

Full Paper

Prediction-based Dynamic Energy Management in Wireless Sensor Networks

Xue Wang *, Jun-Jie Ma , Sheng Wang and Dao-Wei Bi

State Key Laboratory of Precision Measurement Technology and Instruments, Department of Precision Instruments, Tsinghua University, Beijing 100084, P. R. China; E-mails: {wang_sheng00, mjj, bdw02}@mails.tsinghua.edu.cn.

* Author to whom correspondence should be addressed; E-mail: wangxue@mail.tsinghua.edu.cn.

Received: 28 January 2007 / Accepted: 1 March 2007 / Published: 5 March 2007

Abstract: Energy consumption is a critical constraint in wireless sensor networks. Focusing on the energy efficiency problem of wireless sensor networks, this paper proposes a method of prediction-based dynamic energy management. A particle filter was introduced to predict a target state, which was adopted to awaken wireless sensor nodes so that their sleep time was prolonged. With the distributed computing capability of nodes, an optimization approach of distributed genetic algorithm and simulated annealing was proposed to minimize the energy consumption of measurement. Considering the application of target tracking, we implemented target position prediction, node sleep scheduling and optimal sensing node selection. Moreover, a routing scheme of forwarding nodes was presented to achieve extra energy conservation. Experimental results of target tracking verified that energy-efficiency is enhanced by prediction-based dynamic energy management.

Keywords: Wireless sensor network, dynamic energy management, energy efficiency, particle filter.

1. Introduction

Wireless sensor networks (WSNs) utilize a large number of intelligent micro-sensor nodes with sensing, processing and wireless communicating capabilities to implement complicated tasks in the specific sensing area. With the strict energy constraints of wireless sensor nodes, dynamic energy management has become a challenging issue that must be addressed in WSNs. To treat the problem of energy efficiency, it is easy to envision that certain energy optimization mechanisms may be applicable. In particular, the dynamic energy management architecture has been proposed in [1]. The awaken scheme has been carefully discussed at the protocol level [2]. As energy consumption of WSNs is application-oriented, energy management of specified applications, such as target tracking, should be studied. However, very little has been done for specified applications and therefore none of these methods make use of the information about the target state, which contains rich hints for a more reasonable sleeping and sensing scheduling of nodes. Also, optimization performance may be enhanced by making use of distributed computing capability.

Considering the energy efficiency of measurement in WSNs, we present a prediction-based dynamic energy management method, which takes node idle time and communication energy consumption into account. First, particle filters (PF), which can estimate non-linear and non-Gaussian dynamic processes, could be directly applied to the non-linear system model, while other traditional methods, such as extended Kalman filter (EKF), may bring serious model errors [3,4]. To solve the non-linear problem of target state prediction, PF is employed here. Then, idle time is estimated and a sleep schedule is designed for each node so that any node can become a sensing candidate on time. Moreover, we perform in advance reasonable optimization of the sensing process. Since communication energy is a critical aspect of energy consumption, distributed genetic algorithm and simulated annealing (DGASA) are presented to optimize communication energy consumption in WSNs, which assigns computation task to a number of nodes to boost up the optimization ability. In the application to target tracking, target position is predicted by PF and node selection optimization is achieved by DGASA. A routing scheme with forwarding nodes is studied as well. Energy consumption is analyzed during target tracking and efficiency of prediction-based dynamic energy management is demonstrated.

The remainder of this paper is organized as follows: Section 2 discusses collaborative sensing model and energy consumption model in WSNs. In Section 3, our prediction-based dynamic energy management mechanism is presented, where we describe several potential awakening mechanisms, target prediction with PF and energy optimization with DGASA. We have studied the target tracking application with prediction-based dynamic energy management in Section 4. Section 5 provides prediction and optimization results during the procedure of target tracking, and analyzes the network energy consumption with optimized node selection employing dynamic energy management. Finally, our conclusions are presented in Section 6.

2. Basic Models

Let us assume that wireless sensor nodes are deployed randomly in the WSN region with the same sensing range r_{sensing} , and sensing period of WSN is Δt . A sink node is located in the center of the

region to collect the data through the network and maintain the information of node positions and communication paths [5,6]. Wireless sensor nodes are densely placed, so that a number of nodes can potentially make simultaneous observations. This will allow a collaborative sensing model and energy consumption model to be introduced.

2.1. Collaborative Sensing Model

Let us assume that each wireless sensor node can produce bearing angle estimates of the target in the sensing range [7]. For the time instant t , the target is located at (x_{target}, y_{target}) and detected by n nodes, where $n > 2$. Each node i , placed at (x_i, y_i) , can acquire the angle:

$$\theta_i = \arctan \frac{y_i - y_{target}}{x_i - x_{target}} + v_i \quad (1)$$

Where v_i is direction finding (DF) error, which is zero-mean, Gaussian distributed with constant variance σ_{θ}^2 .

DF lines of these nodes can't intersect at a common point due to DF error. Therefore, the non-linear least square estimation is adopted for target location, with the nodes sensing collaboratively [8]. Matrix representation for measurement equation of n nodes is:

$$Y = CX + V \quad (2)$$

where

$$Y = \begin{bmatrix} \theta_1 - \hat{\theta}_1 \\ \theta_2 - \hat{\theta}_2 \\ \vdots \\ \theta_n - \hat{\theta}_n \end{bmatrix} \quad C = \begin{bmatrix} \frac{\partial \theta_1}{\partial x_{target}} & \frac{\partial \theta_1}{\partial y_{target}} \\ \frac{\partial \theta_2}{\partial x_{target}} & \frac{\partial \theta_2}{\partial y_{target}} \\ \vdots & \vdots \\ \frac{\partial \theta_n}{\partial x_{target}} & \frac{\partial \theta_n}{\partial y_{target}} \end{bmatrix} \quad V = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \quad X = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

where $\hat{\theta}_i$ ($i=1, \dots, n$) is the estimation of θ_i .

Since $n > 2$, the least square solution of X exist. According to least-square criteria, choose \hat{X} as an estimation of X to minimize $J(\hat{X})$:

$$J(\hat{X}) = V^T V = (Y - C\hat{X})^T (Y - C\hat{X}) \quad (3)$$

We can acquire $X_{LS} = (C^T C)^{-1} C^T Y$ as the least square estimation of X . Defining the estimation of (x_{target}, y_{target}) as \hat{X}_{target} , then it can be obtained in an iterative manner as $\hat{X}_{target}^{(k+1)} = \hat{X}_{target}^{(k)} + \hat{X}^{(k)}$, where $\hat{X}^{(k)}$ and $\hat{X}_{target}^{(k)}$ denote the estimation of X and (x_{target}, y_{target}) in the k -th iteration, respectively.

The estimation error covariance matrix is defined as:

$$\text{cov}(X_{LS}) = \sigma_{\theta}^2 (C^T C)^{-1} = \sigma_{\theta}^2 \begin{bmatrix} q_{xx} & q_{xy} \\ q_{yx} & q_{yy} \end{bmatrix} \quad (4)$$

We can find $q_{xx} = \sigma_x^2 / \sigma_{\theta}^2$, $q_{xy} = q_{yx} = \rho \sigma_x \sigma_y / \sigma_{\theta}^2$ and $q_{yy} = \sigma_y^2 / \sigma_{\theta}^2$, where σ_x^2 is the variance of error distribution on X-axis, σ_y^2 is the variance of error distribution on Y-axis and ρ is the correlation coefficient. The associated quadratic form of the covariance matrix defines an ellipse depicting the

distribution of error in this two-dimensional case. Semimajor axis σ_l , semiminor axis σ_s and semimajor axis azimuth φ of the error ellipse are [9]:

$$\sigma_l = \sigma_\theta \sqrt{[q_{xx} + q_{yy} + \sqrt{(q_{xx} - q_{yy})^2 + 4q_{xy}^2}] / 2} \quad (5)$$

$$\sigma_s = \sigma_\theta \sqrt{[q_{xx} + q_{yy} - \sqrt{(q_{xx} - q_{yy})^2 + 4q_{xy}^2}] / 2} \quad (6)$$

$$\tan 2\varphi = 2q_{xy} / (q_{xx} - q_{yy}) \quad (7)$$

Collaborative sensing accuracy is defined by the error ellipse.

2.2. Energy Consumption Model

Let us assume that each wireless sensor node consists of sensing, processing, memory and communication components. With multiple power modes, these modules can support different levels of power consumption and functionality. Accordingly, each node can have a set of sleep states based on various combinations of module power states [1]. Each state s_k has power consumption P_k , and the transition time from active state and back is given by $\tau_{d,k}$ and $\tau_{u,k}$, respectively, where $\tau_k = \tau_{d,k} = \tau_{u,k}$. As shown in Table 1, we define five states; “Tx” and “Rx” denotes “Transmit” and “Receive” respectively. And the extra power consumption of radio module for data transmission between wireless sensor node i and j is calculated as [10]:

$$\Psi_{Tx} = \alpha_1 r + \alpha_2 d_{ij}^{n_0} r \quad (8)$$

where r denotes the data rate, n_0 denotes the path loss index, α_1 denotes the electronics energy expended in transmitting one bit of data, $\alpha_2 > 0$ is a constant related to the radio energy, and d_{ij} denotes the distance between the two nodes. If we define $t_{p,q} = |\tau_p - \tau_q|$ as the transition time between state s_p and s_q , then energy consumption due to state transition can be calculated as:

$$E_{p,q} = (P_p - P_q) \cdot \frac{t_{p,q}}{2} \quad (9)$$

where p and q are indices of node state referred in transition.

Table 1. Hardware configuration, power consumption and latency threshold of each sleep state

State	s_0	s_1	s_2	s_3	s_4
CPU	Active	Active	Idle	Sleep	Sleep
Memory	Active	Active	Sleep	Sleep	Sleep
Sensor	On	On	On	On	Off
Radio	Tx, Rx	Rx	Rx	Off	Off
P_k (mW)	$450 + \psi_{Tx}$	450	270	200	10
τ_k (ms)	0	0	15	20	50

3. Principle of Prediction-based Dynamic Energy Management

3.1. Potential Awakening Mechanisms

According to the mentioned energy consumption model, several potential awakening mechanisms can be implemented on each wireless sensor node:

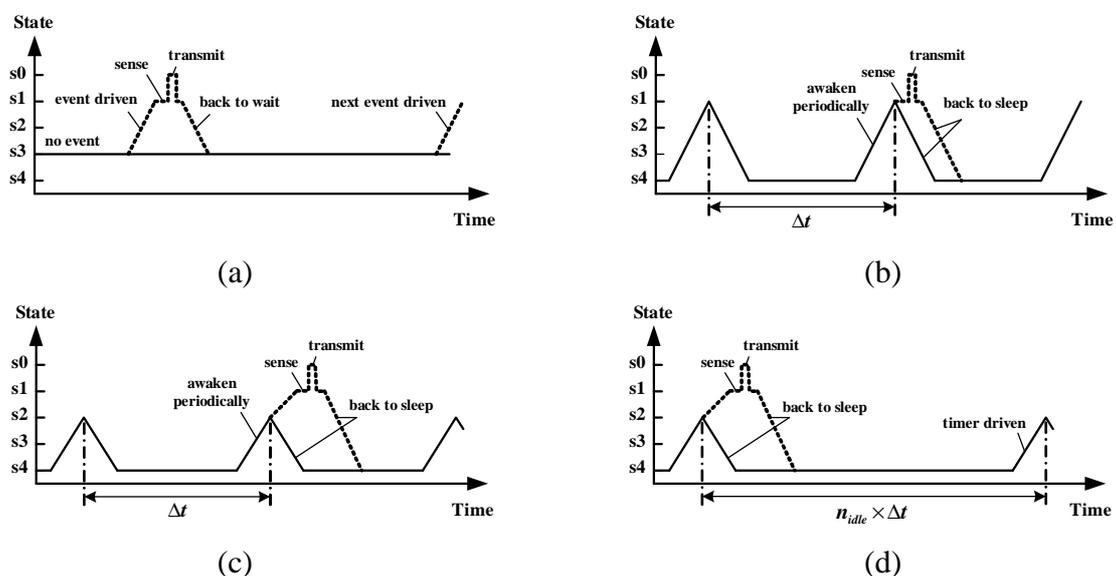
(1) Event-driven on s_3 : As shown in Figure 1(a), when there is no target in the sensing range, the node keeps its state on s_3 . Once any target moves into the range, the sensing module will generate an interrupt to awaken the node to state s_1 . Node will go back to state s_3 after sensing and transmitting.

(2) Awakening periodically to s_1 : The node can go to the deepest sleep state s_4 and is periodically awakened to state s_1 by the timer, as shown in Figure 1(b). If there is any target inside the sensing range, the node will complete measurement and communication, then go back to state s_4 . Otherwise, the node will directly go back to state s_4 .

(3) Awakening periodically to s_2 : This works similarly to the approach of awakening periodically to state s_1 , but the state to which node is awakened periodically is s_2 . Figure 1(c) shows that the node can respond to incoming messages, and expect the ones which have tasks in that period. All the nodes are set back to state s_4 by the announcement of the sink node.

(4) Prediction-based dynamic awakening: As shown in Figure 1(d), we present a dynamic awakening mechanism which adopts the approach of awakening to state s_2 and take node idle time into account as well. The PF algorithm to be introduced will perform the prediction of target state. Each node use predicted target state to estimate idle time so that it can keep state s_4 in as many (n_{idle}) periods as possible and minimize energy consumption.

Figure 1. Comparison of awakening mechanisms: (a) Event-driven on s_3 ; (b) Awakening periodically to s_1 ; (c) Awakening periodically to s_2 ; (d) Prediction-based dynamic awakening.



3.2. Target prediction by Particle Filter

PF is a sequential importance sampling method which is based on Monte Carlo simulation and Bayesian sampling estimation theories and evolved from the Bootstrap nonlinear filtering algorithm [11]. Here, we use PF to draw prior information about the target state, which means the property of target for measurement. For example, we usually take position information as the target state in target tracking applications. With the time index k , we define variable x_k to describe the target state and accordingly the variable y_k can be obtained after observation.

In Bayesian sampling estimation theory, the posterior density $p(x_k | y_{1:k})$ can be inferred from the prior density $p(x_k | y_{1:k-1})$

$$p(x_k | y_{1:k}) = \frac{p(y_k | x_k) p(x_k | y_{1:k-1})}{p(y_k | y_{1:k-1})} \quad (10)$$

where $p(y_k | y_{1:k-1}) = \int p(y_k | x_k) p(x_k | y_{1:k-1}) dx_k$, $p(x_k | y_{1:k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | y_{1:k-1}) dx_{k-1}$.

PF then uses the Monte Carlo simulation method to approximate the posterior density by N particles with the associated weight

$$p(x_{k-1} | y_{1:k-1}) \approx \sum_{i=1}^N \omega_{k-1}^i \delta(x_{k-1} - x_{k-1}^i) \quad (11)$$

For solving the difficulty of sample from the posterior density function, the sequential importance sampling method is used, which samples from a known, easy-to-sample, proposal distribution $q(x_{0:k} | y_{1:k})$, where $x_{0:k}$ is the historical state variables and $y_{1:k}$ is the corresponding observation. The recursive estimate for the importance weights of particle i can then be derived as follows:

$$\omega_k^i = \omega_{k-1}^i \frac{p(y_k | x_k) p(x_k | x_{k-1})}{q(x_k | x_{0:k-1}, y_{1:k})} \quad (12)$$

And then the estimated target state can be approximated by $\hat{x}_k \approx \sum_{i=1}^N \omega_k^i x_k^i$.

3.3. Energy optimization by Distributed Genetic Algorithm and Simulated Annealing

With a predicted target state and awakening mechanism, we can assume that there are N_{sen} candidate nodes for sensing in certain sensing period. As wireless sensor node transfer data to achieve collaborative sensing, we can design a highly efficient and low complexity algorithm to optimize communication energy consumption.

To make use of the distributed computing capability of WSNs, we present a distributed hybrid algorithm. This algorithm is implemented on a number of nodes which work together to enhance optimization of performance. Meanwhile, the advantages of two kinds of optimization method, GA and SA, are combined to get a global optimal solution.

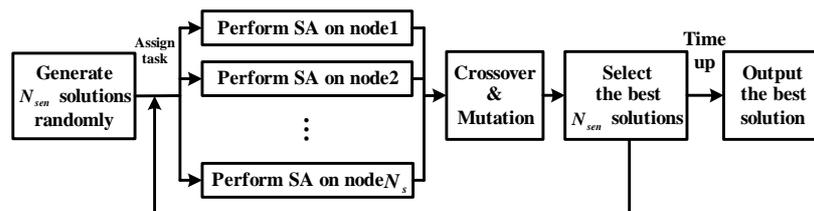
Given an optimization problem, GA encodes the parameters concerned into finite bit strings, each of which presents a possible solution to the problem, and then works with a set of strings, called the population, using reproduction, crossover and mutation operators in a random way but based on the iterative evolution of the fitness function. On the other hand, SA start at an initial random solution state, and a sequence of iterations is generated. A perturbation mechanism is applied, which transforms the

current state into a next state selected from the neighborhood of the current state. If this neighboring state has a lower cost, the neighboring state is accepted as the current state. If this neighboring state has a higher cost, the neighboring state is accepted with a certain probability determined by the acceptance criterion.

GA can retain useful redundant information about what it has learned from previous searches by its representation in individual solutions in the population. Critical components of past good solutions can be captured, and can be combined together via crossover to form high quality solutions. SA retains only one solution in the space and exploration is limited to the immediate neighborhood. However, SA possesses a formal proof of convergence to the global optimal, which GA does not [12]. Therefore, to combine the two methods offers good potential for obtaining an optimum solution.

Figure 2 presents the framework of DGASA. Assigned with the population of solutions from GA, each available node in current period runs SA for a specified time. Then solutions of SA will perform crossover and modulation to generate a new population of GA. An optimal solution for energy consumption is obtained by iteration.

Figure 2. Flow chart of distributed genetic algorithm and simulated annealing.



4. Prediction-based Dynamic Energy Management in Target Tracking

4.1. Target Model

In target tracking applications, we discuss a vehicle target which moves randomly in a two-dimensional sensing field with a maximum speed v_{max} and a maximum acceleration a_{max} [13]. For surveillance purposes, reliable detection should be provided and the position information of target should be reported according to a specified sensing period.

To improve the tracking performance of the maneuvering target, the acceleration inputs must be considered in the system model, so we use the motion of target model which was constructed by Duh [14]. It can compensate the maneuver bias directly and it does not have to assume any *a priori* knowledge of the maneuver target. This process model is given by:

$$X(k+1) = FX(k) + G_1U(k) + G_2V(k) \quad (13)$$

$$E\{V(k)V(j)^T\} = Q(k)\delta_{kj} \quad \delta_{kj} = \begin{cases} 1 & k = j \\ 0 & k \neq j \end{cases} \quad (14)$$

The observation model is:

$$Z(k+1) = HX(k+1) + W(k+1) \quad (15)$$

$$E\{w(k)w(j)^T\} = R(k)\delta_{kj} \quad (16)$$

where k is time index, $X(k) = [x(k), \dot{x}(k), y(k), \dot{y}(k)]^T$ is the state vector representing the relative positions and velocities of the target in the two-dimension plane, $Z(k)$ is the radar measurement vector, $U(k) = [u_x(k), u_y(k)]^T$ is the input vector consisting of the acceleration components in the x and y directions and $V(k)$ and $W(k)$ are the process noise and measurement noise; both sequences are assumed to be uncorrelated white Gaussian noise sequence with zero means and the variance matrices $Q(k)$ and $R(k)$, respectively.

In Equations (13) and (14), F is the model state transition matrix, $G_1(k)$ is the coupling matrix for maneuver inputs, $G_2(k)$ is the process noise input matrix and H is the model output matrix. The related matrices are given by:

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad G_1 = G_2 = \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where T is the sampling time interval. With this target model, the prediction result of PF is the target position at the next sensing instant.

4.2. Sleep Scheduling

When there is no target in the region of WSN, the system is on standby in case of any target gets into it. Any wireless sensor node could be sent to the deepest sleep state. However, it should be awakened on time when target comes around. Then we schedule the sleep time as the following two phases:

(1) Setup phase: Initializing the network, we assume that no target is in the region. The minimum time for a target getting into the sensing range of each node can be estimated according to the shortest distance to the WSN boundary d_{min} and the maximum velocity of target v_{max} :

$$t_{min} = (d_{min} - r_{sensing}) / v_{max} \quad (17)$$

For the possible target position at the next sensing instant, the neighboring nodes should be sent to state s_2 so they became sensing candidates and get ready to be awakened. Accordingly, the idle time is calculated as:

$$t_{idle} = t_{min} - \Delta t \quad (18)$$

Then the number of idle periods is:

$$n_{idle} = \begin{cases} \text{floor}(t_{idle} / \Delta t) & t_{idle} \geq \Delta t \\ 1 & t_{idle} < \Delta t \end{cases} \quad (19)$$

where *floor* is rounding function.

Thus, wireless sensor nodes that are close to the WSN boundary will be awakened periodically. We define these nodes as boundary nodes, and the others are defined as inner nodes.

(2) Tracking phase: Once a target enters the region, sink node announces the prediction of target current position (x_{target}, y_{target}) over the network. All the nodes in state s_2 will receive the message and estimate the idle time according to its coordinates (x_i, y_i) :

$$t'_{idle} = (d_{target,i} - r_{sensing}) / v_{max} - \Delta t \quad (20)$$

For any new target that may possibly enter the region, each node should still be ready for sensing after t_{idle} , so in this case the number of idle periods is calculated as:

$$n_{idle}' = \begin{cases} n_{idle} & t_{idle}' \geq t_{idle} \\ \text{floor}(t_{idle}' / \Delta t) & t_{idle}' < t_{idle} \end{cases} \quad (21)$$

Respectively, n_{idle} and n_{idle}' are used to schedule the state transition of nodes in these two phases. The sleep time of nodes is prolonged as much as possible without missing any events.

4.3. Communication Scheduling

Let us assume that a group of wireless sensor nodes in the neighborhood of the target are supposed to gather the acquired data and transmit it to a sink node during each sensing period. Each node in the sensing group signals with the same transmission power, then the sink node can obtain the sequence of the distance to each node according to the received signal power. As shown in Figure 3(a), the packet is forwarded from far to near. Each node combines its data into the packet, and finally the nearest node sends the packet to the sink node.

Let the sink node be denoted by $Node_0$. The set of nodes from near to far is denoted by $\{Node_i | i = 1, 2, \dots, m\}$, so the energy consumption metric can be calculated according to Equation (8):

$$E = [m\alpha_1 + \sum_{i=0}^{m-1} \alpha_2 d_{i,(i+1)}^{n_0}] \square P \quad (22)$$

where d_{ij} is the distance between $Node_i$ and $Node_j$, and P is the packet size.

However, the neighborhood of the target may be far away from the sink node, so it would require a large amount of energy, even for the nearest node transmitting the packet. Thus, we select a number of nodes randomly from the inner nodes for packet forwarding. These nodes are defined as forwarding nodes. And the selection criterion is: each inner node selects a random number in $[0,1]$; if it exceeds a threshold T , then it is selected to be forwarding node. The threshold T is defined as in [15]:

$$T = \frac{P}{1 - p[r \bmod (1/p)]} \quad (23)$$

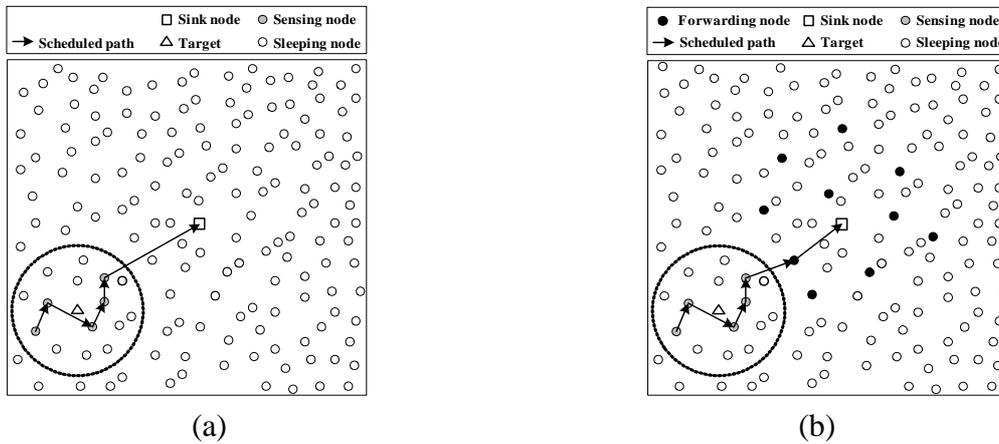
where p is the percentage of forwarding nodes, r is the current period number, and \bmod is the complementation operator. Each inner node should decide whether it is forwarding node in every period during its sleep time and it should wake up to state s_2 on schedule as a forwarding node.

As shown in Figure 3(b), if the nearest node in the sensing group chooses a shortest path adopting one of the forwarding nodes, then the energy consumption metric can be calculated as:

$$E' = [m\alpha_1 + \sum_{i=1}^{m-1} \alpha_2 d_{i,(i+1)}^{n_0}] \square P + \min_{i=p}^q [\alpha_2 (d_{i,1}^{n_0} + d_{i,0}^{n_0}) + \alpha_1 \alpha_2 d_{0,1}^{n_0}] + \Delta E \quad (24)$$

where ΔE denotes the extra energy consumption of the forwarding node.

Figure 3. Routing schemes for target tracking application: (a) Without forwarding nodes; (b) With forwarding nodes.



4.4. Node selection optimization

In this case, the optimization problem is selecting a group of nodes from the candidate sensing nodes in order to minimize communication energy consumption. Let us assume that there are N candidate nodes for selection. The coding scheme for each possible solution of DGASA is described in Figure 4. The i -th code of the solution indicates whether the i -th candidate node is selected for sensing, where “1” denotes node is selected while “0” denotes node is not selected. For example, Figure 4 shows that node i and j are selected for sensing.

Figure 4. Coding scheme for distributed genetic algorithm and simulated annealing.

Node	1	2	...	i	...	j	...	$N-1$	N
Solution	0	0	...	1	...	1	...	0	0

Taking the semimajor axis of the error ellipse as the metric of collaborative sensing error, we assume that sensing error should be less than A_0 in a target tracking application. Then the fitness function of each possible solution is:

$$Fitness = \begin{cases} A + E_0 & A > A_0 \\ A_0 + E(E') & A \leq A_0 \end{cases} \quad (25)$$

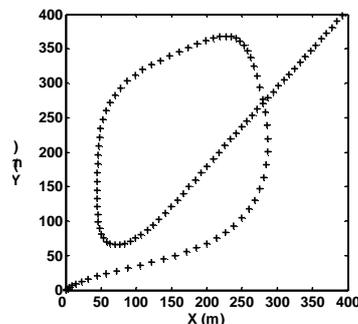
where $A = \sigma_l$, defined as the collaborative sensing error with the possible solution, and E_0 is a constant which is larger than the upper bound of energy consumption metric $E(E')$. Minimizing the fitness function, the sensing accuracy is optimized first. Once accuracy of application is satisfied, the energy consumption metric is optimized for energy efficiency.

5. Simulation Experiments

5.1. Simulation environment

Assume that the sensing field of WSN is 400 m x 400 m, in which there are 256 wireless sensor nodes equipped by peroelectric infra-red (PIR) sensors with sensing range $r_{sensing} = 60$ m and root mean square (RMS) of DF error variance $\sigma_{\theta} = 2^0$. We set $\alpha_1 = 50nJ/b$ \square $\alpha_2 = 100pJ/b/m^2$ and $n_0 = 3$ in the energy consumption model. The sensing period is set as 0.5 s, and then the sample period of PF is 0.5 s accordingly. The performing time for node selection optimization is set as 0.15s for all the following simulations. According to Section 4.1, we generate a trajectory of 120 points as shown in Figure 5, in some part of which the target moves on its maximum acceleration a_{max} and maximum velocity v_{max} for generalization. Here, $a_{max} = 10m/s^2$, $v_{max} = 40m/s$, and sensing error threshold $A_0 = 0.6$ m.

Figure 5. Target trajectory in the sensing field of WSN.

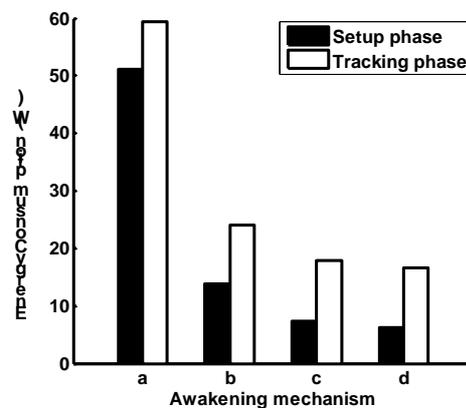


Simulations are performed using Opnet Modeler, which is a simulation platform for communication networks and distribution systems. The network models are established according to the proposed energy management mechanism. We employed CSMA/CA as the MAC protocol. The wireless channel model was bpsk, data rate was 1 Mbps and simulation duration was 60 seconds.

5.2. Network simulations of dynamic energy management mechanism

First, we run simulation of WSN to analyze the energy conserved by prediction-based dynamic awakening mechanism separately. All the wireless sensor nodes available are used for detection in the target vicinity. The awakening mechanisms are compared in Figure 6: (a) Event-driven based on s_3 ; (b) Awakening periodically to s_1 ; (c) Awakening periodically to s_2 ; (d) Prediction-based dynamic awakening. We find that prediction-based dynamic awakening mechanism obtains the lowest energy consumption in both setup and tracking phases.

Figure 6. Comparison of energy consumption in setup and tracking phase with different awakening mechanisms.



Prediction accuracy of PF is discussed and probability density of position prediction error on X-axis and Y-axis is shown in Figure 6. Moreover, the performance of the proposed algorithm for node selection optimization was evaluated. We analyzed the sensing accuracy and energy consumption during the tracking phase, with selected sensor node detecting and reporting. In each sensing period, PF was employed to predict the target position at the next sensing instant, with which DGASA is implemented to select the optimal wireless sensor node in the vicinity for sensing in the next period. GA and SA were also studied for comparison with the same coding scheme.

To prevent the influence of a target entering and leaving the sensing field, we studied the procedure from 10 s to 50 s. Figure 8 shows the collaborative sensing error and energy consumption of GA, SA and DGASA optimization. In Figure 8(a), the collaborative sensing error of GA and DGASA optimization can satisfy the required sensing accuracy, while part of SA optimization results exceed the required sensing error. As shown in Figure 8(b), energy consumption is compared during tracking, and we can find that DGASA optimization gives the lowest energy consumption, while energy consumption of GA optimization is highest.

Figure 7. Probability density of position prediction error on X-axis and Y-axis.

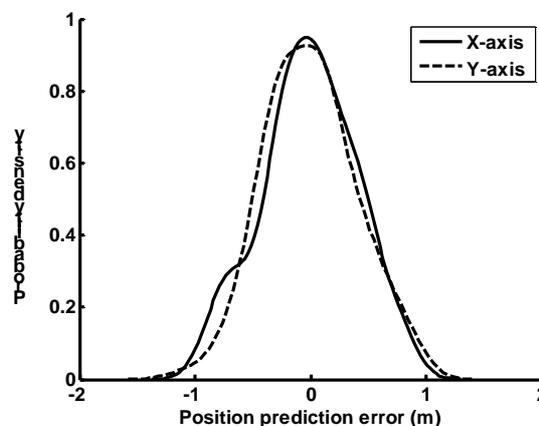
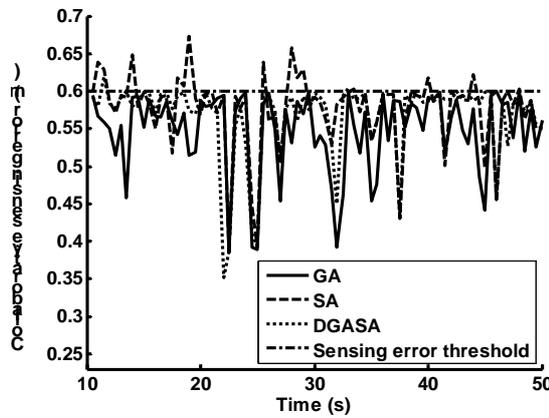
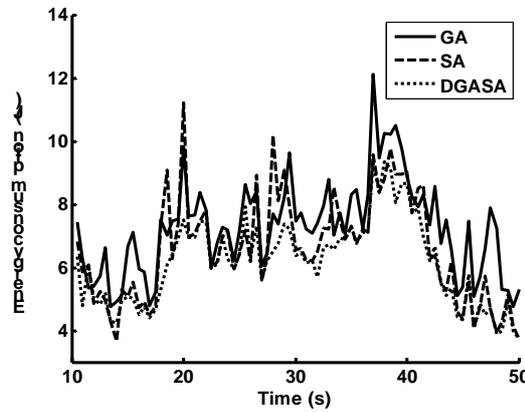


Figure 8. GA, SA and DGASA optimization results: (a) Collaborative sensing error; (b) Energy consumption.



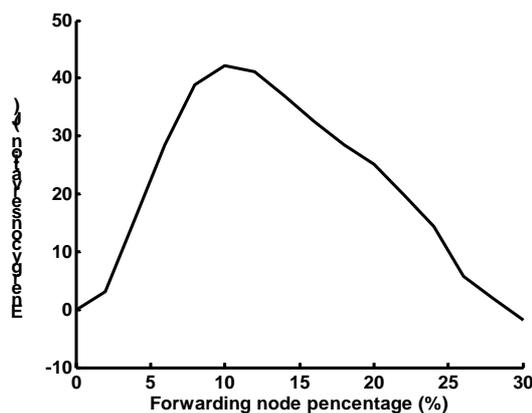
(a)



(b)

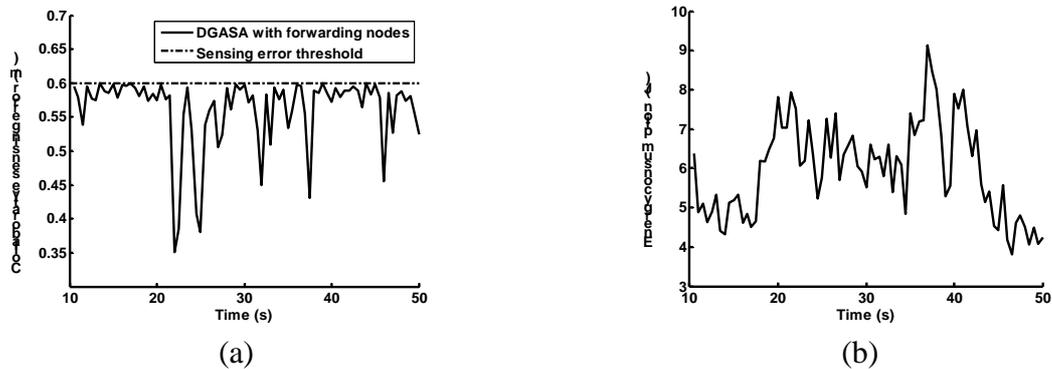
A set of network simulations was also performed in the case that forwarding nodes are utilized in communication. The effect of the forwarding node percentage (changing from 0% to 30%) on energy conservation of DGASA optimization was examined, as shown in Figure 8. It can be seen that we have the most energy conservation with this percentage set at 10%.

Figure 9. Affect of forwarding node percentage on energy conservation of DGASA optimization



When the percentage of forwarding nodes is 10%, the collaborative sensing error of DGASA optimization is shown in Figure 10(a), which can satisfy the required sensing accuracy. Energy consumption of WSN during the tracking is presented in Figure 10(b).

Figure 10. DGASA optimization results with forwarding node percentage set as 10%: (a) Collaborative sensing error; (b) Energy consumption.



Finally, we analyzed energy consumption and conservation ratio of dynamic energy management approaches: dynamic awakening mechanism, node selection optimization and forwarding node routing. Compared to the case that each node keeps in fully active state, total energy conservation of 90.9% can be achieved with all these approaches.

Table 2. Energy consumption and conservation of dynamic energy management approaches.

Dynamic awakening mechanism	Node selection optimization	Forwarding node routing	Energy consumption (J)	Conservation
No	No	No	5020	-
Yes	No	No	668	86.7%
Yes	Yes	No	497	90.1%
Yes	Yes	Yes	455	90.9%

6. Conclusions

Focusing on the energy-efficiency problem in WSNs, we have proposed a dynamic energy management mechanism based on target prediction. Collaborative sensing model and energy consumption model is established, and PF is applied to predict the target state. With the state of the next sensing instant, each wireless sensor node can update its idle time and schedule its sleep without loss of any event. Meanwhile, as the candidate nodes for sensing are known beforehand, we accomplished the optimization of the sensing process with DGASA, which uses the distributed computing capability of WSNs so that energy consumption can be minimized without degrading the performance accuracy. Moreover, we have discussed the routing scheme for data reporting, and an approach utilizing forwarding nodes is presented to conserve more communication energy. Simulations of target tracking demonstrate that PF target prediction-based dynamic awakening mechanism and

DGASA energy management enhance energy efficiency significantly in WSNs target tracking application. The paper gives an adaptive dynamic energy management framework for target monitoring applications in WSNs and has presented details for solving the target tracking problem. The feasibility of prediction-based dynamic energy management mechanisms should be explored on a more practical platform.

Acknowledgements

This work was supported by the National Grand Fundamental Research 973 Program of China under Grant No.2006CB303000 and the National Natural Science Foundation of China (No.60673176; No.60373014; No.50175056).

References and Notes

1. Sinha, A.; Chandrakasan, A. Dynamic power management in wireless sensor networks. *IEEE Des. Test Comput.* **2001**, *18*, 62-74.
2. Chiasserini, C.F.; Rao, R.R. Improving energy saving in wireless systems by using dynamic power management. *IEEE Trans. Wireless Commun.* **2003**, *2*, 1090-1100.
3. Gustafsson, F.; Gunnarsson, F. Particle filters for positioning, navigation and tracking. *IEEE Trans. Signal Process* **2002**, *50*, 425-437.
4. Yu, Y.; Cheng, Q. Particle filters for maneuvering target tracking problem. *Sig. Process.* **2006**, *80*, 195-203.
5. Wang, X.; Jiang, A.; Wang, S. Mobile agent based wireless sensor network for intelligent maintenance. *Lect. Notes Comput. Sci.* **2005**, *3645*, 316-325.
6. Wang, X.; Wang, S.; Ma, J. Dynamic deployment optimization in wireless sensor networks. *Lect. Notes Contr. Inf. Sci.* **2006**, *344*, 182-187.
7. Zhao, F.; Shin, J. Information-driven dynamic sensor collaboration for tracking applications. *IEEE Sig. Process. Mag.* **2002**, *19*, 61-72.
8. Oshman, Y.; Davidson, P. Optimization of observer trajectories for bearings-only target localization. *IEEE Trans. Aero. Elec. Sys.* **1999**, *35*, 892-902.
9. Paradowski, L.R. Uncertainty ellipses and their Application to interval estimation of emitter position. *IEEE Trans. Aero. Elec. Sys.* **1997**, *33*, 126-133.
10. Chhetri, A.S.; Morrell, D. Energy efficient target tracking in a sensor network using non-myopic sensor scheduling. *Proceedings of 8th International Conference on Information Fusion*, **2005**, 558-565.
11. Wang, X.; Wang, S.; Ma, J. An improved particle filter for target tracking in sensor system. *Sensors* **2007**, *7*, 144-156.
12. Calaor, A.E.; Hermosilla, A.Y. Parallel hybrid adventures with simulated annealing and genetic algorithms. *Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks* **2002**, 33-38.

13. Yang, L.; Feng, C. Adaptive tracking in distributed wireless sensor networks. *Proceedings of 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems* **2006**, 103-111.
14. Duh, F.B.; Lin, C.T. Tracking a maneuvering target using neural fuzzy network. *IEEE Trans. Syst. Man Cy.* **2004**, *34*, 16-33.
15. Heinzelman, W.R.; Chandrakasan A. Energy-efficient communication protocol for wireless microsensor networks. *Proceedings of Hawaii International Conference on System Sciences*, **2000**, 1-10.

© 2007 by MDPI (<http://www.mdpi.org>). Reproduction is permitted for noncommercial purposes.