Impact of noise on quality of surveillance and energy efficiency of a wireless sensor tracking system

M. Chuah, P. Yang, Y. Xi, F. Zhou

CSE Technical Report
May, 2005

I. Introduction

Recent advances in sensors, embedded devices and wireless communication technologies have enabled distributed micro-sensing through wireless sensor networks [1]. In a wireless sensor network, each node is capable of sensing phenomena, collecting/processing data, and reporting the sensor data back to a data sink. Wireless sensor networks are envisioned to have significant impacts on many applications e.g. environment monitoring, target surveillance and tracking, industrial process monitoring and tactical systems.

Target tracking application is one of the most important applications of wireless sensor networks. The targets to be tracked range from security attacks in the forms of chemical or biological weapons, to moving objects in civil surveillance, to changes in light, temperature, acoustics in environmental monitoring. In spite of the dfferent targets to be tracked and the various signals that need to be sensed, several common characteristics exist in such tracking applications. First, the tracking system should report the location of the target to data sink(s) in a timely and accurate manner. The effectiveness of a wireless sensor tracking system can be measured in terms of its quality of surveillance which is defined as the reciprocal of the average undiscovered distance of a target [3]. Second, the sensed data may be redundant, correlated, and sometimes inconsistent, so sensors may need to collaborate on processing the data and sending a concise digest to data sink(s). Third, the nodes within a wireless sensor network need to operate in an energy efficient manner so that the lifetime of sensor networks can be prolonged. Typically, nodes within a wireless sensor networks run a sleeping protocol that allow them to save on power consumption but at the same time be able to detect and track targets in a timely manner.

There are many sources of noise in a wireless sensor network. Such noise affects the operation of a wireless sensor network e.g. a sensor node may conclude that a target is present when there is none and hence result in a false detection event. In this report, we investigate how the presence of noise affects the quality of surveillance and the energy efficiency of a wireless sensor tracking system. The report is organized as follows: in Section II, we summarized the work done in the area of wireless sensor network tracking system by other researchers. In Section III, we describe two sleeping protocols, and the experiments we did to evaluate the impact of noise on their performance. In Section IV, we discuss some future work that we intend to explore.

II. Wireless sensor network tracking system

A. Target Tracking System

A target tracking system is a wireless networked system of distributed sensors that can detect one or more uncooperative moving target. In some cases, a robot, the data sink, uses the acquired information to capture the target. Besides the common design issues of a wireless sensor network, additional design issues that should be addressed before a sensor-based tracking system can be deployed successfully are:

(i) accuracy of target location estimation

Each low cost sensor node only has limited memory and computing capacity. Moreover, the sensing range of embedded sensor is also small. It is not likely that one can get accurate information from a single node. Hence, various nodes need to collaborate to determine accurately where the location of the target is.

(ii) limited energy resources

In many situations, neither changing power sources nor re-deploying the entire motes system is a feasible solution. On the other hand, the power constraints are even more critical than the memory and bandwidth constraints in the future since battery technology advancement is slower than advancements in improving memory or bandwidth.. Thus, one often needs to do a tradeoff between the accuracy in the location estimation with energy efficiency when designing a sensor-based tracking system.

(iii) timely delivery of sensor data

Since the target moves, any sensor data that is not timely delivered to the data sink will not be useful. Thus, it is important that the sensor network architecture and its routing protocol be designed to meet the requirements of the sensor application in delivering relevant sensor data in a timely manner.

B. Related work

Several system architectures have been proposed to implement the vehicle tracking system [2][3][4]. In general, there are two architectures: a single-tier network, and a hierarchical network. A fully implemented vehicle tracking and autonomous interception system is discussed in [2]. The whole tracking procedure can be divided into three phrases: (i) calibration and sensing, (ii) generation and sharing of local detection reports, and (iii) leader selection and position estimation. Since the magnetic field is sensitive to the change of environment, magnetometers are integrated into the nodes to detect the presence of target. When a filtered version of the magnetic readings is greater than a threshold, a local report will be generated and broadcasted in its neighborhood. The data in the local report contains the sensing data, and the location of the node itself. The packet should not be long since the airlink bandwidth resource is scarce. After hearing the local report from its neighbors, a node elects itself as leader if it has the maximum magnetometer magnitude. Then, the elected leader propagates the aggregated data to the pursuer by using the landmark routing algorithm. On the pursuer’s side, entity disambiguation service is triggered when it receives the aggregated data so as to distinguish the presence of an intruder from other objects, such as the pursuer itself or noise.

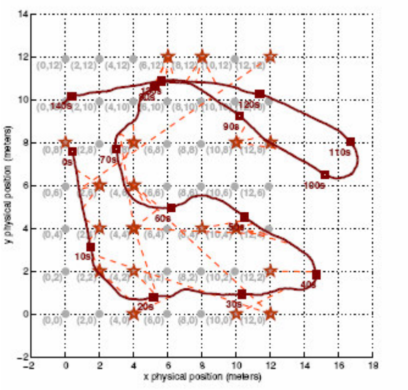

Fig. 1 Intruder tracking [2]

In the PEG system [2], UC Berkeley’s Mica2Dot motes which have a ATMEGA128L microcontroller with 128KB memory. The radio chip is Chipcon CC1000 with 2kB/s bandwidth. The physical phenomena used for object detection is the changing of magnetic field. A total of 100 motes were deployed in a 400 square meter field, the average time for the pursuer to catch the intruder is 26 second based on the results of half a dozen runs. The system is configured such that half of the airlink bandwidth is used for local detection report and the other half is used to support system wide operations e.g. tree building and routing. Fig. 1 shows the trace in the tracking process. The latency incurred in relaying the intrusion location estimate to the pursuer is negligible since the landmark routing algorithm is efficient. One limitation of this system is that there exists the maximum speed of the target that can be detected, which is caused by the bandwidth limitation.

In [2], however, the issue of efficient energy consumption is not considered. In [3], the authors quantify the tradeoff between power conservation and quality of surveillance and evaluate the performance of different energy saving protocols. Several new metrics are defined: (i) the quality of surveillance (QoSv) which is defined as the reciprocal of the distance traveled by the target before it is detected by a sensor node, (ii) Average Linear Uncovered Length (ALUL) which can be (a) a metric denoted as ALUX(x, theta), which is measured at a location x average over all orientation angle, or (b) a metric that is averaged over a two-dimension area (ALUL(X)) with the assumptions that target move continuously and only move for a short distance before being detected. A third metric denoted as ESIL(x) is defined where ESIL(x) is an approximation of ALUL(x). ESIL(x) measures the expected singular intersection length of an area (x). Based on these metrics, this paper evaluated the performance of several sleeping protocols (RIS[7], PEAS[8], PECAS[3], GAF[9], MESH[3]), with the emphasis on the evaluation of a protocol called PECAS which is modified from PEAS. A new parameter, WORK_TIME_DUR, is introduced in PECAS to ensure that each node stays in the working mode for only a period, not its lifetime. Their simulation results show that the performance of PECAS and GAF (measured in terms of QoSv and relative energy saving) are better than other protocols. In addition, a proactive wakeup algorithm was also proposed in [3] to avoid missing target detection when the nodes are in sleep mode. In existing protocols such as RIS, GAF etc, a node switches its states either according to (a) time (RIS), (b) node position (GAF, MESH) or both of them (PECAS), which may result in not detecting a bypass intruder if a node is in the sleep mode. The idea of proactive wakeup is to set each sensor node in one of four states according to its distance to the target, so that the nodes can be woken up in advance. This scheme can work together with any of the sleeping protocols discussed before to meet both the energy-saving and precision requirements.

In [3], the sensor nodes are organized into a flat network but in some scenarios, it will be more beneficial to organize the nodes in a hiearchiacal manner. For example, in [3], all the nodes in fully tracking mode (that is, with strongest level of sensing magnitude) broadcast so that other nodes can update their states according to their distances to the target. One can reduce the overhead of such communication cost by having a clusterhead send the control messages to alert other sensor nodes. A hierarchical sensor network typically consists of a two-layer structure where the lower-tier consists of sensor nodes with limited bandwidth, memory and power capacity and the higher-tier consists of more powerfual nodes that act as clusterheads. Each clusterhead takes the responsibility to collect data from lower tier nodes, and report the information to the sink. Some signal processing may also be done by the higher-tier nodes which have more powerful CPUs, more memory etc. Both static and dynamics approaches can be used to select a clusterhead and build a cluster. In the static approach, the clusterhead and the nodes assignment to a cluster are pre-determined during deployment, while in the dynamic approach, the organization of a cluster is based on the current status of the nodes and the environment. Generally speaking, a dynamic clustering approach is preferred since it is more flexible and adaptive. The only concern is its complexity which may lead to more delay and increased hardware budget for CPU, flash memory etc.

A dynamic cluster formation scheme is described in [4]. Ideally, the node which is closest to the target should be selected to be the clusterhead. When a clusterhead has been selected, it sends a message to let other nearby sensor nodes join its cluster. Once clusterheads are selected within an area, Voronoi diagram, a concept from the graph theory, can be used to divide the whole detection area into several cells (Fig. 2). Each cell is exactly the area that is closer to a particular clusterhead than any other clusterheads. In other words, when a the target is in the Voronoi cell of a particular clusterhead, CH_i, then that clusterhead, CH_i should be selected to report any sensor data related to the target to a data sink. However, deciding whether or not a target is in the Voronoi cell is tricky due to the noisy environment. So, a probabilistic estimate is used; each clusterhead, CH_i, , computes the probability of a target being in its Voronoi cell based on its estimated distance, d, to the target. If the distance from the target to cluster head is smaller than half of the distance from CH_i to a nearest neighbor CH, then the probability Pr(i|d) is 1. If the distance is larger than the distance from CH_i to the farthest vertex of its Voronoi cell, then, the target is considered to be out of the cell. In other words, the probability is 0. Otherwise, the probability is computed as gain/(gain+loss). The variable, gain, is incremented whenever a sample lies within the Voronoi cell of CH_i while the variable, loss, is incremented whenever a sample lies within the Voronoi cell of other clusterheads. A total of 360/resolution samples are taken on the circle that is centered at CH_i and has a radius, d. Each cluster head determines a backoff time that it will wait before transmitting its intention to be the reporting clusterhead using the following equation after determining Pr (i|d)

$$D = W_{\min} + (W_{\max} - W_{\min}) \cdot (1 - \Pr(i \mid d)) + U(W_{ran})$$

where Wmin and Wmax are the minimum and maximum backoff timer values. U() is the uniform distribution in the range [0, Wran-1] that helps to reduce collision if Pr(i|d) are the same for several cluster heads. The above equation shows that a clusterhead with the largest Pr(i|d) will be likely to report first.

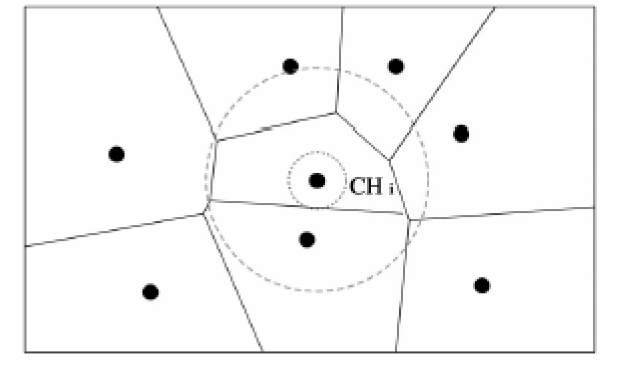

Fig. 2 Voronoi Diagram of CHi [4]

The simulation results in [4] show that the proposed Voronoi diagram-based approach can render accurate estimates of target location as a result of better quality data and less collisions are incurred.

III. Impact of noise on the performance of a sensor tracking system

There are plenty of noise sources outdoors, for example, the noise generated by electronic hardware, sounds of human, birds etc. Such noice affects the inaccuracy of the target tracking system. However, in [3], there is no consideration of the impact of such random noises on the various performance metrics that they use to evaluate the target tracking system. Here, we study the characteristics of different sleeping protocols when additive white Gaussian noise model in present in the system. To conduct such a study, we implemented the RIS, PEAS, PECAS sleeping protocols described in [3] in NS-2 simulator. We also duplicated the same simulation scenario described in [3]. We conduct several experiments to help us understand the impact of noise on the quality of surveillance (QoSv) metric defined in [3]. In the first experiment, we evaluate the maximum number of readings each node will observe during its active period that exceed the detection threshold in the presence of different Gaussian noise magnitude. Based on the results of our first experiment, we pick a reasonable Rsample value that yields low false positive rate for subsequent experiments. Rsample is the number of consecutive samples that need to exceed the threshold before an active node generates a local report that an intruder has been detected. In the 2nd experiment, we evaluated the intruder detection time when different sleeping protocols are used. Rsample is set to 3 in the 2nd experiment. In the third experiment, we compare the relative energy saving using the different sleeping protocols in the presence of Gaussian noise. In subsequent subsections, we first describe the two sleeping protocols that are used in our experiments: the RIS and the PECAS schemes. Next, we describe the noise and energy consumption model we used. Then, we elaborate more on the various experiments we conduct and the results we obtained in each experiment.

A. Sleeping protocols

We used two sleeping protocols in our experiments: (i) the random independent scheme, (ii) the PECAS scheme [3]. For the random independent scheme (RIS), each node follows its own sleeping schedule independently. At each node, the time is divided into time slots of equal length T_{slot} . The interval of each time slot is divided into an active and a sleeping period. The duration of the sleeping period is $p*T_{slot}$. p is referred to as the alertness of the RIS scheme. The start point of the time slots at all the nodes are randomly and independently distributed. The PECAS scheme was modified from the PEAS scheme [8]. In the PECAS scheme, when a node wakes up, it begins probing the environment by broadcasting a PROBE message. Any working node upon receiving the PROBE message should respond with a REPLY message. The probing node waits for a duration indicated by Probe_Wait_Time to collect REPLY messages. If no working node is found, the probing node starts working. Otherwise, it returns to the sleep mode. A node remains in the working mode only for a duration equals to Work_Time_Dur. Each node maintains a variable called Next_Sleep_Time to indicate the length of the remaining time that this working node will start sleeping. The information of the Next_Sleep_Time will be included in the REPLY message if this node hears a PROBE message. A probing node chooses the length of its next sleep period based on the information of all REPLY messages that it receives.

B. Noise and energy consumption models

(1) Noise model

.We assume Gaussian noise is present in the sensor network. The received sensor signal strength in the presence of Gaussian noise is as follows:

$$r = \frac{a}{R^{\alpha}} + N * Gaussian(\mu, \sigma)$$

where

r: received signal strength;
a: target signal strength;
R: distance between target and sensor node;
α: path loss exponent (usually, an integer between 2~5);
N: magnitude of noise;
μ: mean of Gaussian distribution;
σ: deviance of Gaussian distribution;

(2) Energy model

*Energy_Consumption = Current*Voltage*Time*

Here, the voltage of a sensor is considered as a constant (set to the voltage of the batteries). The current of a sensor node varies according to the operation modes. In Table 1, we show the value of the current when the sensor is in different modes of each sleeping protocol. For example, there are four modes in the PECAS protocol, and hence there are four different current values (shown in Table 1) according to the currents drawn by the processor and the radio modules.

A much more comprehensive energy model has been introduced in [6] but we do not use that model. Instead, we obtain the following information from the data sheets provided by the manufacturer of the sensor motes we bought:

Table 1: Current used in Different Modes

Processor /Radio Board	MPR400CB	Remarks
Current Draw for Processor	8mA	Active mode
	<15uA	Sleep mode

Current Draw for Radio	27mA	Transmit with maximum power
	10mA	Receive
	<1uA	Sleep
Voltage	3.3V	
Date rate	38.4kbaud/2	Manchester encoded

Based on the above information, we derive the following energy models for the RIS and PECAS protocols in Table 2.

Table 2: Energy Consumption for the RIS and PECAS schemes

Sleeping Protocols	Mode	Voltage	Current	Remarks (modes of process/radio)
RIS	Sleep	3.3V	16uA	Radio module always in sleep mode for RIS
	Active		8.001mA*	
PECAS	Sleep		16uA	(sleep/sleep)
	Transmit		35mA	(active/transmit)
	Active		8.001mA*	(active/sleep)
	Receive		18mA	(active/receive)
	Date rate	38.4kbaud		Manchester encoded
	Packet size	25.0*8 bits		For probe message

(3) Metrics

We used the following metrics in our simulation study:

- Detection time: the time the target is detected the first time;
- Relative energy saving: the percentage of energy saving with sleeping protocols compared to that without sleeping protocol.
- QoSv [3]: the reciprocal of the uncovered distance (the distance traveled by the target before any detection). Since the speed of the target is constant, the uncovered distance is merely a product of the detection time and target speed.

C. Simulation Setup

The simulation environment is as shown in Table 3: Our simulation study includes two parts: (a) the study of QoSv value with regard to different sleeping protocol in the presence of noise, (b) the relative energy saving of different sleeping protocol in the presence of noise. The network field is a free space of size 400mx400m. 800 nodes are randomly distributed within this area. The target appears in the field at a random location. It chooses a random moving direction and moves in a straight line along the direction at a speed of 10 m/s. The simulation will stop once the target is detected. We run 10 times with different target locations and directions, and average the results to obtain one observation point.

Table 3

Parameters	Values
Simulation Area	400m x 400m
Number of Nodes	800
Default Transmission Range	55.8m

Sensing Range	20m
Mac	802.11DCF
Target Speed	10m/s
Sampling Interval of Sensor	0.001s
Path loss exponent (α)	3
Gaussian (μ, σ)	Gaussian (0, 1)
Target signal strength (a)	3.2e+5
Time slot for RIS	5.0s
Probe Range for PECAS	55.8m

In Figure 3, we show the locations of 800 nodes deployed in our experiment.

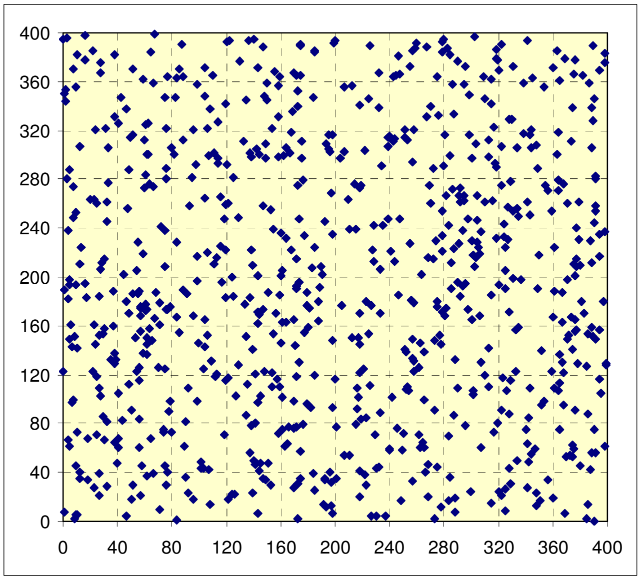

Fig. 3 800 sensor nodes in 400m*400m area

D. Simulation Experiments

(1) Determination of RSample

First, we conduct an experiment to determine a suitable value of RSample for our experiments. RSample is the number of consecutive sensor readings that exceed a certain threshold, Sens_thd, before an active node concludes that an intruder is present. In this experiment, we let the sensor nodes run the RIS sleeping protocol with an alertness of 0.1 for 10 seconds. Each node will record the maximum number of times its sensor readings exceed the threshold during a sleep cycle. T_{slot} is set at 5 second in this experiment. The results in Figure 5 indicate that when the noise magnitude is below 20, at most 3 readings will exceed *Sens_thd* in each T_{slot} in the absence of a target. Thus, we proceed to do the next experiment by setting RSample to 3.

Next, we proceed to determine the detection time with various noise magnitude with RSample set to 1,2, and 3. We repeat each scenario with a certain noise magnitude 6 times by setting the target at a different location each time. The sleeping protocol used is still RIS with an alertness of 0.1. The detection time results were plotted in Figure 5. The results indicate that the detection time is quite stable even with different noise magnitude below 15 when RSample is set to 3. So, for subsequent experiments using the RIS scheme, we set RSample to 3. We repeat the same experiment for the PECAS scheme with Work_Time_Dur set to 10 seconds. Our results indicate that a suitable RSample value is 5 for a noise magnitude less than 15.

(2) Detection Time and QoSv vs Noise Magnitude

We set RSample to 3, and repeat the experiment above using different alertness for RIS. The detection time and QoSv results for RIS with different noise magnitudes were plotted in Figures 6 & 7. For the RIS scheme, the detection time decreases as alertness increases but the decrease is not significant when the alertness increases from 0.2 to 0.4. The QoSv of the RIS scheme increases with increasing alertness. With simple filtering, the detection time and QoSv are not sensitive to the noise magnitudes. We repeat the same experiment using the PECAS scheme. The QoSv results for PECAS-10s with different noise magnitudes are plotted in Figure 8. In this experiment, RSample is set to 5 for the PECAS-10s scheme. Our results indicate that in the presence of noise, the QoSv for PECAS-10s does not change much when the probing range is varied from 55.8 m to 30m but increases when the probing range is set to 20m. The PECAS-10s scheme has better QoSv than that achieved by the RIS scheme when the probing range is set to 20m. For other probing range settings, the QoSv achieved by PECAS-10s are poorer than that achieved using the RIS scheme with alertness 0.1,0.2, and 0.4.(compared Figure 7 with Figure 8).

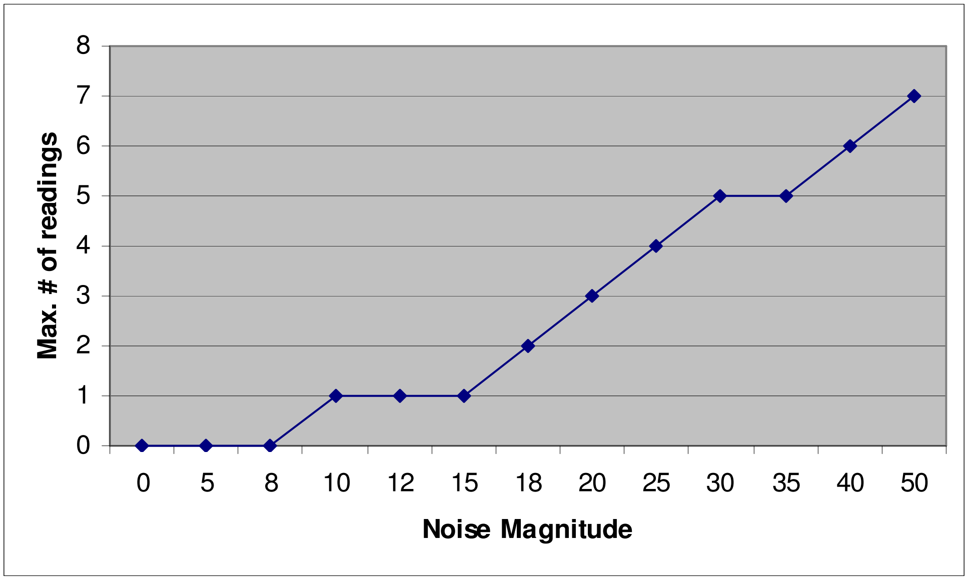

Figure 4: Maximum number of readings exceeding sens_thd vs noise magnitude for RIS-0.10

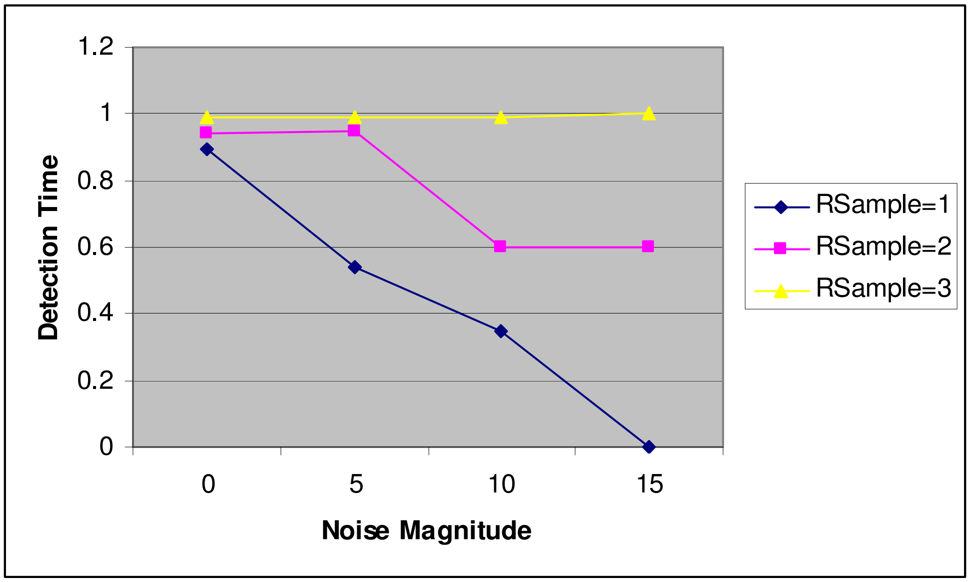

Figure 5: Detection Time vs Noise Magnitude using different RSample (RIS with alertness 0.1)

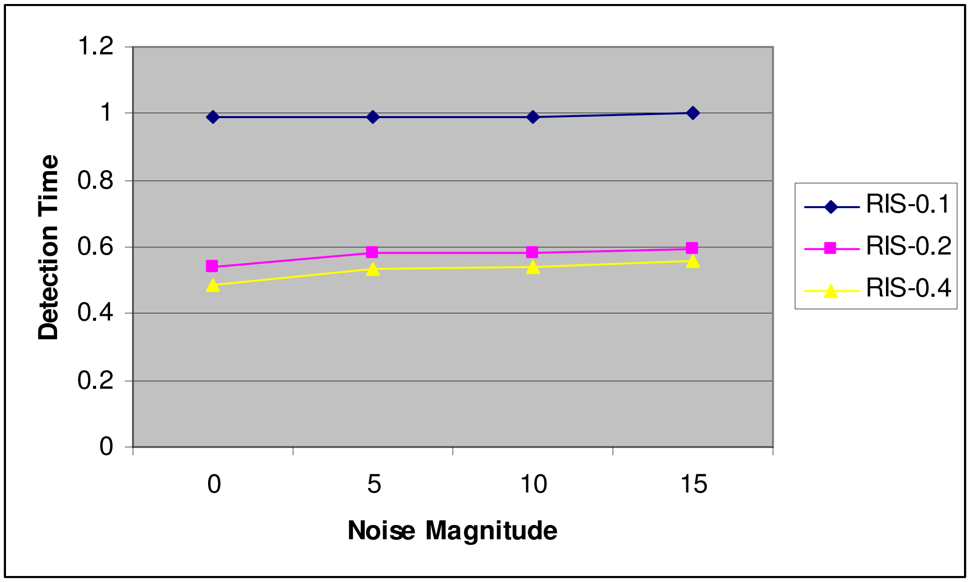

Figure 6: Detection Time vs Noise Magnitude (using RIS)

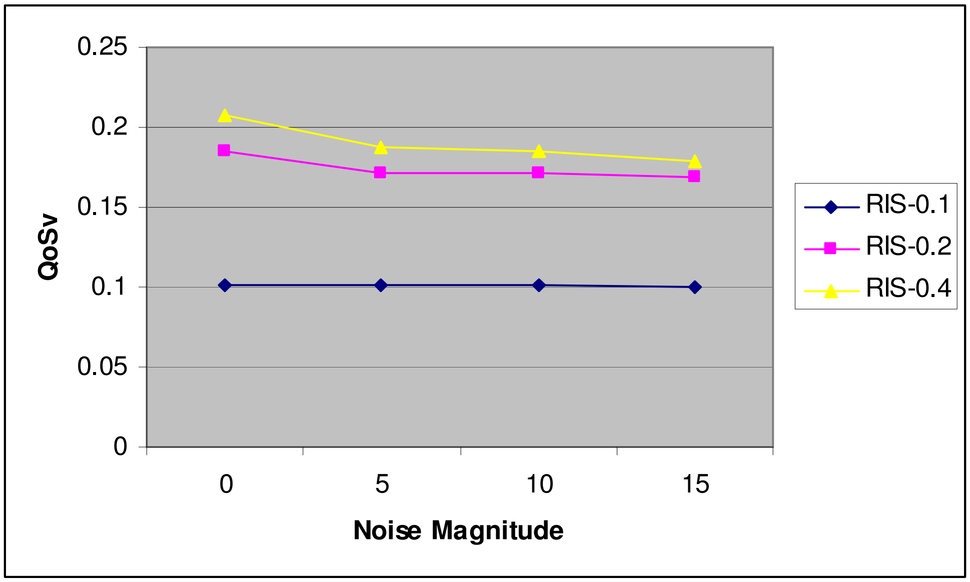

Figure 7: QoSv vs Noise Magnitude (using RIS).

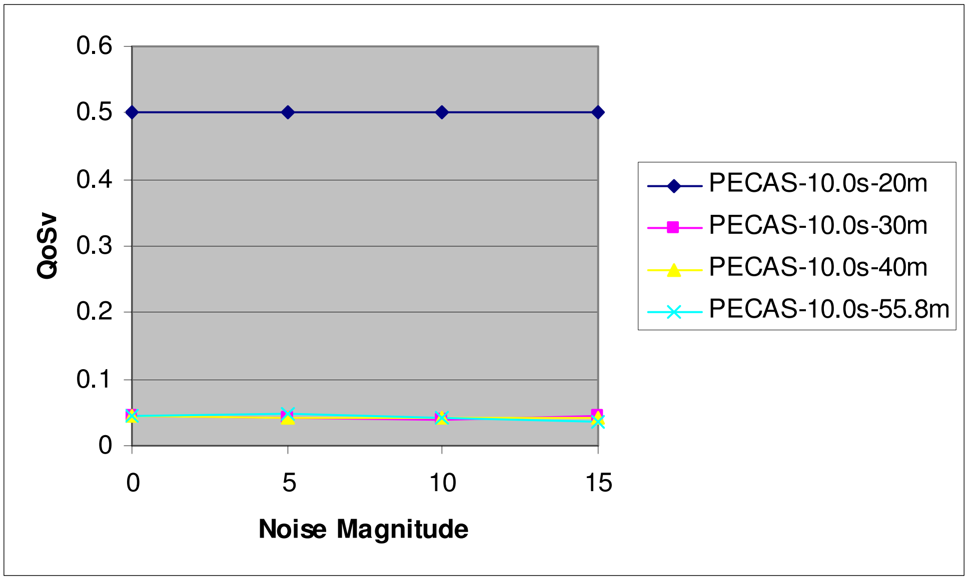

Figure 8: QoSv vs Noise Magnitude with Different Probing Range for PECAS-10s scheme

(3) Relative Energy Saving vs Noise Magnitude

We are interested in knowing if the presence of noise will increase the energy consumption due to false positive events. We are also interested in knowing the relative energy saving of different sleeping protocols. We first plot the relative energy saving for the RIS and PECAS-10s scheme in the absence of noise in Figures 9 and 10 respectively.

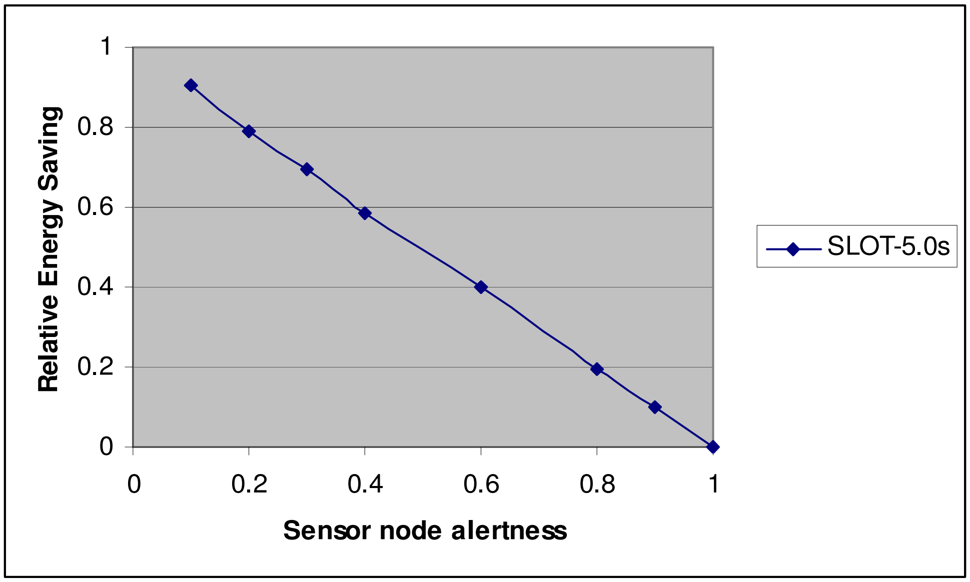

Figure 9: Relative Energy Saving vs Alertness without noise for RIS scheme.

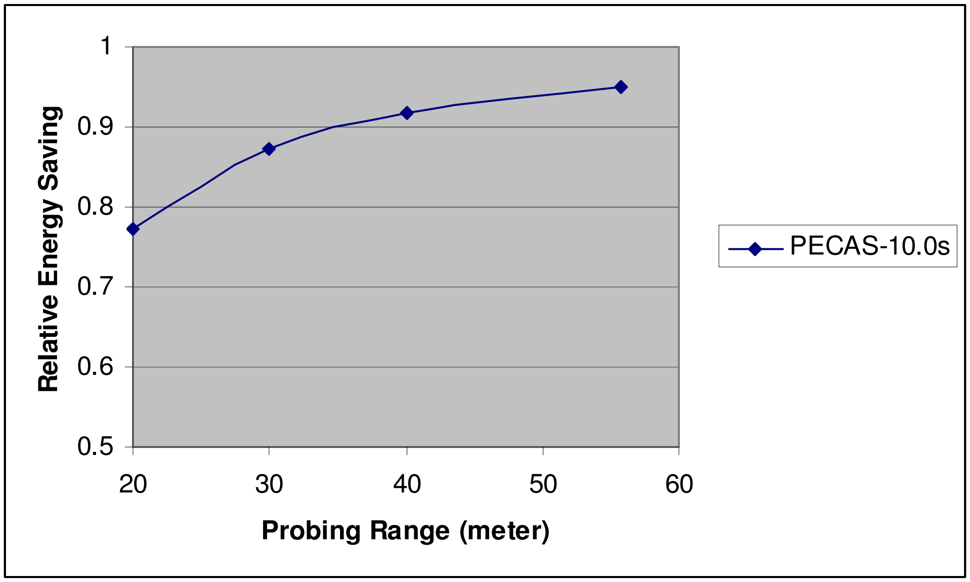

Figure 10: Relative Energy Saving vs Probing Range without noise for PECAS-10s scheme.

Figure 9 indicates that the relative energy saving decreases with increasing alertness for the RIS scheme. Figure 10 indicates that the relative energy saving increases when the probing range for the PECAS scheme is increased. In Figure 11, we plot the relative energy saving when RIS with different alertness and PECAS-10s are used, and noise of different magnitudes is present. The probing range for PECAS-10s is set to 55.8m in this experiment. Our results indicate that the relative energy saving does not change much with increasing noise magnitude because false positive events have been reduced to nearly 0 with the simple filtering mechanism. PECAS-10s provides the highest relative energy saving. Combining the requirement for both high QoSv and high relative energy saving, RIS with an alertness of 0.2 looks more promising than PECAS-10s and RIS with alertness 0.1 or 0.4.

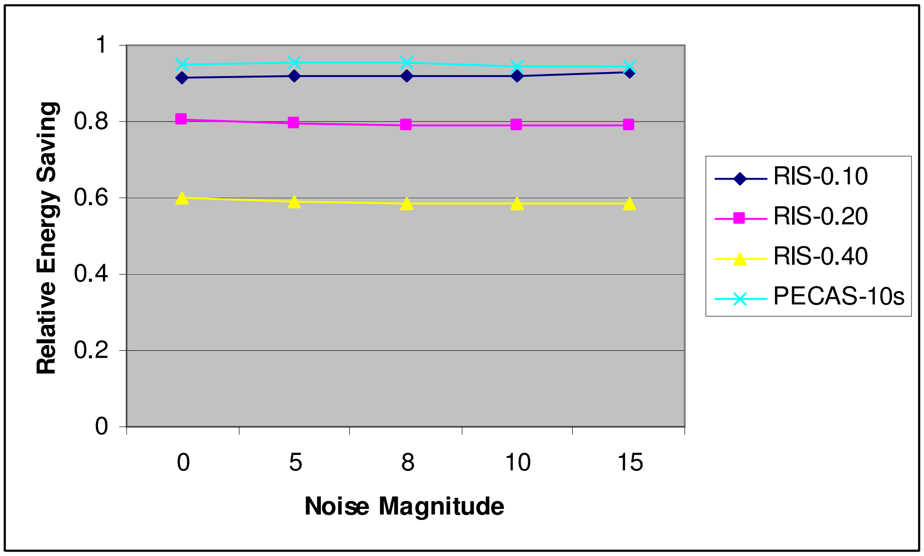

Figure 11: Relative Energy Saving vs Noise Magnitude

The results in this section indicates that with a simple filtering mechanism, sensor nodes can make better decision on whether a target is present, and hence the QoSv is relatively constant irrespective of the noise magnitude present in the environment. The relative energy saving also is relatively constant with different noise magnitudes because there is no false positive event when RSample is chosen correctly.

IV. Future Work

In the sensor tracking system design, location estimate accuracy and energy consumption are two critical issues which must be taken into consideration. In this report, we have investigated the impact of noise on the performance of various sleeping protocols. We concentrate on two metrics: the quality of surveillance, and the relative energy saving. There are several areas that we hope to explore in the near future: (a) impacts of different location estimate protocols on tracking accuracy e.g. in [5], binary decisions are being made by individual nodes to determine if it is close to the target but in [2] the pursuer estimates the intruder's location based on weighted average of readings from individual nodes, (b) we intend to explore a new moving-wave sleeping protocol and compare its performance with existing sleeping protocol, (c) we intend to implement the proactive tracking algorithm proposed in [3] and evaluate the relative energy saving when this algorithm is used, and (d) we intend to explore a sectorized alert protocol where only the nodes within a certain sector are put in alert mode rather than all nodes within the transmission range and compare with the approach we have evaluated in this report. We expect with proactive tracking algorithm, the relative energy saving will drop but the detection time will improve and hence QoSv will improve. Sectorized alert protocol will perform better than the proactive tracking algorithm in terms of relative energy saving but hopefully it will achieve comparable QoSv when compared to that achieved under the proactive tracking algorithm.

V. References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, A survey on sensor networks, *IEEE Communications Magazine*, Vol. 40, No. 8, pp. 102-116, August 2002.

[2] C. Sharp etc, "Design and Implementation of a Sensor Network System for Vehicle Tracking and Autonomous Interception", OSDI 2004

[3] C. Gui, P. Mohapatra, "Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks", Proceedings of Mobicom, 2004

[4] W.Chen, J. Hou, L.Sha, "Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks", IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL.3, NO.3, JULY-SEPTEMBER 2004

[5] S. Pattem, S. Poduri and B. Krishnamachari, "Energy-Quality Tradeoffs for Target Tracking in Wireless Sensor Networks,"in International Workshop on Information Processing in Sensor Networks, 2003

[6] V.Shnayder, M. Hempstead, B.chen, G.Werner and M.Welsh, "Simulating the Power Consumption of Large-Scale Sensor Network Applications", 2004 ACM.

[7] T. Yan etc, "Differentiated Surveillance of sensor networks", ACM International Conference on Embeded Networked Sensor Systems (Sensys), 2003.

[8] F. Ye etc, "PEAS: a robust energy conserving protocol for long-lived sensor networks", IEEE International Conference on Distributed Computing Systems (ICDCS), 2003.

[9] Y. Xu etc, "Geography Informed Energy Conservation for Ad Hoc Routing", Proceedings of ACM Mobicom, 2001.