

# Prediction-based Strategies for Energy Saving in Object Tracking Sensor Networks

Yingqi Xu Julian Winter Wang-Chien Lee  
Department of Computer Science and Engineering  
Pennsylvania State University  
University Park, PA 16802  
E-Mail: {yixu, jwinter, wlee}@cse.psu.edu

## Abstract

*In order to fully realize the potential of sensor networks, energy awareness should be incorporated into every stage of the network design and operation. In this paper, we address the energy management issue in a sensor network killer application - object tracking sensor networks (OTSNs). Based on the fact that the movements of the tracked objects are sometimes predictable, we propose a Prediction-based Energy Saving scheme, called PES, to reduce the energy consumption for object tracking under acceptable conditions. We compare PES against the basic schemes we proposed in the paper to explore the conditions under which PES is most desired. We also test the effect of some parameters related to the system workload, object moving behavior and sensing operations on PES through extensive simulation. Our results show that PES can save significant energy under various conditions.*

## 1. Introduction

The emerging technologies in low-power micro-sensors, actuators, embedded processors, and RF radios have facilitated the deployment of large scale sensor networks. Due to their low cost and capabilities for pervasive surveillance, sensor networks and their applications have tremendous potential in both commercial and military environments. However, in addition to the inherited limitations of the sensor nodes such as scarce power resources, highly distributed operations and unstable wireless communication, the specific requirements of applications may bring additional research challenges and issues to the design of sensor networks. In this paper, we consider one of the sensor network killer applications - *object tracking*.

Among the technical issues to be addressed in developing sensor networks for object tracking, *energy conservation* is probably the most critical one since the sensor nodes are often supported by batteries which could be difficult to replace. A lot of existing researches are focussed on optimizing the communication cost by inactivating radios as

much as possible or by trading off computation for communication [3, 4, 5, 8]. However, these studies neglected a fact that, while the sensing and computing components consume less power than the communication components, they are still important sources of energy dissipation in sensor nodes, especially after the communication cost being optimized. Thus, in this paper, we study the problem of how to reduce the energy consumption in the sensing and computing components of sensor nodes.

We define the problem and the requirements of object tracking applications and develop some basic energy-saving solutions which reduce the number of sensor nodes needed for monitoring the moving objects or decrease the frequency of sampling the environment, respectively. Moreover, we discuss other possible solutions by illustrating a solution space. Based on our analysis and observations, we propose a *Prediction-based Energy Saving scheme*, called PES, that minimizes the number of nodes participating in the tracking activities, while inactivates other nodes into sleeping mode. Simulation based performance evaluation, in terms of *total energy consumption* and *missing rate*, has been conducted. The simulation result shows that the PES can effectively reduce the energy consumption on MCU and sensor components. Moreover, the different heuristics discussed in this paper can be used to balance the energy savings and application requirements.

This paper makes three significant contributions.

- We present and compare several basic energy saving schemes for object tracking sensor networks, and point out the direction for designing an energy-aware OTSN.
- We propose and architect a prediction-based energy saving scheme (i.e., PES) which consists of prediction models, wake up mechanisms and recovery mechanisms. PES activates only the essential sensor nodes needed to track the moving objects, but hibernates other nodes into low-power mode.

- We conducted an extensive performance evaluation by simulating the OTSN and various energy saving schemes we proposed. The study provides profound insights for designing an energy-aware OTSN.

There are many related works that address energy efficiency in sensor networks from various angles. In [6, 16, 22], the analysis of power consumption on sensor nodes provides an important heuristic for power optimization in various areas of the sensor network design. Energy awareness is also studied in the MAC layers, such as TDMA and CDMA protocols [2, 14, 21]. *Low-power paging channel*, employed in this paper as part of the OTSN communication protocols, is widely discussed in [15, 16, 25]. In the network architecture level, the clustering based approach and localized algorithm have been proposed to dynamically cluster the sensor nodes based on their remaining battery level and density of the region [3, 5]. Query processing and data aggregation also provides many energy optimization opportunities that were extensively explored in literatures [12, 13, 20].

The ideas of utilizing predictions to reduce overhead in mobile computing systems have appeared in the literature. Prediction based techniques in cellular network reduce the paging overhead by limiting search space to a set of cells that mobile users may enter [1, 23]. In wireless data broadcast protocol, mobile computers turn on the radio only during the arrival time of the requested data frames, which is predicted based on the indexing information in the broadcast channel [7, 10, 19]. Similarly in sensor networks, the future reading at a sensor is predicted, given the past reading history and the spatial and temporal knowledge of readings from surrounding sensors [4]. In our previous work [24], we also studied a localized network architecture and a dual prediction mechanism, which achieve power savings by allowing most of the sensor nodes stay in sleep mode and by reducing the amount of long-range transmissions.

The rest of the paper is organized as follows: Section 2 provides a background and the assumptions for this paper, and also discusses the system factors that are contribute to the network designs. In Section 3, we clearly define the requirements of OTSN applications and explore some basic schemes which represent the different philosophies for optimizing energy usage in OTSNs. Our prediction-based energy saving schemes are presented in Section 4. Section 5 provides the simulation results; and finally, Section 6 gives the concluding remarks and points out the future work.

## 2. Object Tracking Sensor Networks

Object tracking is considered as one of the killer applications for sensor networks. There are a lot of research issues in design and implementation of the object tracking sensor networks (OTSNs), including data fusion, aggregation,

routing, and energy conservation, etc. Among those, energy conservation is one of the most critical one. Like other sensor networks, the OTSN is driven by scarce energy resource. Therefore, energy saving is the major issue addressed in this paper. In the following, we first provide some background of the OTSNs, describe the assumptions we made in this paper, and discuss the factors that contribute to energy consumption and design complexity of the OTSNs.

### 2.1. Background

In an OTSN, a number of sensor nodes are deployed over an area, called *monitored region*. The approximate geographical boundaries of the monitored region are known to the applications who retrieve the information of interests (such as location, speed, direction, size, and shape) of a tracked moving object. Base station or gateway acts as the interface between the OTSN and applications by issuing the command to the network and collecting the information of interests from the distributed sensor nodes. In this paper, we assume that the sensor nodes are *static* and that a base station has good knowledge of the network topology (in terms of the location of each sensor node) during the operating period. The sensor nodes are enabled for computation, sensing and communication by the Micro-Controller Unit (MCU), sensor components and the RF radio component respectively. To facilitate the energy conservation, most of today's sensor nodes allow these three basic components to be inactivated separately when they are not needed.

These sensor nodes have the responsibility for tracking any moving object which intrudes the monitored region, and reporting the properties of the moving objects to the applications in a specified frequency. Deciding the location, speed, and direction of a moving object needs several sensor nodes to work together, which may require hierarchical technologies and overlapping levels of sensing (this is called sensor fusion). This is an important area of research in the sensor applications, but out of scope of our study. Thus, in this paper, we assume that each sensor node is a *logical* representation of a set of sensor nodes which collaboratively decide the properties of a moving object. In other words, the sensor nodes referred in this paper are the *sensing leaders* or *cluster heads* in a multiple level sensor network. We also assume that the moving objects are identifiable. The objects are electronically tagged or can be identified based on the pre-embedded *object code table* in the sensor nodes, which classifies all the objects such as jaguar, elephant and pedestrians. An unique object\_ID is assigned to each tracked object.

The sensor nodes sample the physical world for a *sampling duration* to obtain the properties of moving objects. During sampling, the MCU and the sensor components are activated for data collecting and processing, but the ra-

dio components can be turned off if no communication is needed. The sampling happens with certain *sampling frequency* adjustable based on the network and application requirements. The sensor nodes which detect the object in their *detection area* have to report to the base station with certain *reporting frequency* which is specified as the application requirements. Otherwise, the sensor nodes do not need to turn on the radio, but keep silent. The reporting frequency decides the rate with which the application receives information regarding to the moving objects from the sensor networks; thus results in different levels of precision for the objects' movement.

Finally, the information exchanges between a sensor node and the base station are based on multi-hop communications. In addition, we assume that a low-energy paging channel exists for a sensor node to wake up some other sensor nodes in sleep. Furthermore, we assume that all the sensor nodes in OTSN are well-synchronized with the applications.

## 2.2. Impacting factors

Like other sensor network applications, OTSN has unattended nodes, wireless transmissions and distributed system architecture. In addition, there are a lot of unique factors to be considered when implementing OTSNs. In this section, we examine those factors that impact the energy consumptions of the OTSNs.

- **Number of moving objects** represents the workload of the networks, which directly affects the energy consumption of the entire network. More moving objects inside the monitored region increase the total number of samplings and reporting.
- **Reporting frequency** is decided based on the application requirements. A higher reporting frequency is set when an application needs timely updates about the moving objects. Otherwise, keeping the reporting frequency low can reduce the number of transmissions, and thus increases the lifetime of the OTSN.
- **Data precision** is closely related with the sampling frequency, the sampling duration and the location models (see below). Therefore, a higher data precision requires more data collection, more intricate computation and larger update packets, which result in more energy consumption on sensing, computing and communication.
- **Sensor sampling frequency** is an important factor for object tracking. An OTSN with low sampling frequency may lose track of objects, but the one with high sampling frequency incurs more energy consumptions.
- **Object moving speed** is relevant to the sensor sampling frequency. To keep track of the object, an OTSN

needs to sample more frequently on an object which moves in high speed.

- **Location models** are used to represent the location information, which is one of the most important information to be obtained at sensor nodes. Other movement information, such as speed and directions, can be derived from the changes of locations and elapsing time. Based on the location identification techniques employed in the system, location model can be categorized as *geometric* model and *symbolic* model [9]. The appropriate location models to be adopted depend on the application requirements.

## 3. Problem Formulation

In this section, we first define the application requirements for object tracking and examine two performance criteria. Then, we investigate some energy saving schemes to meet these requirements and criteria set for our study. All the schemes we discuss below do not consider the energy overhead for transiting between operating modes of all the components on the sensor nodes. Unless specified explicitly, a sensor node is activated when both of MCU and sensor components are in active mode. The operations of radio components is discussed later.

### 3.1. Application requirements and metrics

Here we define the basic application requirements and specify several parameters needed for energy saving schemes. The information of interests regarding to the moving objects includes location, direction, speed, and some unique attributes that can identify the object, such as size, shape, etc.

**Requirements:** *A sensor network with  $S$  sensor nodes is equipped to track  $O$  moving objects. Each sampling duration takes  $X$  seconds. The application requires the sensor nodes to report the objects location (represented by `Sensor_ID`) every  $T$  seconds.*

Based on the above requirements of object tracking applications, the problem to be addressed in this paper can be defined as follows.

**Problem Definition:** *Given the requirements for the object tracking application, develop energy saving schemes which minimize overall energy consumption of the OTSN under an acceptable missing rate.*

In the above, two performance metrics are implicitly given. **Total energy consumption** measures the performance of various energy saving schemes for OTSN, while

the **missing rate** denotes the ratio of sensor node failing to report on time the required information about the moving objects to the application. These two metrics will be formally defined again in Performance Evaluation (see Section 5).

### 3.2. Basic schemes

Based on application requirements, we first introduce some basic energy saving schemes for OTSNs.

**Naive:** In this scheme, all the sensor nodes stays in active mode to monitor their detection areas all the time. As such, the objects in the network are always tracked and their locations are reported to the base station (every  $T$  seconds) by the nodes who have the objects in their detection areas. This scheme, as its name suggested, is not tunable to the application requirements for energy saving. We introduce it here to illustrate the principles used in other energy saving schemes and later use it as a baseline for comparisons with other schemes.

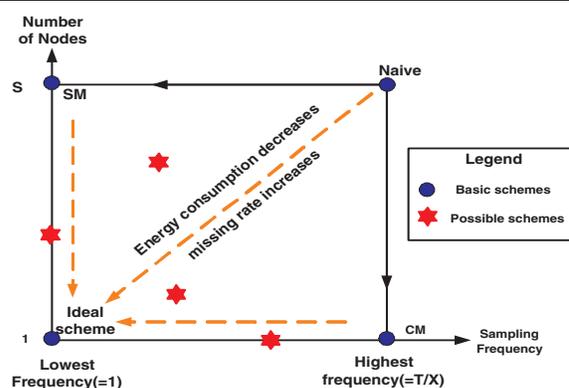
**Scheduled Monitoring (SM):** This scheme addresses the fact that the application does not request the network to report the sensed data all the time. Assuming that all the sensor nodes and base station are well synchronized, all the sensor nodes can turn to sleep and only wake up when it's time to monitor their detection areas and report sensed results. Thus, in this scheme, all the  $S$  nodes will be activated for  $X$  second then go to sleep for  $(T - X)$  seconds. This scheduled monitoring process will be continued for the entire network operation period. The advantage of this scheme is that the sensor nodes spend minimal time in active mode and stay in sleep mode as long as they can. Hence, a significant amount of energy is saved if the applications do not need frequent reports from the network. However, in order to capture the moving objects (i.e., to ensure no missing report), the number of sensor nodes involved in object tracking is more than needed.

**Continuous Monitoring (CM):** This scheme exploits another aspect of energy saving in OTSNs. Instead of having all the sensor nodes in the field wake up periodically to sense the whole area, only the sensor node who has the object in its detection area will be activated. An awake node actively monitors the object until the object enters a neighboring cell. It may wake up the destination node (hand-off)  $W$  seconds before object enters.  $W$ , depending on the transmission rate and object moving speed, is typically very small. Thus, the handoff logically happens when an object reaches the detection area boundary. In this way, the sensor nodes trespassed by moving objects collaboratively monitor the object, and report the locations of the moving object to the base station at the scheduled reporting time. The advantage of this scheme is that it involves only *one* sensor node to monitor each moving object while other sensor

nodes can turn to sleep and save energy. However, to ensure no missing report, the active sensor has to stay awake while there exists an object in its detection area.

### 3.3. Solution space

The above three schemes maintain a 0% missing rate at the cost of extensive energy consumption by keeping MCU and the sensor components unnecessarily active. While the SM and CM schemes improve the naive scheme, they represent different philosophies to achieve energy savings in OTSNs. We envisage that there are other energy saving schemes that can take strengths of both camps and perform better. Figure 1 shows a solution space of energy saving schemes for OTSNs.



**Figure 1. Solution space of energy saving schemes for OTSNs**

In the figure, the x-axis denotes the sampling frequency for each sensor node and y-axis denotes the number of sensor nodes who are activated for objects tracking. All the possible energy saving schemes for OTSNs are restricted within the rectangular area. As illustrated, the naive scheme becomes the SM scheme if we continue to reduce the sampling frequency until it reaches the minimal requirement<sup>1</sup>. On the other hand, the naive scheme becomes the CM scheme if we reduce the number of sensor nodes involved in monitoring to one. Thus, if we reduce both of the sampling frequency and the number of activated nodes to their minimal bounds, we may have an *ideal scheme* which represents the optimal solution we can obtain in this solution

<sup>1</sup> The minimum and maximum sampling frequencies for every  $T$  seconds are 1 and  $\frac{T}{X}$ , respectively. Maximum sampling implies that a sensor node is awake to watch its detection area all the time.

Schemes	Nodes Involved	Continuous?	Energy Consumption
Naive	All(=S)	Yes	$E_{wake} \times TS \times S$
SM	All(=S)	No	$((E_{wake} \times X + E_{sleep} \times (T - X)) \times \frac{TS}{T} \times S$
CM	One for each object	Yes	$E_{wake} \times TS + E_{sleep} \times TS \times (S - 1)$
Ideal	One for each object	No	$E_{wake} \times \frac{TS}{T} \times X + E_{sleep} \times (TS \times S - \frac{TS}{T} \times X)$

**Table 1. Analytical evaluation for energy saving schemes.**

space<sup>2</sup>. There may be many other possible energy saving schemes existing in this solution space (denoted by stars). Now the question is whether there exists such an optimal scheme and how to obtain it? The ideal scheme shows that for each moving object, only one sensor node needs to be woken to monitor. However it requires that node to be activated at the *right time* (when a sensing period starts) and at the *right place* (where the moving object is located). This can only be achieved when the sensor nodes (hence the network) have the ability to predict every single movement of the object. If this is not achievable, can we obtain a near-optimal energy saving scheme based on some accurate predictions? Thus, the problem we have becomes whether we can obtain an energy saving scheme by trading off missing rate (introduced by the less perfect predictions). Based on this analysis, in Section 4, we propose a prediction-based energy saving scheme.

### 3.4. Analysis and comparisons

Properties and analytical models of the basic schemes are presented in Table 1. Based on the description of the basic schemes and later experiments, we find that the basic schemes vary on different ways of monitoring the physical world, but not on how they communicate with the base station. Thus, to manifest the differences of the schemes, in the analytical models we only take into account the energy cost for MCU and sensor components but not the communication cost. In the model,  $E_{wake}$  denotes the energy consumption rate (per second) at a sensor node by having both MCU and sensor components stay in active mode and  $E_{Sleep}$  denotes the energy consumption rate in sleeping mode. We assume that the OTSN has operated continuously for  $TS$  seconds. In all those basic schemes, the missing rate is 0%.

The analysis suggests that naive is the most expensive scheme in terms of energy consumption, since all the sensor nodes keep awake all the time even though transmission only happens  $\frac{TS}{T}$  times. The ideal scheme is the most preferred one since the energy consumption in sleep mode for

MCU and sensor components is much smaller than when they are in active mode.

## 4. Prediction-based Energy Saving Schemes

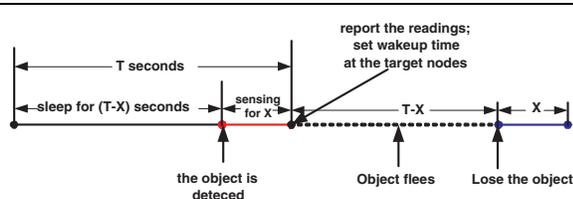
Based on our analysis of the basic and ideal schemes, we propose a *Prediction-based Energy Saving scheme* (PES) for OTSNs. To optimize the energy consumption in OTSN, PES tries to approach to the ideal scheme by minimizing both of the sampling frequency and the number of nodes involved in object tracking, while balances off the overhead caused by missing the objects. PES consists of three parts: 1) a *prediction model* which anticipates the future movement of an object so only the sensor nodes expected to discover the object will be activated; 2) a *wake up mechanism* that, based on some heuristics taking both energy and performance into accounts, sets up which nodes and when they should be activated; 3) a *recovery mechanism* initiated only when the network loses the track of an object.

The basic idea of PES is that a sensor node not performing the duty of object tracking (i.e., there is no object in its detection area) should stay in sleeping mode as long as possible. Meanwhile, a sensor node which has a moving object in its territory, called *current node*, should also try to turn to sleep mode as much as it can. Thus, based on the prediction model used, the current node will predict the possible location(s) of the moving object and determine a group of sensor node(s), called *target nodes*, to help tracking the moving object after certain period of sleeping. The target nodes are chosen depending on some heuristics which will be discussed when we introduce the wake up mechanism. The current node inactivates itself after sending the wake up call to the target nodes.

Based on what we described above, a current node performs sensing for  $X$  seconds and reports the sensed data to the base station. Before going back to sleep, however, the current node predicts the object movement for the next  $(T - X)$  seconds and informs the target nodes. After sleeping for  $(T - X)$  seconds, all the target nodes and the current node wake up together to track the object. The new current node will repeat the above described process, while the other nodes can go to sleep.

The current nodes need to predict the objects' future movement and seek help from the target nodes, because the

<sup>2</sup> The readers should note that the solution space is a logical one. The visually observed distances from two different schemes to the ideal scheme may not accurately reflect their performance closeness to the ideal scheme.



**Figure 2. The current node misses the object**

objects may move out of their detection areas. As shown in Figure 2, an object may flee the detection area of a current node while it is sleeping. When the current node wakes up again, it can not detect the object anymore. Moreover, all the non-current nodes are in the sleep mode and will not turn themselves on until being notified. In this case, the application is failed to receive any report about the object even though the object exists inside the network. Therefore, two primary issues are 1) how to reduce the missing of objects and 2) how to re-locate the object if it happens.

An intuitive solution to the first question of reducing missing rate is to seek helps from other nodes. Since the target nodes are already informed, when the current node goes back to active mode, the target nodes wake up as well to help tracking the object. In this way, even if the current sensor node lose track of the object, hopefully, one of the target nodes is able to find the object in its detection area. Once a target node finds the object, it sends an *acknowledge* message back to the current node. After receiving the acknowledge message from any of the target node within certain time, the current node becomes a non-current node and goes back to sleep if no other objects stay in its detection area. Otherwise, an object missing is confirmed and the recovery process has to be started. Therefore the question comes to which neighboring node should be woken up for collaborative monitoring.

#### 4.1. Prediction model

Prediction results about the movement and destination of the moving objects directly affect the effectiveness of choosing target nodes. If the prediction is accurate, less target nodes need to be activated for collaborative tracking and less overhead is incurred.

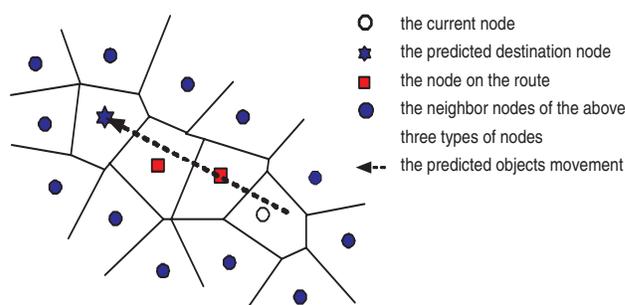
In this paper, we propose a simple prediction model based on an observation that object movement usually remain *constant* for a certain period of time. With this assumption, the current node will predict the movement of the object in the following  $(T - X)$  seconds and the sensor node where the object eventually arrives (called *destination node*), without considering the variance of the speed and direction during this period of time. Therefore the speed and direction used for estimation makes difference for the pre-

diction results. In the following we described three heuristics for selecting the speed and direction used by the prediction models:

- **Heuristics INSTANT.** Based on this heuristic, the current node assumes that moving objects will stay in the current speed and direction for the next  $(T - X)$  seconds. It is a very simple and energy efficient heuristic, since the current node doesn't need other objects movement history for the predictions.
- **Heuristics AVERAGE.** By recording and passing some moving history, the current node derives the object's speed and direction for the next  $(T - X)$  seconds from the average of the object movement history. Thus, heuristics AVERAGE incurs some communication overhead because of the movement history passing among sensor nodes. And the overhead is controlled by the size of the history or the number of the past states recorded.
- **Heuristics EXP\_AVG.** Instead of simply averaging the history, EXP\_AVG assigns different weights to the different stages of history. Compared to AVERAGE, EXP\_AVG is not only able to control the weights to the history which may reflect the objects' future movement, but also compress the history information into a value, thus reducing the transmission overhead.

#### 4.2. Wake up mechanisms

No matter what kind of heuristics we use, we do not expect the prediction to achieve 100% accuracy. In this case, prediction errors means object missing, which causes excessive energy overhead for re-locating the object. To accommodate the prediction errors, a set of target nodes are woken up to help capturing the object, instead of only one destination node.



**Figure 3. Heuristics for wake up mechanisms**

We propose a wake up mechanism that decides the membership of the target nodes based on the different levels of conservativeness. Figure 3 depicts different heuristics used for the wake up mechanism.

- **Heuristic DESTINATION.** The current node only informs the destination node. Hence, the overhead for this heuristic is one node, but with the higher probability of losing the object.
- **Heuristic ROUTE.** In addition to the predicted destination, the target nodes also include the nodes on the route from the current node to the destination node. This heuristic assumes the direction is estimated correctly but leaves a room for errors in speed estimation. Therefore waking up the nodes on the route can efficiently catch up the speed changes.
- **Heuristic ALL\_NBR.** In addition to the nodes on the route and the destination node, the current node also informs the neighboring nodes surrounding the route, current node and the destination. This heuristic assumes the prediction of the moving speed and direction will have observable difference from the actual values. Thus, waking up more nodes reduce the probability of objects missing.

Among the heuristics discussed, All\_NBR is the most conservative one since it assumes the predictions for both of the speed and directions may have errors. Thus, it wakes up all the nodes and neighbors along the predicted route to make up the deficiency. Heuristics ROUTES and DESTINATION are more energy-efficient since they assume at least one of the estimations for moving direction or speed is correct.

### 4.3. Recovery mechanisms

No matter what heuristic used in prediction mechanisms and wake up mechanisms, we are not able to guarantee 0% missing rate, except for waking up all the sensor nodes each time as the SM scheme does. Therefore, PES needs a recovery mechanism to relocate the object, when the object is not found by the current and target nodes.

To be conservative to the energy resource, upon the object miss, the current node first wakes up all the nodes surrounding the estimated route of the moving objects by using the heuristic ALL\_NBR described in Section 4.2<sup>3</sup>. As the ALL\_NBR heuristic in wake up mechanism, the ALL\_NBR recovery does not guarantee the activated nodes can find the missing object. Therefore a more aggressive recovery approach takes place as the second step. In case that ALL\_NBR recovery fails, the current node will initiate

<sup>3</sup> For the PES with ALL\_NBR heuristic, it enters directly into the second stage of the recovery described below.

*flooding recovery* which wakes up all the nodes in the network for object relocation, which ensures 0% missing rate.

The *recovery message* used as wake-up call in both recovery stages is transmitted via ultra low energy paging channel with reasonable communication overhead. One of the information encapsulated in the recovery message is the *activate time*, which regulates the time when all the sensor nodes should wake up to capture the missing object. When the activate time arrives, the nodes receiving the recovery message wake up together and try to find the object. In ALL\_NBR recovery the sensor node who captures the missing object has to notify the current node, thus preventing the second step of the recovery. The notification is not needed in flooding recovery, since at least one sensor can locate the missing object as long as the object is still inside the monitored region.

## 5. Performance Evaluation

In this section we evaluate the performance of PES scheme with various heuristics through simulation. We first describe the parameter settings and performance metrics used in the simulation; then test the sensitivity of PES scheme to three sets of parameters, including *workload*, *moving behavior*, and *sensing operations*. To better understand the timing of using PES and the obtainable energy savings, we also implement and compare the three basic schemes, i.e., naive, scheduled monitoring (SM), and continuous monitoring (CM).

### 5.1. Metrics and settings

We implement a simulator of OTSN and various energy saving schemes using CSIM [18]. We use the shortest path multi-hop routing algorithm for communications between the base station and sensor nodes and adopt energy consumption for WINS sensor nodes [17, 22] as the basis for our simulation (shown in table 2). Two metrics are used for our performance evaluation:

- **Total energy consumption:** the total energy consumed in both active and sleeping modes by the network during the simulated period;
- **Missing rate:** the ratio of the number of missed reports to the total number of reports the application is supposed to receive from the OTSN.

These two metrics measure the energy consumptions of various energy saving schemes and their effectiveness to meet the application requirements.

Our simulation models a OTSN of 95 logical sensor nodes in a  $120 \times 120 \text{meter}^2$  monitored region. We assume the sensing coverage range is 15m. The network is based on a hexagon topology [24], i.e., sensor nodes are

evenly placed in the area such that each sensor node has a hexagon-shaped detection area with exact six neighbor nodes. To model moving behavior of the tracked objects, the Gauss-Markov mobility model [11] is used (with a default  $speed = 5m/s$ ,  $pausetime = 600ms$ , and  $\alpha = 0.8$ )<sup>4</sup>.

Component	Mode	Energy Consumption(mW)
MCU	Activate	360
MCU	Sleep	0.9
Sensor	Activate	23
Radio	Transmission	720
Radio	Receiving	369

**Table 2. Energy consumption on WINS nodes.**

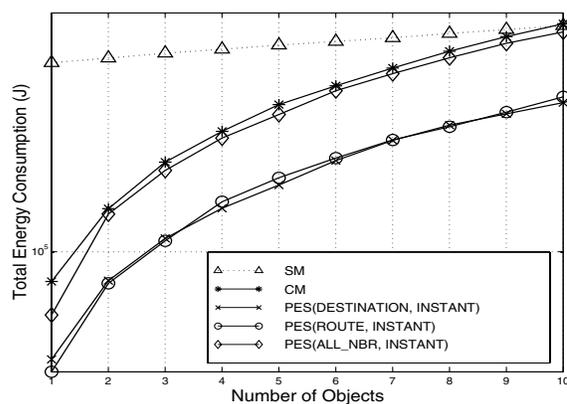
The workload of the network is captured by the number of moving objects at any point of the simulation. For sensing operations, we assume that the sensor nodes need to sample the environment for 100ms to obtain stable data and that the sampling frequency should be greater than the reporting frequency. In addition, we assume that the application requires the OTSN to report the locations of tracked objects (represented by the sensor\_ID) every second. For the prediction model EXP\_AVG, history weight = 0.8. The average size of the data packet sent to the base station is 48 bits; the wake up packet is 40 bits; and the prediction packet used for the heuristic EXP\_AVG is 64 bits. We run 25 trials to obtain the average result, where each trial simulates 120 second of object tracking activities. We assume there is no congestion or transmission conflict in the network. Exploring the behavior of PES under congestion and transmission conflict is the subject of our future research.

## 5.2. Impact of the network workload

Our first experiment (see Figure 4) compares PES scheme (with three combinations of heuristics) to the basic schemes (i.e., SM and CM) in the simulated OTSN by varying the total number of moving objects from 1 to 10 to generate the increased workload to the network. While not shown in the figure, the naive scheme is simulated to serve as a baseline for energy savings of all the schemes. If a scheme does not consume significantly lower energy than the naive scheme, it is not considered as an enhancement for OTSNs. Our experiment result shows that all the schemes evaluated save at least 90% of the energy consumed by naive scheme when there is one object

<sup>4</sup>  $\alpha$  is the tuning parameter used to control the randomness of the next move. Total randomness is obtained by setting  $\alpha = 0$ , while linear motion is obtained by setting  $\alpha = 1$ .

in the area. Thus, the curve for the naive scheme is omitted here and in all the other figures for the clarity of presentation. In addition, we use logarithmical scale for Y-axis (i.e., total energy consumption) in Figure 4. We did simulate the heuristic EXP\_AVG for prediction models with three wake up heuristics (i.e., heuristic DESTINATION, ROUTE and ALL\_NBR) for this experiment. However, the results show that there is no noticeable effect of network workload on the prediction models, thus we only show the prediction model with heuristic INSTANT. Furthermore, missing rate was also studied in this experiment. The results match our instinct that the missing rate for ALL\_NBR is much lower than ROUTE and DESTINATION, but the metric does not affect by the network workload at all. Therefore, we omit it here due to the limited space.



**Figure 4. Impact of the network workload**

CM and SM schemes are two basic schemes evaluated for comparisons. PES is not expected to be the scheme of choice under all the conditions. Thus, exploring the situation where the PES is more efficient than these basic schemes is an important exercise for network planning and deployment. Our simulation result verifies that CM, SM and PES dissipate a very small amount of energy. They achieve such energy saving over the naive scheme by not keeping all the sensor nodes awake all the time. The Figure 4 also indicates that SM consumes more energy than CM and PES when the number of objects is small. However, the difference shrinks radically when more objects are in the network. Since SM activates all the sensor nodes periodically, its energy consumption is not affected much by the number of moving objects. The only extra cost for SM to monitor more objects in the network is the additional transmission overhead to the base station. Thus, SM is the scheme of choice if there are a lot of moving objects in the net-

work. For CM and PES schemes, increasing the number of moving objects not only raises communication (transmissions and receiving) energy but also the energy consumed in the MCU and sensor components. The simulation result shows that when the total number of objects is small, the total energy consumption increases linearly with the increasing number of objects (note that the y-axis is in logarithmical scale). Thus, for the rest of experiments, we only consider one moving object for simplicity.

The energy consumptions of PES with heuristics INSTANT for prediction models and three wake up groups of nodes (i.e., DESTINATION, ROUTE, and ALL\_NBR) are illustrated. The PES curves for heuristic ALL\_NBR spend almost the same energy as CM, but the one with the heuristic ROUTE and DESTINATION dissipate approximately 45% energy of CM by cutting off the number of target nodes. The reason that there is no big difference between heuristic ROUTE and DESTINATION is because that heuristic DESTINATION incurs more recovery overhead due to object misses, which balances off the wake up overhead in the heuristic ROUTE.

### 5.3. Impact of moving behavior

To observe the impact of moving behavior of tracked objects on energy consumptions and missing rate of PES scheme, we vary two important control parameters for our simulation, the pause time and the average speed of the moving objects. The pause time controls the frequency a moving object changing its state in terms of speed and direction, while the average speed control an object's possible moving range. The average direction change, also controllable in our simulation, is fixed to simplify our study.

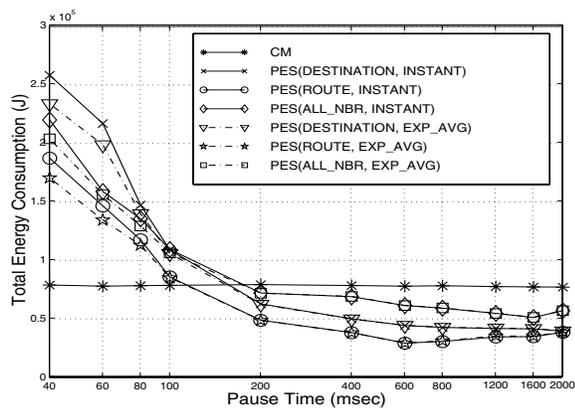
A small pause time makes the moving pattern of an object dynamic and thus difficult to predict. On the other hand, a moving object with long pause time keeps constant movement state longer and thus is easier to predict. Since the CM scheme significantly outperforms the SM scheme when there is a limited number of objects in the network, we use the total energy consumption of CM as the watermark to compare the PES scheme with various heuristics. In the remaining experiments, we only include CM for comparison.

**5.3.1. Study of the pause time.** Figure 5(a) plots the energy consumption as the pause time varies from 40 to 2000ms. The pause time in x-axis is logarithmical scaled for clarity. As shown in the figure, CM outperforms PES (with all combinations of different heuristics) when the moving object changes its speed and direction very frequently (e.g., when pause time is below 200ms). Also, as shown in Figure 5(b), the corresponding missing rates of PES are high. Thus, we can easily conclude that a moving object should be continuously tracked using the CM scheme when its moving behavior is highly dynamic and unpredictable.

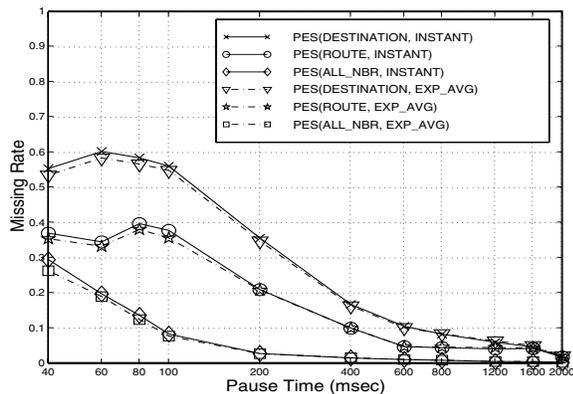
On the other hand, observed from the above two figures, the energy consumptions and missing rates of the PES scheme dramatically decrease (which can be observed in regularly scaled figures) as the pause time increases. When the pause time equals to 200ms (which still represents a pretty dynamic moving behavior), the PES scheme employing ALL\_NBR heuristic in the wakeup mechanism outperforms the CM scheme while maintaining a very low missing rate (below 2.5%). When the pause time further increases, the PES scheme with heuristics DESTINATION and ROUTE both lower their missing rates with even more energy savings. When the pause time is 800ms, heuristic ROUTE saves more than 60% energy over the CM scheme with less than 5% of missing rate. Thus, we can conclude that, a PES scheme (based on application requirements) can be used to replace the CM scheme to achieve a considerable energy saving when the moving behavior of a tracked object is not drastically dynamic.

Figure 5(a) and Figure 5(b) also compare the PES schemes with both of INSTANT and EXP\_AVG prediction heuristics. Basically, the different heuristics for wakeup mechanism have a much higher impact on performance than the heuristics for prediction since we can see that the PES curves are paired based on the wakeup heuristics. Even so, we can still observe that the EXP\_AVG outperforms INSTANT. This is particularly visible when the moving behavior of the object is highly dynamic. However, the difference quickly disappears when the pause time increases. Since the CM is the scheme of choice for a highly dynamic moving object and there is no noticeable performance difference between EXP\_AVG and INSTANT, we will not show performance result of the PES scheme corresponding to EXP\_AVG model for the rest of experiments (where the moving behavior of object is set to be reasonably dynamic to explore the impacts of other factors).

**5.3.2. Study of the moving speed.** Figure 6 shows that the impact of moving speed on energy consumption for scheme CM and PES with INSTANT prediction heuristic. We do have the result for the missing rate with various moving speed as well but omit it due to the limited space. The figure shows that the energy consumption increases linearly as the objects' moving speed increases. In addition, the missing rate also increases linearly (not shown here). This is because the faster an object moves, the more target nodes need to be woken up (for ROUTE and ALL\_NBR). In addition, the destination node is farther away from the current node and thus more difficult to predict. As shown in the figure, when the speed is under 25 m/second, all three of wake up heuristics are able to achieve energy savings compared with CM. Among the three PES schemes, the ALL\_NBR has a higher energy consumption but a much lower missing rate than the other two. ROUTE outperforms DESTI-



(a) Total Energy consumption



(b) Missing Rate

Figure 5. Impact of the pause time

NATION in terms of both energy consumption and missing rate. This is because the overhead incurred by waking up more (target) nodes in heuristic ROUTE is still less than the energy overhead for relocating the object in heuristic DESTINATION. This fact indicates that ROUTE is a good choice for monitoring high-speed moving objects by balancing energy consumption and object missing.

## 5.4. Impact of sensing operations

In addition to the moving behavior of tracked objects, we also study the impact of sensing components on energy saving schemes for OTSNs. We vary two control parameters, i.e., sampling duration and sampling frequency, to observe their effect on total energy consumption and missing

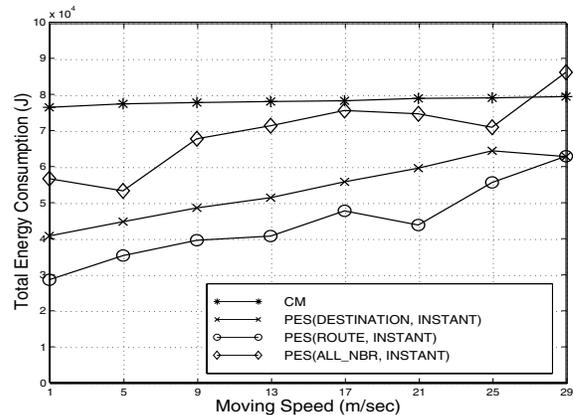


Figure 6. Impact of the moving speed

rate. Due to the space constraint, the figure of missing rate is not shown.

**5.4.1. Study of the sampling duration.** The sampling duration needed to obtain required data for a moving object directly decides the energy consumption by MCU and sensing components. As shown in Figure 7, the increase of sampling duration has a noticeable effect on PES with various wakeup heuristics, but not on CM. Meanwhile, the missing rates (not shown here) are low and decreasing linearly when the sampling duration increases. For PES, the longer a sensor node samples its detection area, the shorter it can turn to sleep for energy saving. On the other hand, this also implies a shorter period of uncertainty in estimating the objects' next movement. Thus, sampling duration increase has an effect on raising the sensing operation cost and decreasing the prediction inaccuracy.

The figure shows that ALL\_NBR heuristic, while maintaining a very low missing rate, is not a good choice for the OTSN when it takes more than 175msec for each sampling. It implies that the operation cost for the longer sampling period overwhelms the energy saving obtained by accurate predictions.

**5.4.2. Study of the sampling frequency.** Sampling frequency is the number of sampling performed during a reporting period. To meet the application requirements, a sensor node has to perform sampling at least once per reporting period. However, increasing the sampling frequency at a sensor node may increase the freshness of an object's moving history (for AVERAGE and EXP\_AVG prediction models), but reduce the duration the node can turn to sleep and thus save energy. Thus, sampling frequency represents a tradeoff between the accuracy of predictions and the cost of sensing operation. With a small prediction period (i.e., sleep time), the probability that an object stays in the same speed and direction is high; thus, the prediction is more likely to

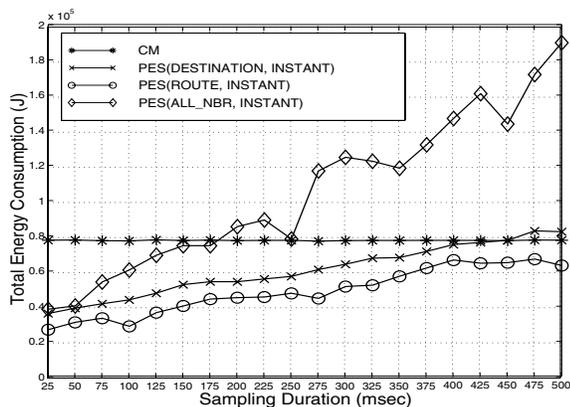


Figure 7. Impact of the sampling duration

be correct. However, a higher sampling frequency also implies unnecessary monitoring (because the results are not needed for reporting).

Figure 8 shows that the energy consumption is linearly increased when the sampling frequency is increased. However, the missing rate (not shown here) is dramatically decreased. Based on our simulation result, the missing rates of the PES scheme with all three wakeup heuristics are below 1% when the sampling frequency equals to 3. However, continue to increase the sampling frequency is not a good strategy, because the extra operation cost will eventually overwhelm the gain achieved by the higher prediction accuracy. As we observed from the above figure, all the PES schemes consume more energy than CM at some point. From this experiment, we can observe that the sampling frequency is a tunable parameter between energy consumption and objects missing rate that may worth further study.

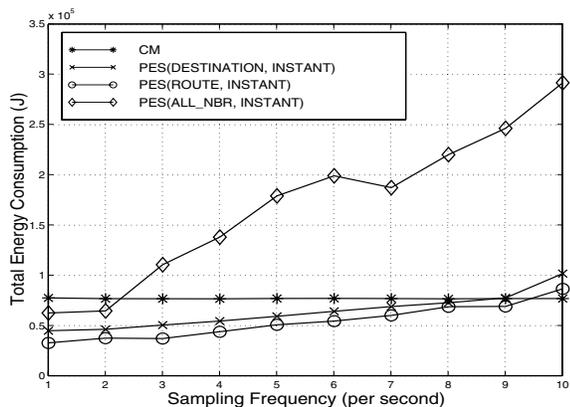


Figure 8. Impact of the sampling frequency

## 6. Conclusion and Future Work

A fundamental challenge for sensor networks comes from the fact that sensor nodes are supported by scarce battery resources. Most of the existing work reaches energy saving by optimizing the communication cost of radio components. However, MCU and sensor components also play critical roles in sensor nodes energy consumption. In this paper, a prediction-based energy saving (PES) scheme is proposed to exploit the energy management issues in object tracking sensor networks. Energy awareness is incorporated into computing and sensing operations by inactivating the sensing and MCU components whenever possible.

A prediction model in PES predicts the future movement of the tracked objects, which provides the knowledge for a wake up mechanism to decide which nodes need to be activated for object tracking. Different heuristics are discussed for both prediction and wakeup mechanisms. An extensive simulation study has been conducted to evaluate the performance (in terms of total energy consumption and missing rates) of the proposed PES scheme and other basic energy saving schemes for object tracking sensor networks. The experimental result confirms that PES can effectively reduce the energy consumption on MCU and sensor components. The different heuristics discussed in this paper provide a space for object tracking applications to balance between the energy consumption of the entire network and required data quality (in terms of missing rate).

As for the future work, we are looking into the moving patterns of the mobile objects, and are exploring more sophisticated prediction models to improve the prediction accuracy, thus further minimizing the communications between the sensor nodes and the base station. In addition, we plan to incorporate the energy saving techniques in radio communication into the current work. Based on PES, we are building a prediction-based, energy-aware communication protocol for sensors to adapt their communications with the base station. As such, we can employ the same prediction-based approach in different layers of the object tracking sensor networks to optimize the energy consumption and increase the lifetime of the entire network.

## References

- [1] A. Bhattacharya and S. K. Das. Lezi-update: An information-theoretic framework for personal mobility tracking in PCS networks. *ACM/Kluwer Journal on Wireless Networks*, 8(2-3):121–135, March-May 2002.
- [2] L. Clare, G. Pottie, and J. Agre. Self-organizing distributed sensor networks. In *The International Society for Optical Engineering*, pages 229–237, Orlando, FL, April 1999.
- [3] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the ACM/IEEE International Con-*

- ference on Mobile Computing and Networking*, pages 263–270, Seattle, Washington, USA, August 1999.
- [4] S. Goel and T. Imielinski. Prediction-based monitoring in sensor networks: taking lessons from MPEG. *ACM Computer Communication Review*, 31(5), October 2001.
  - [5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *IEEE Proceedings of the Hawaii International Conference on System Sciences (HICSS)*, January 2000.
  - [6] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for network sensors. In *ACM SIGOPS Operating Systems Review*, volume 34, pages 93–104, 2000.
  - [7] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Data on the air - organization and access. *IEEE Transactions of Data and Knowledge Engineering*, 9(3):353–372, June 1997.
  - [8] Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 66–75, 1998.
  - [9] D. Lee, W.-C. Lee, J. Xu, and B. Zheng. Data management in location-dependent information services. *IEEE Pervasive Computing*, 1(3), 2002.
  - [10] W.-C. Lee and D. L. Lee. Using signature techniques for information filtering in wireless and mobile environments. *Special Issue on Database and Mobile Computing, Journal on Distributed and Parallel Databases*, 4(3):205–227, July 1996.
  - [11] Ben Liang and Zygmunt J. Haas. Predictive distance-based mobility management for PCS networks. In *Proceedings of the 18th IEEE The Conference on Computer Communications (INFOCOM)*, pages 1377–1384, March 1999.
  - [12] S. Madden, R. Szewczyk, M. Franklin, and D. Culler. Supporting aggregate queries over ad-hoc wireless sensor networks. In *4th IEEE Workshop on Mobile Computing Systems and Applications*, June 2002.
  - [13] Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, , and Wei Hong. The design of an acquisitional query processor for sensor networks. In *2003 ACM SIGMOD international conference on on Management of data*, pages 491 – 502, San Diego, California, June 2003.
  - [14] G. Pei and C. Chien. Low power TDMA in large wireless sensor networks. In *Military Communications Conference*, pages 347–351, 2001.
  - [15] J. Rabaey, J. Ammer, T. Karalar, S. Li, B. Otis, M. Sheets, and T. Tuan. Picoradios for wireless sensor networks: the next challenge in ultra-low-power design. In *Proceedings of the International Solid-State Circuits Conference*, San Francisco, CA, February 2002.
  - [16] J. M. Rabaey, M. J. Ammer, J. L. da Silva Jr., D. Patel, and S. Roundy. Picoradio supports ad hoc ultra-low power wireless networking. *IEEE Computer*, 33(7):42–48, 2000.
  - [17] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava. Energy aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, March 2002.
  - [18] H. Schwetman. *CSIM user's guide (version 18)*. Mesquite Software, Inc, <http://www.mesquite.com>, 1998.
  - [19] N. Shivakumar and S. Venkatasubramanian. Efficient indexing for broadcast based wireless systems. *ACM/Baltzer Mobile Networks and Applications (MONET)*, 1(4):433 – 446, December 1996.
  - [20] S.Madden, M. J. Franklin, J. Hellerstein, and W. Hong. Tag: Tiny aggregate queries in ad-hoc sensor networks. In *ACM Symposium on Operating System Design and Implementation (OSDI)*, Dec. 2002.
  - [21] K. Sohrabi and G. J. Pottie. Performance of a novel self-organization protocol for wireless ad-hoc sensor networks. In *IEEE Vehicular Technology Conference*, volume 2, pages 1222–1226, 1999.
  - [22] WINS project. Electrical engineering department, ucla. <http://www.janet.ucla.edu/WINS/>.
  - [23] G. Wan and E. Lin. A dynamic paging scheme for wireless communication systems. In *Proceedings of the Third Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 195–203, September 1997.
  - [24] Y. Xu and W.-C. Lee. On localized prediction for power efficient object tracking in sensor networks. In *In Proc. 1st International Workshop on Mobile Distributed Computing (MDC)*, pages 434–439, Providence, Rhode Island, May 2003.
  - [25] L. C. Zhong, R. Shah, C. Guo, and J. Rabaey. An ultra-low power and distributed access protocol for broadband wireless sensor networks. In *IEEE Broadband Wireless Summit*, Las Vegas, N.V., May 2001.