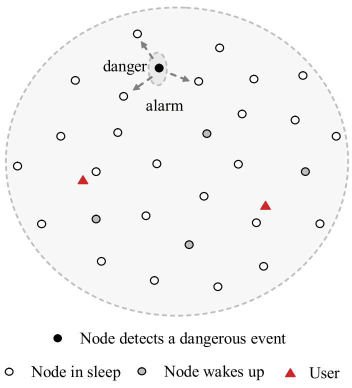


Fig. 1. Critical event monitoring with a WSN.

in the proposed sleep scheduling method. First, when a node detects a critical event, it originates an alarm message and quickly transmits it to a center node along a predetermined path with a *level-by-level offset* way. Then, the center node broadcasts the alarm message to the other nodes along another path also with a *level-by-level offset* way. Through designing a special wake-up pattern, the two possible traffics could be both carried by a node, and the node just needs to be awake for no more than τ time in each duty cycle, where τ is the minimum time needed by a node to transmit an alarm packet. To eliminate the collision in broadcasting, a colored connected dominant set (CCDS) in the WSN via the *IMC* algorithm proposed in [12] is established. Each node transmits or receives packets in a specific channel according to the color assigned.

In summarization, characteristics of the proposed sleep scheduling scheme are:

1. The upper bound of the broadcasting delay is $3D + 2L$, where D is the maximum hop of nodes to the center node, and L is the length of duty cycle, the unit is the size of time slot. As the delay is only a linear combination of hops and duty cycle, it could be very small even in large scale WSNs.
2. The broadcasting delay is independent of the length of the duty cycle, but it increases linearly with the number of the hops.
3. The broadcasting delay is independent of the density of nodes.
4. The energy consumption is very low as nodes wake up for only one slot in the duty cycle during the monitoring.

The rest of the paper is organized as follows: We describe the problem scenario in Section 2, and present the proposed sleep scheduling method in Section 3, respectively. In Section 4, we analyze the performances of the proposed method, and then make extensive simulations to validate, followed by the conclusions in Section 5. Existing related works are introduced in Section *Related works* of *supplementary file*, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPDS.2011.165.

2 Problem Description

We assume that a certain node, called as center node, in the network has obtained the network topology in the initialization (e.g., sink node). The center node computes the sleep scheduling according to the proposed scheduling scheme and broadcasts the scheduling to all the other nodes. The implementation of obtaining topology and broadcasting scheduling is introduced in Section *Experiments* of *supplementary file*, which can be found on the Computer Society Digital Library. The following terms are defined in this paper.

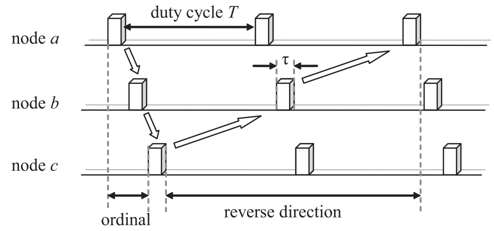


Fig. 2. The *level-by-level offset schedule*.

- **Event detection:** For the critical event monitoring in a WSN, sensor nodes are usually equipped with passive event detection capabilities that allow a node to detect an event even when its wireless communication module is in sleep mode. Upon the detection of an event by the sensor, the radio module of the sensor node is immediately woken up and is ready to send an alarm message.
- **Slot and duty cycle:** Time is partitioned into time slots. The length of each slot is about the minimum time needed by sensor nodes to transmit or receive a packet, which is denoted as τ. For example, to transmit a simple packet with a size of several bytes using the radio chip $Chipcon\ CC2420$, τ could be less than 2 ms. The length of each duty cycle is $T = L * \tau$, i.e., there are L slots in each duty cycle.
- **Network topology:** For the sake of simplicity, we assume the network topology is steady and denote it as a graph G.
- **Synchronization:** Time of sensor nodes in the proposed scheme is assumed to be locally synchronous, which can be implemented and maintained with periodical beacon broadcasting from the center node.

We define $f(n_i)$ as the slot assignment function. If $f(n_i) = s, s \in \{0, \dots, L-1\}$, it means that node n_i wakes up only at slot s to receive packets. Meanwhile, we define $F(n_i)$ as the channel assignment function which assigns a frequency channel to node n_i.

3 The Proposed Scheduling Method

3.1 Basic Idea

It is known that the alarm could be originated by any node which detects a critical event in the WSN. To essentially reduce the broadcasting delay, the proposed scheduling method includes two phases: 1) any node which detects a critical event sends an alarm packet to the center node along a predetermined path according to *level-by-level offset schedule*; 2) the center node broadcasts the alarm packet to the entire network also according to *level-by-level offset*

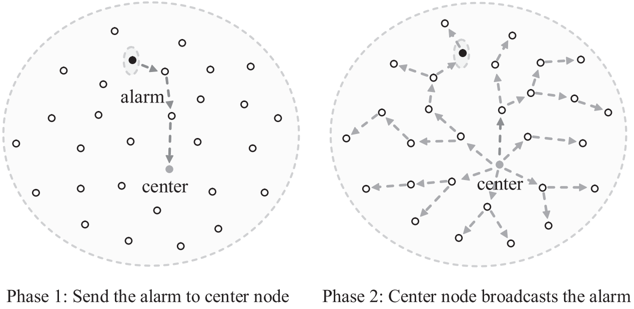


Fig. 3. Two phases of the alarm broadcasting in a WSN.

schedule. As an example, Fig. 3 illustrates these two phases of the processing.

We define the traffic paths from nodes to the center node as *uplink* and define the traffic path from the center node to other nodes as *downlink*, respectively. Each node needs to wake up properly for both of the two traffics. Therefore, the proposed scheduling scheme should contain two parts: 1) establish the two traffic paths in the WSN; 2) calculate the wake-up parameters (e.g., time slot and channel) for all nodes to handle all possible traffics.

To minimize the broadcast delay, we establish a breadth first search (BFS) tree for the uplink traffic and a colored connected dominant set for the downlink traffic, respectively.

3.2 Traffic Paths

First of all, we choose a sensor node as the center node c. Then, we construct the BFS tree which divides all nodes into layers H_1, H_2, $H_3, \ldots, H_D$, where H_i is the node set with minimum hop i to c in the WSN. With the BFS tree, the uplink paths for nodes can be easily obtained.

To establish the second traffic path, we establish the CCDS in G with three steps: 1) construct a maximum independent set (MIS) in G; 2) select *connector nodes* to form a connected dominated set (CDS), and partition *connector nodes* and *independent nodes* in each layer into four disjoint sets with IMC algorithm proposed in [12]; 3) color the CDS to be CCDS with no more than 12 channels. The details are described as follows, and the variables therein are defined in Table 1.

First, we construct a MIS I. As all nodes have been divided into H_1, H_2, $H_3, \ldots, H_D$ with the BFS tree, the MIS can be established layer by layer (i.e., hop by hop) in the BFS as follows: Start from the $0th$ hop, we pick up a maximum independent set, then, move on to the $first$ hop, pick up another maximum independent set. Note that, *independent nodes* of the $first$ hop also need to be independent of those in the previous hop. Repeat this process until all hops of nodes have been worked on. The pseudocode of the MIS construction is given in *Algorithm 1* of *supplementary file*, which can be found on the Computer Society Digital Library.

Second, we construct the CDS by selecting *connector nodes* C from $V \backslash I$ to interconnect *independent nodes* as follows: Obviously, for any two 2-hop neighboring *independent nodes*, at least one node in G is adjacent to both of them. Hence, the node is possible to be selected as a *connector node*. We use the idea of the IMC algorithm in [12] to select the *connector nodes*, which partitions *independent nodes* $I \cap H_i$ in each layer into four disjoint subsets $U_{i,j}$ $(0 \leq j \leq 3)$, and selects four disjoint subsets $W_{i-1,j}$ $(0 \leq j \leq 3)$ among $(H_{i-1} \cup H_{i-2}) \cap \overline{I}$ as *connector nodes* to cover $I \cap H_i$. When nodes in $W_{i-1,j}$ broadcast simultaneously, they will not cause any collision among nodes in $U_{i,j}$. By this way, the CDS is established. The pseudocode of *connector nodes* selection is given in *Algorithm 2* of *supplementary file*, which can be found on the Computer Society Digital Library.

TABLE 1
Definitions of Some Variables

c	The center node in the networks
H_i	The nodes with minimal hop i to c in G
H'_i	The nodes with minimal hop i to c in CDS
I_i	The *independent nodes* with minimal hop i to c in CDS
C_i	The *connector nodes* with minimal hop i to c in CDS
B_i	The *dominated nodes* dominated by I_i

We further color the CDS to be CCDS as follows: We divide all nodes in CDS into several sets according to their minimum hops to c in CDS. As CDS is based on $G^2(I)$, the number of hops from *independent nodes* to c in the CDS is even, and the number of hops from *connector nodes* to c in the CDS is odd. Therefore, we obtain $I_0, I_2, I_4, \ldots$ and $C_1, C_3, C_5, \ldots$. In addition, *dominated nodes* B could be divided into $B_0, B_2, B_4, \ldots$. They are dominated by $I_0, I_2, I_4, \ldots$, respectively. Since any two *independent nodes* cannot be adjacent, the distribution of *independent nodes* is actually sparse. It has been proved that each *independent node* has less than 12 neighbors in I within 2-hop distance [12]. Therefore, G' could be colored with $ch_1, \ldots, ch_{12}$. Hence, when *independent nodes* in each layer broadcast simultaneously, they will not cause any collision at *connector nodes*. We define *sending channel* as $ch_s(n_k)$ and *receiving channel* as $ch_r(n_k)$ for each node n_k, corresponding to channels in which n_k sends packets and receives packets, respectively. Each node n_k in I_i gets its $ch_s(n_k)$ according to its color, and each node n_t in C_i obtains its $ch_r(n_t)$ according to the color of one of its parents in I_{i-1}. In addition, we color the subsets $U_{i,j}$ and $W_{i-1,j}$ with cl_j $(0 \leq j \leq 3)$ in each layer. Hence, when *connector nodes* in each layer (i.e., $W_{i-1,j}$, $0 \leq j \leq 3$) broadcast simultaneously, they will not cause any collision at *independent nodes* in the next layer (i.e., $U_{i,j}$, $0 \leq j \leq 3$). Each node n_k in I_i gets its $ch_r(n_k)$ according to the color of $U_{i,j}$ that it belongs to, and each node n_t in C_i obtains its $ch_s(n_t)$ according to the color of $W_{i,j}$ that it belongs to. While, each node n_s in B_i obtains its $ch_r(n_s)$ according to the *sending channel* of an *independent node* in I_i which dominates n_s. The pseudocode of the coloring is given in *Algorithm 3* of *supplementary file*, which can be found on the Computer Society Digital Library.

3.3 Wake-Up Patterns

After all nodes get the traffic paths, *sending channels* and *receiving channels* with the BFS and CCDS, the proposed wake-up pattern is needed for sensor nodes to wake-up and receive alarm packet to achieve the minimum delay for both of the two traffic paths.

As described above, there are two traffic paths for the alarm dissemination, and sensor nodes take two *level-by-level offset schedules* for the traffic paths. Fig. 4 shows the two *level-by-level offset schedules*: 1) sensor nodes on paths in the

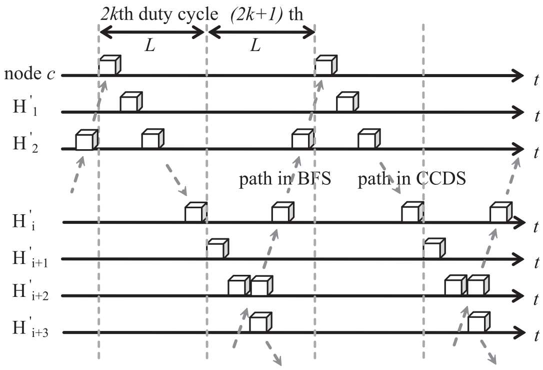


Fig. 4. Two periodic *level-by-level offset schedules*.

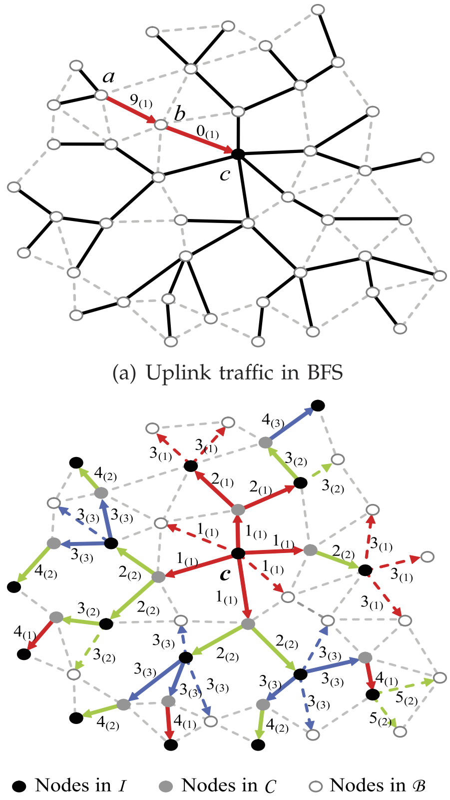
BFS wake up *level-by-level* according to their hop distances to the center node; 2) after the center node wakes up, the nodes in the CCDS will go on to wake up *level-by-level* according to their hop distances in the CCDS. Hence, when an alarm packet is originated, it could be quickly forwarded to the center node along a path in the BFS, then, the center node immediately broadcasts it along the paths in the CCDS.

Since it is hard to predict when the alarm occurs, the two *level-by-level offset schedules* are taken periodically as shown in Fig. 4. Moreover, it is needed to effectively arrange time slots for sensor nodes at different positions in the topology, so that the two *level-by-level offset schedules* can periodically work without interfering with each other. The assignment of time slots is summarized in Table 2, which can be briefly described as follows: 1) all nodes in H obtain slots for uplink traffic according to their hops in H and the sequence number of duty cycles; 2) nodes in H' obtain slots for downlink traffic according to their hops in H' and the sequence number of duty cycle; 3) nodes in B_i obtain the same slot as C_{i+1} for downlink traffic. For example, a sensor node n_j in H_1 obtains slot $L-1$ in odd duty cycles for uplink traffic. On the other hand, n_j may also be in H'_2, and it obtains slot 2 in even duty cycles for downlink traffic. In addition, it is obvious that, whenever a sensor node detects a critical event, it waits for no more than two duty cycles before its time slot for uplink traffic comes.

Furthermore, for nodes which are both in H_{2mL+s} and H'_{2nL+t}, when $s+t=L$, nodes will be assigned the same slot for uplink traffic and downlink traffic, i.e., nodes need to wake up for only one time slot every two duty cycles and it can receive the possible alarm transmitted both in uplink and downlink. Therefore, their *receiving channels* need to be modified. Suppose n_j is a node with the same slot for uplink traffic and downlink traffic. It should wake up in its ch_w channel, and its child in the BFS also should send the possible alarm to n_j in n_j's ch_w channel instead of ch_1.

TABLE 2
Wake-Up Patterns

Node $(0 \le s \le L-1)$	*Time Slot for wake-up* in $2k$th duty cycle	in $(2k+1)$th duty cycle
$n_j \in H_{(2m+1)L+s}$	$f(n_j) = L - s$	
$n_j \in H_{2mL+s}$		$f(n_j) = L - s$
$n_j \in H'_{2mL+s}$	$f(n_j) = s$	
$n_j \in H'_{(2m+1)L+s}$		$f(n_j) = s$
$n_j \in B_i$	$f(n_j) = f(n_t)$, *where* n_t *is any node in* C_{i+1}	



(a) Uplink traffic in BFS

(b) Downlink traffic in CCDS

Fig. 5. An example of the alarm broadcast with the proposed scheduling method.

3.4 An Example

In order to show the assignment more clearly, we give an example shown in Fig. 5, where the numbers in brackets denote the frequency channels, and the numbers in front of brackets denote the time slots in a duty cycle. The length of duty cycle is set 10. Consider two nodes a and b (shown in Fig. 5a), which are in H_2 and H_1, respectively, in the BFS. Suppose node a detects a critical event. It will originate an alarm packet and sends it to node b at time slot 9 in the earliest odd duty cycle in channel ch_1. When node b wakes up at time slot 9 in channel ch_1 and receives the alarm, it sends the alarm to the center node c which wakes up at time slot 0 in each even duty cycle in channel ch_1. After receiving the alarm, node c begins to broadcast the alarm packet among the CCDS, as shown in Fig. 5b. The solid lines are the paths in the CCDS. In the broadcasting phase (i.e., in even duty cycle for nodes a and b), node a and node b are in H'_3 and H'_1, respectively, in the CCDS. Therefore, they wake up at time slots 3 and 1, respectively, in each even duty cycle in their *receiving channels* (channel 3 and channel 1, respectively). When receiving the alarm packet, node a broadcasts it in its *sending channel* (channel 2), while node b does not broadcast the packet as it is a *dominated node*.

From Fig. 5b, all the transmissions at the same time slot do not cause any collision, and the broadcast is executed level-by-level without waiting. Furthermore, since the alarm can be quickly relayed to center node in an uplink path and center node could immediately begin to broadcast it, the broadcasting delay is much lower. In addition, the energy consumption of nodes is also very low, since most

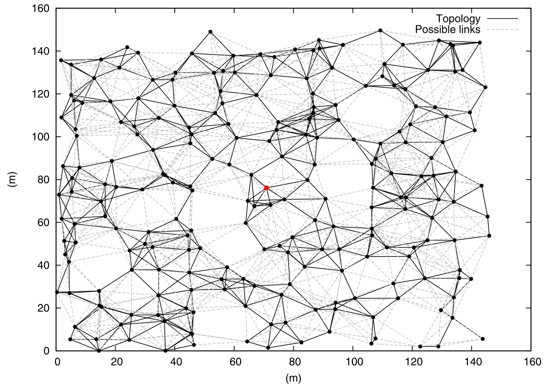


Fig. 6. The distribution of nodes in an unsteady WSN.

nodes stay awake for only one time slot in each duty cycle. Moreover, the center node and nodes with the same wake-up slots for uplink traffic and downlink traffic, stay awake for one time slot every two duty cycles.

Obviously, I_i, C_i, and B_i are used only for downlink traffic to solve the collision. Nodes in I_i broadcast alarms to C_{i+1} and B_i, and nodes in C_{i+1} broadcast alarms to I_{i+2}. While, nodes in B_i do not need to send alarms.

4 Analysis and Simulation

4.1 Performance Analysis

Lemma 1. *The maximum hop of the shortest path in the CCDS from any node to the center node is no more than $2D$.*

Proof. Consider any *independent node* n_j, there must be a parent in C connecting another *independent node* which is closer to the center node than n_j. If the parent is in the same layer with n_j in the BFS, then, it increases the hops of n_j to c in the CCDS. Otherwise, the number of hops does not increase. Consider the worst case for each hop with one increment on the shortest path from a node in layer H_D to c, the maximum length of the shortest path in the CCDS is consequently $2D$. □

Lemma 2. *The upper bound of alarm broadcasting delay in WSN is no more than $3D + 2L$.*

Proof. According to the proposed scheme, alarm packet can be transmitted along the uplink traffic path in the BFS without waiting. When the center node gets the packet, it immediately broadcasts the packet along the downlink traffic paths in the CCDS without waiting. Since the maximum hops of the shortest path in the BFS is no more than D, the upper bound of the delay to transmit an alarm from any node to the center node is D. Similarly, the upper bound of the delay to broadcast the alarm from the center node to all other nodes is no more than $2D$, according to *Lemma 1*. In addition, because the alarm may be originated at any time by a node and it has to wait for a duration until the time for its uplink schedule comes. The duration is no more than $2L$. Hence, the total delay is no more than $3D + 2L$. □

4.2 Simulations in Unreliable Environment

We use *ns-2* simulator to evaluate the performances of the proposed scheduling method in unsteady WSNs. In Fig. 6, 225 sensor nodes are randomly deployed in an area of $150 * 150\ \mathrm{m}^2$. The successful communication probability p to characterize the wireless link between any two nodes is employed. Considering the interference caused by non-neighboring nodes, we define the worse link quality than that in practice with assumption $p = 1 - (\frac{d}{20})^2$, where d is the distance between two nodes and $d < 20$. The links with $p \geq 50\ \%$ are chosen to form the topology of network for the proposed scheme, as shown in Fig. 6. The dashed lines are the links with $p < 50\%$. The duty cycle is $1\ \mathrm{s}$.

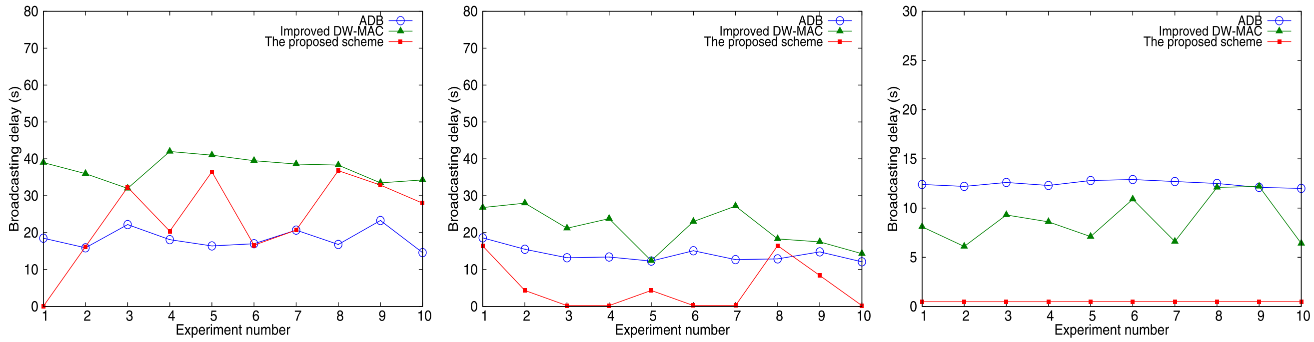
TABLE 3
Duty Cycle Configuration

	Active time	Duty cycle
Our scheme	$T_{data} = timeslot$	$1s$
DW-MAC	$T_{sync} = 0ms$, $T_{data} = timeslot$	$1s$
ADB	$T_{beacon} = 0ms$, $T_{data} = timeslot$	$1s$

For comparison, we also conduct some simulations for the *ADB* [10] and *DW-MAC* [11] schemes. In *ADB* scheme, as sensor nodes work with asynchronous duty cycles, the average transmission delay in each hop is actually about half of the duty cycle in the ideal environment. For fair comparison, we suppose a determined broadcast tree is established with a priori knowledge of the link quality so as to simplify the overhead in *ADB*. In *DW-MAC* scheme, as sensor nodes have to reserve data transmissions by transmitting the scheduling frame (SCH) in the short active time, the packet transmission delay is actually determined by the maximum hop counts of SCH transmission within the active time. For fair comparison, we improve *DW-MAC* scheme for critical event monitoring as follows (called as improved *DW-MAC*). As few data packets are transmitted in the network and the alarm is just a sign with small size, we regard the SCH in *DW-MAC* as the alarm packet and assume their sizes are the same. Hence, we are concerned just with the SCH (i.e., alarm) broadcasting delay in the network. We ignore the SYNC duration and SIFS duration in DW-MAC and assume multiple channels are already appropriately assigned among nodes to avoid collision in the SCH broadcasting. The configuration of nodes' duty cycle in three schemes are shown in Table 3. The sizes of active time in three schemes are set the same so as to compare their performance in the same level of energy consumption during the monitoring.

4.2.1 *Different Sizes of Time Slot*

We first set the size of the time slot to be the minimum time τ for sensor nodes to transmit an alarm packet, e.g., $2\ \mathrm{ms}$. When an alarm transmission fails between two adjacent nodes with the proposed scheme, the sender node has to retransmit the alarm after 2 duty cycles. While, for the *ADB* and the improved *DW-MAC* schemes, the sender node retransmits the alarm after 1 duty cycle. Fig. 7a shows the broadcasting delay with the three schemes in the WSN shown in Fig. 6. Obviously, the proposed scheme does not exhibit good performance in the case of minimum time slot. To improve it, we set the size of the time slot to be $10\ \mathrm{ms}$. Each sensor node still listens for $2\ \mathrm{ms}$ during each duty cycle. When a sensor node wakes up to listen to the channel and detects a collision or a failing reception during the $2\ \mathrm{ms}$, it will keep listening and receiving till the end of this time



(a) Time slot = $2ms$ (b) Time slot = $10ms$ (c) Time slot = $20ms$

Fig. 7. Broadcasting delay with different sizes of time slot.

slot. Accordingly, when the sender node finds that it fails to transmit the alarm packet during the 2 ms, it will keep retransmitting the packet till the end of the time slot. With this improvement, sensor nodes may successfully retransmit packets within a time slot and do not need to retransmit packets after 2 duty cycles. Hence, the transmission delay could be largely reduced.

Fig. 7b shows the simulation results in the same network shown in Fig. 6. It can be seen, the broadcasting delay with the proposed scheme becomes much lower when the size of time slot is 10 ms. Similarly, it is shown in Fig. 7b that, as the links in the network are unsteady, transmissions in the experiments are randomly successful, which affects the results in the experiments. For example, in experiments 1 and 8, packets usually cannot be successfully transmitted within a time slot, and have to be retransmitted after 2 s in next duty cycle. Therefore, the delay becomes large. Compared with the proposed scheme, the delay with the *ADB* and the improved *DW-MAC* schemes is even larger in most experiments. As sensor nodes in *ADB* wake up asynchronously, the average transmission delay in each hop is at least about half a duty cycle even if all transmissions were successful. While, for the improved *DW-MAC*, because the SCH (i.e., alarm) is forwarded within synchronous time slots, the number of hop counts of SCH transmission in each duty cycle is restricted by the size of time slot T_{data}. In ideal case, the number is T_{data}/τ. However, due to unsteady links, the number is dynamic. Hence, the number of duty cycles needed for the broadcasting in the network is random, resulting in highly dynamic results in the experiments.

We further enlarge the time slot to be 20 ms for the three schemes and corresponding results are shown in Fig. 7c. It can be seen, the proposed scheme achieves a distinct predominance to the other two schemes. Moreover, the broadcasting delay with the proposed scheme and the *ADB* scheme becomes much steadier in 10 experiments, as almost each packet can be successfully transmitted within 20 ms. While, the delay with the improved *DW-MAC* is still dynamic, because the number of hop counts of alarm transmission in each duty cycle is still uncertain due to unsteady links.

It is unnecessary to further enlarge the size of time slot, because the performance of the proposed scheme could not be further promoted. On the other hand, further enlargement of time slot increases energy consumption of sensor nodes, especially for the improved *DW-MAC* as nodes have to keep awake during the whole of the synchronous time slot.

We conduct more experiments with the schemes in several networks. All the networks are generated randomly with 225 sensor nodes. In each network, we made 20 experiments and the average broadcasting delay with the standard deviation is shown in Tables 4 and 5. For example, the average broadcasting delay in network 1 with the proposed scheme is 5.4 s and the standard deviation of the delay is 4.8 s, which is denoted as 5.4/4.8 in Table 4. From Tables 4 and 5, the average broadcasting delay of the proposed scheme is always much lower than that of the other two methods. When $timeslot = 0.02$ s, the delay of the proposed scheme almost keeps invariable in experiments in each network.

4.2.2 Multiple Alarms

In some cases, the critical event may trigger several alarms in the network, and they may be sent to a parent node when it wakes up. To deal with the collision, we design a mechanism for the proposed scheduling as follows: Suppose the time slot is denoted as $k * \tau$. When a sensor node having detected the event is going to send an alarm packet, it keeps transmitting the packet randomly with the probability 1/2 during the time slot. However, if the node detects some others are transmitting alarm packets during the same time slot, it gives up its transmission. Through this way, the nodes sending alarms could be decreased gradually. Note that, the parent node just needs to successfully receive one alarm. The parent node cannot judge whether there is an alarm packet by just detecting the

TABLE 4
Average Delay/Standard Deviation in Different Networks ($timeslot = 0.01$ s)

Network	1	2	3	4	5
Our scheme	5.4/4.8	3.1/4.7	4.2/5.3	5.9/5.7	3.6/3.8
DW-MAC	15.5/4.3	16.4/6.4	13.4/5.4	15.7/6.6	12.1/5.8
ADB	14.2/2.1	13.7/2.1	11.9/2.2	14.9/2.7	10.8/2.0

TABLE 5
Average Delay/Standard Deviation in Different Networks ($timeslot = 0.02$ s)

Network	1	2	3	4	5
Our scheme	0.48/0	0.52/0	0.42/0	0.50/0	0.38/0
DW-MAC	9.2/2.5	9.6/3.3	8.5/1.8	9.9/2.9	8.3/1.9
ADB	12.4/0.3	13.4/0.3	10.9/0.2	13.0/0.3	9.5/0.2

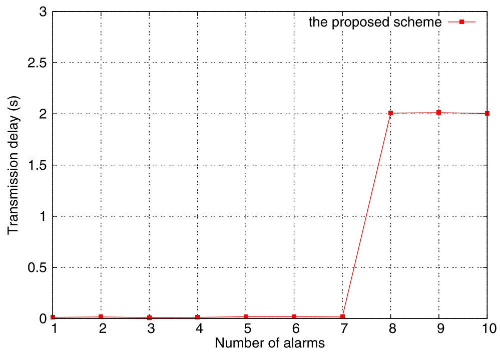


Fig. 8. Transmission delay for multiple alarms.

channel, because some configuration packets also need to be transmitted in the network and the alarm packet needs to be exactly received to avoid misinformation.

We evaluate the performance of the mechanism with a simple and typical network model. Suppose there are M ($1 \leq M \leq 10$) nodes that need to send packets to a parent node which keeps awake for 20 ms every two duty cycles periodically. The quality of the link between the parent node and each child is 70 percent. Suppose the range of the event region is smaller than that of nodes' radio detection. Fig. 8 shows the time when the parent node successfully receives a packet. For each value of M, we conduct 20 experiments and give the maximum time. It is obviously from Fig. 8 that, when $M < 8$, the M children nodes can successfully send one packet to their parent within the 20 ms. When $M = 9$ or 10, it needs two duty cycles to send the packet, resulting in 2 s extra delay. However, the total broadcasting delay is still much lower than that of the improved *DW-MAC* and *ADB* schemes, according to Fig. 7c.

4.2.3 Energy Consumption

We also analyze the energy consumption of sensor nodes with the proposed scheme in WSN. Since the energy consumption is mainly due to the idle listening when there is no critical event most of the time, it is reasonable for us to approximatively calculate the energy consumption according to the length of wake-up duration in a duty cycle. For example, when a MicaZ node turns on its radio module, its current is about 20 mA. Hence, the energy consumption within 5 ms wake-up duration is about $3.3\ \mathrm{V} * 20\ \mathrm{mA} * 5\ \mathrm{ms} = 3.3\ \mathrm{mJ}$.

As described above, most sensor nodes with the proposed scheme stay awake for only τ in a duty cycle and some even stay awake for τ every two cycles. Therefore, the energy consumption of the proposed scheme could not be higher than that of most existing schemes. In addition, when the size of time slot is enlarged, each sensor node with the proposed scheme does not need to keep awake during the whole of the time slot. If a node does not detect the busyness of channel when it wakes up, it will go to sleep immediately.

Note that, before applying the proposed scheme, some initialization works need to be done first, such as obtaining the topology and broadcasting the assignment, which would cost sensor nodes some extra energy. However, this kind of initial cost could be amortized by the afterward low energy consumption of the proposed scheme in the long term of monitoring.

5 Conclusions

In this paper, we proposed a novel sleeping scheme for critical event monitoring in WSNs. The proposed sleeping scheme could essentially decrease the delay of alarm broadcasting from any node in WSN. The upper bound of the delay is $3D + 2L$, which is just a linear combination of hops and duty cycle. Moreover, the alarm broadcasting delay is independent of the density of nodes in WSN.

Theoretical analysis and conducted simulations showed that the broadcasting delay and the energy consumption of the proposed scheme is much lower than that of existing methods.

Acknowledgments

This research was supported in part by the National *S&T* Major Project with Grant 2010ZX03006-002, the NSF of China with Grant 60903171 and 60933012, grants from RGC under the contracts CERG 623209 and 622410, and Huawei-HKUST joint lab.

References

[1] N. Bouabdallah, M.E. Rivero-Angeles, and B. Sericola, "Continuous Monitoring Using Event-Driven Reporting for Cluster-Based Wireless Sensor Networks," *IEEE Trans. Vehicular Technology,* vol. 58, no. 7, pp. 3460-3479, Sept. 2009.

[2] M.I. Brownfield, K. Mehrjoo, A.S. Fayez, and N.J. Davis IV., "Wireless Sensor Network Energy-Adaptive Mac Protocol," *Proc. Third IEEE Consumer Comm. and Networking Conf.,* pp. 778-782, Jan. 2006.

[3] T. Zheng, S. Radhakrishnan, and V. Sarangan, "PMAC: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," *Proc. 19th IEEE Int'l Parallel and Distributed Processing Symp.,* pp. 224-231, Apr. 2005.

[4] S.C. Ergen and P. Varaiya, "TDMA Scheduling Algorithms for Wireless Sensor Networks," *Wireless Networks,* vol. 16, no. 4, pp. 985-997, 2010.

[5] G. Lu, B. Krishnamachari, and C. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Wireless Sensor Networks," *Proc. 18th IEEE Int'l Parallel and Distributed Processing Symp.,* pp. 224-230, Apr. 2004.

[6] A. Keshavarzian, H. Lee, and L. Venkatraman, "Wakeup Scheduling in Wireless Sensor Networks," *Proc. Seventh ACM Int'l Conf. Mobile Ad Hoc Networking and Computing,* pp. 322-333, May 2006.

[7] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, "Delay Efficient Sleep Scheduling in Wireless Sensor Networks," *Proc. 24th IEEE Int'l Conf. Computer Comm.,* pp. 2470-2481, Mar. 2005.

[8] N.A. Vasanthi and S.A., "Energy Efficient Sleep Schedule for Achieving Minimum Latency in Query Based Sensor Networks," *Proc. IEEE Int'l Conf. Sensor Networks, Ubiquitous, and Trustworthy Computing,* pp. 214-219, June 2006.

[9] N.A. Vasanthi and S. Annadurai, "AWS: Asynchronous Wakeup Schedule to Minimize Latency in Wireless Sensor Networks," *Proc. IEEE Int'l Conf. Sensor Networks, Ubiquitous, and Trustworthy Computing,* pp. 144-151, June 2006.

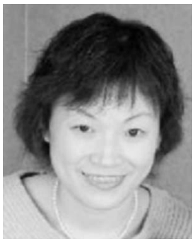
[10] Y. Sun, O. Gurewitz, S. Du, L. Tang, and D.B. Johnson, "ADB: An Efficient Multihop Broadcast Protocol Based on Asynchronous Duty-Cycling in Wireless Sensor Networks," *Proc. Seventh ACM Conf. Embedded Networked Sensor Systems,* pp. 43-56, Nov. 2009.

[11] Y. Sun, S. Du, O. Gurewitz, and D.B. Johnson, "DW-MAC: A Low Latency, Energy Efficient Demand-Wakeup MAC Protocol for Wireless Sensor Networks," *Proc. Ninth ACM Int'l Conf. Mobile Ad Hoc Networking and Computing,* pp. 53-62, 2008.

[12] S.C.-H. Huang, P.-J. Wan, X. Jia, H. Du, and W. Shang, "Minimum-Latency Broadcast Scheduling in Wireless Ad Hoc Networks," *Proc. 26th IEEE Int'l Conf. Computer Comm.,* pp. 733-739, May 2007.

Peng Guo received the MS and PhD degrees from Huazhong University of Science and Technology, Wuhan, China, in 2003 and 2008, respectively. He is currently a lecturer in Department of Electronics and Information Engineering in Huazhong University of Science and Technology.

Tao Jiang is currently a full professor in Wuhan National Laboratory for Optoelectronics, Department of Electronics and Information Engineering, Huazhong University of Science and Technology. He has authored or co-authored more than 70 technical papers in major journals and conferences. His current research interests include cognitive wireless access, OFDM, cooperative networks, nanonetworks and wireless sensor networks. He served or is serving as symposium technical program committee membership of many major IEEE conferences, including INFOCOM, ICC, and GLOBECOM, etc. He is invited to serve as TPC Symposium chair for the International Wireless Communications and Mobile Computing Conference 2010. He is serving as associate editor of some technical journals in communications, including in *IEEE Communications Surveys and Tutorials*, *Wiley's Wireless Communications and Mobile Computing*, etc. He is a recipient of the Best Paper Awards in IEEE CHINACOM09 and WCSP09. He is a senior member of the IEEE, members of IEEE Communication Society, IEEE Broadcasting Society, and IEEE Signal Processing Society.



Qian Zhang received the BS, MS, and PhD degrees from Wuhan University, China, in 1994, 1996, and 1999, respectively, all in computer science. She joined the Hong Kong University of Science and Technology and now is a full professor in the Department of Computer Science and Engineering. Before that, she was with Microsoft Research Asia, Beijing, China, from July 1999, where she was the research manager of the Wireless and Networking Group. She has published more than 250 refereed papers in international leading journals and key conferences in the areas of wireless/Internet multimedia networking, wireless communications and networking, and overlay networking. She is the associate editor for the *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Multimedia*, *IEEE Wireless Communications Magazine*, *Computer Networks*, and *Computer Communications*. She also served as editor for *IEEE Transactions on Wireless Communications* (2005-2009) and *IEEE Transactions on Vehicular Technologies* (2004-2008). She is a Fellow of the IEEE.

Kui Zhang received the BS and MS degrees from Huazhong University of Science and Technology, Wuhan, China, in 2007 and 2009, respectively. He has been a research assistant in Department of Computer Science and Engineering in Hong Kong University of Science and Technology. Now, he is working toward the PhD degree in University of Twente, the Netherlands.