INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

OBSERVING TARGET IN MOBILE SENSOR NETWORKS

[1]K. Pushpalatha, [2]N V N Sowjanya

[1]PG Scholar, Department of Computer Science and Engineering Bharath Institute of Science and Technology Hyderabad, A.P-500 097, India

[2]Assistant Professor, Department of Computer Science and Engineering Bharath Institute of Science and Technology Hyderabad, A.P-500 097, India

Abstract

Target tracking systems, comprising of thousands of low cost sensor nodes, have been used in many presentation provinces such as skirmish surveillance, nature monitoring and border sanctuary. These applications need to meet certain real time constraints in response to momentary events, such as fast moving targets. While the real time presentation is a major concern in these applications, it should be compatible with other important system properties such as energy consumption and accuracy. Hence, it is desirable to have the ability to exploit the tradeoffs among them. This work presents the real-time design and analysis of Vigil Net, a large-scale sensor network system which tracks, detects and classes targets in a timely and energy efficient manner. Based on a deadline partition method and theoretical derivations of each sub-deadline, we are able to make guided engineering decisions to meet the end-to-end tracking deadline. To concern our design and obtain an empirical understanding of these tradeoffs, we invest sign cant efforts to perform large-scale simulations with 10,000 nodes as well as a test with 200 XSM motes, running Vigil Net. The results from both analysis and evaluation can serve as general design guidelines to build similar real-time systems.

Keywords: wireless sensor networks, mobile target tracking, quality of service

1. INTRODUCTION

The advances in computation, communication and sensing capabilities, large scale sensor based distributed environments are emerging as a predominant mobile computing infrastructure. Sensor applications typically monitor real world phenomena and utilize the accurate knowledge of current state to dynamically optimize application execution. Traditionally, a critical issue in enabling wide-spread use of mobile computing is that of effective power management to extend battery life. This is further exacerbated in sensor environments where it is often difficult or infeasible to replenish power supplies of wireless devices. Power supply in sensors is used for a variety of purposes for basic sensing operations, for powering the memory and CPU, and for communication. While the specific rate of energy consumption for each of these operations is sensor and application specific, much of the existing literature indicates that communication constitutes a major source of power drain.

While well engineered sensor technology can significantly conserve power, for a variety of applications much further gain can be accrued by exploiting the natural tradeoff between application quality and energy conservation. In this paper, we explore a framework that exploits resource versus quality tradeoff to reduce communication in the context of information collection for mobile target tracking in sensor environments. The proposed framework exploits the application tolerance by collecting data at only the accuracy levels needed to meet the application's requirements. The greater the application's tolerance to error in sensor values, the lower the communication overhead between sensors and server, and hence higher the resulting savings.

2. FEATURE RESPONSIVE STATISTICS COLLECTION AND TRACKING APPLICATIONS

2.1 Sensor Nodes and Sensor Networks

The overall environment for adaptive information col-lection is illustrated in Figure 1, where applications employ information collection protocol to obtain the desired information from a distributed sensor network infrastructure.

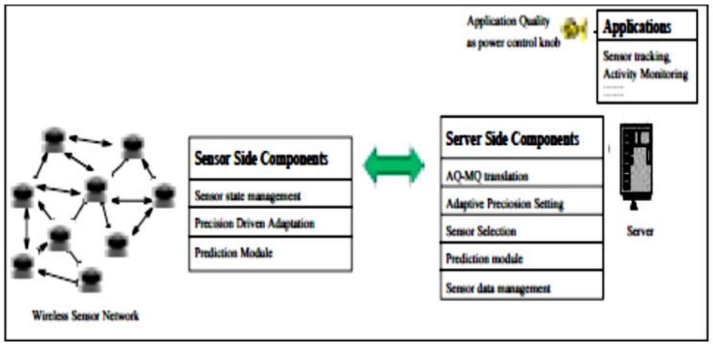


Figure 1. Information collection architecture

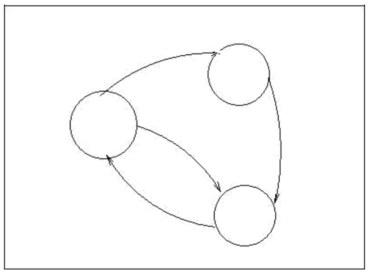


Figure 2. State transition diagram for sensor node in target tracking

A sensor node consists of the embedded sensor, a processor with some limited memory and the radio circuitry. Each component is controlled by a micro operating system that decides which device to turn off and on. A power aware sensor node has 3 different states of operation (see Figure 2). While in state S0, a sensor node transmits and receives messages at every time instant. State S1 (quasi-active state) is a reduced energy state in which the sensor sends a message to the server at a reduced frequency as explained later. The most energy efficient state S2 is the monitor state in which the sensor turns its radio off and senses the environment for signals. The state transitions of the sensor are dictated by the adaptive information collection protocols. A sensor senses the environment at periodicity tense and communicates its readings to the server at periodicity send. For simplicity we assume that tend is the same as tsense. We consider time to be discretized into units of length tsend each. A sensor reading at time unit is represented as $I_i(\tau)$.

The system architecture considered in this paper is logical. Physically, the server architecture may itself be centralized, distributed or hierarchical. Furthermore, a part of the server module might reside on a computation-ally superior sensor node, which may be responsible for both sensing as well as fusing the readings from other sensor nodes. In the latter case, the server may host both the sensor module as well as the server module. Furthermore, the issue of wireless net-work routing of data between the sensors and servers is orthogonal to our work and is abstracted out.

2.2 Mobile Target Tracking

In our application scenario, each sensor has a unique identifier, coordinates, and visibility radius. For low cost and low energy consumption, we assume passive sensors that operate only on the received acoustic or seismic waveforms from non-cooperative sources. A sensor senses the environment and communicates its readings to the server periodically; the server triangulates the location of the object using the readings. The signal attenuation of the target source power P is governed by equation I = P=4 r2, where r is the Euclidean distance between the object and a sensor node. This relation is the basic equation used in triangulation.

Let $\{\langle o_x[1], o_y[1]\rangle, \cdots \langle o_x[t_s], o_y[t_s]\rangle\}$ track of object $\{\langle o'_x[1], o'_y[1]\rangle, \cdots \langle o'_x[t_s], o'_y[t_s]\rangle\}$
approximate track perceived by the server.

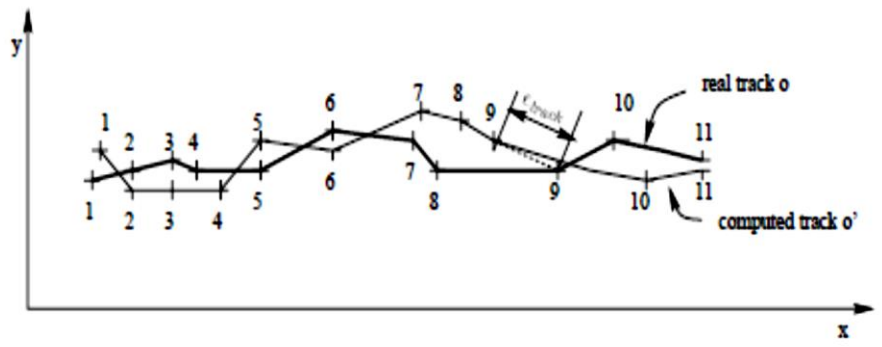


Figure 3. Tracking quality

Track Quality: Track quality, track, is defined as the maximum distance between the real track o of the target object and the approximate track generated at the user end.

$$\epsilon_{track} = max_{0 \leq \tau \leq n} |o[\tau] - o'[\tau]|.$$

Sensor Measurement Quality: If we consider the measurement at a sensor S as a time series I[1 : t] which is

- If $\Delta I_i < 0$, then scale $|\Delta I_i|$ by $\frac{I_i}{I_i + \Delta I_i}$. Thus the new $|\Delta I_i|$ becomes $\frac{I_i^2 \xi}{1 + I_i \xi}$, where $\xi = \sqrt{\frac{\Delta d^2_{max}}{C}}$.
- If $\Delta I_i > 0$, then scale $|\Delta I_i|$ by $\frac{I_i}{I_i - \Delta I_i}$. Thus the new $|\Delta I_i|$ becomes $\frac{I_i^2 \xi}{1 - I_i \xi}$, where $\xi = \sqrt{\frac{\Delta d^2_{max}}{C}}$.

Approx. Mated measurement quality

j Ij is the maximum divergence between I[ ] and I0[ ] at any time i.e :

$$|\Delta I| = max_{0 < \tau < t} |I[\tau] - I'[\tau]|$$

Given a user defined track quality parameter track, the sensor network tracks an object with an error bound of track with small communication overhead. We map track quality to measurement quality below, where we utilize a simple tracking approach of triangulating 3 sensor readings to determine the location of an object. Consider three sensors whose locations ((x1; y1); (x2; y2); (x3; y3)) are known. The intensity readings of the b sensors at the time of triangulation are I1; I2 and I3. We then have the following set of equations:

$$\begin{cases} (x-x_1)^2+(y-y_1)^2 = \frac{p}{4\pi I_1} \\ (x-x_2)^2+(y-y_2)^2 = \frac{p}{4\pi I_2} \\ (x-x_3)^2+(y-y_3)^2 = \frac{p}{4\pi I_3} \end{cases}$$

Solving for x and y :

$$x = \frac{-b_2c_1+b_1c_2}{b_2a_1-b_1a_2} \qquad (1)$$

$$y = \frac{-a_2c_1+a_1c_2}{a_2b_1-a_1b_2} \qquad (2)$$

Where, $a_1 = 2(x_2-x_1), b1 = 2(y_2-y_1), c_1 = x_1^2 - x_2^2 + y_1^2 - y_2^2 - \frac{p}{4\pi}\left(\frac{1}{I_1}-\frac{1}{I_2}\right), a_2 = 2(x_3-x_2), b2 = 2(y_3-y_2)$, **and** $c_2 = x_2^2 - x_3^2 + y_2^2 - y_3^2 - \frac{p}{4\pi}\left(\frac{1}{I_2}-\frac{1}{I_3}\right)$.

For the above triangulation scheme, we can show that the tracking error tolerance track is proportional to the track. This provides, for each sensor Si, a relative measurement error tolerance which is subsequently used in our adaptive protocols.

There are two issues with the above approach. Firstly, we observe that first order approximations of x and y work well only when Ii is small. As x and y are linear x is under-estimated and when Ii > 0, x is over-estimated. We overcome this problem by scaling j Iij as follows:

- **If** $\Delta I_i < 0$, **then scale** $|\Delta I_i|$ **by** $\frac{I_i}{I_i+\Delta I_i}$. **Thus the new** $|\Delta I_i|$ **becomes** $\frac{I_i^2\xi}{1+I_i\xi}$, **where** $\xi = \sqrt{\frac{\Delta d_{max}^2}{C}}$.
- **If** $\Delta I_i > 0$, **then scale** $|\Delta I_i|$ **by** $\frac{I_i}{I_i-\Delta I_i}$. **Thus the new** $|\Delta I_i|$ **becomes** $\frac{I_i^2\xi}{1-I_i\xi}$, **where** $\xi = \sqrt{\frac{\Delta d_{max}^2}{C}}$.

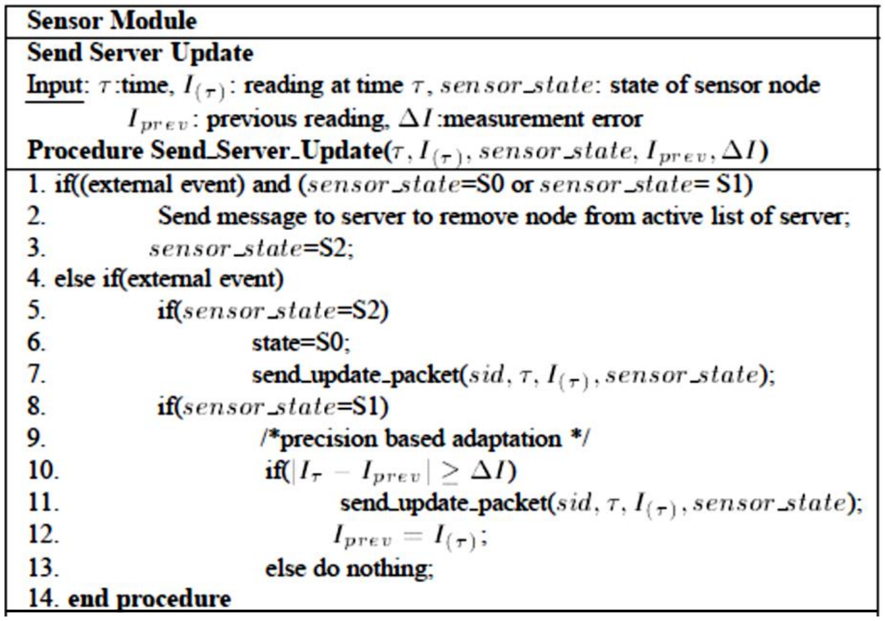
Secondly, in our tracking algorithm, the target source power was assumed to be constant. However, when the source power is unknown, tracking and error translation are more complicated.

3. QUALITY AWARE INFORMATION COLLECTION PROTOCOLS

As per the state transition diagram of a sensor node, at a particular instant of time some sensors will be ac-tive while the others will be in a *monitor* or *quasi-active* state. The sensor selection phase determines which sensors should be active at a particular time instant, and controls the state transition of the sensors from active to quasi active and monitor states. It also implements precision based measurement update. The following sub-sections describe the sensor selection process in detail at the server and the sensor side.

3.1 The Sensor Module

A sensor may be triggered by an external event to move from a monitor state S2 to an active state S0. An external event in a tracking application would be an object moving within the visible radius of a sensor node. In its active state, a sensor sends continuous updates at each time interval to the server module. An update packet consists of the sensor reading, timestamp and state of the sensor. The server module may decide to transit the sensor to a *quasi-active* state S1 by sending the measurement tolerance I corresponding to the desired application tolerance track. The sensor now makes a transition to a *quasiactive* state, and sends an update to the server module only if its reading exceeds its previous reading by the measurement tolerance.



```
Sensor Module
Send Server Update
Input: $\tau$:time, $I_{(\tau)}$: reading at time $\tau$, $sensor\_state$: state of sensor node
       $I_{prev}$: previous reading, $\Delta I$:measurement error
Procedure Send_Server_Update($\tau, I_{(\tau)}, sensor\_state, I_{prev}, \Delta I$)
1. if((external event) and ($sensor\_state$=S0 or $sensor\_state$= S1)
2.         Send message to server to remove node from active list of server;
3.         $sensor\_state$=S2;
4. else if(external event)
5.         if($sensor\_state$=S2)
6.                 state=S0;
7.                 send_update_packet($sid, \tau, I_{(\tau)}, sensor\_state$);
8.         if($sensor\_state$=S1)
9.                 /*precision based adaptation */
10.                if($|I_\tau - I_{prev}| \geq \Delta I$)
11.                        send_update_packet($sid, \tau, I_{(\tau)}, sensor\_state$);
12.                        $I_{prev} = I_{(\tau)}$;
13.                else do nothing;
14. end procedure
```

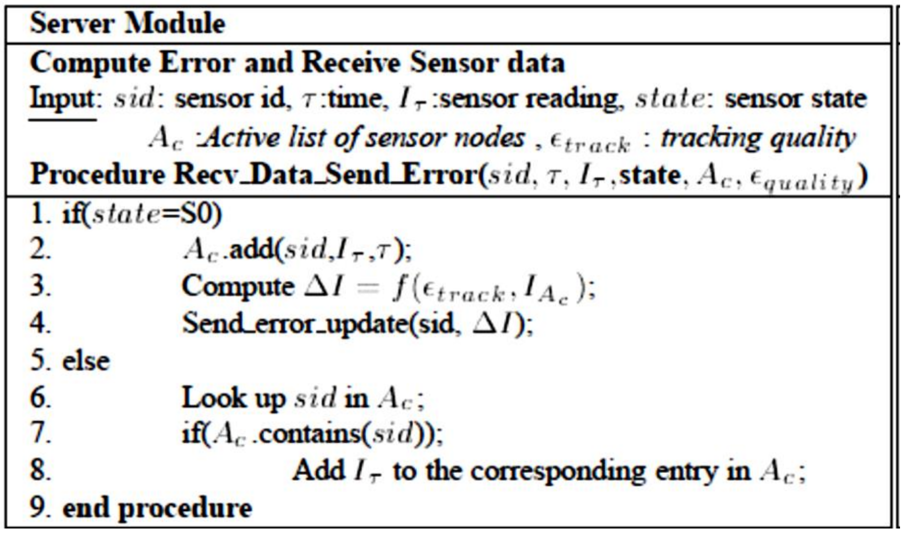
Table 1. Sensor side algorithm for adaptation and state transition

The server on receiving the value update will make the corresponding update for the sensor measurement in its sensor database. A sensor may transit from the *active* or *quasi-active* states to the *monitor* state due to either an external event (e.g., sensing measurement falling below a threshold), or a message from the server. If the transition happens due to an external trigger, the sensor sends a corresponding message to the server. Table 1 shows the sensor module for precision based adaptation and state transition at a sensor node.

3.2 The Server Module

The server maintains an active list of sensor nodes which contains an array of sensors and a history of their readings over a time period. For the sake of simplicity, we bound the communication latency in the sensor network, i.e. the time taken for an intensity reading to reach the server by Ætl . Thus the network latency is less than Ætl for any node in the network. At time t >= 0, when the server receives an update from sensor Si, it waits for Ætl (to receive all possible sensor readings at that time instant) and checks to see if the sensor is in mode S0(active state). If so, it adds this sensor to its active list and sends an error update message containing the tolerable measurement error I. However, if the sensor is in state S, the server looks up the active list to determine whether the sensor readings are already in use by applications at the server. If so, it adds the latest reading of sensor Si along with its timestamp to the sensor's history list. Alternatively, if the sensor is not in the active list of the server the server sends a message to the sensor to shift to the monitor state. On receipt of the message from the sensor that it has transited into monitor state, the server looks up its active list and removes the corresponding entry in the list.

The framework described above ensures that for each *active* sensor, the server maintains in its database an approximate measurement that is within the tolerable threshold of the actual sensor measurement. The reduced rate of communication between the sensor and the server in the *quasi-active* state and switching off of the radio in the *monitor* state result in energy conservation.



```
Server Module
Compute Error and Receive Sensor data
Input: $sid$: sensor id, $\tau$:time, $I_\tau$:sensor reading, $state$: sensor state
          $A_c$ :Active list of sensor nodes , $\epsilon_{track}$ : tracking quality
Procedure Recv_Data_Send_Error($sid$, $\tau$, $I_\tau$,state, $A_c$, $\epsilon_{quality}$)
1. if($state$=S0)
2.         $A_c$.add($sid$,$I_\tau$,$\tau$);
3.         Compute $\Delta I = f(\epsilon_{track}, I_{A_c})$;
4.         Send_error_update(sid, $\Delta I$);
5. else
6.         Look up $sid$ in $A_c$;
7.         if($A_c$.contains($sid$));
8.                 Add $I_\tau$ to the corresponding entry in $A_c$;
9. end procedure
```

Table 2. Server side sensor selection protocol

A triangulation based tracking application program sits at the server above the proposed middleware frame-work. It takes n readings from the active list of the server module and pinpoints the location of the moving object with bounded tracking quality guarantee

3.3 Estimate Models and Local Aggregation

The communication cost of the precision based approach can be further reduced by exploiting the predictability of readings of a sensor. If we consider a precision based approach, whenever the current location of the object deviates from its previous reported position by $_{track}$, the sensor has to communicate its readings to the server. If the sensor and the server can decide on the mobility model of the target object, they can predict its location in the near future and further communication savings can be achieved. In such a case, our approach can be modified such that a sensor reports its readings only if the observed value deviates from the expected value based on the prediction model by a pre-defined error bound pred. Such prediction models may be integrated into the client and server modules of the information collection frame-work to gain further savings in energy. Many issues need to be resolved in incorporating prediction models, such as who determines the prediction model, the sensor or the server; the protocol by which a sensor or the server choose a model M; dynamic model switching when the prediction model needs to be changed In order to further reduce the long range communication cost between sensors and the server, local agree ration of information within the sensor network can be exploited. The key advantage of local aggregation is that a partial result obtained at a localized point may only be communicated to the server only if the change in the result is significant enough to violate the quality requirement. In target tracking, the result is a location from early triangulation.

4 PERFORMANCE ANALYSIS

4.1 Simulation

The sensor network is modeled as a grid with nodes located at integer coordinate points. Time is modeled as a discrete quantity and the sensor nodes and the access points are assumed to be time synchronized with respect to each other. The server has superior computation and communication capabilities compared to the sensor nodes. Though we refer to a single grid and access point in our experiments and tracking protocols, the same set of protocols may be running on several such grids and access points to track a moving object in space and time. A grid based network topology was simulated to evaluate the performance of our adaptive tracking protocols with 400 sensor nodes with an internode distance d = 10m. The performance of our algorithms in terms of the energy consumption at each individual sensor node and the entire grid over the period of time the object is tracked. For this purpose we use an energy model (AMP S project[3, 5]) where the energy dissipation for sensing and computation is negligible compared to that for communication overhead of the radio subsystem. The performance is defined for the movement patterns for a target object. For this purpose we used two elementary and feasible movement patterns of a moving object as shown in Table 3.

4.2 Outcomes

Four separate protocols to compare the total energy consumption in the sensor network over the period of tracking. We use a non-adaptive algorithm NON-ADP, in which each sensor sends a reading at a fixed time

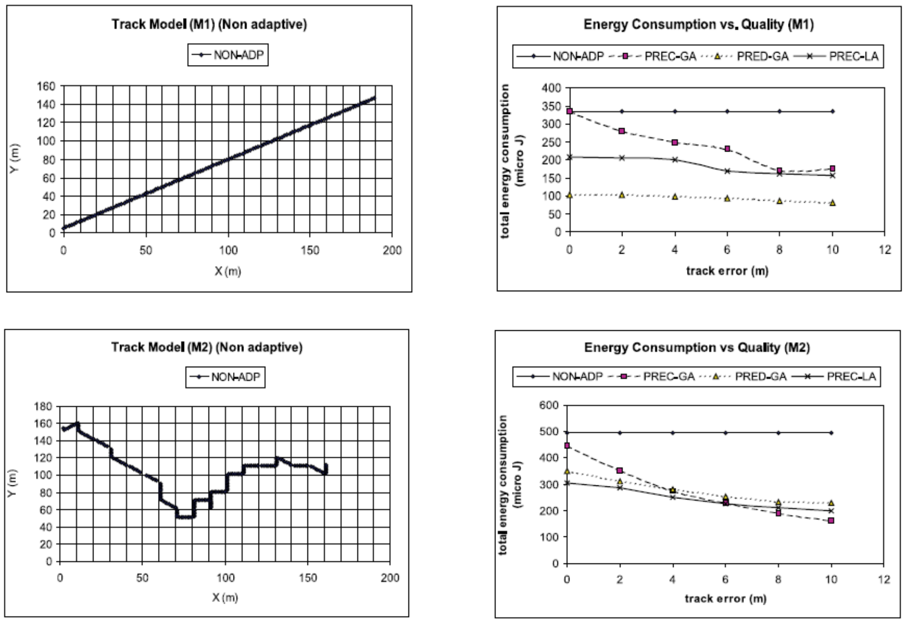


Figure 5. Precision and Prediction based experiments for mobility model M2

Interval of send =0.2 seconds to the server. **PREC-GA** is a precision based adaptive protocol which has a user defined tolerance parameter associated with it. This means that the user specifies at the application level the quality at which he wants the object to be tracked. The application quality is mapped to the measurement quality, based on which a sensor sends an update to the server. In **PRED-GA**, we allow the sensor to fit a constant velocity based prediction model to track the object. The fourth algorithm **PREC-LA** implements the local aggregation mechanism where an leader node is elected among a group of sensors and each active sensor in the group sends its update based on its measurement error to the leader, who forwards it to the server only if the overall group quality is violated. Also, for the sake of simplicity we abstract our algorithm from the underlying routing algorithm which routes a packet from a sensor node to the leader.

We vary the error tolerance from 2 to 10 meters, and measure the energy consumption at each node and over the entire grid for the period of simulation. In each of these graphs the energy consumption is measured in terms of micro joules per bit. Figure 4 shows the results obtained for mobility model M1. The graphs show (a) the track of the object, (b) the variation of the total energy consumption per bit using the four algorithms de-scribed above. We get a significant savings (50-60%) in communication costs over the network. PRED-GA outperforms the precision based PREC-GA since the object follows a linear velocity model throughout its trajectory. However, at a certain track error, energy savings from prediction and precision based algorithms reach a saturation level as further optimizations are not feasible. PREC-LA outperforms PRED-GA since it per-forms lesser number of global server updates based on local computation at the leader node.

Figure 5 shows the same results for mobility model M2. We notice that using a prediction based algorithm PRED-GA, we quickly reach a saturation point and the precision based algorithm PREC-GA starts performing better at a certain value of track error. This is because the object now follows a more random path than in mobility model M1 which leads to model switching and more communication overhead at a sensor node when we use a prediction based approach.

Figure 6 shows the impact of send time interval and velocity of the moving object to energy savings at the sensor node. The first graph plots the energy savings with an increase in send or the time interval at which a node sends an update in the non-adaptive algorithm. We notice that at lower values of tsend we obtain significant energy efficiency. However, when tsend is sufficiently large, the quality of the true track as governed by tsend is not fine grained enough to be optimized. In the second graph, the velocity of the object was varied from 5 m/s to 15m/s while the track error was fixed at 10m and tsend=0.2 seconds for object model M1. At lower velocities, our energy savings are very high, since the sensor nodes in a grid cell send much fewer readings than in the non-adaptive case (both in case of **PREC-GA** and **PRED-GA**).

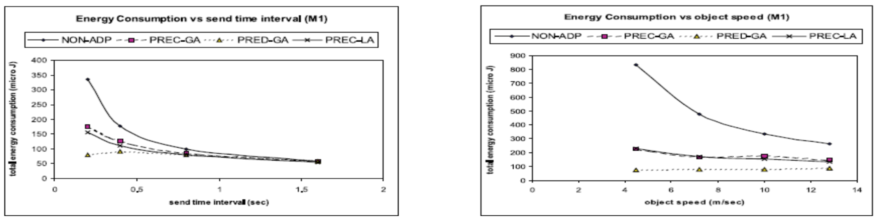


Figure 6. Effect of tsend and object velocity in mobility model M1

m/s while the track error was fixed at 10m and tsend=0.2seconds for object model M1. At lower velocities, our energy savings are very high, since the sensor nodes in a grid cell send much fewer readings than in the no adaptive case (both in case of **PREC-GA** and **PREDGA**).

Conclusion

The feasibility to design a complex real-time sensor network, using the deadline partition method, which guarantees an end-to-end tracking deadline by satisfying a set of sub-deadlines. And also analytically identify the tradeoffs among system properties while meeting the real time requirements. The validate our design and analysis through both a large scale simulation with 10,000 nodes as well as a test with 200 XSM nodes. We contribute a set of tradeoffs that are useful for the future development of real-time sensor systems. Given real-time constraints, a system designer can make guided engineering judgments on the system parameters shahs the network density, the appropriate detection algorithm and the duty cycle settings for the sensor nodes.

REFRENCES:

[1] C. Olston, B. T. Loo, and J. Widom. Adaptive precision setting for cached approximate values ACM Sigmod, 2001

[2] M. Bhardwaj, A. P Chandrakasan. Bounding the Lifetime of Sensor networks via optimal Role assignments Proceedings of INFOCOM,2002

[3] A. Sinha, A. Chandrakasan. Dynamic Power Management in Wireless Sensor networks IEEE DesignandTestComputers,2001

[4] G.J. Pottie andW.J. Kaiser. Wireless Integrated network sensors

[5] R.Min et al. An architecture of a power aware distributed micro sensor workshop on Signal Processing systems, 2000.

[6] T. Rappaport. Wireless Communications: Principles and Practice Prentice Hall

[7] S. Singh and C. S Raghavendra. PAMAS- Power Aware Multi-access protocol with signaling for adhoc networks SIGCOMM Computer Communication Review, July 1998

[8] X. Yu, K. Niyogi, S. Malhotra and N. Venkatasubramanian.An Adaptive Information Collection Middleware for Sensor Network Applications.

[9] F. Zhao, J. Shin and J. Reich. Information driven Dynamic Sensor Collaboration for tracking applications IEEE Signal processing magazine, 2002

[10] D. Estrin, R. Govindan, J. Heinemann and S. Kumar. Next Century Challenges: Scalable Coordination in Sensor Networks MobiCOM, 1999

[11] W. R. Hein Zelman, J. Kulik and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks ACM /IEEE MobicomConference, 1999

[12] C. Olston and J. Widom. Best-Effort Cache Synchronization with Source Cooperation SIGMOD 2002

[13] http://www.darpa.mil/ito/research/sensit/index.html

Authors:

1). K. Pushpalatha - PG Scholar, Department of Computer Science and Engineering Bharath Institute of Science and Technology, Hyderabad, Telangana, India.

2) N V N Sowjanya - Assistant Professor, Department of Computer Science And Engineering Bharath Institute of Science and Technology, Hyderabad, Telangana, India