Energy-Quality Optimization Model for Target Tracking in Wireless Sensor Networks

Yongcai Wang, Dianfei Han, Qianchuan Zhao, Xiaohong Guan and Dazhong Zheng

{wangyongcai,handianfei99}@mails.tsinghua.edu.cn

CFINS, Automation Department

8/23/2004

Abstract: For target tracking in wireless sensor networks, reducing energy consumption and improving tracking accuracy are two main concerns. In this paper, by describing the essence of tracking accuracy as the number of nodes involved in detection, an energy-quality optimization model for cluster-based tracking is developed. This model is a two objectives and two variables optimization model, in which sensing radius and the number of clusters interact and affect the energy-quality performances. Based on this model, simple solution strategy is proposed to obtain Pareto Fronts designs for the number of clusters and sensing radius. This result can be used as design guidance in network setup and sensors configuration. Simulation results validate its efficiency. We also discuss the advantages and disadvantages of cluster-based activation by comparing with naïve activation and random activation. Target missing problem is explored. Early wake-up and large-scope probe mechanisms are proposed and shown good performance in avoiding target missing.

1 Introduction

Advances in MEMS and integration of sensing and communication technologies have enabled the deployment of large-scale wireless sensor networks (WSN) for intelligence information gathering and environment monitoring [2]. One of the most important and promising areas where the advantages of sensor networks can be exploited is the tracking of mobile targets [3]. Different from traditional tracking systems like radar systems, sensor networks have some unique features and specific application requirements. Three primary features are 1) the high nodes density, 2) tiny, cheap and multifunctional hardware, and 3) ad-hoc deployment. These features make WSN more suitable for many tracking applications both in military and civilization. In this paper, we study the tracking problem in WSN. We propose an energy-quality optimization model and discuss its application in cluster-based tracking system.

Specific application requirements of WSN rise from two aspects: the inherent properties and the demands due to the tracking application. First, inherent properties of sensor nodes mainly include limited power resource, small transmission range, limited computation capacity and distributed deployability, etc. These properties result in several specific requirements in applying WSN. Among all the requirements, extension of the network’s lifetime is the most important. Therefore, energy efficiency consideration should go through all the designing process in tracking protocols. Also, Ad-Hoc deployment leads to important requirements in sensor nodes coordination, which may affect every aspect of tracking performance. Second, the demands due to tracking application lead to the requirements for the high tracking accuracy and low missing rate. Lots of efforts and investigations have been devoted in this area to design tracking protocols or other mechanisms to satisfy these requirements [3]-[4]. Among them, cluster-based prediction tracking protocols are the most promising solutions [5][6][7]. In this kind of tracking protocols, only a subset of sensors are active when tracking targets, and other nodes keep sleeping until a predicted waking up. Besides this, naïve activation and random on/off activation also have been investigated. These efforts and studies have provided effective techniques in designing tracking protocols and also take important insights into the behaviors of the WSN in tracking application. Among these insights, one of the well recognized is the tradeoff between the energy efficiency and the tracking accuracy [3][8]. Energy consumption of a WSN occurs in four domains: message sending, message receiving, data processing and sensing. Tracking accuracy is affected by a number of aspects as: sensor types, sensing accuracy, data fusion technique, nodes density, target moving pattern, routing protocol etc. So Energy-Quality tradeoff involves almost every
aspect of WSN tracking system.

In this paper, we address the problem of establishing an energy-quality optimization model. The basic inspiration is to describe the essence of tracking accuracy as the number of nodes involved in detection. We investigate its validity for different kinds of detection mechanisms by proving that the variance of estimation error converges at $O(1/\sqrt{n})$, where $n$ is the number of sensors which are involved in target detection. This approach helps us to formulate a unified quality metric, which is independent to the target trajectories and detection schemes. The energy metric is formulated as an average energy consumption model, counting the sensing, communication, data processing energy for a tracking action. Quality metric and energy metric interact as tradeoffs. The overlapping area of active cluster region and target detectable region is critical in analyzing the two metrics’ relationship. We calculate the convolution of the overlapping area and derive out the expression of the expected number of sensors that can detect the target. Then a two-variable-two-objective optimization model is built and a performance evaluation framework is set up. The two variables are the number of clusters and the sensing radius of nodes. Multi-variable optimization strategy is applied to find Pareto Fronts solutions for the two variables. Pareto Fronts solutions are non-bad solutions candidates in multivariable optimization. Obtaining these Pareto Fronts, the analyzing results work as design guidance for sensors configurations and topology construction. Efficiency of analytical results is verified by extensive simulation. In simulation, cluster-based tracking protocol described in [3] is used. We simulate for different net scales and different target patterns. The simulation results show that the Pareto Fronts obtained from analytical results are well in accordance with those in the simulation results. Further analyzing reveals that the high consistency is mainly because of the robustness of order, and the difference is mainly caused by the target missing. Advantage and disadvantage of cluster-based tracking is also discussed by comparing with naive activation and random activation [1]. Target missing problem is explored and two recovery mechanisms: early wake-up and large-scope are proposed. They show good performance in simulation.

The organization of this paper is as follows. In section 2, we introduce the network configuration, the energy consumption model and three basic activation schemes used in mobile target tracking. In section 3, we present the definition of the energy metric and a unified quality metric. In section 4, we present the derivation process of the optimization model. In section 5, experiments results are illustrated to show the performance of optimization model. In section 6, we discuss the strengths and defects of cluster based tracking and propose a target recovery scheme. In section 7, we conclude this paper and outline future directions.

2 Models and Tracking Schemes

2.1 WSN tracking model

We consider a similar network model as proposed in LEACH [5] and PEGASIS [7]. A network consisting of $N$ homogeneous nodes is deployed in a $(m) \times (m)$ square area. The nodes are randomly distributed in the area with uniform density $\rho$, so $N = \rho \cdot L \cdot L$. Sensors are designed to cooperate in detecting one moving target and send the acquired information to a distant base station located at $(x, y)$. Nodes are assumed to know their self-locations, which can be obtained by applying locating process at initialization phase [10] [11]. We assume every sensor has two configured sensing radiuses, normal radius $s$ and long radius $S$. The value of $s$ and $S$ can be selected within a range $[s_{\text{small}}, s_{\text{large}}]$. Sensors usually use radius $s$ to detect target and $S$ is only used for target recovery. Each sensor is also assumed to have two configured transmission radius. The first is a basic transmission radius, denoted by $c$, which can cover the one hop transmission within the cluster. The second is a larger transmission radius, denoted by $C$, which can cover the long-range transmission to the base station. $c$ and $C$ can be selected within a range $[c_{\text{small}}, c_{\text{large}}]$. The selectable sensing and transmission range assumptions are closely based on the function of current sensor devices. For example, Mica2 mote provides totally 100 different transmission radiuses [12] and the sensing range can be controlled by power. Sensors have two states: “active” and “sleep”. In “active” state, sensors can sense and communicate, while in “sleep state”, they stay in a power saving mode and wait for
waking up. The target moves in an unknown pattern, while active sensors keep tracking and report their detection in a time round scheme [5]. Figure 1 demonstrates the network configurations.

![Wireless sensor network model for target tracking](image)

**Figure 1. WSN model for tracking application**

### 2.2 Energy consumption model

Based on [1][5][7], we use a simplified energy consumption model to evaluate the energy dissipation. Basically, energy consumption is made up of four parts:

1. **Message sending**
   
   We use the First Order Radio Model mentioned in [5]. The radio dissipates $E_{\text{elec}} (nJ/\text{bit})$ per bit to run the radio circuitry and consumes additionally $E_{\text{amp}} (pJ/\text{bit}/m^2)$ for the amplifier. We assume an $c^\alpha$ energy loss due to channel transmission. Thus, to transmit a $k$ bit message to a distance $c$, a node consumes:
   
   $$T(c) = E_{\text{elec}} \cdot k + E_{\text{amp}} \cdot k \cdot c^\alpha$$  \hspace{1cm} (2.1)

2. **Message receiving**
   
   The receiving circuit differs from the sending part in that no amplifier is needed. So to receive a $k$-bit message, a node consumes:
   
   $$L = E_{\text{elec}} \cdot k$$ \hspace{1cm} (2.2)

3. **Detection**

   Based on [1], we assume an $s^\beta$ energy loss. In order to cover an area with radius $s$, a node consumes:
   
   $$S(s) = E_{\text{sensors}} \cdot s^\beta$$ \hspace{1cm} (2.3)

4. **Data processing**

   It is reasonable to assume energy dissipation for processing is proportional to length of the message. So we have:
   
   $$F = F_{\text{cpu}} \cdot k$$ \hspace{1cm} (2.4)

These four aspects work as components in composing the energy consumption of WSN.

### 2.3 Tracking protocols

Besides model introduction, we briefly introduce the existing tracking protocol in the literature.

2.3.1 Native activation

In this strategy, all nodes are configured to sense all the time.

2.3.2 Randomized activation

In randomized activation, each node is configured to be on with a probability $p$ and be off with a probability $1-p$. When a node is off, it is in sleep mode. Although it cannot sense, it can be waken up by other nodes. This kind of tracking protocols are discussed in [1].
2.3.3 Clustered activation

This strategy is introduced in [5] [6] [7]. At any time, only subset of nodes are active, while the others are “sleep” to save energy. The cluster members detect the target and send the information to the cluster head. After data processing, the cluster head transmits the results to base station.

In this paper, we will mainly focus on cluster-based tracking schemes and build up quality-energy optimization model. The other two activation schemes will be discussed in section 6.

3 Performance metrics

For a target tracking application, two performance metrics are mainly concerned. 1) Energy consumption of WSN, which determines how long the system can work. 2) Tracking accuracy, which reflects the application objective.

3.1 Energy metrics

For energy metric, we define it as the average energy expenditure of the network for one target report. It varies for different tracking protocols. In this section, we formulate the energy metrics based on the WSN model discussed in section 2 for different tracking schemes.

For cluster-based tracking, figure 2 demonstrates a tracking scenario. If the radius of cluster is \( R \), an active cluster averagely contain \( n_c \) sensors, where \( n_c = \pi R^2 \rho \). Not all of these sensors can detect the target. Only \( n_s \) of them in the target detection area can detect the object. Since the overlapping area is not a circle, supposing its area is \( S_{\delta} \), then \( n_s = \rho S_{\delta} \). \( n_s \) is the number of sensors in the active cluster that can detect target. According to the energy consumption model, and assuming hop-by-hop data processing and good transmission scheduling, there are totally \( n_s \) times of transmissions, \( n_s \) times of receiving, \( n_s \) times of data processing and one time of long range transmission to the base station. So we the energy consumption is

\[
E = n_s E_{\text{sens}} s^\beta + n_s (E_{\text{elec}} k + E_{\text{amp}} k C^\alpha + E_{\text{elec}} k + F_{\text{cpu}} k) + (E_{\text{elec}} k + E_{\text{amp}} k C^\alpha)
\]

(3.1)

The energy consumption is composed with three parts. \( n_s E_{\text{sens}} s^\beta \) is the sensing energy, \( n_s (E_{\text{elec}} k + E_{\text{amp}} k C^\alpha + E_{\text{elec}} k + F_{\text{cpu}} k) \) is the energy consumed in the cluster, and \( (E_{\text{elec}} k + E_{\text{amp}} k C^\alpha) \) is long range communication energy.

![Figure 2. Cluster-based target tracking in WSN](image)

In naïve and random activation, for the fair of comparison, we suppose there are appropriate routing schemes making communication energy comparable to the cluster-based strategies. Only one time of long range communication is needed. As cluster-based tracking, energy consumption operations for a sensor to report a message include once transmission, once receiving and once data processing. We formulate it as report energy.

\[
E_{\text{report}} = E_{\text{elec}} k + E_{\text{amp}} k C^\alpha + E_{\text{elec}} k + F_{\text{cpu}} k
\]

(3.2)

In naïve activation, all the nodes are in sensing mode and only the nodes within the detection range can detect the object. The detection range is a circle of radius \( s \) around target. Suppose there are
$n_s$ nodes that can detect the object, then $n_s = \rho \pi s^2$, so the expected energy consumption is:

$$E = N \cdot E_{\text{sensor}} \cdot s^\beta + n_s \cdot E_{\text{report}} + E_{\text{elec}} k + E_{\text{amp}} k C^\alpha$$

(3.3)

where $E_{\text{elec}} k + E_{\text{amp}} k C^\alpha$ means only one time of long range communication is needed.

Similarly, in randomized activation, there are on average $p \cdot N$ nodes active in sensing mode and only $P \cdot n_s$ sensors can detect the target, where $n_s = \rho \pi s^2$. The expected energy dissipation is:

$$E = P \cdot N \cdot E_{\text{sensor}} \cdot s^\beta + P \cdot n_s \cdot E_{\text{report}} + E_{\text{elec}} k + E_{\text{amp}} k C^\alpha$$

(3.4)

3.2 Quality metrics

Average tracking deviation is commonly used as quality metric as introduced in [1] and [3], which can be formulated as:

$$Q = \frac{1}{T} \int_0^T \sqrt{(x_a(t) - x_e(t))^2 + (y_a(t) - y_e(t))^2} \, dt$$

(3.5)

where $(x_a(t), y_a(t))'$ is the actual position of the target at time $t$, and $(x_e(t), y_e(t))'$ is the estimated position of the target. Although this metric directly describes average tracking performance, it is dependent on the target’s trajectory. It is more suitable for the simulation based performance evaluation. For theoretical analysis, the target pattern is unknown. In order to give a proper metric, we give a close examination of the tracking algorithms. There are many tracking algorithms available in the literature, but according to the ability of the sensors, they can be categorized into three classes.

1) If the sensor can detect the location of the target, every detectable sensor provides an estimation of the target position. Weighted average, LSE(Least Square Estimation) or Kalman Filter can be used in fusing the target position.

2) If the sensor can only measure the distance to the target, geometrical methods are applied to calculate the position of the target.

3) If the sensor can only detect the appearance of target, it is binary detection. The location of sensor is used as estimation of the target. In the case centroid method can be applied to fuse the position.

Note that although these methods differ greatly in implementation, they are common in the principle: the more the better. The more sensors are involved in tracking target, the more accurate the tracking is. So we propose our quality metric as the number of nodes involved in target detection. This quality metric is essential to different tracking strategies and it is suitable for theoretical analysis for its independence to the target trajectory. The validity of this idea can be formulated as theorem 1.

**Theorem 1.** If the observation of the sensors are independent and the measurement error is irrelevant with distance, then the variance of estimation error converges at $O(1/\sqrt{n})$, where $n$ is the number of sensors which can detect target.

**Proof.** In the first kind of tracking algorithms, since each node can give an independent estimation of the target position, according to statistics, the estimation error converges at $O(1/\sqrt{n})$. In the second kind, the estimation error can be proved converges at least at $O(1/\sqrt{|n/3|})$, which is also $O(1/\sqrt{n})$. Detail proof is in Appendix. In the third mechanism, location of the sensors are used as the estimations, they work as independent observations of target position. If the detection area is fixed, and $n$ is the number of nodes in this area, then from statistics, the estimation error converges at $O(1/\sqrt{n})$.

One point to mention is that in the third mechanism, the way to increase tracking accuracy is to increase the nodes density. Because for this kind of detection, minimum expected error is dominated by inter node distance.
4 Optimization Model

After introducing the energy and quality metrics, we derive the optimization model in this section.

4.1 Calculation of the overlapping area

Recall figure 2. In clustered activation scheme, \( n_i = \rho S_d \). The radius of cluster is \( R \) and the sensing radius is \( s \). We denote \( x \) as the distance between the cluster head and the target position. For given \( R \) and \( s \), the overlapping area \( S_d \) is a function of \( x \). In order to formulate the relationship of \( S_d \) and \( x \), we use convolution to calculate the expression of \( S_d(x) \).

\[
S_d(x) = 2 \int_{-R-s}^{R+s} \sqrt{s^2 - \tau^2} \sqrt{R^2 - (x-\tau)^2} \, d\tau
\] (4.1)

Figure 3 illustrates how (4.1) works. We use convolution to calculate the overlapping area of two half circles, and then double the result to obtain \( S_d(x) \).

![Figure 3. The convolution method](image)

From convolution, we get the expression of \( S_d(x) \):

\[
\begin{cases} 
S_d(x) = \pi s^2 & \text{when } R - s > x > s - R \\
S_d(x) = -f_1 \left( \frac{R^2 - s^2 + x^2}{2x} \right) + f_1(s) + f_2(R) - f_2 \left( \frac{-R^2 + s^2 + x^2}{2x} \right) & \text{when } R + s > x > R - s \\
S_d(x) = 0 & \text{otherwise}
\end{cases}
\] (4.2)

in which, \( f_1(\bullet) = (\frac{\bullet}{2} \sqrt{s^2 - \bullet^2} + \frac{s^2}{2} \arcsin \frac{\bullet}{s}) \), \( f_2(\bullet) = (\frac{\bullet}{2} \sqrt{R^2 - \bullet^2} + \frac{R^2}{2} \arcsin \frac{\bullet}{R}) \). We write \( f_1(\bullet), f_2(\bullet) \) for simplifying expression. We can see \( S_d(x) \) is a piece-wise function. The function curves of \( S_d(x) \) with different parameters are shown in figure 4, in the solid curves. The upper solid curve shows \( S_d(x) \) with parameter \( s=1.5 \) and \( R=4 \). The lower solid curve shows \( S_d(x) \) with parameter \( s=1 \) and \( R=2 \).

To simplify \( S_d(x) \), we use a piece-wise linear function \( \tilde{S}_d(x) \) to approximate \( S_d(x) \), where:

\[
\begin{cases} 
\tilde{S}_d(x) = \pi s^2 & \text{when } R - s > x > s - R \\
\tilde{S}_d(x) = -\frac{1}{2} \pi s(x - R - s) & \text{when } R + s > x > R - s \\
\tilde{S}_d(x) = 0 & \text{otherwise}
\end{cases}
\] (4.3)

The function curve of \( \tilde{S}_d(x) \) is compared with \( S_d(x) \) in figure 4, in dashed line. We analyze the approximation error for different \( s \) and \( R \) to check approximating accuracy. Figure 5 shows the result of approximation error analysis. In Figure 5, we set \( s \) to 1 and vary \( R \) and \( x \) to calculate the bound of the approximation error over \( x \). More specifically, we have:

\[
delta S(R) = \max_x (S(x) - \tilde{S}(x))
\] (4.4)
\[ \delta_2(R) = \min_x (S_y(x) - \bar{S}_y(R)) \] (4.5)

\[ \delta_3(R) = \delta_1(R) - \delta_2(R) \] (4.6)

Figure 5 shows the curve of \( \delta_1, \delta_2 \) and \( \delta_3 \). It indicates that for different parameter settings, the difference between the upper bound and the lower bound is bounded. So we use \( \bar{S}_y(x) \) to represent the expression of the overlapping area in our following analysis.

4.2 Derivation of the optimization model

In (4.3), \( x \) is the distance from active cluster head to target real position. It is a random variable. If the probability density function (pdf) of \( x \) is \( f(x) \), since we have gotten the expression of \( \bar{S}_y(x) \), the expected overlapping area can be expressed as:

\[ E(S_y) = \int_0^\infty f(x)\bar{S}_y(x)dx \] (4.7)

So that the quality metric, the expected number of nodes involved in detection can be expressed as:

\[ n_y = \rho \cdot E(S_y) = \rho \int_0^\infty f(x)\bar{S}_y(x)dx \] (4.8)

And the energy metric, the average energy consumption can be expressed as:

\[ E = \rho \pi R^2 E_{\text{sensor}}^d + n_y (E_{\text{elec}}k + E_{\text{amp}} k^\alpha + E_{\text{cpu}} k + E_{\text{elec}} k + E_{\text{amp}} k C^\alpha) \] (4.9)

From (4.8) and (4.9), we can see the quality metric and the energy metric are connected by two parameters \( R \) and \( s \). \( R \) determines the size of cluster and \( s \) is sensing radius. They are both important parameters in tracking protocols design.

But the pdf of \( x \), \( f(x) \) is unknown. It depends on the topology of network and the target’s motion patterns. To present the optimization model theoretically, we discuss two kinds of general distributions: uniform and normal. Later we will check the distribution of \( f(x) \) using simulation data, and will show that the optimization results are robust to the distribution of \( f(x) \) for the robustness of order.

4.2.1 Uniform distributed \( f(x) \)

From (4.3), we know that when \( x > R+s \), \( \bar{S}_y(x) = 0 \), so we assume that \( x \) is uniformly distributed on the close set \([0, R+s]\). Then the mean overlapping area can be calculated as:

\[ E(S_y) = \int_0^{R+s} f(x)\bar{S}_y(x)dx = \frac{1}{R+s} \int_0^{R+s} \bar{S}_y(x)dx = \pi Rs^2 / (R+s) \] (4.10)

So the optimization model with uniform distributed \( f(x) \) can be written as:
\[
\begin{align*}
\text{minimize} & \{ -\rho \pi R s^2 / (R + s) \} \\
\text{minimize} & \{ n_c E_{\text{sensor}} s^0 + n_s (E_{\text{elec}} k + E_{\text{amp}} k C^a + E_{\text{elec}} k + F_{\text{cpu}} k) + E_{\text{elec}} k + E_{\text{amp}} k C^a \} 
\end{align*}
\]

(4.11)

where \( n_c = \pi R^2 \rho \) and \( n_s = \rho \pi R s^2 / (R + s) \). In (4.11), minimize \{-\rho \pi R s^2 / (R + s)\} is converted from maximize \{\rho \pi R s^2 / (R + s)\}, which is the quality objective.

4.2.2 Normal distributed \( f(x) \)

For normal distributed \( f(x) \) with parameters \( [\mu, \sigma] \), we set \( \mu = (R + s) / 2 \) and \( \sigma = (R + s) / 6 \). So that the probability that \( 0 \leq x \leq R + s \) is equal to \( P(\mu - 3\sigma \leq x \leq \mu + 3\sigma) = 99.73\% \). Then the mean overlapping area can be calculated as:

\[
E(S_d) = \frac{1}{C} \left[ \frac{1}{2} \sqrt{\frac{\pi}{2}} s \sigma \left[ \exp\left(-\frac{(R + s - \mu)^2}{2\sigma^2}\right) - \exp\left(-\frac{(R - s - \mu)^2}{2\sigma^2}\right) \right] + \frac{1}{2} \pi s^2 \text{erf}\left(\frac{\mu}{\sqrt{2}\sigma}\right) - \frac{1}{4} \pi s (R - s - \mu) \text{erf}\left(\frac{R - s - \mu}{\sqrt{2}\sigma}\right) - \frac{1}{4} \pi s (R + s - \mu) \text{erf}\left(\frac{R + s - \mu}{\sqrt{2}\sigma}\right) \right] \quad (4.12)
\]

where \( C = \frac{1}{\sqrt{2\pi\sigma^2}} \int_0^\infty \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) dx = \frac{1}{2} + \frac{1}{2} \text{erf}\left(\frac{\mu}{\sqrt{2}\sigma}\right) \) is the normalization constant, and \( \text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z \exp(-t^2) dt \) is the "error function" encountered in integrating the normal distribution [13]. So the optimization model with normal distributed \( f(x) \) can be expressed as:

\[
\begin{align*}
\text{minimize} & \{ -\rho E(S_d) \} \\
\text{minimize} & \{ n_c E_{\text{sensor}} s^0 + n_s (E_{\text{elec}} k + E_{\text{amp}} k C^a + E_{\text{elec}} k + F_{\text{cpu}} k) + E_{\text{elec}} k + E_{\text{amp}} k C^a \} 
\end{align*}
\]

(4.13)

where \( n_c = \pi R^2 \rho \) and \( n_s = \rho E(S_d) \). In (4.13), we also convert the quality objective from maximize \{\rho E(S_d)\} into minimize \{-\rho E(S_d)\}.

Analyze the two optimization models. Both (4.11) and (4.13) are two-variable-two-objective optimization problems. We need to choose \( R \) and \( s \) appropriately to optimize energy metric and quality metric corporately. Because the number of clusters must be integer, the value of \( R \) can only be selected on discrete points. If the WSN is organized into \( n \) clusters, then \( R = D / \sqrt{n} \). As introduced in section 2.1, the value of \( s \) can be also selected on discrete points in the range [s_small, s_large]. It is two-variable-two-objective problem with discrete solution space. Traditional solution technique as penalty function etc. are not suitable. Because it is difficult to select the penalty parameter, assign weights or calculate differentials.

4.3 Solution of optimization models

Pareto frontier optimization technique, which is often used in industry optimization [14], is a good choice to tackle this kind of problems. For saving of space, we only choose the optimization model (4.13) to demonstrate the solution process. For (4.11), it is the same process. Since in application, we care of the number of clusters \( n \) instead of the cluster radius \( R \). To solve the optimization model, we vary \( n \) and \( s \) to see how they affect the quality and energy metrics. Table 1 is the parameter setting used in performance analysis. Figure 6 and Figure 7 show how \( n \) and \( s \) affect quality metric and energy metric respectively.

<table>
<thead>
<tr>
<th>variable</th>
<th>value</th>
<th>meaning</th>
<th>variable</th>
<th>value</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L )</td>
<td>100(m)</td>
<td>Width/length of sensing field</td>
<td>( E_{\text{elec}} )</td>
<td>5.0e-9</td>
<td>Circuitry coefficient</td>
</tr>
<tr>
<td>( N )</td>
<td>1000</td>
<td>Total number of sensors</td>
<td>( E_{\text{amp}} )</td>
<td>1.0e-10</td>
<td>Amplifier coefficient</td>
</tr>
<tr>
<td>( \rho )</td>
<td>0.1</td>
<td>Sensor density</td>
<td>( E_{\text{sensor}} )</td>
<td>1.0e-10</td>
<td>Sensing coefficient</td>
</tr>
<tr>
<td>( (x_b, y_b) )</td>
<td>(50,-100)</td>
<td>Coordination of base station</td>
<td>( F_{\text{cpu}} )</td>
<td>1.0e-8</td>
<td>CPU coefficient</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Value</td>
<td>Parameter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------</td>
<td>--------</td>
<td>-----------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E_{ini}$</td>
<td>Initial energy of sensor nodes</td>
<td>$10J$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$C$</td>
<td>Transmission radius to base</td>
<td>$200(m)$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Radio attenuation</td>
<td>$2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\beta$</td>
<td>Sensing attenuation</td>
<td>$2$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c$</td>
<td>Common transmission radius</td>
<td>$k$</td>
<td>$1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R$</td>
<td>Data packet size</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Energy consumption and tracking quality both increase with sensing radius and decrease with the number of clusters. This accords our general knowledge. But our aim is to minimize energy consumption while maximizing tracking accuracy. To solve this optimization problem, we apply Pareto optimization technique. In figure 8, every point in the figure stands for a $(n, s)$ pair. Its corresponding quality objective $-n_s(n, s)$ and energy objective $E(n, s)$ work as $x, y$ coordinates respectively. So in figure 8, every point, a $(n, s)$ pair, is a possible solution for our optimization problem. Since our aim is to maximize quality, while minimize energy consumption, the optimal solution should lie on the left-down border. The points on this border are called Pareto Fronts. In the figure, all the Pareto Fronts are found and displayed with a circle. This kind of point is defined as there is no any other point can be better than it for both metrics. Pareto Fronts are the not-bad solution candidates of the optimization model. For the contradiction of the two objectives, there is no optimal solution. We have to trade off between the two metrics. But for a given application preference, the constraint optimal can be found. Figure 9 shows an example. An application demands that at least 10 nodes can detect the target on average, while reducing energy consumption as possible. Converting into constraint optimization problem and the optimal solution is easily found and displayed with a star.

So, solving the optimization model, we can obtain the Pareto Fronts, which are not-bad solution...
candidates, and this solution technique is adaptable for different application preferences. Applying this optimization process before system implementation will effectively direct our cluster topology design and sensing radius configuration for sensors. In the next section, we will use simulation to validate our analytical results.

5 Simulation

In simulation, we use the network model introduced in section 2.1, and follow the cluster-based prediction tracking protocol as introduced in [3]. Firstly, we will demonstrate the simulation results using same parameter settings as Table 1, and compare the results with those obtained from analytical model. After that, we will simulate with different scale networks and different target motion trajectories to verify the analytical model.

In [5] [9], at the setting up phase, \( n \) cluster heads are uniformly generated, and the corresponding clusters are adaptively organized by broadcasting and coordination. After that, during the tracking phase, only one cluster is active to track the target. Sensors in this cluster keep sensing, and only the sensors that can detect the target send the detected information to the cluster head. Cluster head fuses the information to estimate the target’s position, and send it back to the base station. In order to wake up next cluster in advance, linear prediction model is used to predict target’s next position.

\[
\begin{bmatrix}
\hat{x}_{t+1} \\
\hat{y}_{t+1}
\end{bmatrix} = \begin{bmatrix}
\hat{x}_t \\
\hat{y}_t
\end{bmatrix} + \begin{bmatrix}
\hat{x}_{t-1} \\
\hat{y}_{t-1}
\end{bmatrix}
\]  

(5.1)

where \( \begin{bmatrix}
\hat{x}_t \\
\hat{y}_t
\end{bmatrix}, \begin{bmatrix}
\hat{x}_{t-1} \\
\hat{y}_{t-1}
\end{bmatrix} \) are the observed target positions at time \( t \) and \( t-1 \), and \( \begin{bmatrix}
\hat{x}_{t+1} \\
\hat{y}_{t+1}
\end{bmatrix} \) is the predicted position of target at time \( t+1 \). If the target is predicted to leave this cluster, a downstream cluster, whose cluster head is closest to the predicted position, will be waked up. And it keeps the tracking process.

5.1 Simulation results

The target motion function used in this simulation is:

\[
\begin{align*}
x(t) &= x_0 + v_x \cdot t \\
y(t) &= y_0 + v_y \cdot t + \sin(t/10) \cdot 10
\end{align*}
\]  

(5.2)

where \( x_0=0, y_0=0; v_x=1, v_y=1 \). We assume active sensors report messages every 0.5 second, and the simulation time is totally 100 s. For different \( (n, s) \) pairs, we simulate the target tracking process. The mean energy consumption and mean number of detectable sensors are illustrated in figure 10 and figure 11. In these figures, we also plot the results from the analytical model, and compare them with the simulation results.
Figure 10. Comparing of the mean number of nodes that can detect target between simulation and analyzing.

Figure 10 compares the results of quality metric. The lighter surface is the result from simulation, and the darker surface is the result from analytical model. The simulation surface isn’t smooth for the reason of the randomness of sensors’ deployment. The comparison indicates the effective of the analytical model.

Figure 11. Comparing of average energy consumption of WSN between simulation and analyzing.

Figure 11 compares the results of energy consumption. The analytical surface well approximates the simulation surface, which also indicates the effective of the analytical model.

To further compare the analytical model and the simulation result, we compare their Pareto Fronts. As introduced in section 4.3, we have obtained the Pareto Fronts from the analytical model. They are \((n, s)\) pairs, which are recognized as not-bad solution candidates. By simulation results, we can obtain a new set of Pareto Fronts. They may differ from those of analytical model, for the reason of randomness in simulation and the analyzing assumptions. To check how the analytical model can describe the real tracking process, one way is to compare how many Pareto Fronts obtained by analytical model are still Pareto Fronts in simulation. For this experiment, we compare the two sets of the Pareto Fronts. 91.11\% of the Pareto Fronts obtained from analytical model are still Pareto Fronts in simulation. Figure 12 shows the vision of the comparison. The circles are Pareto Fronts obtained from analytical model. We plot them in the simulation results. Only a small part of them are no longer Pareto Fronts, but they are still close to the border. This result shows consistency of optimization model and simulation model.

To further explore this consistency, we test the normal distribution hypotheses. Figure 13 shows the frequency histogram of the distance between the target and the active cluster head for \(n=40\) \((R=9)\) and \(s=5\). We use Kolmogorov method [15] to test the normal distribution hypotheses. The acceptable level is 0.01. The normal distribution hypotheses can be accepted limpingly. Although the distribution hypothesis is not very accurate, the analytical model can still well approximate the tracking process. Later we will discuss the reason for the robustness of the order.

5.2 More simulation results

To further verify the analytical model, we compare the simulation results and analytical results with more network settings and for different target patterns. The percent of coincident Pareto Fronts is used as a metric to measure how the analytical model works. Table 2 summarizes these results. The first row shows different target motion patterns. In the second row are different parameter settings. The last two rows are the percent of coincident Pareto fronts comparing with normal assumption and uniform assumption separately. For example as in the second column, the target moves in a linear pattern in \(x\) direction and a sine wave pattern in \(y\) direction. The network area is 100m*100m and...
sensors density is 0.05. The speed of target are 2m/s both in x and y direction. And the coincident percent of Pareto Fronts in simulation is 84.6% comparing with normal hypothesized analytical model, and 83.1% comparing with uniform hypothesized analytical model.

<table>
<thead>
<tr>
<th>Target Pattern</th>
<th>Parameters setting</th>
<th>Percent of coincident Pareto Fronts comparing with Uniform hypothesized analytical model</th>
<th>Percent of coincident Pareto Fronts comparing with Normal hypothesized analytical model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear motion</td>
<td>(100 \times 100) (\rho=0.1) (v_x=1, v_y=1)</td>
<td>95.2%</td>
<td>97.2%</td>
</tr>
<tr>
<td>Linear trajectory</td>
<td>(100 \times 100) (\rho=0.05) (v_x=2, v_y=2)</td>
<td>84.6%</td>
<td>83.1%</td>
</tr>
<tr>
<td>Linear in x and sine wave in y</td>
<td>(200 \times 200) (\text{rou}=0.03) (v_x=2, v_y=2)</td>
<td>61.4%</td>
<td>63.3%</td>
</tr>
</tbody>
</table>

From these results we can see that, analytical model can describe target tracking process more accurately, when target motion is less dynamic and sensors are more densely deployed. And it is interesting to find that for different target motions and different network settings, the two analytical models can both provide rather good solution candidates. We check the inconsistence, and find that it is mainly caused by the target missing. In the next section we will discuss on it. The uniform hypothesized analytical model works a little worse than the normal hypothesized analytical model, but its results are still acceptable.

We analyze the reason for the high consistence between the analytical models and simulation results. The Robustness of the order helps us to get good results even when the distribution hypothesis is not very good. Since the variables \((n, s)\) are discrete, an increasing from \(n\) to \(n+1\), or a decreasing from \(s\) to \(s-1\) will remarkably affect the quality and energy metrics. Although the analytical models differ a little from true application for the assumptions, they can still describe the trends accurately.

So, to direct \((n, s)\) selection in protocol design phase, we can easily apply the analytical model and obtain the optimal \((n, s)\) pair based on application preferences.

6 Advantage and Disadvantage of Cluster-Based Tracking

In the last two sections, we mainly focus on the optimization of cluster-based tracking. In this section we will discuss the advantage and disadvantage of the cluster activation.

6.1 Advantages

![Figure 14. Tracking quality comparison for different protocols](image1)

![Figure 15. Energy consumption comparison for different protocols](image2)
Cluster-based tracking use subset of active sensors to track the trace of target. By keeping most of nodes in sleeping mode, it can effectively reduce network energy consumption, and prolong system’s lifetime. To explore this advantage, we compare cluster activation to naïve activation and random activation with fairly energy metrics as defined in section 2. We assume in naïve activation and random activation, when reporting detection, nodes only consume comparable communication energy as in cluster-based tracking. So naïve activation and random activation won’t be defeated by the suffering of more long range communication. Using parameter settings as Table 1, we compare energy and quality performances for different activation schemes. In the comparison, cluster activation hasn’t been optimized. We simply fix \( n \) and increase \( s \). Figure 14 shows the results of quality comparison. Naïve activation performs best as expected. Cluster activation’s performance is between 60% sensors randomly on and 80% sensors randomly on. Figure 15 compares energy performance. Naïve activation costs the most energy. Cluster activation costs energy much less than 60% sensors randomly on. For smaller \( s \), it performs even better than 40% and 20% sensors randomly on. From these comparisons, we can see the cluster activation can provide acceptable quality performance, while effectively saving network’s energy consumption. But cluster activation is not suitable for all application. It has its disadvantages.

**Disadvantages**

Disadvantage of the cluster activation is the target missing problem. Under the cluster-based tracking scheme, only the sensors in a small area are active. If the target suddenly changes its direction or suddenly speeds up and the active cluster cannot detect it, target missing will occur. Using the network settings as Table 1, to track a target with trajectory described as (5.2), the target missing ratio is shown in figure 16. Recall figure 11. Although using smaller \( s \) and larger \( n \) consume less energy, target missing is often. Target missing phenomenon is also the main course for the inconsistency between the analytical model and the simulation results. In naïve activation, all sensors sense all the time. The target will not miss until it runs out the network’s sensing region. In random activation, active sensors are randomly distributed in the network. And the target is detected with the same probability whatever its motion patterns. So target missing is the main disadvantage of cluster activation. This disadvantage makes it not suitable for tracking highly diverse target.

![Figure 16. target missing ratio for cluster activation](image1)
![Figure 17. target missing ratio with recovery mechanisms](image2)
![Figure 18. Energy consumption with recovery mechanisms](image3)

To make up the target missing problem we propose two mechanisms: early wake-up and enlarged scope probe. In early wake-up, cluster is early switched before the target leaving the current cluster. Figure 19 demonstrates the idea of early wake-up. A tolerant distance \( \tau \) is set. When the target has reached the ring region, next cluster will be waked up. Large scope probe is a recovery mechanism, and it is first proposed in [3]. During the tracking, when the active cluster fails to locate the target, the cluster-header notifies the cluster members to change the probe scope to \( s \)-large. This will efficiently decrease the lost ratio without much energy consumption. Figure 20 illustrates the idea of large scope probe.
We apply these two mechanisms in simulation. Figure 17 shows the lost ratio for the same tracking experiment. The lost ratio is effectively decreased, which shows the effectiveness of these two mechanisms. But the performance improves with the cost of more energy. Figure 18 shows the energy consumption. Comparing with figure 11, energy consumption for small $s$ and large $n$ is increased. This is because the sensors often switch to large sensing radius to recover the target.

7 Conclusion

This paper focuses on energy-quality optimization problem for target tracking application using wireless sensor networks. We propose the quality metrics as the number of nodes involved in target detection. We prove its validity for different location methods, and by this basic understanding, we have unified the quality evaluation schemes for different tracking scenarios. Then a quality-energy optimization model is built up for cluster-based tracking protocols. For different assumptions of prediction error, the formulations and solution techniques of the optimization model are presented. We use simulation result to validate the analysis of the quality-energy models. The simulation results demonstrate the relationship of the energy-quality, with the variation of sensing range and cluster radius, which validate our analysis. Target missing problem in cluster-based tracking is analyzed and the early wake-up scheme is proposed to counter this problem.

Through simulation, we noticed that the distribution of the prediction error in cluster based tracking protocols is nearly a normal distribution with parameter $(\mu, \sigma)$, in our further study, we will analyze what the optimization model should like under this kind of error distribution. Another interesting work to do is the design of a practically usable protocol. Though many efforts have been focused in target tracking, many details (eg. network initialization) are neglected in analysis. To make the cluster-based scheme practical, these issues must be tackled.

Reference

Appendix:

Proof of the limited variance of the estimation in mechanism 2:

Suppose the three sensors in one group are at \((x_1, y_1), (x_2, y_2), (x_3, y_3)\) respectively and are not on the same line. The object is at \((x, y)\), and is estimated to be at \((x_e, y_e)\). For each sensor, the object is estimated at a distance of \(d_i\) to the sensor. And the actual location of the target is at a ring with the inner radius \(d_i - \delta/2\) and the outer radius \(d_i + \delta/2\). Let \(\theta_i\) denotes the angular between the x axis and vector \((x_i - x_0, y_i - y_0)\). Since the position of the object is at the overlay area of the three rings, we have:

\[
|x_0 - x_e| \leq \delta \cos \theta_1 + \delta \cos \theta_2 + \delta \cos \theta_3 < 3\delta
\]

and

\[
|y_0 - y_e| \leq \delta \sin \theta_1 + \delta \sin \theta_2 + \delta \sin \theta_3 < 3\delta
\]

which indicates the estimation \((x_e, y_e)\) is limited in variance.