An Energy-Efficient Collaborative Target Tracking Framework in Distributed Wireless Sensor Networks

Lin Shang, Kang Zhao, Zhengguo Cai, Dan Gao, and Maolin Hu
Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China

Correspondence should be addressed to Lin Shang; shanglin.0921@gmail.com
Received 10 December 2013; Revised 25 May 2014; Accepted 28 May 2014; Published 8 July 2014

Energy consumption and tracking accuracy are two significant issues for collaborative tracking in distributed wireless sensor networks (DWSNs). To obtain a benefit from those issues, most of the recent work tends to reduce the spatial redundancy, while ignoring utilizing the attribute of time redundancy. In this paper, a novel energy-efficient framework of collaborative signal and information fusion is proposed for acoustic target tracking. The proposed fusion algorithm is based on neural network aggregation model and Gaussian particle filtering (GPF) estimation. And the neural network based aggregation (NNBA) can reduce spatial and time redundancy. Furthermore, a fresh cluster head (CH) selection method demanding less task handover is also presented to decrease energy consumption. The analyzed framework coupled with simulations demonstrates its excellent performance in tracking accuracy and energy consumption.

1. Introduction
With the fast development of microelectromechanical system (MEMS), digital signal processing, and embedded computing, sensor nodes are becoming smaller and cheaper. Therefore, it is possible to deploy large-scale distributed wireless sensor networks (DWSNs) to “achieve quality through quantity” in many complex applications [1, 2].

However, challenges and difficulties also exist in target tracking DWSNs [3–5]: (1) sensor resources limitation [6] (especially communication bandwidth, energy, sensing range, and processing capabilities); (2) deployment and coordination of a mountain of sensors; (3) redundant data caused by similar measurements of adjacent sensors. Accomplishing the tracking task requires not only the methods for data fusion and collaborative signal and information processing (CSIP), but also an efficient scheme of energy consumption. In the literature, various researches have been proposed for target tracking in DWSNs [7].

In [8, 9], the nodes detect the mobile target and send measurements directly to the sink node so that it can estimate the target location using all data. But the centralized schemes, regarded as the first CSIP (CSIP1), depend on the sink node to collect and process measurements, as the sink node will consume a lot of energy and be apt to die quickly with constraint battery; the network lifetime is consequently shortened. Energy aware probabilistic target localization algorithm for cluster-based WSN is proposed in [10]. Sensors detecting a target report a short message to the cluster head (CH). Then the CH localizes and determines the subset of sensors in the vicinity of target. This scheme can reduce communication bandwidth and save lots of energy. Nevertheless, the CH undertakes a heavy tracking task, which shortens its lifetime. The work in [11] proposed an adaptive dynamic cluster-based tracking (ADCT) protocol in which CHs are selected dynamically; meanwhile sensor nodes surrounding the target are waken up to construct a cluster with the target’s moving in the network. Such scheme can be called the second CSIP (CSIP2). However, the election of the first CH is achieved by competition, which causes an additional burden in communication bandwidth.

A distributed predictive tracking (DPT) protocol for tracking moving objects with separate algorithms for nodes and CHs is presented in [12]. The CH adopts the target descriptor to identify target and predicts its next location. Then it informs predicted area CH who activates $k$ sensors
to sense the target. But most of the efforts have been spent on selecting the node nearest to the predicted target position for the CH. This scheme can be considered as the third CSIP (CSI3).

Moreover, the work in [13] showed three algorithms for tracking a mobile target in WSNs utilizing cluster-based architecture, namely, adaptive head, static head, and selective static head. In [14], an intensive review was conducted to improve energy efficiency in collaborative target tracking in DWSNs.

Previous work was more focused on the spatial redundancy of the distributed sensor network for target tracking. However, the study on the time redundancy is still important. The purpose of this paper is to design a novel energy-efficient collaborative signal and information fusion processing (CSIFP) framework for acoustic target tracking in DWSNs, to collaboratively estimate the position and velocity of a moving target. In our framework, sensor nodes detecting a target are activated and dynamically form a cluster around the target. Each member node transfers the deredundant data to the CH. CH aggregates and processes these messages to estimate the current state of the target and reports the result to the sink. The CH is reselected relying on certain rule in Sections 4.1 and 4.2. To estimate the target position, the Gaussian particle filter (GPF) algorithm [15] is adopted. Compared with the particle filter (PF) [16–18], it omits resampling to reduce the computational complexity and avoid transmitting all particles and weights when task handover happens. Furthermore, the neural network based aggregation (NNBA) model relies on the cooperation of the nodes, which can reduce the spatial and time redundancy.

The paper is organized as follows. Section 2 presents the system assumptions and the system models. Section 3 proposes the framework of the distributed collaborative target tracking framework is illustrated. Section 4 discusses a neuron network based aggregation model for target tracking. Section 5 describes the simulation environment and Section 6 shows and analyzes the results. Finally, Section 7 summarizes the work and gives a conclusion.

2. Problem Statements

2.1. Assumptions. First, several assumptions about the sensor nodes and the underlying networks are defined:

(i) All sensor nodes are assumed to be homogeneous and synchronized. Cluster-based structure is used for the nodes’ deployment. The CH is responsible for task decision, data fusion, and routing tracking results to the sink node.

(ii) Each node knows its own position after initial deployment.

(iii) The maximum communication range of each sensor node is twice larger than the maximum sensing range; that is $R_c \geq 2R_s$. This guarantees that all activated nodes can communicate with CH.

(iv) All sensor nodes are initialized with the same fixed amount of battery energy.

(v) As the paper focuses on the energy consumption of target tracking in DWSNs, the energy consumed by sensor nodes in the sleeping state is not considered. However, it does not mean that the energy consumption of idle sensor nodes can always be ignored.

2.2. Target Dynamic Model and Measurement Model. Target tracking aims to estimate the real-time state of a moving object, which is mainly based on the target dynamic models. The appropriate models can not only describe the evolution of the target state with respect to time, but also represent a large number of observations, reducing the data transmission in the DWSNs [19].

The target state is denoted by a four-dimensional vector in 2D Cartesian coordinate plane:

$$x_k = \begin{bmatrix} \xi_k \\ \eta_k \\ \xi_{k}' \\ \eta_{k}' \end{bmatrix}^T,$$  

(1)

where $\xi_k$ and $\eta_k$ denote the target position in $x$-axis and $y$-axis at time $t_k$ and $\xi_{k}'$ and $\eta_{k}'$ denote the corresponding velocities.

We describe the target dynamic model with additive noise as the state-space model in the following form:

$$x_k = f_k(x_{k-1}, u_{k-1}) + w_k, \quad z_k = h_k(x_k) + v_k,$$  

(2)

(3)

where $x_k$, $z_k$, and $u_k$ are the target state, observation, and control input vectors, respectively, at the discrete time $t_k$. $w_k$ is the system noise at $t_k$, and $v_k$ is the observation noise, assumed to be statistically independent from $w_k$. $f_k$ and $h_k$ are state transition and observation functions, which are vector valued (possibly time-varying).

For nearly constant velocity (CV) or white noise acceleration model [19], $u_k$ is approximated to zero, and (2) can be rewritten by

$$x_k = F_k x_{k-1} + w_k,$$  

(4)

where

$$F_k = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T^2 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$  

(5)

Note that $T$ is the sampling period.

2.3. Measurement Model [20–22]. Energy based acoustic sensors are widely used in DWSNs due to low cost and universality. The intensity of the source attenuates at a rate that is inversely proportional to the distance from the source. Given $N_s$ sensors, the signal intensity measured on the $n$th sensor is given by

$$z_{k,n} = h_k(x_k) + v_{k,n} = g_n \cdot \frac{s}{\| r_t - r_n \|^2} + v_{k,n},$$  

(6)

where $s$ is the strength of the acoustic emitted by the target during $T$. The vectors $r_t$ and $r_n$ denote the coordinates of the
sensing distance \( R \) corresponding to the calculated strength at the maximum acoustic intensity measurement exceeds a threshold, the acoustic energy sensor node, it can detect a target once the \( V \) is approximately 2 and this value is assumed for this paper. 

### 3. Distributed Collaborative Target Tracking Framework

In this paper, we develop an energy-efficient distributed collaborative target tracking framework. 

This effective framework involves four modules: the tracking task initialization, CH election, node selection, and tracking task handover. Before specifying the aforementioned contents, several messages are listed in Table 1.

#### 3.1. Tracking Task Initialization

Assume that a network with \( N_i \) sensors is deployed on a 2D area. For a given acoustic energy sensor node, it can detect a target once the acoustic intensity measurement exceeds a threshold, \( Z_i \) th, corresponding to the calculated strength at the maximum sensing distance \( R_i \). Sensor nodes detecting the target are activated, whereas the other nodes remain in the sleeping state for saving energy. Considering the influence of the measurement noise, we define a set of candidate nodes whose measurements of acoustic intensity are not lower than \( Z_i \) th/\( \varepsilon \) and remaining energy is greater than zero. Only the candidate nodes could participate in the target tracking process and transmit the measurements to the CH. The candidate nodes set \( S \) is denoted as follows: 

\[
S_k = \left\{ j \mid z_{k,j} \geq \frac{Z_{i,\text{th}}}{\varepsilon} \cdot E_j > 0 \right\}, \tag{7}
\]

where \( z_{k,j} \) is the measurement of sensor node \( j \) at time \( t_k \) and the energy threshold \( Z_{i,\text{th}} \) is \( g \cdot (s/R_i^2) \). Parameter \( \varepsilon \in (0, 1] \) is customized, and a bigger \( \varepsilon \) means more candidate nodes will be chosen. \( E_j \) is the remaining energy of the node \( j \), and \( R_i \) is the communication radius.

At initial time phase as well as the target detection restart phase, each candidate sensor node starts a timer \( T_C \) and will broadcast CHInfoMesg after the timer expires. The length of \( T_C \) is 

\[
T_C = \frac{d \cdot s}{z_{k,j}}. \tag{8}
\]

According to (6), the sensor with the largest measurement is nearest to the target; thus, \( T_C \) is the smallest. So the sensor node nearest to the target will first send its CHInfoMesg. The other nodes which have received this message register the broadcaster as their CH, terminate the timer, and give up broadcasting message. Through the above process, the candidate closest to the target is elected to be the CH. The cluster member nodes transmit their observation to CH, and the CH aggregates the collected information and performs the tracking task using GPF.

#### 3.2. CH Selection

It is expectable that CHs will consume much more energy than others because they undertake more burdensome tasks. For balancing the energy consumption, each node needs to take turn as CH by certain scheduling algorithm. However, CH rotation process consumes a lot of energy for transmitting message between CHs. Therefore, we propose a novel CH selection scheme considering these two factors.

At time \( t_{k-1} \), it is assumed that the CH is \( CH_{k-1} \). According to the predicted target position at time \( t_k \) and cluster members information, \( CH_{k-1} \) builds a candidate CH set \( C_k \):

\[
C_k = \left\{ j \mid \text{dis} (j, x_k) \leq \rho \cdot R_i, E_j \geq E_{\text{th}} \right\}, \tag{9}
\]

where \( \rho \in [0, 1] \) is customized and a smaller \( \rho \) means CH changes more frequently, \( E_j \) is the remaining energy of node \( j \), and \( E_{\text{th}} \) is the threshold energy. If \( E_j \) is lower than \( E_{\text{th}} \), a node is not suitable for being a cluster head.

The current CH checks itself whether it belongs to the candidate set \( C_k \) or not. If so, it will continue taking charge of the tracking task; otherwise, it will hand over the CHHandoverMesg including tracking information and necessary parameters to \( CH_k \). In order to select an appropriate new CH, we define the attraction force \( F_i \) of the node \( j \), exerting on the current CH. The larger value means the node \( j \) has greater probability to be chosen as next CH:

\[
F_i = \alpha \cdot F_{\text{dir},i} + \beta \cdot F_{\text{dis},i} + \gamma \cdot F_{\text{power},i} \tag{10}
\]

Note that \( F_{\text{dir},i} \), \( F_{\text{dis},i} \), and \( F_{\text{power},i} \) are the direction attraction component force, distance attraction component force, and power attraction component force exerting on \( CH_{k-1} \) by node \( j \) respectively. \( \alpha \), \( \beta \), and \( \gamma \) are three nonnegative constants which adjust the ratios of the above three component forces, and \( \alpha + \beta + \gamma = 1 \). Consider the following:

(i) Direction attraction component force \( F_{\text{dir},i} \): the direction from the current cluster head \( CH_{k-1} \) to node

| Table 1: Messages used in the target tracking framework. |
|-----------------------------------------------|-----------------------------|
| Message type                   | Purpose |
| CHInfoMesg                     | CH informs its member nodes to join this cluster |
| CHHandoverMesg                 | CH transmits the parameters to the new CH when task handover happens |
| DataMesg                       | Nodes send their measurements to CH |
| DataDegrangeMesg               | DataDegrangeMesg transmits the target position estimation to the sink |
| TargetToSinkMesg               | CH transmits the target position estimation to the sink |

International Journal of Distributed Sensor Networks
and the direction from CH$_{k-1}$ to the predicted target position $\hat{x}_k$ are more closer, which means that the node $j$ is nearly on the direction of the target motion and has a high probability to be the cluster head at time step $t_k$. We define the cosine value of this angle to indicate the force $F_{dir,j}$:

$$
F_{dir,j} = \left\{ \begin{array}{ll}
\frac{\langle r_{CH_{k-1}}, r_j \rangle}{\|r_{CH_{k-1}}\| \cdot \|r_j\|}, & \text{if } j \in S_{k-1}, \\
0, & \text{else},
\end{array} \right.
$$

where $r_{CH_{k-1}}$ is a vector from CH$_{k-1}$ to the predicted target position $\hat{x}_k$ and $r_j$ is vector from the current cluster head CH$_{k-1}$ to node $j$. $(r_{CH_{k-1}}, r_j)$ is the inner product. $\|r_{CH_{k-1}}\|$ and $\|r_j\|$ represent the length of $r_{CH_{k-1}}$ and $r_j$, respectively.

(ii) Distance attraction component force $F_{dis,j}$: for the sake of reducing the frequency of CH reselection, we choose the candidate node as far as the current CH on the premise of an acceptable tracking accuracy. $F_{dis,j}$ is defined as follows:

$$
F_{dis,j} = \left\{ \begin{array}{ll}
\frac{\text{dis}(j, \text{CH}_{k-1})^2}{R_c^2}, & \text{if } \text{dis}(j, \text{CH}_{k-1}) < R_c, \\
0, & \text{else},
\end{array} \right.
$$

where dis($j, \text{CH}_{k-1}$) is the Euclidian distance between the node $j$ and the current cluster head CH$_{k-1}$ and $R_c$ is the communication radius.

(iii) Power attraction component force $F_{power,j}$: $F_{power,j}$ is defined to indicate the node battery level, which is given by

$$
F_{power,j} = \left\{ \begin{array}{ll}
\frac{E_j}{E_0}, & \text{if } E_j \geq E_{th}, \\
0, & \text{else},
\end{array} \right.
$$

where $E_0$ is the initial energy of the node $j$.

Finally, CH$_k$ is selected according to the following rule:

$$
\text{CH}_k = \left\{ \begin{array}{ll}
\text{CH}_{k-1}, & \text{CH}_{k-1} \in C_k, \\
\{ j \mid \text{dis}(j, \text{CH}_{k-1}) < R_c, \arg \max \left( F_j \right) \}, & \text{otherwise}.
\end{array} \right.
$$

Noting that it is possible for multiple candidate nodes to have the same maximum attraction force, in this case, we will select one node randomly among these nodes as the new CH.

3.3. Member Node Selection. The node selection determines the total energy consumption and data fusion accuracy, which has a significant impact on the performance of the whole sensor network.

To implement an optimal cluster with member nodes, more nodes should be selected to improve the tracking accuracy. On the other hand, fewer nodes should be selected to reduce the energy consumption for transmission. So we need to make a compromise. Considering the influence of measurement noise, we choose the nodes located in the communication range of the current CH from the set of candidate nodes $S_k$. The set of member nodes are denoted by

$$
N_k = \{ j \mid \text{dis}(j, \text{CH}) \leq R_c, j \in S_k \},
$$

where dis($j, \text{CH}$) is the distance between the node $j$ and the position of the current CH at $t_k$. The constraint dis($j, \text{CH}$) $\leq R_c$ is resistant to the measurement noise.

3.4. Tracking Task Handover. Once the target is far away from the current CH and CH$_{k-1}$ $\notin C_k$, CH$_{k-1}$ cannot efficiently manage. CH$_{k-1}$ should hand over the tracking task to a new CH. This operation can not only shut down the old nonsignificant cluster nodes to save energy in time but also improve the tracking accuracy.

The current CH selects a member node from $C_k$ as the new CH, CH$_k$, which is along the direction of estimated target motion as well as far away from the current CH. Then CH$_{k-1}$ sends a CHHandoverMesg message to CH$_k$. This message contains the current target state estimation as well as some necessary parameters. After receiving the CHHandoverMesg from CH$_{k-1}$, CH$_k$ will continue to take charge of the target tracking task and broadcast CHInfoMesg to inform its cluster members.

3.5. Working Scheduling. The distributed collaborative target tracking framework of the CH and member nodes is shown in Figure 1. Consider the following.

(i) The cluster head: CH is generated or reselected according to (14), and it broadcasts CHInfoMesg message to inform its cluster members. CH receives the DataMesg packet or DataDegradeMesg packet from its members, then chooses three messages with the biggest acoustic intensity measurement, and performs the GPF algorithm described in Section 4.2 to estimate the current target state and predict the possible state at next time step. The CH also takes charge of sending the tracking results to the sink node. If the difference between estimated target position at $t_k$ and the one at $t_{k-1}$ is greater than the threshold $\theta_2$, CH uploads TargetToSinkMesg packet containing packet type, CH ID, tracking result; otherwise, CH sends TargetDegradeMesg packet without tracking result. The sink keeps the previous value once receiving TargetDegradeMesg.

(ii) Member node $j$: after receiving CHInfoMesg from the current CH, it checks whether it belongs to the member nodes set $N_k$. If so, the node transfers DataMesg including packet type, ID, and measurement to the CH. Otherwise it goes to sleep state. When the difference of the calculated distance between the node $j$ and the target at $t_k$ and $t_{k-1}$ is smaller than difference threshold $\theta_1$, it sends a DataDegradeMesg, omitting measurement. The CH holds its previous measurement once receiving DataDegradeMesg.
Note that, at any time, only several sensor nodes in the vicinity of the target need to collect the acoustic signal and the others keep sleeping. The CH and member nodes run in the neural network based aggregation tracking model, which will be expatiated in Section 4.

4. Neural Network Based Aggregation Tracking Model

This section details the fusion processing algorithm based on neural network based aggregation tracking model. It combines the neural network of multilayer perceptron with clustering routing structure. This algorithm reduces time redundancy and spatial redundancy to reduce the traffic, save energy, and prolong the network lifetime.

4.1. Collaborative Target Tracking Scheme. Figure 2 gives the diagram of NNBA model structure [23], which uses four-perceptron neural network. For DWSNs, each sensor node works as an underlying neuron; CH is like a central neuron, and the whole network is a complex neuron network system.

Cluster member nodes are in the input layer and the first hidden layer. CH nodes are in the second and the third
hidden layer and output layer. For instance, each cluster with n members collects data of m species by different types. Consequently, this NNBA model has $n \times m$ input layer nodes and the same number of first hidden layer neurons. Here, we only use the acoustic sensor, so $m = 1$. The quantity of neurons in the second hidden layer, the third hidden layer, and the output layer can be adjusted by the users according to the practical application. The hidden layers have no inevitable relationship with n. It only deals with separate data of different types and need not be fully connected between the first and the second hidden layer. So do the second and the third hidden layer. But the treated data between the third hidden layer and output layer can be integrated, for they are full-connected.

According to this NNBA model, firstly sensor nodes collect the acoustic intensity measurements from the target and preprocess them according to the neuron function of the first hidden layer. Then the nodes send the results to the CH. The CH further processes to track the target using GPF algorithm detailed in Section 4.2. GPF is embedded in the second hidden layer, the third hidden layer, and output layer neurons. Finally, the CH transfers the estimated target state to the sink. Note that neural network model can do some modification according to the specific requirements in various applications.

4.2. Gaussian Particle Filter. Recalling the theory of particle filtering [16–18], we estimate the target a posteriori distribution $p(x_{0:k} \mid z_{0:k})$ by a Gaussian pdf, whose mean and covariance are calculated from the weighted particle set. In contrast to the standard PF [16–18], GPF algorithm does not require the process of resampling, which reduces the computational complexity.

Assume that the initial target state is known and its distribution accords with Gaussian distribution $N(x_0; \bar{x}_0, \bar{P}_0)$. The steps of GPF algorithm are described as follows.

(1) Time step $t = 0$;

Initialize the parameter: $\mu_0 = \bar{x}_0, \Sigma_0 = \bar{P}_0$.

(2) Time Update (Prediction)

Time step $t_{k-1}$;

(a) Draw samples from the distribution $p(x_{k-1}) \sim N(x_{k-1}; \mu_{k-1}, \Sigma_{k-1})$, and denote them as $\{x_{i,k-1}\}_{i=1}^N$.

(b) For $i = 1, \ldots, N$, sample from $p(x_k \mid x_{k-1} = x_{i,k-1})$ to obtain predicted particles’ set $\{x_{i,k}^{j,k-1}\}_{j=1}^N$.

(c) Calculate the mean $\bar{x}_{k|k-1}$ and covariance $\bar{P}_{k|k-1}$ as

$$
\bar{x}_{k|k-1} = \frac{1}{N} \sum_{i=1}^N x_{i,k|k-1},
$$

$$
\bar{P}_{k|k-1} = \frac{1}{N} \sum_{i=1}^N \left( x_{i,k|k-1} - \bar{x}_{k|k-1} \right) \left( x_{i,k|k-1} - \bar{x}_{k|k-1} \right)^T.
$$

(3) Measurement Update

Time step $t_k$;
\[
p_1 = f(I, w) = I \ast w = I
\]
\[
ad_1(k) = g_1 p_1, ad_2(k - 1) = \begin{cases} 
  a_1(k) - a_2(k - 1), & a_1(k) < \theta_1 \\
  0, & \text{else} \\
  a_2(k - 1), & a_1(k) \geq \theta_1
\end{cases}
\]
\[
a_2(k) = g_2 a_1(k), a_2(k - 1) = \begin{cases} 
  a_2(k), & a_1(k) < \theta_1 \\
  a_2(k - 1), & a_1(k) \geq \theta_1
\end{cases}
\]
\[
Y_1(k) = h(a_1(k), a_2(k)) = \begin{cases} 
  \text{none}, & a_1(k) < \theta_1 \\
  a_2(k), & a_1(k) \geq \theta_1
\end{cases}
\]

**Figure 3:** NNBA model first hidden layer neuron function.

(a) Draw samples from the importance function \( q(x_k \mid z_{0:k}) \), and denote them as \( \{x^i_k\}_{i=1}^N \).

(b) Obtain the respective particle weights by
\[
w^i_k = \frac{p(x_k \mid x^i_{k-1}) N(x_k \mid \mu_{k-1}, \Sigma_{k-1})}{q(x_k \mid z_k)}, \quad (17)
\]

(c) Normalize the weights as
\[
w^i_k = \frac{w^i_k}{\sum_{i=1}^N w^i_k}. \quad (18)
\]

(d) Calculate the mean \( \mu_k \) and the covariance \( \Sigma_k \) of approximate Gaussian distribution by
\[
\mu_k = \frac{1}{N} \sum_{i=1}^N x^i_k, \\
\Sigma_k = \frac{1}{N} \sum_{i=1}^N (x^i_k - \mu_k) (x^i_k - \mu_k)^T. \quad (19)
\]

(e) Estimate the target position at time \( t_k \) and predict the target position at time \( t_{k+1} \) by
\[
\hat{x}_k = \mu_k, \quad \hat{x}_{k+1} = F_{k+1} \cdot \hat{x}_k. \quad (20)
\]

**4.3. NNBA Neuron Model**

**4.3.1. The First Hidden Layer.** Figure 3 shows the first hidden layer neuron model. The \( i \)th input is the measurement \( z_{k,i} \) of node \( i \). The input function, which is a weighted sum, actually becomes an identity function. Because each of the neuron only has one input signal, the operation result is input signal itself. The process area has two functions. The first calculates the difference between the measurement and previous record if the measurements satisfy \( p_1 \geq Z_{th}/\epsilon \) (\( \epsilon \in (0, 1] \)). Based on the measurement model (6), the sensor data is valuable when the distance \( \|r_t - r_n\| \) between sensor node and the target is within certain limits, so the threshold \( Z_{th}/\epsilon \) guarantees the accuracy of measurement. And the second function checks and decides whether to record this current input data. The output function is a specific threshold function, which is different from the standard threshold function with output 0 or 1. This function outputs \( a_2(k) \) if \( a_1(k) \geq \theta_1 \), else output none, to reduce the time redundancy.

**4.3.2. The Second Hidden Layer.** The second hidden layer on the CH merges with the GPF algorithm (see Figure 4). CH chooses three maximum measurements and puts them in neurons, respectively, in the second hidden layer. According to GPF algorithm, CH draws samples \( \{x^i_{k-1}\}_{i=1}^N \) from the distribution \( p(x_{k-1}) \sim N(x_{k-1}; \mu_{k-1}, \Sigma_{k-1}) \), in which the mean \( \mu_{k-1} \) and covariance \( \Sigma_{k-1} \) can be computed by (16). In the process function, each neuron continues to complete the time update process, draws samples from the importance function \( q(x_k \mid z_{0:k}) \), and obtains the respective particle weights by (17) based on input parameter \( p, \mu_{k-1}, \Sigma_{k-1} \) and the same particle set \( \{x^i_{k-1}\}_{i=1}^N \). At last, each neuron outputs its own normalization weights.

**4.3.3. The Third Hidden Layer.** One neuron is used to complete the subsequent GPF algorithm in the third hidden layer. As shown in Figure 5, CH averages value of the input \( Y_2 \), the weight of the three particle sets. The process function is used
to estimate the target position $\hat{x}_k$ and covariance $\Sigma_k$ according to (19). The output of the function equals the estimated target position $\hat{x}_k$ represented by $Y_3$, which will be transferred to the output layer.

The GPF tracking algorithm is completely performed by the second and third hidden layer. Note that these two can be combined into a union, in which case the input data should be aggregated into a vector.

4.3.4. **The Output Layer.** The output neuron model resembles the input neuron. Different from the input neuron, the output neuron model can be multi-input neuron, which is determined by the sensor type. In this study, only acoustic energy sensor is used, so there is one neuron in output layer.

As Figure 6 shows, CH compares the input $p_o$ and $a(k-1)$. If the difference exceeds the threshold $\theta_2$, that is, $\Delta(k) \geq \theta_2$, CH outputs $p_o$ and sends the TargetToSinkMesg containing target state estimation to the sink; otherwise, the neuron outputs none, and CH sends TargetDegradeMesg message to the sink. The shortened messages reduce traffic and lower the energy consumption especially when the target stops or moves slowly.

Neurons in NNBA algorithm need some operation parameters (such as weight, thresholds, etc.) at runtime; as
a result, the neuron network model parameters should be trained after the clustering and CH election. In general, the neuron network can learn, train, and adjust these parameters by itself. However, due to the limited resources, it is unadvisable to do the training inside the DWSNs. Hence, NNBA model does this training process at the sink out of the neuron network. And then the sink would send the model parameters to the CHs and member nodes.

5. Simulation Environment and Setup

In this section, we build a simulation platform to evaluate the performance of the proposed distributed collaborative target tracking framework. Firstly, we describe our simulation environment and parameters. Then we define the performance metrics to compare the proposed algorithm and others’ work.

5.1. Simulation Environment. We simulate a wireless sensor network with \( N_s = 200 \) randomly deployed acoustic sensors to monitor a \( 800 \text{ m} \times 800 \text{ m} \) square field. All the sensor nodes within the monitoring region (excluding sink) have the same structure and can detect the target within the sensing range. The measurements collected by sensor nodes are corrupted by zero-mean i.i.d. Gaussian noise with variance \( \sigma^2_w = 0.00005 \). The parameters of the measurement model in (6) are \( s = 1000, g_n = 1. \) Besides that, for the CH selection phase in the proposed framework, weight constants in (10) are \( \alpha = 0.4, \beta = 0.3, \) and \( y = 0.3; \) energy threshold is set to 10% of the initial battery level; namely, \( E_{th} = 0.1 \times E_0. \)

We should also determine the thresholds \( \varepsilon \) and \( \rho \), for they are playing an important role in selecting the candidate nodes and CH. The bigger the \( \varepsilon \) is, the higher the tracking accuracy is, but the more energy the whole network consumes. The bigger the \( \rho \) is, the less frequently the CH hands over, and the more energy consumption the whole network has. On the other hand, the tracking accuracy is not closely tied with \( \rho \). According to large number of experiment results, a best tradeoff between good tracking accuracy and low energy consumption is achieved in the case of \( \varepsilon = 0.64 \), \( \rho = 0.7 \). Simulation parameters are summarized in Table 2.

| Parameter               | Value                      |
|-------------------------|----------------------------|
| Monitoring region       | \( 800 \text{ m} \times 800 \text{ m} \) |
| Number of nodes \( N_s \) | 200                         |
| Sink location           | \((400, 400)\) in the middle of the area |
| Sensing radius \( R_s \) | 150 m                       |
| Communication radius \( R_c \) | 300 m                   |
| Capability of power \( E_0 \) | 1 J                     |
| \( E_{\text{elec}} \)    | 50 nJ/bit                  |
| \( \varepsilon_{fs} \)   | 10 pJ/bit/m^2              |
| \( \varepsilon_{smp} \)  | 0.0013 pJ/bit/m^4          |
| Member nodes threshold \( \varepsilon \) | 0.64                   |
| CH threshold \( \rho \)  | 0.7                        |

5.2. Performance Metrics

5.2.1. Tracking Performance. Assume a target moves into the monitoring field at time \( t = 0 \) with initial state \([50 6 0 6]^{\top}\). The simulation tracking time is \( T_{\text{sim}} = 100 \). The system process noise \( w_k \) is assumed as zero-mean Gaussian distribution with variance \( \Sigma_w = \text{diag}([0.125 0.5 0.125 0.5]) \). The number of particles used in each algorithm is \( M = 600 \).

To evaluate the tracking performance for different algorithms, we calculate the root of mean square errors (RMSEs)
of target position and velocity estimations at each time step under 50 Monte Carlo runs. The RMSEs metric [24] is defined as

$$\text{RMSE}(t_k) = \left\{ \begin{array}{ll} \frac{1}{N_{\text{mc}}} \sum_{j=1}^{N_{\text{mc}}} \left( (\hat{x}_{k}^{j} - x_{k}^{\text{true}})^2 + (\hat{\eta}_{k}^{j} - \eta_{k}^{\text{true}})^2 \right) & \text{for position,} \\ \frac{1}{N_{\text{mc}}} \sum_{j=1}^{N_{\text{mc}}} \left( (\hat{x}_{k}^{j} - \dot{x}_{k}^{\text{true}})^2 + (\hat{\eta}_{k}^{j} - \dot{\eta}_{k}^{\text{true}})^2 \right) & \text{for velocity,} \end{array} \right. \quad (21)$$

where $\hat{x}_{k}^{j}, \hat{\eta}_{k}^{j}$ are the target position estimation at time step $t_k$ in $j$th Monte Carlo run, and $x_{k}^{\text{true}}, \eta_{k}^{\text{true}}$ are the true position at $t_k$. Similarly, $\dot{\hat{x}}_{k}^{j}, \dot{\hat{\eta}}_{k}^{j}$ are the target velocity estimation at time step $t_k$ in $j$th Monte Carlo run, and $\dot{x}_{k}^{\text{true}}, \dot{\eta}_{k}^{\text{true}}$ are the true velocity at $t_k$.

5.2.2. Energy Consumption. The node energy depletion results from sensing module, communication module, and the processing module. The total energy consumption at time $t_k$ can be summarized as follows:

$$E(k) = E_{\text{sen}}(k) + E_{\text{comm}}(k) + E_{\text{proc}}(k). \quad (22)$$

Among the three parts, wireless communication module plays the dominant role in the energy consumption. Many researches have been carried out on the low-energy radios. The first-order radio energy consumption model [25] is adopted for simulation. To transmit $m$-bit message over a distance $d$, the radio expends

$$E_{\text{Tx}}(m,d) = E_{\text{Tx-elec}}(m) + E_{\text{Tx-amp}}(m,d) = \begin{cases} mE_{\text{elec}} + m\varepsilon_{fs}d^2, & d < d_0 \\ mE_{\text{elec}} + m\varepsilon_{mp}d^4, & d \geq d_0. \end{cases} \quad (23)$$

To receive $m$-bit message, the radio expends

$$E_{\text{Rx}}(m) = E_{\text{Rx-elec}}(m) = mE_{\text{elec}}. \quad (24)$$

where $E_{\text{Tx-elec}}$ and $E_{\text{Rx-elec}}$, separately denote the energy to activate the baseband circuits when transmitting and receiving. $E_{\text{Tx-amp}}$ is the energy to run the RF module. The electronic energy, $E_{\text{elec}}$, denotes the energy spent to activate the baseband circuits to transmit or receive 1 bit. And $\varepsilon_{fs}$ and $\varepsilon_{mp}$ denote the energy spent to run the RF module to transmit 1 bit in free space and multipath space separately. $d_0$ is a threshold indicating that if the distance is less than $d_0$, the free-space model is used; otherwise, the multipath model is used. For the experiments, the communication energy parameters are set: $E_{\text{elec}} = 50 \text{ nJ/bit}$, $\varepsilon_{fs} = 10 \text{ pJ/bit/m}^2$, and $\varepsilon_{mp} = 0.0013 \text{ pJ/bit/m}^4$.

From (23), we can derive $d_0$ as

$$d_0 = \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}} = 87.7 \text{ m}. \quad (25)$$

The packet is defined in Table 3.

In all algorithms, the HEADER of the listed messages in Table 3 is 8 bits.

6. Simulation and Analysis

This section shows experimental studies for verifying the proposed framework.

6.1. Performance of Target Tracking. In order to demonstrate the GPF’s satisfied performance of tracking accuracy, a comparison with PF [17] is carried out below. Here, we consider two cases: in PF-1, the mean and covariance for Gaussian noise are modeled precisely; the noise statistical parameters in PF-2 have large model error.

Figure 7 displays a typical trajectory and the tracking obtained by three different filtering algorithms. Figure 8 demonstrates the comparison of RMSEs of the estimated position and velocity obtained by three different filtering algorithms at each time step under 50 Monte Carlo runs ($N_{\text{mc}} = 50$).

As shown in Figures 7 and 8, we can see that the PF-1 algorithm has the best tracking accuracy, while PF-2 works badly. It is because that PF is greatly influenced by model parameters.

It is noted that the GPF algorithm outperforms the PF-2 with a close performance to PF-1 algorithm. Though GPF omits the resampling process, it preserves the desired accuracy of target estimation. Instead of handing over all particles to the new CH in a general PF algorithm, GPF only needs to transmit parameters (the mean and the covariance of the posterior distribution), which saves amount of energy. Further discussion will be detailed in Section 6.2.

To further validate the effectiveness of our proposed algorithm CSIPF, we compare it with those obtained from previous work [8, 11, 12], including CSIP1, CSIP2, and CSIP3 algorithms. All of them use GPF estimation method.

The tracking trajectories obtained by these algorithms are shown in Figure 9. Figure 10 demonstrates the average result of 50 Monte Carlo runs from up to down: (a) the RMSE
Table 3: Packet descriptions.

| Message type               | Direction | Details                                                                                      | Length (bit) |
|----------------------------|-----------|----------------------------------------------------------------------------------------------|--------------|
| CHInfoMesg                 | CH → node | 8 + 16 + 32 + HEADER, 8 → type, 16 → CH ID, 32 → coordinate                                  | 64           |
| CHHandoverMesg (for GPF algorithm) | CH → CH  | 8 + 16 + 16 + 320 + HEADER, 8 → type, 16 → previous CH ID, 16 → new CH ID, 320 → CH handover parameters (10 * 16 bits), including estimated target state and covariance | 272          |
| DataMesg                   | Node → CH | 8 + 16 + 32 + 32 + 16 + HEADER, 8 → type, 16 → node ID, 32 → node coordinate, 32 → measurement at time step \( t_k \), 16 → remaining energy | 112          |
| DataDegradMesg             | Node → CH | 8 + 16 + 16 + HEADER, 8 → type, 16 → node ID, 16 → node remaining energy                     | 48           |
| TargetToSinkMesg           | CH → sink | 8 + 16 + 64 + HEADER, 8 → type, 16 → CH ID, 64 → target position estimation                  | 96           |
| TargetDegradMesg           | CH → sink | 8 + 16 + HEADER, 8 → type, 16 → CH ID                                                        | 32           |
| CHHandoverPFMesg (for PF algorithm) | CH → CH  | 8 + 16 + 16 + 43200 + HEADER, 8 → type, 16 → previous CH ID, 16 → new CH ID, 43200 → CH handover parameters including 600 particles and weights | 43248        |

Figure 8: (a) The position RMSE. (b) The velocity RMSE.
in the estimated target position and (b) the RMSE in the estimated target velocity.

It is clear that the tracking performance of CSIFP is comparable to the referenced methods (CSIP1, CSIP2, and CSIP3). CSIP1 performs best because of relying on the largest set of sensors’ measurement. However, a rise in energy consumption appears. CSIP2 and CSIP3 approach a moderate precision due to their own strategies of CH selection and spatial fusion. It makes sense that the proposed CSIFP introduces not only spatial fusion but also time fusion, sacrificing the tracking accuracy in some degree. It is noteworthy that the slight performance degradation is worthy in contrast to the significant energy saving benefit in Section 6.2.

6.2. Performance of Network Energy Efficiency. In this subsection, we will firstly explain why the GPF algorithm is adopted by comparing generic PF’s and GPF’s energy consumption. Then the performance of energy efficiency in our proposed CSIFP is validated as well as the CSIP1, CSIP2, and CSIP3.

From Figure 11, the cumulative energy consumptions in PF-1 algorithm and PF-2 algorithm are nearly the same. It is obvious that energy consumption is uncorrelated with prior model of system noise. The GPF, by contrast, consumes least energy of the three. The theoretical proof is derived.

In the PF algorithm, $M$ particles and their weights are transferred when tracking task handover happens, and total quantity of data transmission is

$$Q_1 = K \cdot M + M,$$  

where $K$ represents the dimension of the target state vector and $1 \leq K \leq 6$ for the 2-D target tracking. $M$ is the number of the particles, usually in the $10^2$ order of magnitude or more. In the GPF algorithm, the current CH transmits nothing but the Gaussian mean and covariance of the target estimation. Considering the symmetry of the covariance, the quantity of the communication data is

$$Q_2 = K + K \cdot \frac{K + 1}{2}.$$  

**Figure 9:** Target tracking trajectory using CSIP1, CSIP2, CSIP3, and CSIFP algorithm.

**Figure 10:** (a) The position RMSE. (b) Velocity RMSE.
Figure 11: The cumulative communication energy consumption of PF and GPF after the entire tracking process.

Figure 12: Traffic in different particle numbers and state dimensions for PF and GPF.

It can be seen from Figure 12, the traffic of general PF algorithm is much bigger than GPF, and the latter’s traffic has nothing to do with the particle number. The accuracy of the algorithm can be improved without a heavier traffic load by increasing the number of particles, if the computation complexity is accepted.

Figure 13 presents the total communication energy consumption of four different collaborative processing schemes. We can see that the CSIP1 consumes the most energy because all the active nodes with the $R_i$ range send their measurements to CH. The energy consumption of CSIP2 and CSIP3 is less than CSIP1 by choosing the candidate sensor nodes nearby the target to participate the tracking process. The former saves more energy due to selecting the farther member node as the new CH when CH changes, which reduces the handovers. The proposed CSIFP scheme consumes the least energy, because the CH selection method makes advantage of CSIP2, and the NNBA model is utilized to reduce the time redundancy as well. Consequently, the network payload approaches an expectable level.

6.3. Performance of Robustness. Furthermore, for investigating the robustness of our proposed algorithm CSIFP, 6 independent Monte Carlo simulations with different trajectories are carried out. Figure 14 presents these random trajectories and tracking results obtained by these four algorithms. It is clear to us that CSIFP can keep good tracking accuracy when the trajectory changes.

Table 4: Average result of performance.

|                | CSIFP | CSIP1 | CSIP2 | CSIP3 |
|----------------|-------|-------|-------|-------|
| Position RMSE (m) | 7.7646 | 6.9960 | 7.0706 | 7.3298 |
| Velocity RMSE (m/s) | 1.3077 | 1.4051 | 1.3125 | 1.2685 |
| Energy cost (J)      | 0.3299 | 0.7775 | 0.3664 | 0.4001 |

To further verify the tracking performance of the proposed algorithm, the position RMSE and velocity RMSE are, respectively, illustrated in Figure 15 and Figure 16. The position RMSE curve of CSIFP has obvious fluctuation. It is because that CSIFP reduces time redundancy by NNBA model. NNBA model will be inertial if a slight shift happens in contrast to the previous measurement. Figure 17 demonstrates the total communication energy consumption of the tracking under four different collaborative processing schemes. We can see that CSIFP is energy efficient for it consumes the least amount of energy.
Moreover, we carry out statistics by average RMSE and total energy consumption, as illustrated in Table 4. CSIFP achieves nearly the same good tracking accuracy. It obtains a gain of 56.83% in the energy cost in contrast with CSIP1, with a sacrifice of 12.88% in precision. Compared to CSIP2 and CSIP3, CSIFP consumes the least energy. This gain benefits from not only the CH selection method but also the fusion algorithm based on NNBA. From the above results, a good tradeoff between tracking accuracy and energy cost is achieved.
7. Conclusions

In this paper, we have presented a high energy-effective strategy of collaborative signal and information fusion processing for acoustic target tracking applications in DWSNs. Within this framework, the whole network is assumed to be a neural network, in which the member nodes correspond to the neurons and the CH corresponds to the neural center. The nodes detect the target and transmit measurements to the CH. CH receives the information, responds to different neurons’ stimulation, estimates the target state based on the GPF, and sends the estimation to the sink finally.
Besides, a novel clustering scheme and CH selection method are implemented to balance the energy cost and tracking accuracy. The framework saves energy consumption through three aspects: (1) GPF algorithm transmits fewer but critical messages; (2) the NNBA model reduces the information redundancy; (3) the CH selection method demands less frequency of task handovers.

Last but not least, the numerical simulations directly validate that this collaborative tracking fusion framework can decrease the total network energy consumption and guarantee high tracking accuracy.

There have been several problems unresolved. Our proposed algorithm may not apply to the highly maneuverable case. The framework refers to the prior knowledge of
the covariance matrices and noise variances, whereas they are usually unknown and time-varying in practical.

**Conflict of Interests**

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

**Acknowledgments**

This work is supported in part by Shanghai Science and Technology Commission (SSTC) research projects under Grant no. 12DZ2293200 and the “Strategic Priority Research Program” of the Chinese Academy of Sciences (CAS) under Grant no. XDA06020300.

**Figure 17**: The cumulative communication energy consumption of CSIFP, CSIP1, CSIP2, and CSIP3 after the entire tracking process of 6 different trajectories.
References

[1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, “Wireless sensor networks for habitat monitoring,” in Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, pp. 88–97, September 2002.

[2] O. Karaca and R. Sokululu, “A cross-layer fault tolerance management module for wireless sensor networks,” Journal of Zhejiang University: Science C, vol. 13, no. 9, pp. 660–673, 2012.

[3] H. Tsai, C. Chu, and T. Chen, “Mobile object tracking in wireless sensor networks,” Computer Communications, vol. 30, no. 8, pp. 1811–1825, 2007.

[4] O. Younis and S. Fahmy, “Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach,” in Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM ’04), vol. 1, IEEE, March 2004.

[5] Y. Xu and H. Qi, “Mobile agent migration modeling and design for target tracking in wireless sensor networks,” Ad Hoc Networks, vol. 6, no. 1, pp. 1–16, 2008.

[6] A. Arora, P. Dutta, S. Bapat et al., “A line in the sand: a wireless sensor network for target detection, classification, and tracking,” Computer Networks, vol. 46, no. 5, pp. 605–634, 2004.

[7] 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.

[8] S. K. Sarna and M. Zaveri, “ERTA: energy efficient real time target tracking approach for wireless sensor networks,” in Proceedings of the 4th International Conference on Sensor Technologies and Applications (SENSORCOMM ’10), pp. 220–225, Venice, Italy, July 2010.

[9] K. Tsukamoto, H. Ueda, H. Tamura, K. Kawahara, and Y. Oie, “Design of wireless sensor network for multi-point surveillance of a moving target based on the relationship between tracking probability and sensor density,” in Proceedings of the 6th IEEE Consumer Communications and Networking Conference (CCNC ’09), pp. 1–5, January 2009.

[10] Y. Zou and K. Chakrabarty, “Target localization based on energy considerations in distributed sensor networks,” Ad Hoc Networks, vol. 1, no. 2-3, pp. 261–272, 2003.

[11] W. C. Yang, Z. Fu, J. H. Kim et al., “An adaptive dynamic cluster-based protocol for target tracking in wireless sensor networks,” in Advances in Data and Web Management, pp. 157–167, Springer, Berlin, Germany, 2007.

[12] H. Yang and B. Sikdor, “A protocol for tracking mobile targets using sensor networks,” in Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications, pp. 71–81, Anchorage, Alaska, USA, May 2003.

[13] K. A. Darabkh, S. S. Ismail, M. Al-Shurman, I. F. Jafar, E. Alkhaled, and M. F. Al-Mistarihi, “Performance evaluation of selective and adaptive heads clustering algorithms over wireless sensor networks,” Journal of Network and Computer Applications, vol. 35, no. 6, pp. 2068–2080, 2012.

[14] O. Demigha, W. Hidouci, and T. Ahmed, “On Energy efficiency in collaborative target tracking in wireless sensor network: a review,” IEEE Communications Surveys and Tutorials, vol. 15, no. 3, pp. 1210–1222, 2013.

[15] J. H. Kotecha and P. M. Djuric, “Gaussian particle filtering,” IEEE Transactions on Signal Processing, vol. 51, no. 10, pp. 2592–2601, 2003.

[16] B. Ristic, S. Arulampalam, and N. J. Gordon, Beyond the Kalman Filter: Particle Filters for Tracking Applications, Artech House Publishers, 2004.

[17] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 174–188, 2002.

[18] N. Kantas, A. Doucet, S. S. Singh, and J. M. Maclejowski, “An overview of Sequential Monte Carlo methods for parameter estimation in general state-space models,” in Proceedings of the 15th IFAC Symposium on System Identification (SYSID ’09), vol. 15, pp. 774–785, Saint-Malo, France, July 2009.

[19] X. R. Li and V. P. Jilkov, “Survey of maneuvering target tracking—part I: dynamic models,” IEEE Transactions on Aerospace and Electronic Systems, vol. 39, no. 4, pp. 1333–1364, 2003.

[20] D. Li and Y. H. Hu, “Energy-based collaborative source localization using acoustic microsensor array,” Eurasip Journal on Applied Signal Processing, vol. 2003, no. 4, pp. 321–337, 2003.

[21] X. Sheng and Y. Hu, “Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks,” IEEE Transactions on Signal Processing, vol. 53, no. 1, pp. 44–53, 2005.

[22] C. Mesoookho, U. Mitra, and S. Narayanan, “On energy-based acoustic source localization for sensor networks,” IEEE Transactions on Signal Processing, vol. 56, no. 1, pp. 365–377, 2008.

[23] W. Wang, T. Huang, Y. Zhou, and H. Liu, “Improved data fusion model for wireless sensor network,” in Proceedings of the 6th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM ’09), IEEE, September 2009.

[24] Z. Yu, J. Wei, and H. Liu, “An energy-efficient target tracking framework in wireless sensor networks,” Eurasip Journal on Advances in Signal Processing, vol. 2009, Article ID 524145, 14 pages, 2009.

[25] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” IEEE Transactions on Wireless Communications, vol. 1, no. 4, pp. 660–670, 2002.