<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Extending the Lifetime of Sensor Networks Using Prediction and Scheduling</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authors(s)</strong></td>
<td>Lim, Jong Chern; Bleakley, Chris J.</td>
</tr>
<tr>
<td><strong>Publication date</strong></td>
<td>2008-12-18</td>
</tr>
<tr>
<td><strong>Conference details</strong></td>
<td>International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Sydney, Australia, 15 - 18 December, 2008</td>
</tr>
<tr>
<td><strong>Publisher</strong></td>
<td>IEEE</td>
</tr>
<tr>
<td><strong>Item record/more information</strong></td>
<td><a href="http://hdl.handle.net/10197/7103">http://hdl.handle.net/10197/7103</a></td>
</tr>
<tr>
<td><strong>Publisher's statement</strong></td>
<td>© 2008 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.</td>
</tr>
<tr>
<td><strong>Publisher's version (DOI)</strong></td>
<td>10.1109/ISSNIP.2008.4762049</td>
</tr>
</tbody>
</table>
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.

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 resources 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’ whereas 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 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)](image-url)
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 $\Sigma$, the measure of entropy is:

$$
\text{Entropy} = \ln\left(\frac{1}{\sqrt{2\pi e}}\right) \sqrt{\Sigma}
$$

(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 $\mu_{T|O}$ (predicted value of the inactive node) over the remaining attributes is given by:

$$
\mu_{T|O} = \mu_T + \Sigma_{TO} \Sigma_{O0}^{-1} (O - \mu_O)
$$

(2)

where $O$ is the current observed value of the active node, $\mu_O$ and $\mu_T$ is the mean of the active node and the inactive node. $\Sigma_{TO}$ and $\Sigma_{O0}$ 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 MGSCH and ENTSCHC 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Weight</th>
<th>Active Node Selection</th>
<th>Data Prediction Method</th>
<th>Scheduling</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG</td>
<td>Data Average</td>
<td>Mean</td>
<td>Active Node Value</td>
<td>No</td>
</tr>
<tr>
<td>ENT</td>
<td>Entropy</td>
<td>Mean</td>
<td>Active Node Value</td>
<td>No</td>
</tr>
<tr>
<td>DIS</td>
<td>Distance</td>
<td>Mean</td>
<td>Active Node Value</td>
<td>No</td>
</tr>
<tr>
<td>MG</td>
<td>Entropy</td>
<td>Entropy</td>
<td>Multivariate Gaussian</td>
<td>No</td>
</tr>
<tr>
<td>MGSCE</td>
<td>Entropy</td>
<td>Entropy</td>
<td>Multivariate Gaussian</td>
<td>Yes</td>
</tr>
<tr>
<td>ENTSCE</td>
<td>Entropy</td>
<td>Entropy</td>
<td>Active Node Value</td>
<td>Yes</td>
</tr>
</tbody>
</table>

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:

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

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.

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.

TABLE II. DATASETS

<table>
<thead>
<tr>
<th>Name</th>
<th>Date</th>
<th>Attribute</th>
<th>Intervals</th>
<th>Period</th>
<th>Number of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RH</td>
<td>1st Dec</td>
<td>Relative Humidity</td>
<td>Every 15 Minutes</td>
<td>16 days</td>
<td>53</td>
</tr>
<tr>
<td>RH</td>
<td>1st Jan</td>
<td>Relative Humidity</td>
<td>Every 15 Minutes</td>
<td>16 days</td>
<td>53</td>
</tr>
<tr>
<td>ST</td>
<td>1st Dec</td>
<td>Surface Temperature</td>
<td>Every 15 Minutes</td>
<td>15 days</td>
<td>53</td>
</tr>
<tr>
<td>ST</td>
<td>20th Dec</td>
<td>Surface Temperature</td>
<td>Every 15 Minutes</td>
<td>15 days</td>
<td>53</td>
</tr>
<tr>
<td>AT</td>
<td>1st Dec</td>
<td>Ambient Temperature</td>
<td>Every 15 Minutes</td>
<td>16 days</td>
<td>51</td>
</tr>
<tr>
<td>AT</td>
<td>1st Jan</td>
<td>Ambient Temperature</td>
<td>Every 15 Minutes</td>
<td>15 days</td>
<td>53</td>
</tr>
</tbody>
</table>
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 MGSCHEN is only 13.7% and 1.5% respectively.

Fig. 7 shows the performance of MGSCHEN and ENTSCHEN. 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 MGSCHEN 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. MGSCHEN 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.

Fig. 9 shows the performance of ENTSCHEN when used to monitor surface temperature (StdDevE > 0.5). As expected it performs better than MGSCHEN. 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 MGSCHEN 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:

\[
E_t = \frac{1}{N} \sum_{i=1}^{N} |X_m(i, t) - X_p(i, t)|
\]

(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.
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-MGSCHIE provides an average error of less than 1.5% in relative humidity and 0.5°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