Extending the Lifetime of Sensor Networks Using Prediction and Scheduling

Jong Chern Lim
UCD School of Computer and Informatics,
University College Dublin,
Dublin, Ireland
Jong.Lim@ucdconnect.ie

C.J. Bleakley
UCD School of Computer and Informatics,
University College Dublin,
Dublin, Ireland
chris.bleakley@ucd.ie

***Abstract*— Power consumption in Wireless Sensor Networks (WSNs) is a very important issue. Using measured sensor network data, this paper shows that it is possible to conserve a significant amount of energy through the proper use of data prediction and node scheduling without a significant loss in accuracy. Results show that it is possible to increase lifetime by up to 2600% at the cost of increasing average error by 0.5°C for temperature or 1.5% for humidity measurements. The four main design issues tackled are clustering, prediction, scheduling, and spike errors.**

Keywords-wireless sensor networks; gaussian predictor; scheduling; entropy;

I. INTRODUCTION

Wireless Sensor Networks consist of nodes which are used to detect or track real world quantities [1]. These nodes are autonomous and are able to self organize into intelligent networks. Each autonomous node contains a micro controller, memory, a radio transceiver, and sensors. The disadvantage of being autonomous is that nodes need to function without an external power source. Most nodes are battery powered. This limited supply of energy makes power consumption a major issue in WSNs.

One of the ways that this problem can be tackled is through the use of scheduling. Fundamentally, scheduling determines when nodes are switched on or off. There are many ways in which scheduling decisions for the network can be made. In this paper a method, called MulS, for developing efficient schedules is presented. The method is assessed using measured data obtained from the Lausanne Urban Canopy Experiment (LUCE) deployment [3].

The general approach is that the WSN first gathers exploratory data which is used to find relationships between the data sensed at different nodes. These data relationships allow prediction of the data for the entire WSN by only measuring the data at a subset of the nodes. Nodes with strong data relationships are clustered together. Next, within each cluster, a node is selected as an active node. Using these active nodes, the value at all other nodes can be predicted. A schedule is made up of multiple sets of active nodes (subsets) with each taking measurements in turn in a round-robin fashion. When a schedule is running only the selected nodes are active while the other nodes are switched off thus saving energy.

This work was supported by Enterprise Ireland.

The novel aspect of this research is the use of multiple subsets. Previous research mostly deals with up to two subsets which, at most, can double network lifetime. Through the use of multiple subsets we aim to increase network lifetime and to share the workload fairly among all the nodes.

As production cost of a wireless node decrease, it is a fair presumption that the network density of deployments will also increase. This makes results presented here relevant for the future as using multiple subsets will be highly advantageous in high density networks.

MulS is designed for use in a two tier networks such as TENET [2]. TENET is a WSN architecture consisting of lower tier nodes and master nodes. MulS provides an effective way to reduce energy consumption of the lower tier motes through the use of scheduling. Results show that MulS is able to give an improvement of up to 2600% in lifetime with an average error of less than 1.5% in humidity and 0.5°C in temperature. In addition, we assess a number of prediction algorithms, showing that multivariate Gaussian prediction is only effective for certain quantities. Finally, we address the problem of spike errors.

The remainder of this paper is broken into parts. In Section 2 we discuss related work and point out the novelty of our approach. In Section 3 we describe MulS in detail. In Section 4 we examine the results and implications they have. Finally, the paper ends with a conclusion and suggestions for future work.

II. RELATED WORK

Scheduling is an important research topic in wireless networks. The purpose of scheduling is to coordinate resourses within a network in order to reach a certain goal. In the case of WSNs the resources are nodes and the goal is usually to conserve energy while maintaining an acceptable degree of sensing accuracy. Current research on WSN scheduling can be broken into two main categories - the sampling schedule and the communication schedule. Communication scheduling deals with transmission timing. Sampling schedules can be adjusted based on data relationships such as temporal and spatial correlation. This research falls under the sampling schedule category hence communication scheduling is not further elaborated on.

Sampling scheduling uses correlations to reduce the volume of data sent back to the master node while maintaining an acceptable degree of error. Scheduling using temporal correlation has been dealt separately by [10] and [11] (Contour maps). Contour Maps sets thresholds where nodes only send data if it is above a threshold. In [10], a method to evaluate sensor data characteristics, using a Kalman Filter is presented. This allows each node the capability to autonomously adjust its sampling rate.

Work on spatial correlation for scheduling can be broken into two categories; coverage-based and data similarity-based. In the data similarity approach, nodes make decisions based on data correlation between neighbours. For instance, in Contour Maps, nodes suppress data transmission if its neighbouring node is transmitting a similar value. In CAG [12], clusters are formed when the forwarding tree is built using a user-specified error threshold which is sent during the query phase. Nodes join a cluster if their reading is within the error threshold of the clusterheads reading. Both CAG and Contour Maps are data similarity methods with coverage bounds, the drawback of such methods is that data correlations which aren't bounded by distance will be missed.

The coverage approach [13], [9] and [14], tries to maintain a certain degree of sensing coverage over the monitored area while switching off as many nodes as possible. The disadvantage of this approach is that sometimes neighbouring nodes aren't correlated. For instance it is feasible that two nodes with close proximity have a large dissimilarity in readings (e.g if they are separated by a wall). Because of this we decided to investigate the data relationship approach.

The data relationship approach has been dealt with in [5], [6] and [8]. BBQ [5] is a model-based querying approach, which chooses a data acquisition plan for the sensor network to best answer the query. The algorithm first builds a PDF (Probability Density Function) model based on historical data. Based on this model, values are estimated to answer queries. The degree of uncertainty willing to be accepted by the user is defined in the query. For estimates with high uncertainty the system may retrieve updated sensor readings from the sensor network. BBQ also implements a cost model which is able to compare the relative cost of executing different plans.

KEN [6] also uses probabilistic methods to predict data, the major difference is that BBQ is 'pull based' where as KEN is 'push based'. In KEN, data is acquired at a steady rate in order to detect anomalies. A dynamic probabilistic model of nodes in the sensor network and the sink is kept synchronized. Whenever data is not within the error bounds of the model, a node will send the data back to the sink. However this only deals with temporal correlations. In order to use spatial correlations KEN uses distributed clustering at the cost of higher intra-source communication.

Through the use of a multivariate Gaussian model, the system described in [8] builds two subsets where one subset can predict readings in the second subset. Measurements are ping-ponged between the subsets.

Like the systems reported in [8], BBQ, and KEN, the MulS system proposed herein relies on probabilistic methods to predict data. Like KEN it is 'push based'. The major difference with KEN is that MulS runs on multiple subsets in a round robin fashion and is a centralized approach thus no intra-source communication is needed. The advantage of MulS is a huge in increase in lifetime. Through the proper use of scheduling, MulS ensures that the workload is fairly distributed over the whole network.

A similarity with [8] is that MulS relies on multivariate Gaussian for prediction. We show that the multivariate Gaussian predictor is not as efficient when monitoring certain attributes and we give a simple alternative method. We also present a method to evaluate different attributes in deciding which prediction method to use. Finally we propose a technique for reducing error spikes.

III. OVERVIEW OF APPROACH

MulS has two main functions, scheduling and prediction. As shown in Fig.1, a schedule is made up of multiple subsets of nodes which activate in turns in a round robin fashion. It is the purpose of the scheduling algorithm to determine the optimum active nodes for each subset. Prediction methods are used to estimate the values of other nodes base on the active subset.

Once network connectivity is established an exploratory phase is initialized to gather data from every node in predetermined intervals. After the exploratory phase, scheduling is performed. Fig. 2 shows the scheduling method. Using the data obtained during the exploratory phase, nodes are clustered into groups. The purpose is to group nodes with strong data relationships. This allows more accurate prediction to be performed for inactive nodes.

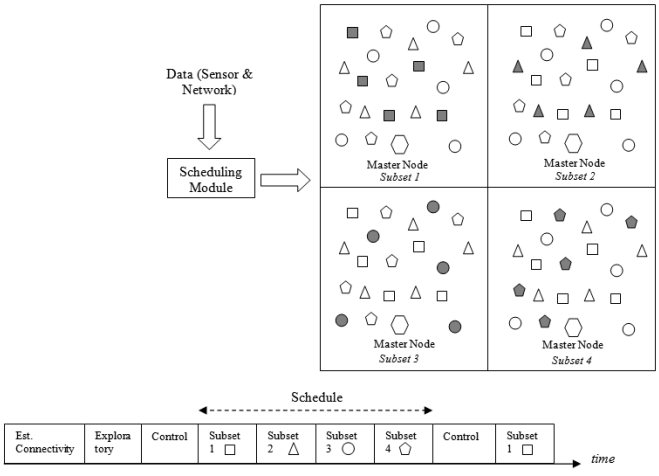

Figure 1. Scheduling Framework (MulS)

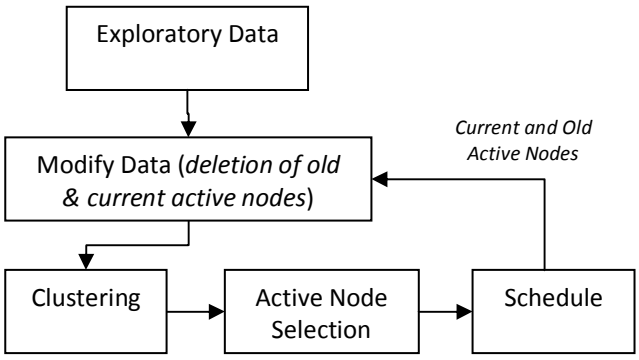

Figure 2. Scheduling Method

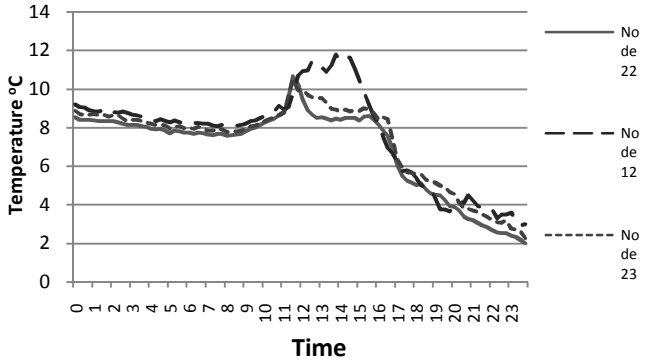

Figure 3. Data Relationship of Three Nodes Measured using Entropy as the Weight (Ambient Temperature)

Fig. 3 shows the data relationship of three nodes. Of the three, Node 22 and Node 23 have a strong data relationship (low entropy) thus they will be grouped together. Node 12 has a big difference in values especially between 1200 and 1600.

A subset is formed by selecting one node to represent each group (which in this case either node 22 or 23 can be picked to be the Active Node). The size of the subsets n, can be determined by the user based on the degree of error and lifetime that the user wants to achieve. To find the second subset, active nodes of the first subset are deleted from the data. Using this modified data, clustering and active node selection is done again. Once selected, nodes from the first subset are reassigned to the second subset nodes based on data relationship. To get the third subset the process is repeated, only this time it is the active nodes of subsets 1 and 2 which are deleted from the data. This process is repeated till the full number of subsets needed is met. Each node is only allocated once in a subset. The subsets are applied in a round robin fashion to maintain accuracy and the graceful degradation of the network.

Normalized cut (N-cut) [4] was used as the clustering algorithm. Nodes are grouped based on weights. Nodes with similar weights have a higher chance of being grouped together. In this paper, three different measures of weight (distance, data difference, and entropy) were used and evaluated.

The distance weight is based on the physical distance between two nodes. Hence nodes which are closer in proximity are grouped together. Data difference is the average difference between the data values of two different nodes. Thus nodes with less difference are clustered together. The third method used for clustering is calculation of the entropy between two nodes. Given the covariance matrix of the data obtained from two nodes Σ, the measure of entropy is:

$$Entropy = \ln(\sqrt{(2\pi e)^2\ |\Sigma|\ }) \qquad (1)$$

After all the nodes have been clustered, the active node which will represent the other nodes within the group for that particular subset is picked. Two methods were used. In the first method, active nodes are selected by choosing the node which is closest to the mean value of the other nodes within the group. The second method is by choosing the node which has the smallest total entropy within the group.

Two methods are used to predict the data which would be obtained by inactive nodes. The first method is simply that every node within the cluster takes the value of the current active node. The second method is through the use of the multivariate Gaussian model constructed during the exploratory phase to predict the data. In [5] it states that if o is observed for attributes O (O being the data obtained during the exploratory phase), then the mean $\mu_{Y|_o}$ (predicted value of the inactive node) over the remaining attributes is given by:

$$\mu_{Y|_o} = \mu_Y + \Sigma_{YO}\Sigma_{OO}^{-1}\left(o - \mu_O\right) \qquad (2)$$

where o is the current observed value of the active node, μ_O and $\mu_Y$is the mean of the active node and the inactive node. Σ_{YO} and Σ_{OO} are formed by selecting the corresponding rows and columns from the original covariance matrix.

Six different clustering and prediction algorithms were studied. Table 1 shows the breakdown of the different - weight, active node selection method, and data prediction method used by the different algorithms. The first four, MG, AVG, ENT, and DIS, test the effectiveness of the different prediction and clustering methods. They only run on a single subset of active nodes. The best clustering and prediction methods, MG and ENT are extended to run in a round-robin fashion. The algorithms are called MGSCHE and ENTSCHE respectively. Both scheduling methods use clustering based on entropy.

The purpose of this paper is to find the best scheduling and prediction method. To achieve these, two assumptions were made. The first is that using multiple master nodes will allow every lower tier node to transmit directly to the master. Secondly, the energy cost of every function (e.g sensing, waking up and transmitting) for every node is the same. These assumptions on the power model allow us to directly relate the percentage of network lifetime improvement to the number of subsets used in a schedule. The simple assumption used herein is that lifetime will increase as the number of subsets increases. It is anticipated that using multiple subsets in a round robin fashion will increase the error.

TABLE I. ALGORITHMS

Name	Weight	Active Node Selection	Data Prediction Method	Scheduling
AVG	Data Average	Mean	Active Node Value	No
ENT	Entropy	Mean	Active Node Value	No
DIS	Distance	Mean	Active Node Value	No
MG	Entropy	Entropy	Multivariate Gaussian	No
MGSCHE	Entropy	Entropy	Multivariate Gaussian	Yes
ENTSCHE	Entropy	Entropy	Active Node Value	Yes

IV. RESULTS & DISCUSSION

The six different scheduling and prediction methods shown in Table 1 were tested using data obtained from [3]. A summary of the dataset is given in Table 2. For AVG the average data of one day was used as the weight.

Fig. 4 shows the average absolute error obtained over the four temperature datasets when using the four predictors (AVG, ENT, DIS and MG). Given that N is the number of nodes in the network and T the total duration of the dataset, then the average absolute error is:

$$E = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left|X_m(i,t) - X_p(i,t)\right| \qquad (3)$$

where i is the node id which runs from 1 to N, t is the time instance, X_m is the actual measured value and X_p is the predicted value.

TABLE II. DATASETS

Name	Date	Attribute	Intervals	Period	Number of Nodes
RH 1st Dec	1/12/2006	Relative Humidity	Every 15 Minutes	16 days	53
RH 1st Jan	1/1/2007	Relative Humidity	Every 15 Minutes	16 days	53
ST 1st Dec	1/12/2006	Surface Temperature	Every 15 Minutes	15 days	53
ST 20th Dec	20/12/2006	Surface Temperature	Every 15 Minutes	15 days	53
AT 1st Dec	1/12/2006	Ambient Temperature	Every 15 Minutes	16 days	51
AT 1st Jan	1/1/2007	Ambient Temperature	Every 15 Minutes	15 days	53

Results show that ENT is the best predictor when measuring temperature. Fig. 5 shows the average absolute error obtained for relative humidity, where ENT is the worst performer. MG is the best algorithm for relative humidity. The performance of both ENT and MG show that clustering using entropy is the best method for these data sets. The question arises of when to use multivariate Gaussian as the predictor and when to use the simplistic method of taking the active node's value as the value of the other nodes within the cluster.

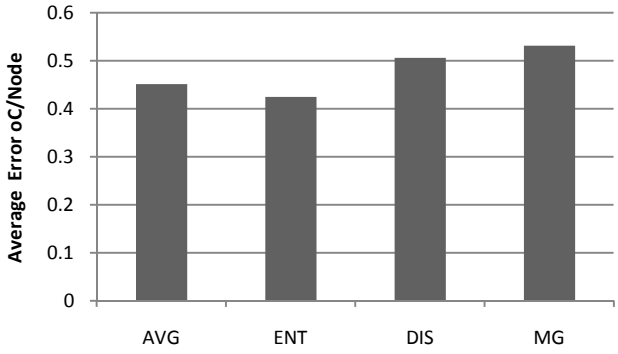

Figure 4. Average Error (Four Temperature Datasets)

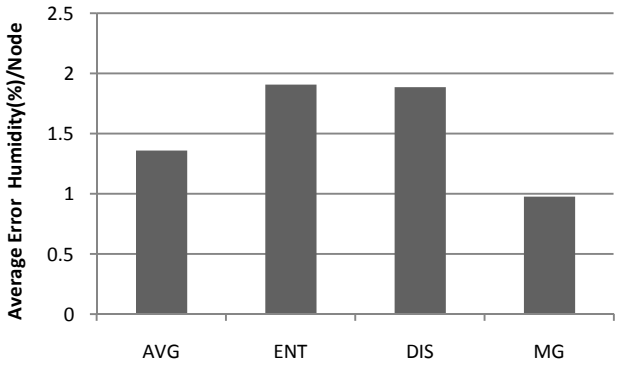

Figure 5. Average Error (Two Humidity Datasets)

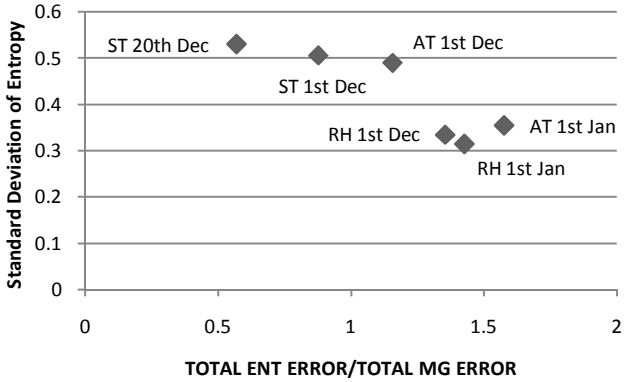

Figure 6. Performance of ENT compared to MG vs. Standard Deviation of Entropy

The trends of the datasets have been identified as the reason for this performance difference between the two algorithms. Further investigation revealed that there is a relationship between their performance and the standard deviation of the entropy (StdDevE) calculated over every node. As shown in Fig. 6, ENT only outperforms MG when the standard deviation is more than 0.5. This results points to two main conclusions; 1) the best method for clustering is using entropy as a weight and 2) the effectiveness of the prediction is governed by StdDevE. Next we tested MG and ENT running on multiple schedules (MulS).

MG and ENT operate using the best predictor nodes within the network hence using multiple subsets increases error. Simulation results show that the average error increase of using ENTSCHE and MGSCHE is only 13.7% and 1.5% respectively.

Fig. 7 shows the performance of MGSCHE and ENTSCHE. The percentage improvement in network lifetime is directly related to the number of subsets used. For instance an improvement of 2600% means 26 subsets were used in a round robin fashion. The number of operating nodes per-subset is found by dividing the total number of nodes (53 nodes) by the number of subsets (26 subsets) and rounding down the answer. In this case the number of active nodes per-subset is 2.

What is interesting about Fig. 7 is that the average error in humidity when running MGSCHE on 2 nodes per subset is only around 1.5%. Which means that at the expense of only average error of 1.5%, the network lifetime can be improved by 26 times. MGSCHE achieves similar results when tested with ambient temperature as shown in Fig. 8. Results show that an improvement of 2500% is possible with an increase average error of less than 0.5°C.

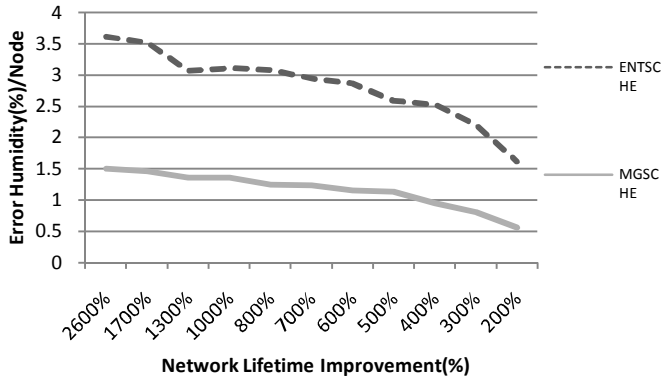

Figure 7. Relative Humidity 1st – 16th January 2007 (RH 1st Jan)

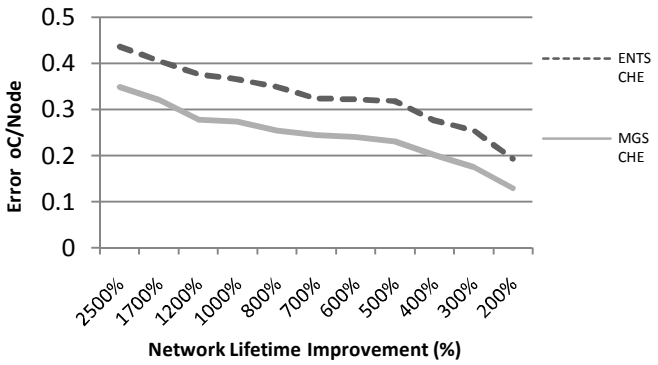

Figure 8. Ambient Temperature 1st – 16th January 2007 (AT 1st Jan)

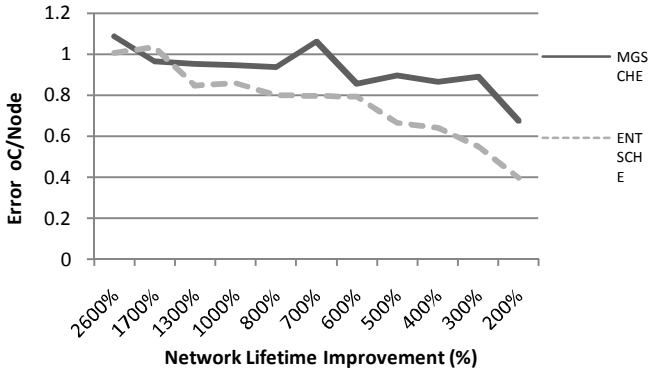

Figure 9. Surface Temperature 20th Dec 2006 – 4th Jan 2007 (ST 20th Dec)

Fig. 9 shows the performance of ENTCHE when used to monitor surface temperature (StdDevE > 0.5). As expected it performs better than MGSCHE. It shows that the maximum increase in network lifetime it can support while maintaining an average error of 0.5°C is only 200%. This is poor when compared with the performance of MGSCHE on other datasets.

Next we take a fine-grained look at the performance of MulS. Fig. 10 shows the difference in average absolute error over all nodes at time *t* when there are only two active nodes compared to when there are 26 active nodes. The average absolute error at each interval *t* is:

$$E_t = \frac{1}{N}\sum_{i=1}^{N}\left|X_m(i,t) - X_p(i,t)\right| \qquad (4)$$

This shows that to limit error spikes to 2% it is best to use more active nodes during certain times of the day. The important thing is to know when to employ additional nodes to avoid spike errors. It was found that most errors occur between 1100 and 1500. Using this knowledge the system can then specify that measurements between 1100 and 1500 should have more active nodes to reduce the maximum error. Similar results were also obtained in test with surface temperature.

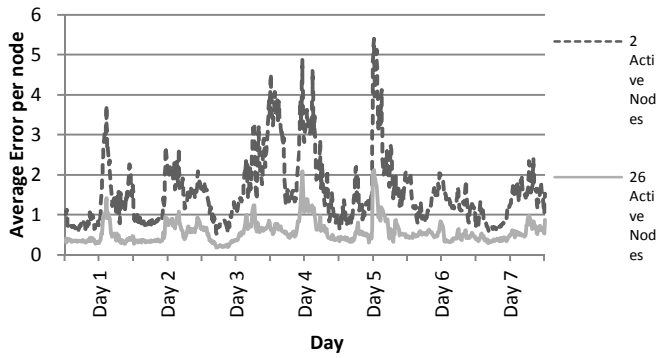

Figure 10. MGSCHE Average Error - Relative Humidity 10th – 16th January 2007 (RH 1st Jan)

V. CONCLUSION & FUTURE DIRECTION

In this research the focus was on developing a means of deriving an efficient schedule from training data. By simulation, it is shown that clustering using an entropy metric performs best. In terms of predictors both MG and ENT have their strengths. The choice of predictor can be made based on the standard deviation of the entropy.

In terms of performance we show that with just two nodes per-subset, giving energy savings of up to 2600%, using MulS-MGSCHE provides an average error of less than 1.5% in relative humidity and 0.5^{o}C in ambient temperature. Even though this is good performance, spike errors can be significant. We have demonstrated a simple method of preventing them simply by adding more active nodes during certain periods of the day.

As initial results are promising we plan to further this research in four ways. We plan: 1) to work on improving the predictors, 2) to consider temporal correlations, 3) to experiment with transmission, 4) to implement the system in a WSN testbed.

ACKNOWLEDGMENT

We would like to thank Enterprise Ireland for funding this project.

REFERENCES

[1] I.Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks”, IEEE Communication Magazine, 40(8), 102–114, 2002.

[2] O. Gnawali, B. Greenstein, K.-Y. Jang, A. Joki, J. Paek, M. Vieira, D. Estrin, R. Govindan, and E. Kohler, “The tenet architecture for tiered sensor networks”. In Proceedings of the 4th ACM Conference on Embedded Networked Sensor Systems (Sensys ’06). ACM Press, November 2006.

[3] http://sensorscope.ep.ch, Sensorscope - Wireless Distributed Sensing System for Environmental Monitoring.

[4] J. Shi and J. Malik, “Normalized cuts and image segmentation”. In IEEE Conference on Computer Vision and Pattern Recognition, pages 731–737, 1997.

[5] A. Deshpande, C. Guestrin, S. Madden, J.M. Hellerstein, and W. Hong, “Model-driven data acquisition in sensor networks”. In Proceedings of the International Conference on Very Large Data Bases (VLDB). Morgan Kaufmann, San Francisco, CA, 2004.

[6] D. Chu, A. Deshpande, J. Hellerstein, and W. Hong, “Approximate data collection in sensor networks using probabilistic models”. In Proc. of the 2006 Intl. Conf. on Data Engineering, Apr. 2006.

[7] P. Liaskovitis,C. Schurgers, “A Distortion-Aware Scheduling Approach for Wireless Sensor Networks”. In: Gibbons, P.B., Abdelzaher, T., Aspnes, J., Rao, R. (eds.) DCOSS 2006. LNCS, vol. 4026, Springer, Heidelberg (2006)

[8] Y. Le Borgne and G. Bontempi, “Round Robin Cycle for Predictions in Wireless SensorNetworks”. In Proceedings of the 2nd International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP ’05), Melbourne, Australia, December 2005.

[9] B. Pazand and A. Data, “An energy-efficient node-scheduling scheme for wireless sensor networks based on minimum dominating sets”. International Journal of Network management, 2008.

[10] A. Jain and E.Y. Chang, “Adaptive Sampling for Sensor Networks”. In Proceeedings of the 1st International Workshop on Data Management for Sensor Net works (DMSN ’04). Toronto, Canada, June 2004.

[11] X. Meng, T. Nandagopal, L. Li and S. Lu, "Contour Maps: Monitoring and Diagnosis in Sensor Networks," Computer Networks, 2006.

[12] S. Yoon and C. Shahabi, “The Clustered AGgregation (CAG) Technique Leveraging Spatial and Temporal Correlations in Wireless Sensor Networks,” ACM Trans. Sensor Networks, 2006.

[13] C.-F. Huang and Y.-C. Tseng, “The coverage problem in a wireless sensor network,” in Proc. ACM Int. Conf. Wireless Sensor Networksand Applications (WSNA), pp. 115–121, 2003.

[14] M. A. M. Vieira et al., “Scheduling Nodes in Wireless Sensor Network: A Voronoi Approach,” Proc. 28th Annual IEEE Local Comp. Net., Bonn/Konigswinter, Germany, pp. 423–29, Oct. 2003.