On Localized Prediction for Power Efficient Object Tracking in Sensor Networks

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

Abstract

Energy is one of the most critical constraints for sensor network applications. In this paper, we exploit the localized prediction paradigm for power-efficient object tracking sensor network. Localized prediction consists of a localized network architecture *and a prediction mechanism called* dual prediction*, which achieve power savings by allowing most of the sensor nodes stay in sleep mode and by reducing the amount of long-range transmissions. Performance evaluation, based on mathematical analysis, shows that localized prediction can significantly reduce the power consumption in object tracking sensor networks.*

1 Introduction

Object tracking, widely deployed for military area intrusion detection and wildlife animal monitoring, is a representative application of wireless sensor networks [9, 16]. To develop sensor networks for object tracking, *battery power conservation* is one of the most critical issues since the sensor nodes are often supported by batteries which could be difficult to replace.

Most of today's sensor boards provide four different modes for radio transmissions: *Transmit, Receive, Idle* and *Sleep* [5]. Studies of sensor power consumption on WINS nodes developed by Rockwell and UCLA show that 1) long distance transmission dominates the energy dissipation of sensor networks; 2) idle mode consumes nearly as much power as receiving mode; and 3) sleeping mode consumes only around one-sixth of the power in active mode [13]. This analysis of radio power consumption provides important hints for power optimization in various areas of sensor network design. Energy efficiency of the sensor networks can be improved by reducing long distance transmissions at the cost of more localized communications among nearby sensor nodes and inactivating radio components as much as possible [2, 3, 4]. In this paper, we present a prediction based approach, called *localized prediction*, for power efficient object tracking sensor networks, by exploiting the above hints.

The localized prediction consists of two parts: a *localized sensor network architecture*, where most of the sensor nodes keep sleeping until waken up by an active sensor node, via a low power paging channel, to anticipate the task of object tracking, and a prediction mechanism called *dual prediction*. Predictions about future movement of a tracked object are calculated at both of a sensor node and its cluster head (which will be defined later). Information collected at a sensor node is not sent if the object's movement is consistent with the prediction. This reduction of long distance transmissions is at the cost of handing off moving history of an object (needed for calculating predictions) among neighbor sensors.

The ideas of utilizing predictions to reduce overheads is not new in mobile computing systems. Prediction based techniques has been proposed to reduce the paging overhead in cellular network by limiting search space to a set of cells that mobile users may enter [1, 17]. In wireless data broadcast systems, mobile computers turn on the radio only during the arrival time of requested data frames, which is predicted based on the indexing information provided in broadcast channels [6, 8, 15]. Similarly in sensor networks, Goel and Imielinski argued that readings at a sensor node can be predicted based on the past reading history and spatio and temporal relationships of readings from surrounding sensors. They proposed a prediction based monitoring mechanism, called *PREMON*, to reduce the number of transmissions at the cost of more receptions [3]. The dual prediction mechanism in our proposal is different from PREMON in an important aspect. Instead of calculating predictions at a cluster head and sending predicted readings to a sensor via long distance transmission (as PREMON does), dual prediction trades off local computation at the sensor node for reduction of long distance transmission.

We have conducted a performance evaluation, via mathematically analysis, to explore the potential power savings by localized prediction and make comparison with PREMON and a system without using predictions. Our result concludes that the localized prediction can significantly reduces the power consumption in object tracking sensor networks and outperforms the compared approaches.

The rest of paper is organized as follows. Section 2 describes the system architecture of localized sensor networks for object tracking. Section 3 discusses prediction based object tracking. In Section 4, performance of localized prediction is evaluated by comparing with other mechanisms. Finally, Section 5 concludes this paper and depicts future research directions.

2 Object Tracking Sensor Networks

In this section, we describe a general system architecture, set up assumptions on location model and topology for illustration and later analysis, and finally present the localized algorithm for power efficient object tracking.

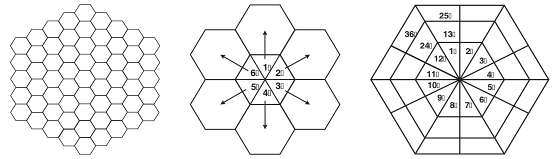
2.1 System Architecture

In wireless sensor networks, clustering techniques are frequently used to construct self-organized network hierarchy in order to address communication, power conservation, and information aggregation problems in the network layers [2, 4]. In this paper, we deploy hierarchical cluster as network architecture. All the sensor nodes within a cluster send data to their *cluster head*. Furthermore, we assume TDMA is used as MAC protocol for the communication between a sensor node and its cluster head [10, 14], and low power paging channel is used for communications among sensor nodes [11, 12, 18]. The time slots of a TDMA channel for a cluster are evenly allocated to members of the cluster. Thus, a sensor node may save power by staying in sleep and only waking up when its time slot arrives. Comparing to TDMA, the paging channel is more flexible and power efficient since the sensor nodes are activated on demand.

2.2 Location Models and Topology

Objects location can be represented in a geometric model (e.g., coordinates) or a symbolic model (e.g., id of a sensor node) [7]. With knowledge of sensor network topology, those two models can be transformed based on the application requirements. In this paper, without losing generality, we assume a symbolic representation of object locations for its simplicity.

To facilitate our discussion and later analysis, as shown in Figure 1, we assume an ideal, hexagon shaped



(a) Hexagon (b) Six triangles (c) Smaller subarea

Figure 1. Ideal sensor network topology

sensor topology. All sensor nodes have the same detection radius r, the maximum distance within which a sensor node can detect the existence of objects. We also assume that sensor radio range d satisfies $d = \sqrt{3}r$. Hence, each sensor node is surrounded by six neighboring nodes. In our model, we reduce the overlap of two neighboring sensor detection areas to the common edges of detection areas. Based on these assumptions, each of the sensor detection area can be modelled as a hexagon[1] and is further divided into six identical equilateral triangles numbered 1 to 6, representing the symbolic location of objects (see Figure 1(b)). The neighbors are identified by the numbers of triangles that they are next to. The precision of this model can be enhanced by dividing detection area into smaller pieces. For example, as shown in Figure 1(c), a detection area consists of 36 subareas.

2.3 Localized Object Tracking

Based on our assumption in Section 2.2, the location of a moving object is represented by the triangle number and the moving trail is represented as a sequence of triangles numbers. Thus, a moving trail for Figure 2 could be $\langle 5, 5, 6, 1, 1, 1, 2, 2 \rangle$.

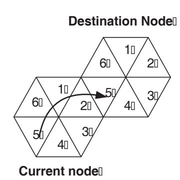


Figure 2. Moving trail in a sensing area

The sensor node, where an object is currently monitored, is called the *current node*. It assumes that an object will leave for the neighboring sensor node next to the triangle where the object is located (called target node), and thus wakes up this target node. The target node where the object eventually enters is called *destination node*. For example, in Figure 2, the target nodes next to triangle 6 and 1 of the current node are waken up, even though the object enters the destination node next to triangle 2. To prevent target nodes being idle for a long period of time, the wake-up messages from the current node come with a TTL value, which repre-

[1] A sensor detection area can also be modelled as geometry shapes with more edges, such as heptagon and octagon with seven and eight neighbors, respectively.

sents the period of time a target node should stay awake before going back to sleep.

3 Prediction Based Object Tracking

In this section, we first discuss some prediction heuristics of object moving behavior and then describe the dual prediction mechanism. Finally, three prediction models that can be deployed for dual prediction, namely, *constant model, average model,* and *exponential average model* are presented.

3.1 Heuristics for Prediction

For object tracking applications, the state of a moving object, such as direction, velocity and route, is particularly important. Heuristics can be derived by collecting moving patterns of the tracked objects. For example, if an object's movement is a reflection of the patterns of its whole trip, a sensor node may be able to predict the object's future moving directions by using a directional prediction model. Speed range can be used to replace actual speed since it is difficult to predict accurately. Furthermore, an object's moving route within a detection area could be derived from predictions of velocity and direction, with the geographical knowledge of detection areas.

3.2 Dual Prediction

The basic idea for dual prediction is to have sensor nodes and their cluster heads both calculate the next states of tracked objects. Algorithm 1 and Algorithm 2 show specific actions taken at sensor nodes and cluster heads for predicting an object's future movement. The sensor nodes do not send an update of object movement to its cluster head unless it is different from the prediction. In addition, no prediction values need to be sent from cluster heads to sensor nodes. However, the saving of long distance transmissions between a sensor node and its cluster head comes with a small price, i.e., transfer of moving history from a current node to the destination node. As we will show later in the performance evaluation, this cost is well justified because it consumes less power for transmission to a neighbor sensor node and it occurs only when the tracked object moves into a new detection area.

Algorithm 1 Prediction algorithm at sensor nodes.

Incoming Message: Hist_Msg($Hist$)
Local Variables: Sen_Read, $Pred$
System Functions: $Predictor()$
Procedure:

{Once the object enters the detection area, the sensor predicts object's movement from history}
$Pred \leftarrow Predictor(Hist)$
while object is inside the detection area **do**
  monitor the object, record the sensor readings to Sen_Read
end while
if ($Sen\_Read \neq Pred$) **then**
  Send Update_Msg(Sen_Read) to cluster head
end if
{Calculate object's movement history from the previous history and movement in its detection area}
$Hist \leftarrow (Sen\_Read, Hist)$
send Hist_Msg($Hist$) to destination node

Algorithm 2 Prediction algorithm at cluster heads.

Incoming Message: Update_Msg(Sen_Read)
Local Variables: $Hist$, $Pred$
System Functions: $Predictor()$
Procedure:

while object is inside the area cluster covers **do**
  for object's future movement in sensor i, $Pred \leftarrow Predictor(Hist)$
  wait for the object leaving detection area of sensor i
  if (get Update_Msg(Sen_Read) from sensor i) **then**
    $Hist \leftarrow (Sen\_Read, Hist)$
  else
    $Hist \leftarrow (Pred, Hist)$
  end if
end while

3.3 Prediction Models

Prediction models refer to prediction functions that incorporate heuristics and strategies to predict object movement. In the following, we describe three prediction models based on object's moving history:

- **Constant Model:** By assuming that the object movement in terms of direction and velocity remains the same[2], this approach does not need to record and pass any history data to the destination node.
- **Average Model:** By recording and passing an object moving history, the average model derives its future movement by averaging the history.
- **Exponential Average Model:** Instead of simply averaging the history states, this model assigns more weights to the recent history states.

All the above models may compress the history information into a value, so it can be passed to the destination node without incurring excessive overhead.

4 Performance Evaluation

In this section, we use mathematical analysis to evaluate localized prediction. Firstly, we show the potential performance improvement of a localized sensor network system over a non-localized system. Then, we compare the performance of dual prediction mechanisms with naive (i.e., no prediction), PREMON in a non-localized system in order to filter out the power saving due to localization. *Power consumption* is the metric used in

[2] the route can be calculated accordingly.

our evaluation. The parameters used in our analysis are summarized in Table 1.

Parameter	Description
T	Running time (in seconds) for object tracking
S	Number of sensors in the networks
N	Number of sensors involved in object tracking
C	Average number of sensor nodes in a cluster
K	Number of transmissions and receptions between sensors and their cluster heads
M	Total number of radio turn-on's by sensors
I	Interval between allocated TDMA time slots for a sensor node.
L	Length of a TDMA time slot (L=I/C)
W	The average number of target nodes activated by a current node.
TTL	The period of time a target node stays awake
P	Size of the message in transmissions
D	Distance of transmissio

Table 1. Analytical Parameters.

4.1 Evaluation of Localization Effect

Based on the parameters defined above, the radio components in a non-localized system are turned on for a total number of M ($= \frac{S \cdot T}{I}$) times. However only K time slots are effectively used for communications between sensors and cluster heads, and the remaining $M - K$ time slots are spent in idle state. Localized system tries to reduce the number of idle time slots by paying overhead on target nodes, waken up by the current node and staying idle for TTL. The other power overhead is incurred in low-power paging channels, via which a current node wakes up target nodes and the destination node. However, [18] shows that a paging channel consumes less than 1μW running at full duty cycle. Thus, we only consider power consumed in idle mode of target nodes as the system overhead. Assuming the sensor topology as described in Section 2.2, then $0 \le W \le 5$ (since the object will eventually move into a destination node). Thus, there are $N \cdot W$ target nodes. During each TTL, a target node turn on its radio $\frac{TTL}{I}$ times (and thus is idle for $\frac{TTL}{I} \cdot L$ seconds) before going back to sleep. Hence, the total number of radio turn-on's in target nodes during the running time of the sensor network is $N \cdot W \cdot \frac{TTL}{I}$.

To simplify our evaluation, we adopt the power consumption data of Rockwell's WINS nodes obtained in [13]. For each time slot of L seconds, power consumption at a sensor node for idle state is $E_{idle} = 727.5 \cdot L\ nJ$, and the one for transmission is $E_{Tx} = 771.1 \cdot L\ nJ$. Therefore, power consumption in non-localized system is represented as $E_{nonloc} = K \cdot L \cdot 771.1 + (M - K) \cdot L \cdot 727.5\ nJ$, and one in localized system is $E_{loc} = K \cdot L \cdot 771.1 + Min(W \cdot N \cdot \frac{TTL}{I}, M - K) \cdot L \cdot 727.5\ nJ$, where $Min(W \cdot N \cdot \frac{TTL}{I}, M - K)$ implies the upper bound for the total number of radio turn-on's in target nodes (i.e., radio in idle state).

In our evaluation, we fix some parameters, i.e., $S = 100, T = 100, I = 1, C = 4, TTL = 5$, and hence derive $L = 0.25, M = 10000$. In each subfigure of Figure 3, K is increased from 0 to its upper bound values (i.e., $N \cdot \frac{T}{I}$), and W is varied within its possible values. All the subfigures represent the above comparison with the number of nodes involved in object tracking, N, being assigned to $20\%, 80\%$, and 100% of the total number of nodes in the network, respectively.

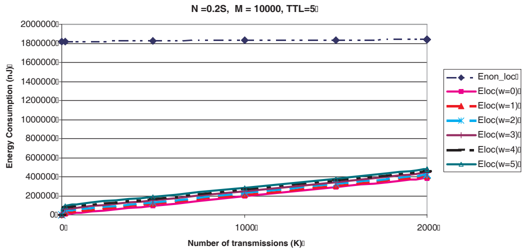
K contributes to the power consumption in transmission. As K increases (in all the subfigures) from 0 to its upper bound values, the power consumption increases slowly in the non-localized system but increases dramatically in the localized system. This is because, for the non-localized system, the extra power consumption incurred as K increases is due to the small difference between radio transmission and idling. As for the localized system, the extra power consumption incurred as K increases is due to the increases of transmissions.

N and W have impact on power wasted in the idle state. As shown in Figure 3(a)-(c)), power consumption for the idle state in non-localized system is much higher than that in the localized system. Only when the values of N and K reach their upper bound values, the power consumption of the localized system reaches the level of the non-localized system. Otherwise, the localized system always outperforms the non-localized system.

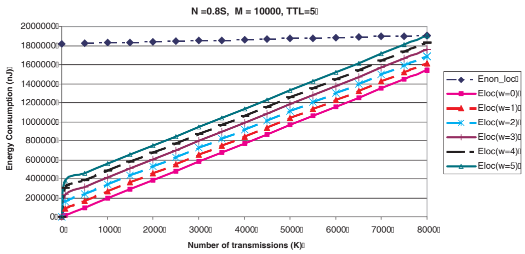
4.2 Evaluation of Prediction Effect

In the following, we analyze the power performance of naive, PREMON and dual predictions. We assume the average distance between neighboring sensor nodes is D_{nbr}, and the average distance between a sensor node and its cluster head is D_{cls}. Our cost formulas are based some numeric parameters obtained in [4]. Power consumed in transmitting or receiving messages is $E_{elec} = 50nJ/bit$. For transmission amplifier to achieve an acceptable ratio-of-signal-noise, $\epsilon = 0.1nJ/bit/m^2$, at a distance D, there is an extra power consumption of $\epsilon \cdot D^2$. Thus, energy consumption in transmitting a P-bit message in a distance D is $E_{Tx}(P, D) = E_{elec} \cdot P + \epsilon \cdot P \cdot D^2$, and energy consumed for receiving this message is $E_{Rx}(P) = E_{elec} \cdot P$. As shown in [5], the energy cost for executing 208 cycles (i.e., roughly 100 instructions) is 1.6 times of the energy consumed for receiving a single bit. Thus, in our evaluation, computation energy consumption per 100 instructions is $E_{comp} = 1.6 \cdot E_{Rx}(1) = 80$ nJ per 100 instructions.

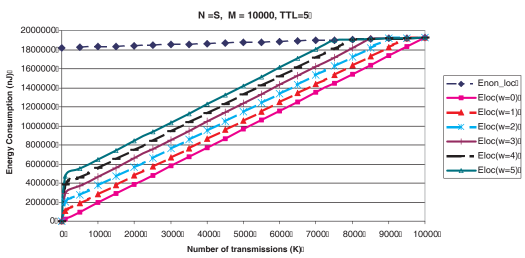
In naive system, sensor nodes report their readings with P_{naive} bits message in their scheduled TDMA slot periodically. Therefore, transmission from sensors and receptions at cluster heads are both $\frac{K}{2}$. Total energy consumed in the naive system is $E_{naive} =$



(a) $N = 0.2S$



(b) $N = 0.8S$



(c) $N = S$

Figure 3. Comparison of non-localized and localized mechanisms

$\frac{K}{2} \cdot (E_{Tx}(P_{naive}, D_{cls}) + E_{Rx}(P_{naive}))$.

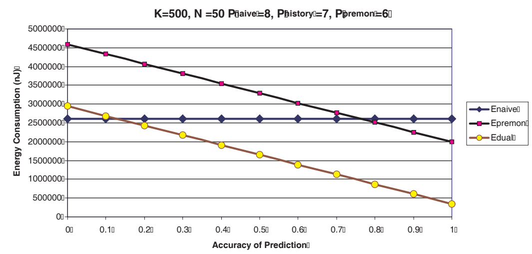
In PREMON, the prediction is performed at the cluster heads, and is passed to the sensor nodes as a P_{premon} bits message $\frac{K}{2}$ times . The sensor node communicates with its cluster head only when the readings differ from the prediction value received from cluster heads. Let the average accuracy for predictions to be α (in terms of percentage of total number of predictions). The total energy consumed in PREMON is:

$$\begin{aligned} E_{Premon} &= \frac{K}{2} \cdot (E_{Rx}(P_{premon}) + E_{Tx}(P_{Premon}, D_{cls})) \\ &+ \frac{K}{2} \cdot (1-\alpha) \cdot (E_{Tx}(P_{naive}, D_{cls}) + E_{Rx}(P_{naive})) \\ &+ \frac{K}{2} \cdot E_{comp} \end{aligned}$$

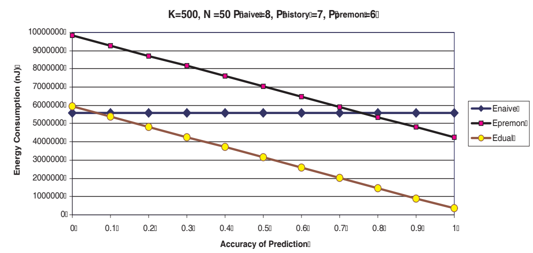
In dual prediction model, to predict an object's future movement, sensor nodes need to obtain the object's moving history from its neighbors. We use P_{hisory} to denote the size of history packet. Like PREMON, the dual prediction approach makes $\frac{K}{2}$ predictions throughout the system running time, T. Thus, each sensor node makes an average of $\frac{K}{2 \cdot N}$ predictions, assuming only one object being tracked by the sensor network. We also use α to denote accuracy of the dual prediction approach. Thus, the total power consumption for dual prediction is as follows.

$$\begin{aligned} E_{Dual} &= N \cdot (E_{Tx}(P_{history}, D_{nbr}) + E_{Rx}(P_{history})) \\ &+ \frac{K}{2} \cdot (1-\alpha) \cdot (E_{Tx}(P_{naive}, D_{cls}) + E_{Rx}(P_{naive})) \\ &+ K \cdot E_{comp} \end{aligned}$$

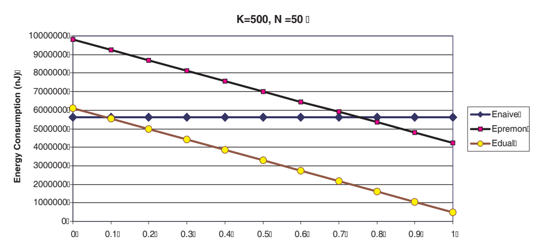
We compare power consumption in naive, PREMON, and dual predictions, by varying D_{nbr}, D_{cls}, and α. Let K and N be 500 and 50, respectively, and assume $P_{naive}, P_{history}, P_{premon}$ to be 8 bytes, 7 bytes, and 6 bytes, respectively. Figure 4 shows evaluation results.



(a) $D_{nbr} = 10, D_{cls} = 25$



(b) $D_{nbr} = 10, D_{cls} = 50$



(c) $D_{nbr} = 20, D_{cls} = 50$

Figure 4. Power consumption of prediction mechanisms

From Figure 4, we observe that the dual prediction needs a much lower prediction accuracy than PREMON to outperforms the naive system. This is because, for PREMON, the reduction of transmitting readings from sensor nodes is at the cost of transmitting predictions from the cluster heads and receiving them at the sensor nodes. Thus, unless cluster head has extremely high prediction accuracy, it's hard to offset the overhead.

By fixing D_{nbr} but increasing D_{cls} (see Figure 4(a) and (b)), the overhead of dual prediction over the naive system is decreased since the accuracy of predictions required to overcome the overhead is decreased, while PREMON prediction remains same. This is because the predictions in dual prediction is enabled by the short distance transmission of history information, while the predictions in PREMON rely on long distance transmissions between cluster heads and sensor nodes.

By fixing D_{cls} but increasing D_{nbr} (see Figure 4(b) and (c)), we found that the dual prediction is the only one affected by this parameter change. This is because the dual prediction transfers moving history between two neighboring sensor nodes while the other two approach purely relies on communications between cluster heads and sensor nodes, which costs much higher than the transmission between neighboring sensor nodes. Thus, even when the distance of neighboring sensor nodes doubled (as shown in Figure 4(b), (c)), the increased overhead is limited.

5 Conclusion

In this paper, we described a localized prediction approach for minimizing global power consumption object tracking sensor networks. The proposed approach makes most of the sensor nodes stay in sleep mode as long as possible and only wakes up needed sensor nodes to ensure seamless tracking of the object. In addition, predictions are performed at both of sensor nodes and their cluster heads to reduce message transmissions. As a result, as long as the prediction models maintain certain level of accuracy (e.g., 10%), a significant amount of power can be saved. Based on mathematical analysis, our performance evaluation shows that the localized prediction significantly outperforms non-localized system and existing prediction approach in power conservation.

As for the next step, we plan to further investigate prediction models based on application requirements and heuristics. In addition, we are looking into the tradeoff of power consumption with various system issues, such as sampling frequency, location models, and objects moving speed etc.

6 Acknowledgments

The authors would like to thank Dr. Gail Mitchell of BBN Technologies for her valuable comments and suggestions which significantly improved this paper.

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] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, pages 263–270, Seattle, Washington, USA, August 1999.

[3] S. Goel and T. Imielinski. Prediction-based monitoring in sensor networks: taking lessons from MPEG. *ACM Computer Communication Review*, 31(5), October 2001.

[4] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocols for wireless microsensor networks. In *Proceedings of the Hawaii International Conference on Systems Sciences*, January 2000.

[5] 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.

[6] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Data on the air - organization and access. *IEEE Transactions of Data and Knowledge Engineering*, June 1997.

[7] D. Lee, W.-C. Lee, J. Xu, and B. Zheng. Data management in location-dependent information services. *IEEE Pervasive Computing*, 1(3), 2002.

[8] 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.

[9] J. Nemeroff, L. Garcia, D. Hampel, and S. DiPierro. Application of sensor network communications. In *Military Communications Conference*, 2001.

[10] G. Pei and C. Chien. Low power TDMA in large wireless sensor networks. In *Military Communications Conference*, 2001.

[11] 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.

[12] 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.

[13] 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.

[14] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. B. Srivastava. Optimizing sensor networks in the energy-latency-density design space. *IEEE Transactions on Mobile Computing*, 2002.

[15] 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.

[16] WINS project. Rockwell science center. http://www.sins.rsc.rckwell.com.

[17] 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.

[18] 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.