Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2014, Article ID 641720, 8 pages
http://dx.doi.org/10.1155/2014/641720

Research Article

Smart Power Management and Delay Reduction for Target Tracking in Wireless Sensor Networks

Juan Feng, Baowang Lian, and Hongwei Zhao

School of Electronic Information Northwestern Polytechnical University, Xi'an, China

Correspondence should be addressed to Juan Feng; fengjuankh@hotmail.com

Received 3 November 2013; Revised 16 January 2014; Accepted 7 February 2014; Published 11 March 2014

Academic Editor: Vincenzo Eramo

Energy efficiency and data transmission delay are critical issues for mobile target tracking wireless sensor networks, in which abundant sensor nodes are deployed to collect the target information from the sensing field. At present, many existing works have been concentrated on extending network lifetime, while less emphasis was placed on both transmission delay reduction and the adaptive sleep of sensor nodes considering the application constraints. In this paper, we propose a smart power management and delay reduction approach for target tracking based on a grid network structure, where sensor nodes adjust their sleep intervals according to the distance between the node and the moving target. Sensor nodes can distributedly decide their sleeping time using the information from their neighbors. Furthermore, we propose a real-time chain to relay the sensed data for transmission delay reduction. The proposed approach allows sensor nodes that are far away from the target to sleep more and make the target information forward to the sink in time. Experimental results verify that, in contrast to adaptive coordinate and local power management protocols, the proposed approach achieves a significant energy saving while maintaining a short transmission delay.

1. Introduction

Wireless sensor networks (WSNs) have emerged as an attractive technology which can gather information by the collective effort of numerous small sizes, low-cost, and wirelessly connected sensor nodes. Due to many attractive characteristics of sensor nodes, WSNs have very extensive applications. One of the important applications is target tracking, such as vehicle tracking and migration behaviour of animals tracking. In such application scenarios, the sensor nodes collectively monitor the path of moving targets in the sensing area. To inform the sink as quickly as possible when the target occurs, the data transmission delay should be short. Since enormous sensor nodes are always deployed in unattended environment, it is very difficult to replace their battery after the deployment. Therefore, balancing the transmission delay and energy consumption to meet the requirement of the application is the most critical issue.

There are some proposals that also consider these issues, such as [1–4], which propose some power management (PM) approaches. The main idea of PM is dynamically getting nodes to sleep to reduce energy consumption since the power consumption in sleep state is usually much smaller than that in the active or idle mode. Thus, the sensor nodes may be put into a sleep mode as long as possible in order to conserve energy. In order to devise a more efficient PM mechanism, the application constraints should be considered. For instance, in target tracking application, a target moves randomly. We do not know the next position of the target. If we blindly turn the sensor node off during each idling time, we will miss some events. In detail, the existing works are not efficient for target tracking because the sleep time is not adjusted in accordance with the position of the target. Most of the time the nodes consume more energy in active or idle mode, but they do not detect any target.

Moreover, the data transmission delay, which is defined as the time between the moment a source sends a packet and the moment a sink receives the packet, is also a critical issue to be considered in target tracking WSNs. If all the sensor nodes adopt an appropriate sleep time according to the position of the target, each node has the asynchronous wake-up instant. When a node wants to send data, it has to wait until the relay nodes wake up from the sleep mode so that there will be a long data transmission delay. In [5, 6], the authors adopt chain-based network architecture to transmit sensed data. It is an efficient way for data gathering and achieves high energy efficiency since every node just communicates with its closest neighbours. However, it is not fit for the target tracking because it involves all the sensor nodes in the chain for the data gathering so that long data transmission delay is brought, whereas just the nodes around the target need to report their sensed data in the target tracking application. Lots of unnecessary nodes should be moved to reduce the transmission delay.

Based on the these concepts, we propose a smart power management and delay reduction approach (SPM/DR) that considers the application constraints to exploit sleep and idle states. Our main goal is to choose an optimal sleep time for each node so as to make the system adaptive and energy-efficient without degrading the system performance. This paper is based on a grid network structure where a node is selected as grid head (GH) in each grid and we assumed that a sleep node cannot be communicated with or woken up so that the sleep duration has to be determined based on all the available information when the sensor node goes to sleep. We proposed a real-time chain for transmitting sensed data. In a time instant, small parts of GHs in the chain are kept active to relay data and the other GHs are still in sleep to save energy. The information of the target can be transmitted through the chain without delay. In tracking course, only GHs instead of all the sensor nodes send the PM information to their neighbouring GHs, which improves the distributed and coordinated approaches in the existing work. Then each GH weighs the received information according to the distance between itself and its neighbor GH to calculate the sleep time for its grid members (GMs). Therefore, this way decreases the network traffic of PM information transmission and has the better balance between the transmission delay and energy efficiency.

Moreover, in [7], hierarchically coordinated PM is adopted which divides the network into hierarchical layers. Just one layer is in tracking state in one time instance and the others are in sleep state to save energy. This paper improves the approach proposed in [7] from the following aspects: (1) this paper proposed a real-time chain for transmitting sensed data in order to have the better balance between the transmission delay and energy efficiency; (2) for the sleep strategy, this paper adopts the distributed approach instead in which the central sink controls the type of hierarchical layer and sleep strategy in each hierarchical layer, in which the information of moving target needs to be transmitted to the sink in time in order to make decision. Nevertheless, in SPM/DR, each GH decides its sleep interval by the information from its neighbour GHs; (3) SPM/DR approach improves the sleep strategy of GHs by allowing GHs to have a fixed sleep interval with local sleep strategy.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 specifies the energy consumption and management model. In Section 4, SPM/DR approach is described in detail. Then, Section 5 shows the experimental results of SPM/DR compared with the state-of-the-art approaches. Finally, we conclude the paper in Section 6.

2. Related Works

The lifetime of a sensor network depends highly on the power consumption performed at each sensor node. Recently, many target tracking and sensed data gathering approaches have been proposed for WSNs to achieve satisfactory network lifetime and data transmission delay. In [8, 9], the authors design energy efficient communication process at hardware and system levels. In [1, 10], dynamic power management schemes have been proposed to reduce the power consumption.

In [2, 11], the authors introduce some local PM policies, in which nodes reduce the power consumption by selectively shutting down idle components. Each node keeps a timer recording how long no event has been detected and goes to sleep after this timer times out. After a fixed sleep time, the node returns to active state. The authors in [3] use an adaptive learning tree scheme such that the quality of the shutdown control algorithm depends on the knowledge of the user behaviors. However, these policies are intended for general networks. They do not consider the characteristics of target tracking and use the application constraints in a DPM scheme.

For target tracking in WSN, the authors in [8] propose an adaptive coordinate PM policy, which extends the local PM decision to include the timeout values of the neighbours in the network. Once a timeout occurs within a node, it piggybacks a “timeout” message onto its other regularly scheduled messages for wireless broadcast. The receiving nodes strip this timeout message from the packet and forward the packet to subsequent nodes if necessary. In this way, nodes will be aware of the timeout status of their neighbours and can enter into low-power mode if it and each of its neighbours are simultaneously in a timeout state. In [9], a voting PM policy is proposed, in which each node broadcasts periodically a summary of target detection information. Each node collects this summary from its neighbors. If enough of the neighbors vote that there is no event being detected, then the node can enter the low power state. However, due to the dense nodes in WSN, the nodes in an adjacent area have the similar detecting information. If every node broadcasts its detecting information to its neighbours, it will result in more transmission energy consumption and information redundancy. In addition, when nodes make a PM decision, they do not consider different effect of each neighbor. Because of random network distribution, each neighbor has different distance to the current node so that each neighbor has different effect on the current node. In [10], the authors dynamically change the sleep schedule of sensor nodes considering how many hops from the node to the target. However, when a node far from the sink detects a target, it has to wait until the nodes which are far from the target and close to the sink wake up from sleep mode to transmit data. Thus, there will be a long transmission delay for sensed data.

Moreover, in [12], the authors switch off the idle nodes based on the prediction algorithms which predict the next position of the target by the current data from the sensor node to track the target movement. Nevertheless, the prediction algorithms always have high computation complexity and communication energy cost and need to be implemented on the sink after the sink receives all the sensed data.

3. Energy Model

The energy models of data transmitting are similar to that in [5], which is described in (1). $E_{Tx}(k,d)$ and $E_{Rx}(k)$ represent energy consumption of transmitting and receiving k bits data over a distance d

$$E_{Tx}(k,d) = \left(E_{Tx\text{-elec}} + \varepsilon_{\text{amp}} * d^{\alpha}\right) * k,$$
$$E_{Rx}(k) = E_{Rx\text{-elec}} * k, \tag{1}$$

where $E_{Tx\text{-elec}}$ and $E_{Rx\text{-elec}}$ are distance independent terms that take into account overheads of transmitter and receiver electronics. ε_{amp} [Joule/(bit $\cdot$ m^{α})] is a constant which represents the energy needed to transmit one bit to achieve an acceptable signal-to-noise ratio over a distance d, and α is path loss exponent ($2 \leq \alpha \leq 5$) which depends on the channel characteristics. According to [5] we assume that $E_{Tx\text{-elec}} = E_{Rx\text{-elec}} = E_{\text{elec}}$.

Sensor node has several sleep states because the components of the nodes can be turn on or shut off. In target tracking, if a target will appear within the sensing area of a node, the node should be awake in advance to sense the target and transmit data. For the other nodes, they remain in the sleep state s_k most of the time and switch to active state s_0 at specified time slots to check if there are sensing or relay tasks in the next time instant. Therefore, if sensor nodes cannot change their states in time, the energy will be wasted or the tracking performance is reduced. Now, the problem is how to formulate a policy for sensor node to transfer between these sleep states to maximize the lifetime of the network.

Each sleep state s_k has power consumption P_k. The transition times to it from active state and back are denoted by $\tau_{d,k}$ and $\tau_{u,k}$, respectively [1]. We define the node sleep states as $i > j$, $P_j > P_i$, $\tau_{d,i} > \tau_{d,j}$, and $\tau_{u,i} > \tau_{u,j}$. From that we can see deeper sleep state has less power consumption but incurs a longer latency and a higher energy to awaken. We assume an event is detected by N_i at some time. N_i finishes processing the event at time t_1 and predicts that the next event occurs at time $t_2 = t_1 + t_i + \tau_{u,k}$. At time t_1, N_i decides if it transfers to sleep. So a sleep time threshold $T_{\text{th},k}$ is utilized to avoid losing event

$$T_{\text{th},k} = \tau_{d,k} + \tau_{u,k}. \tag{2}$$

If $(t_2 - t_1) > T_{\text{th},k}$, N_i can go to sleep state s_k at time t_1 and wake up at t_2. Otherwise, when $(t_2 - t_1) \leqslant T_{\text{th},k}$, N_i should not go to the sleep state. So the saving energy from a state transition can be calculated as follows:

$$\begin{aligned} E_{\text{save},k} &= P_0\left(t_i + \tau_{u,k}\right) - \frac{P_0 + P_k}{2}\left(\tau_{d,k} + \tau_{u,k}\right) \\ &\quad - P_k\left(t_i - \tau_{d,k}\right) \\ &= \frac{P_0 - P_k}{2}\left(2t_i - \tau_{d,k} + \tau_{u,k}\right). \end{aligned} \tag{3}$$

The energy saving makes sense when $E_{\text{save},k} > E_c$, where E_c is the additional energy consumption for the sleep states transition. So we can find out the threshold,

$$T_{\text{th},k} = \frac{1}{2}\left(\tau_{d,k} - \tau_{u,k}\right) + \frac{E_c}{P_0 - P_k}. \tag{4}$$

Then, it is clear that the node should get into a sleep state only when its idle period will be long enough. In this paper, the nodes can estimate their idle period more accurately by information and weights from its neighbours.

4. Smart Power Management and Delay Reduction

Our idea is to exploit the long intervals when there is no target in sensing regions and offer the more sleep time to sensor nodes without degrading the system performance. We divide the PM approach into two subapproaches. First, only fringe nodes are kept alert while interior nodes have more sleep time in surveillance stage. Second, each GH adjusts the sleep time for its GMs in light of the information from its neighbour GHs in tracking stage, and the sensed data is sent to the sink through a real-time chain in order to reduce the transmission delay.

4.1. Network Initialization. We consider a static WSN which is composed of one sink and some randomly distributed sensor nodes N_i, $i \in [1, P]$ in a two-dimensional sensing field, where P is the number of the deployed nodes. The sink has an infinite power supply, and it gathers the sensed information from sensor nodes. The distribution of sensor nodes is mutually independent with density $\lambda = P/S_{\text{sensing}}$, where S_{sensing} is the area of the sensing field. We assume each node is aware of its location after deployment (e.g., using some localization techniques). Let $X_i(x_i, y_i)$, $1 \leq i \leq P$ be the location of node N_i.

Furthermore, the whole sensing field is divided into small equal size grids and in each grid one node which has the most energy is selected as the GH. In the definition of virtual grid, each pair of nodes in neighbour and diagonal grids can communicate directly with each other [1, 11]. Assume that the transmission range of sensor node is R_t. We size each grid as a $\alpha \times \alpha$ square. In order to meet the definition of virtual grid, in any two adjacent grids, the distance between two possible farthest nodes must not be larger than R_t. Therefore, we get

$$(2\alpha)^2 + (2\alpha)^2 \leq R_t^2 \quad \text{or} \quad \alpha \leq \frac{R_t}{2\sqrt{2}}. \tag{5}$$

Initially, all the sensor nodes are in idle state and they have the same initial energy. One node in each grid is selected as the GH randomly by broadcasting an announcement (carrying its position and grid ID) after waiting for a random time period $R(T_{\text{ran}})$, which is a discrete random variable with the uniform distribution in $[0, T_{\text{ran}}]$. The node who first broadcasts its GH announcement in one grid will be the GH. After the process, there is one GH in each grid. In addition, each grid has a grid ID, and the nodes in one grid have the same grid ID. A sensor node can calculate its grid ID (u, v) from its location $X_i(x_i, y_i)$ as

$$u = \left[\frac{x_i - x_0}{\alpha} \right], \qquad v = \left[\frac{y_i - y_0}{\alpha} \right], \tag{6}$$

where [ ] is a symbol which stands for the integer part of the number in it. (x_0, y_0) is the location of the network origin, which is a system parameter set in the network initialization stage. For simplicity, we assume that u and v are positive. Through the initial process, each node can learn the information about other nodes in the same grid and maintains the information, for example, the location, and so on.

4.2. A Real-Time Chain for Transmitting Sensed Data. For target tracking application, another important factor to be considered is the detecting delay. Clearly, when any sensor node detects a target, it needs to report to the sink through multihop routing in time. However, increased energy savings generally come with a penalty of increased detecting delay. To resolve this problem, we consider an important characteristic of the tracking scenario in which once a target is detected, the detecting sensor nodes need to continuously send the move information of the target to the sink. As a result, each detecting node has to calculate the routing for data transmission per round, and it has to wait for the next routing node to be in active mode in order to transmit data. That involves an amount of calculations and long delay.

We propose a real-time chain for transmitting the sensed data. When a target appears, the node who detects the target first will broadcast a message "*found_target*" to other nodes including its GH in SS. Then the GH broadcasts the "*found_target*" message to its neighbour GHs to inform them to get ready for detection. The receiving GHs forward the message to the subsequent GHs until all the GHs receive the message. Then a real-time chain from the first detected node to the sink is formed using any routing algorithm (such as some routing protocol in [13]) as Figure 1 shows. The GHs in this transmission chain should keep in active mode to guarantee the short delay for sensed data transmission. Otherwise, the other GHs keep synchronization and use the local PM policy, which keep a timer recording how long no event has been detected and goes to sleep after this timer times out. After a fixed sleep time, they return to active state. If they detect a target, they keep active at the next time instant. Thus, only the nodes around the target and GHs in the real-time chain should keep active and other nodes can sleep adaptively to save energy. Moreover, the sink can estimate the target position by received data. If the moving distance of the target is larger than a threshold T_d, the sink will inform the GHs in the real-time chain and the adjacent GHs to reform a chain. In this way, the sensed data can be sent to the sink with minimum delay. In addition, a lot of energy is saved because computational costs and the changing frequency between sleep and active modes of routing nodes are reduced.

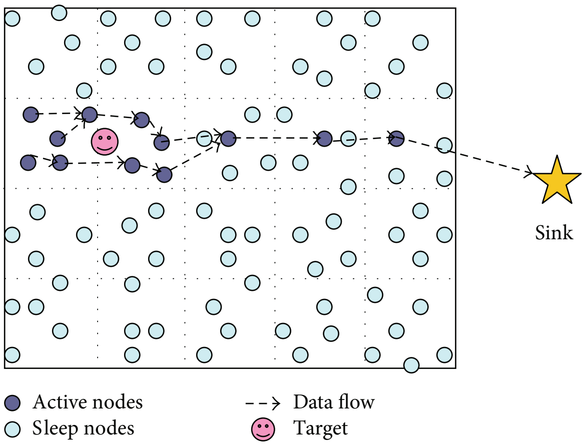


Figure 1: A real-time chain for transmitting sensed data.

4.3. Sleep Policy in Tracking Stage. For tracking stage, the position of the moving target in this time instant has strong correlation with that in next time instant. Therefore, to more effectively make PM decisions and estimate if the target will be in itself sensing area in the next instant, each node can use the results of the motion detection from its neighbors in a coordinate way. As Figure 2 (on the left side) shows, each node broadcast its detected information to its neighbors periodically. When the current node N_c decides if it should go to sleep state, it will use the detected information from its neighbors. However, due to the dense nodes in WSN, the nodes in an adjacent area have the similar detected information. If every node broadcasts its detected information periodically, it will result in more transmission energy consumption and information redundancy. In our approach (on the right side in Figure 2), only the GH who detects a target or receives the target information from its GMs broadcasts the detected information to their neighboring GHs. Each GH can decide if the sleep time of its GMs needs to be changed based on the information it received. If it needs to be changed, the GH recalculates the sleep time for its GMs and informs them in the next active instant, which reduce the energy consumption of the detected information transmission.

4.3.1. One Hop Neighbors Coordination. One hop neighbors coordination means when a GH makes PM decisions, it just considers the detected information from its immediate neighbor GHs. The detected information broadcasted by a GH includes the location of the sender and the distance between the sender and the target. It can be estimated by the strength of the sensing signal. A GH calculates the sleep time for its GMs in three different cases.

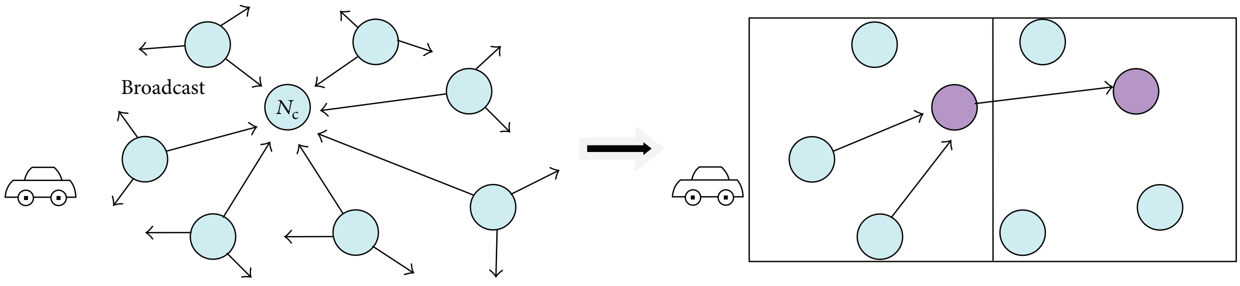


Figure 2: The transmission of the target detecting information.

The first case is that the GH neither detects any target nor receives any target information from others during an active time period. The maximum sleep time is decided by the following equation:

$$\max t_{\text{sleep}}^{\text{tk}} = \frac{\left((\alpha/v_{\max}) - \tau_{d,\text{sp}} - \tau_{u,\text{sp}}\right)}{\eta}, \qquad (7)$$

where α is the side length of the grid and η is the success ratio of detecting required by application.

The second case is that the target is detected by the GH or its GMs. The GH keeps active until the target moves out of sensing area of its grid, and it decides the sleep time of its GMs $t_{\text{sleep}}^{\text{tk}} = 0$. Then the GH broadcasts a detected information message to its neighbour GHs so that they can prepare to track the target at the next moment.

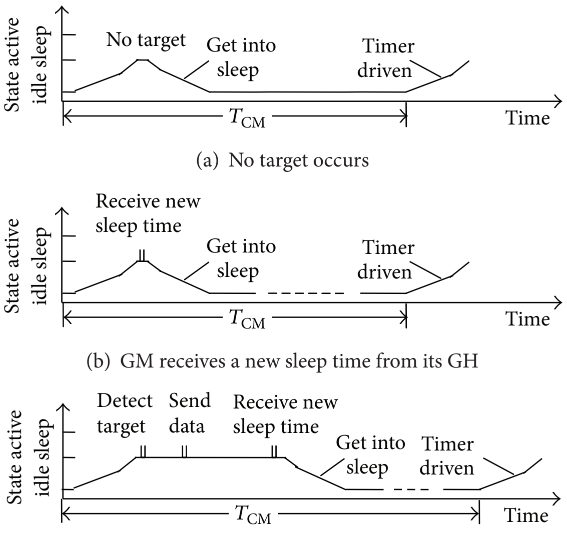
The third case is that the GH receives the target information from its neighbors. Thus, the GH can calculate the maximum sleep time for its GMs by the following equation:

$$\max t_{\text{sleep}}^{\text{tk}} = \frac{\left((\min\left\{d_{\text{GMs},\mathit{detectNode}}\right\}/v_{\max}) - \tau_{d,\text{sp}} - \tau_{u,\text{sp}}\right)}{\eta}, \qquad (8)$$

where $\min\{d_{\text{GMs},\mathit{detectNode}}\}$ is the minimum distance between the GMs of the current grid and the node who detects the target. Similarly, to overlap the active parts of GMs and their GH so that GMs and their GH communicate with each other, we get

$$t_{\text{sleep}}^{\text{tk}\,\prime} = \left[\frac{\max t_{\text{sleep}}^{\text{tk}}}{T_{\text{GH}}}\right] T_{\text{GH}} - \tau_{u,\text{sp}}, \qquad (9)$$

where T_{GH} is the period of GHs. The sleep time of GM is set to the integer multiple of T_{GH} to overlap the active parts of GMs and their GH so that the GMs can send their sensing data to their GHs or receive the PM information from their GHs to adjust their sleep time accordingly. When $t_{\text{sleep}}^{\text{tk}\,\prime} > T_{\text{th}^1}$ and $t_{\text{sleep}}^{\text{tk}\,\prime} > T_{\text{th}^2}$ hold, the sleep time of the GMs in tracking stage $t_{\text{sleep}}^{\text{tk}}$ can be set as $\min\{t_{\text{sleep}}^{\text{tk}\,\prime}, t_{\text{sleep}}^{\text{sur}}\}$, in which $t_{\text{sleep}}^{\text{sur}}$ is the sleep time of the GMs in the surveillance stage, otherwise $t_{\text{sleep}}^{\text{tk}} = 0$. When there is change on the sleep time value of GMs, the GH informs its GMs about the updated sleep time. The GMs receive a new sleep time from their GH and adjust the current sleep mode. The sleep state transition of the GM is shown in Figure 3.



(a) No target occurs

(b) GM receives a new sleep time from its GH

(c) GM detects a target

Figure 3: The sleep state transition of the GM.

In this way, sensor nodes will get the target movement information from the GHs in neighboring area and accurately estimate the sleep state and time interval. In addition, transmitting the target information only among GHs instead of that among all the sensor nodes can reduce the communication cost and information redundancy.

4.3.2. Multihop Policy Considerations. In our aforementioned policies, when GHs make the PM decisions, they consider the detected information from the immediate neighbor GHs. We can also increase the coordination to the neighbor GHs located multihop away from the current GH. This requires a balance between the energy consumption of the detected information transmitted to multihops and the energy saving of GMs that can sleep more. We assume that the energy consumption of detected information sending and receiving is E_{br} and E_{re}, respectively. When GHs consider the detected information only from the immediate neighbor GHs, the energy consumption of the detected information transmission is

$$E_{\text{cons}}^{h=1} = E_{\text{br}} + 8E_{\text{re}}, \qquad (10)$$

when the detected information is transmitted to the neighbors which are $h = H$ hops far away from the current node

$$E^{h>1}_{\text{cons}} = E^{h=1}_{\text{cons}} + \sum_{h=2}^{H} (8h - 8) E_{\text{br}} + \sum_{h=2}^{H} 8hE_{\text{re}}. \quad (11)$$

When $h = 1$, the energy saving by GMs sleeping is

$$\begin{aligned} E^{h=1}_{\text{save}} &= \left[(P_i - P_s) t^{tk}_{\text{sleep}} + (P_i - P_{\text{tr}^1}) \tau_{d,\text{sp}} + (P_i - P_{\text{tr}^2}) \tau_{u,\text{sp}} \right] \\ &\quad \times N_{\text{CMs}}, \end{aligned} \quad (12)$$

where N_{CMs} is the number of the GMs in the current grid. Similar to $h = 1$, when $h = H$, the sleep time of GMs is

$$t^{ml-h'}_{\text{sleep}} = \left[\frac{(H\alpha/v_{\max}) - \tau_{d,\text{sp}} - \tau_{u,\text{sp}}}{\eta T_{\text{GH}}} \right] T_{\text{GH}} - \tau_{u,\text{sp}}, \quad (13)$$

and when $h = H$, the energy saving by GMs sleeping is

$$\begin{aligned} E^{h>1}_{\text{save}} &= \left[(P_i - P_s) t^{ml-h'}_{\text{sleep}} + (P_i - P_{\text{tr}^1}) \tau_{d,\text{sp}} + (P_i - P_{\text{tr}^2}) \tau_{u,\text{sp}} \right] \\ &\quad \times N_{\text{CMs}}; \end{aligned} \quad (14)$$

therefore, when $h = H$, we can obtain the energy saving as follows:

$$\begin{aligned} E^{h>1}_{\text{save}} - E^{h>1}_{\text{cons}} &= (P_i - P_s) N_{\text{CMs}} \left[\frac{(H\alpha/v_{\max}) - \tau_{d,\text{sp}} - \tau_{u,\text{sp}}}{\eta T_{\text{GH}}} \right] T_{\text{GH}} \\ &\quad - 4H^2 (E_{\text{br}} + E_{\text{re}}) - 4H (E_{\text{re}} - E_{\text{br}}) + C, \end{aligned} \quad (15)$$

where C is a constant which is not related to H,

$$C = N_{\text{CMs}} \left[(P_i - P_{\text{tr}^1}) \tau_{d,\text{sp}} + (P_s - P_{\text{tr}^2}) \tau_{u,\text{sp}} \right] - E_{\text{br}}. \quad (16)$$

To optimize the energy saving, the value of $E^{h>1}_{\text{save}} - E^{h>1}_{\text{cons}}$ needs to be the maximum. Since $P_i - P_s > 0$, $N_{\text{CMs}} > 0$ and $\tau_{d,\text{sp}}$, $\tau_{u,\text{sp}}$, T_{GH}, and η are all the constants. Thus, when the value of y in (17) is maximized, we can get the maximum value of $E^{h>1}_{\text{save}} - E^{h>1}_{\text{cons}}$:

$$\begin{aligned} y = {} & (P_i - P_s) N_{\text{CMs}} \frac{\alpha}{v_{\max}} H - 4H^2 (E_{\text{br}} + E_{\text{re}}) \\ & - 4H (E_{\text{re}} - E_{\text{br}}) + C. \end{aligned} \quad (17)$$

To find the maximum y, H is differentiated as

$$\frac{\partial y}{\partial H} = (P_i - P_s) N_{\text{CMs}} \frac{\alpha}{v_{\max}} - 8H (E_{\text{br}} + E_{\text{re}}) - 4 (E_{\text{re}} - E_{\text{br}}); \quad (18)$$

```
(1) while (after initiating or received "req-replace")
(2) do
(3)   node $N_i$ set a timer
(4)   $T_i \leftarrow$ CalculateWaitTime ($E^i_{\text{residual}}$)
(5)   Wait ($T_i$)
(6)       if wait time expired
(7)         Broadcast ("finish_election")
(8)         select $N_i$ as GH
(9)       end if
(10)      if received ("finish_election" from $N_j$)
(11)        cancel wait ( )
(12)        select $N_j$ as GH
(13)      end if
(14)    end if
(15) end while
```

Algorithm 1: Selecting GH in one grid.

thus, when $\partial y/\partial H = 0$, we have the maximum y and get H as follows:

$$H = \frac{(P_i - P_s) N_{\text{CMs}} (\alpha/v_{\max}) + 4 (E_{\text{br}} - E_{\text{re}})}{8 (E_{\text{br}} + E_{\text{re}})}; \quad (19)$$

therefore, the maximum saving energy is

$$E_s = \max \left\{ \left(E^{h=1}_{\text{save}} - E^{h=1}_{\text{cons}} \right), \left(E^{h>1}_{\text{save}} - E^{h>1}_{\text{cons}} \right) \right\}. \quad (20)$$

We can select appropriate H to obtain the optimal energy saving policy according to the network parameters. When $t^{ml-h'}_{\text{sleep}} \geq T_{\text{th}^1}$ and $t^{ml-h'}_{\text{sleep}} > T_{\text{th}^2}$ hold, the sleep time of the GMs in tracking stage with multihop GHs coordination can be set as $\min\{t^{ml-h'}_{\text{sleep}}, t^{\text{sur}}_{\text{sleep}}\}$.

4.4. Grid Maintenance. To avoid energy overfull consuming on GH, if the energy of any GH is lower than a threshold T_e, the GH will broadcast a message "*req-replace*" to its GMs. Then the nodes in the same grid reselect the GH. A node with more remaining energy is expected to become a new GH.

The pseudo code of the GH selection is shown in Algorithm 1. When GM receives "*req-replace*" message, it waits a random time to broadcast "*finish_election*" message to compete the GH election. For each GM the residual energy E^i_{residual} is converted to a waiting time T_i (line 4). More residual energy leads to a shorter T_i. Therefore, the GM with the most E^i_{residual} waits for the shortest time before sending "*finish_election*" message to others. Therefore, it can be selected as GH (line 8 and 12). The other nodes stop waiting and give up the GH election as soon as they receive the message. T_i is calculated by the following equation:

$$T_i = T_{\min} + (T_{\max} - T_{\min}) \left(1 - \frac{E^i_{\text{residual}}}{E^i_{\text{initial}}} \right) + R(T_{\text{ran}}), \quad (21)$$

where $T_{\min}$ and $T_{\max}$ are two design parameters, which are used to control the waiting time in a reasonable range.

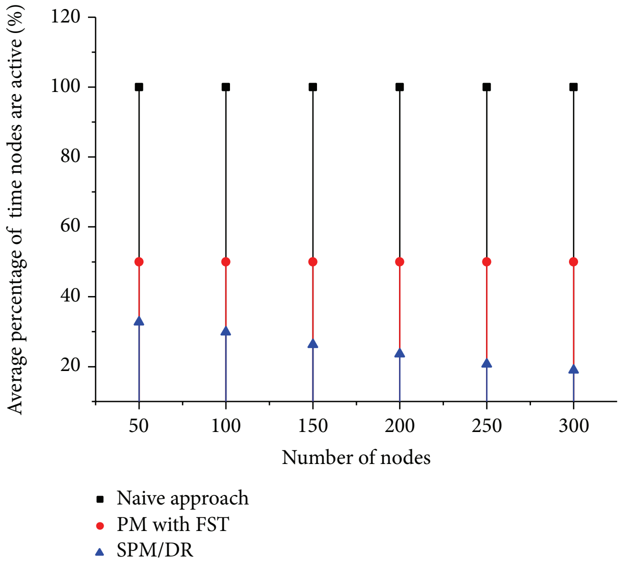


FIGURE 4: Average percentage of time nodes are active.

E_{residual}^{i} and E_{initial}^{i} are the residual and initial energy of node, respectively. E_{residual}^{i} is divided by E_{initial}^{i} in order to avoid the case that a node waits too much time when its residual energy becomes very low. Two nodes might have the same residual energy and communication costs and therefore have the same T_i. To avoid this a random time $R(T_{\text{ran}})$ is added. $R(T_{\text{ran}})$ is a discrete random variable with the uniform distribution in $[0, T_{\text{ran}}]$, which is an order of magnitude smaller than $T_{\max}$.

If GH election failed, maybe due to the loss of the broadcast messages, the old GH will continue its role and broadcast the reelection request periodically. For reliability purpose, when a GM fails to send data to its GH for several times (e.g., a GH dies suddenly), it will send a GH reelection request to the GMs.

5. Simulations and Analysis

5.1. Experiment Environment. We assume that there are 400 sensor nodes distributed randomly over an area of 150 m by 150 m. And the sensing range of each sensor node is $r = 15$ m. We also assume each node has an initial energy of 100 J (Joules).

We set $T_e = 1/3$ (initial energy) to avoid energy overfull consuming on GH. T_d is set to 30 m. If the moving distance of the target is larger than T_d, the real-time chain will be adjusted or reformed. The bandwidth of wireless channel is 2.4 Kbps, and the data packet size is 512 bytes.

5.2. Simulation Results. To compare the performance of SPM/DR with other schedules, we implemented the other three approaches, namely, (1) No PM approach (nodes are always on), (2) local PM approach (all the nodes sleep periodically if they did not detect any event in 2 s and the fixed sleep time is also set as 2 s), and (3) adaptive coordinate PM approach.

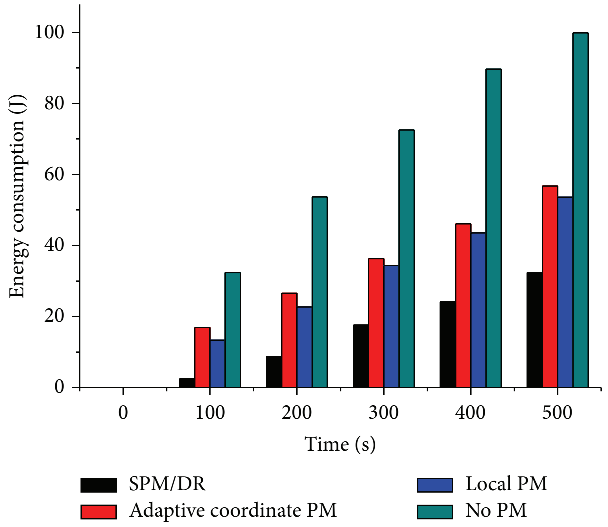


FIGURE 5: Average energy consumption in surveillance stage.

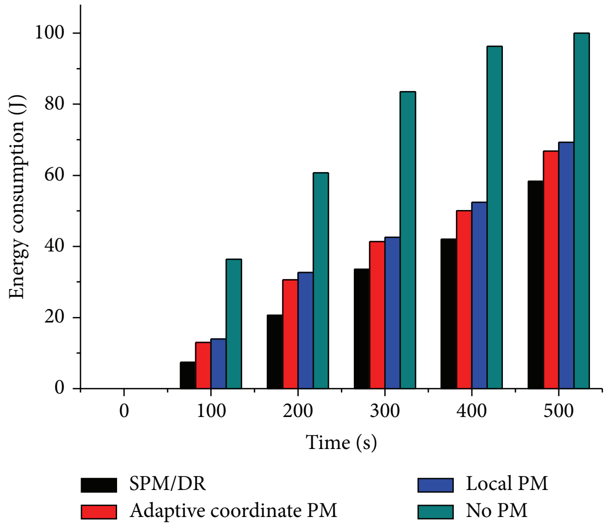


FIGURE 6: Average energy consumption in tracking stage.

Figure 4 shows the average percentage of time when nodes are in active mode in surveillance stage. It can be seen that most of the nodes in SPM/DR approach are in sleep mode because we just keep the network border nodes alert and the other nodes have a lot of sleep time.

Figure 5 shows the average energy changed with simulation time of the approaches in our simulation. It can be seen that SPM/DR approach can save about 35% more energy compared to adaptive coordinate PM approach in the surveillance stage. No PM approach network consumes the most energy because nodes are always active.

Next, we assume the target enters the field at a random location and moves at a constant speed of 5 m/s. Figure 6 shows the average energy comparison of three approaches. Here, x-axis displays the time of simulation and y-axis displays the average energy consumption of every node at the corresponding time. It can be seen that SPM/DR saves

Table 1: Average transmission delay.

PM approach	Delay
SPM/DR	0.76 ms
Adaptive coordinate PM	0.93 ms
local PM	1.25 ms
No PM	0.59 ms

about 15% more energy compared to adaptive coordinate PM approach.

We can see that adaptive coordinate PM approach consumes more energy than SPM/DR due to more communication cost among neighbor nodes. The local PM approach also has more energy consumption because the nodes far from the target cannot have more sleep time. However our SPM/DR approach consumes the least energy because the sensor nodes in our approach have adaptive sleep time which allowed the nodes far from the target sleep longer.

Table 1 shows the average transmission delay in different approaches. From that we can see no PM approach has the shortest transmission delay since all the nodes are always in active mode. SPM/DR approach performs better than the other two approaches because the GHs in the real-time chain keep active for the sensed data transmission.

In a word, SPM/DR approach can conserve more energy for target tracking WSN. Moreover, it did not degrade tracking performance since we evaluated sleep time adaptively by target move information before the target comes up.

6. Conclusions

This paper proposed a novel smart power management and delay reduction approach (SPM/DR) for tracking target in WSN. It can reduce energy consumption and increase the network lifetime with short data transmission delay. SPM/DR outperforms the other PM approaches by allowing more nodes to have longer sleep intervals and tracks the target by dynamically changing the schedule. Moreover, we use a real-time chain for transmitting the sensed data so that the transmission delay can be reduced. Simulation results proved that SPM/DR performs better than local and adaptive coordinate PM approach for target tracking.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61301094) and NPU Foundation for Fundamental Research (NPU-FFR-JCY20130135).

References

[1] A. Sinha and A. Chandrakasan, "Dynamic power management in wireless sensor networks," *IEEE Design and Test of Computers*, vol. 18, no. 2, pp. 62–74, 2001.

[2] P. S. Sausen, M. A. Spohn, and A. Perkusich, "Broadcast routing in wireless sensor networks with dynamic power management and multi-coverage backbones," *Information Sciences*, vol. 180, no. 5, pp. 653–663, 2010.

[3] S. Bhatti and J. Xu, "Survey of target tracking protocols using wireless sensor network," in *Proceedings of the 5th International Conference on Wireless and Mobile Communications (ICWMC '09)*, pp. 110–115, August 2009.

[4] R. Olfati-Saber and P. Jalalkamali, "Collaborative target tracking using distributed Kalman filtering on mobile sensor networks," in *Proceedings of the American Control Conference (ACC '11)*, pp. 1100–1105, San Francisco, Calif, USA.

[5] Q. Mamun, S. Ramakrishnan, and B. Srinivasan, "Selecting member nodes in a chain oriented WSN," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '10)*, April 2010.

[6] Q. Mamun, S. Ramakrishnan, and B. Srinivasan, "An efficient localized chain construction scheme for chain oriented wireless sensor networks," in *Proceedings of the 10th International Symposium on Autonomous Decentralized Systems (ISADS '11)*, pp. 3–9, March 2011.

[7] Juan Feng, Baowang Lian, and Hongwei Zhao, "Hierarchically coordinated power management for target tracking in wireless sensor networks," *International Journal of Advanced Robotic Systems*, vol. 10, article 347, 2013.

[8] N. H. Zamora, J.-C. Kao, and R. Marculescu, "Distributed power-management techniques for wireless network video systems," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, pp. 564–569, April 2007.

[9] N. H. Zamora and R. Marculescu, "Coordinated distributed power management with video sensor networks: analysis, simulation, and prototyping," in *Proceedings of the 1st ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC '07)*, pp. 4–11, September 2007.

[10] S. Anandamurugan and C. Venkatesh, "Power saving method for target tracking sensor networks to improve the lifetime," *International Journal of Recent Trends in Engineering*, vol. 1, pp. 594–596, 2009.

[11] B. Brock and K. Rajamani, "Dynamic power management for embedded system," in *Proceedings of the IEEE International System-on-Chip (SOC) Conference*, pp. 416–4419, September 2003.

[12] Y. Xu, J. Winter, and W.-C. Lee, "Dual prediction-based reporting for object tracking sensor networks," in *Proceedings of 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS '04)*, pp. 154–163, August 2004.

[13] S. K. Singh, M. P. Singh, and D. K. Singh, "Routing protocols in wireless sensor networks—a survey," *International Journal of Computer Science & Engineering Survey*, vol. 1, no. 2, 2010.

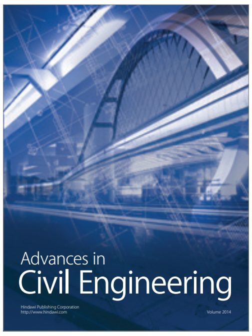
Journal of
Engineering

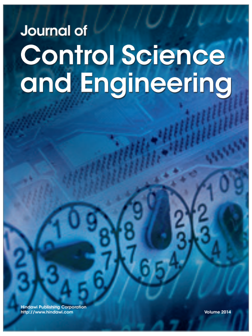
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

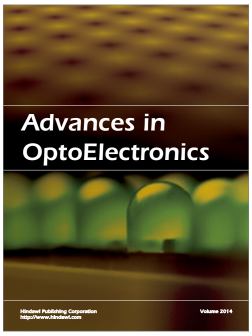
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

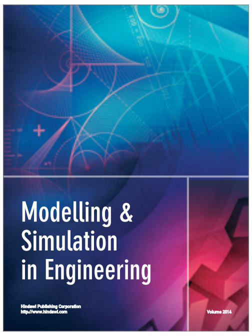
Journal of
Robotics

Journal of
Electrical and Computer Engineering

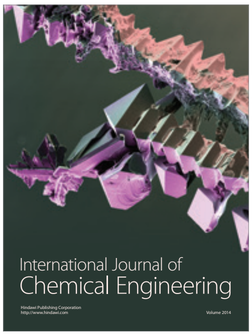

Advances in
OptoElectronics

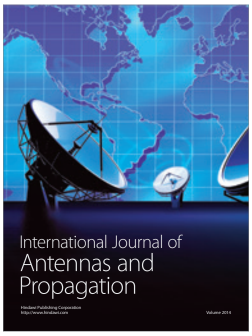
VLSI Design

International Journal of
Navigation and Observation

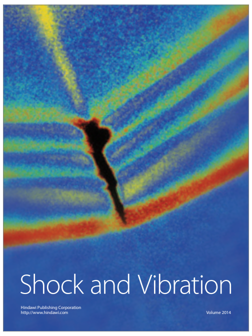

Modelling & Simulation in Engineering

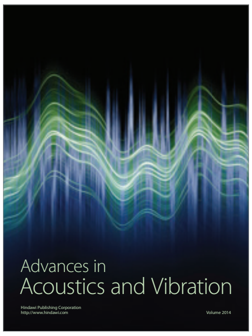
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration