

An Energy-Efficient Adaptive Sampling Scheme for Wireless Sensor Networks

Alireza Masoum, Nirvana Meratnia, Paul J.M. Havinga

Pervasive Systems, Department of Computer Science University of Twente, The Netherlands

{a.masoum, n.meratnia, p.j.m.havinga}@utwente.nl

Abstract— Wireless sensor networks are new monitoring platforms. To cope with their resource constraints, in terms of energy and bandwidth, spatial and temporal correlation in sensor data can be exploited to find an optimal sampling strategy to reduce number of sampling nodes and/or sampling frequencies while maintaining high data quality. Majority of existing adaptive sampling approaches change their sampling frequency upon detection of (significant) changes in measurements. There are, however, applications that can tolerate (significant) changes in measurements as long as measurements fall within a specific range. Using existing adaptive sampling approaches for these applications is not energy-efficient. Targeting this type of applications, in this paper, we propose an energy-efficient adaptive sampling technique ensuring a certain level of data quality. We compare our proposed technique with two existing adaptive sampling approaches in a simulation environment and show its superiority in terms of energy efficiency and data quality.

I. INTRODUCTION

A wireless sensor network (WSN) consists of a large number of spatially distributed battery-powered wireless sensor nodes which are usually deployed randomly in an unattended environment. In a data collection mode, sensor nodes continuously sense the environment and send back data to the Base Station (BS) either directly or through some intermediate nodes for further analysis. Efficient energy consumption has the highest priority in WSNs in order to allow network operate for a long time.

Dealing with sampling policies is one of possible approaches for energy management. Determining which sensor nodes and how often should collect data in such a way that application quality of service requirements are satisfied is the challenge faced by resource management solutions. To do so, exploring data correlation is a promising concept which is widely utilized in data collection policies [1]. Adaptive sampling is one of the most comprehensive ways of data collection which tries to readjust sampling frequency and sensor node scheduling from time to time, in order to adapt to changes resulting from high system dynamics.

Current adaptive sampling strategies mostly aim to make a trade-off between being energy-efficient and having high data quality. These approaches are interested in tracking every change in data readings to satisfy data quality requirements. In other hand, in most of monitoring applications the environmental parameters such as temperature usually follow a normal distribution and thereby the data fall in a specific range. In these applications, having data in a given range is interpreted as normal. Existing adaptive sampling techniques

ignore this property and change their sampling frequently upon detection of these changes even when measurements fall within the given range tolerated by applications.

In this paper we target applications that can tolerate changes in sensor values as long as measurements falling out of a specific range are reported. By focusing on these applications, we propose an energy-efficient adaptive sampling mechanism, which employs spatio-temporal correlation among sensor nodes and their readings to determine which nodes and how often should sample and transmit their measurement. The main idea behind our approach is to carefully select a dynamically changing subset of sensor nodes to sample and transmit their data. To do so, each sensor node participates in sensor node scheduling procedure and shows its interest to be sampling node.

The rest of this paper is structured as follows: Section II reviews related work on adaptive sampling algorithms. Network architecture and our spatio-temporal correlation based adaptive sampling mechanism are described in Section III and IV, respectively, while simulation results and performance evaluations are presented in Section V. Finally, Section VI provides the conclusion.

II. RELATED WORK

Current adaptive sampling techniques utilize temporal, spatial, or spatio-temporal correlations to adapt sampling frequency. Padhy et al. [2] consider temporal correlation while defining a utility-based sensing and communication protocol. They model temporal correlations as a piecewise linear function and use a predefined confidence threshold to find an appropriate sampling frequency. Each node uses a linear regression model for its prediction. Time series forecasting methods are employed by [6] to predict sampling and transmission rate. Wood et al. [7] propose an architecture for context awareness which uses prediction of future contexts to minimize the energy required for sensing them. To do so, they make a tradeoff between energy consumption and context identification accuracy. A neural network based adaptive sampling approach is proposed in [10]. To predict sampling period, historical time delay values, network load, and throughput are considered as input layer vectors for the neural network.

Our approach has two main differences with existing approaches. Firstly, our approach considers monitoring applications which are not interested in all value changes and can tolerate them as long as measurements falling out of a specific range are reported. Secondly, each sensor node can

participate in sensor node scheduling process. In our approach, each sensor node defines an interest parameter which can be utilized in sampling node selection procedure.

III. PRELIMINARIES

Before explaining our adaptive sampling mechanism, we first explain our network setup and introduce the basic definitions and concepts used in this study.

A. Network setup

We employ a two-tier network model. In the first tier, we consider a network composed of N stationary sensor nodes which are deployed over n disjoint clusters. The location of sensor nodes, cluster heads, and the base station are fixed and are known a priori. All sensor nodes are homogeneous in terms of their sensors. Cluster heads are more powerful than sensor nodes in terms of their processing capability. The second tier of our network model is composed of n clusters. We partition each cluster into several sub-clusters based on correlation that exists between readings of sensor nodes. For each cluster, there is a set of sub-clusters. Each node only belongs to one sub-cluster and the information about sub-clusters is known only by the local sensor nodes located in that sub-cluster.

B. Data Quality Metric

Data quality is an important QoS parameter in this work. Current adaptive sampling approaches are interested in keeping track of every change in sensor readings to satisfy data quality requirements. They define a function to predict data when no real measurement is available and define data quality as the difference between predicted data and real sensor measurement. In fact, They calculate the error between predicted data and real measurement. If this error is larger than a predefined threshold, it means that the environmental conditions cannot be well-covered by the current sampling rate/nodes and sampling frequency need to be changed.

There are, however, a number of applications that are not interested in keep tracking of all changes as long as data measurements remain within a predefined range. Although in these applications no change within the defined range is of interest, sensor measurements closer to the boundaries of the pre-defined range are more important to be reported than the rest. The reason behind this is that it is quite likely that sensor measurements closer to the boundary indicate a trend, following which next measurements fall out of the pre-defined range. Focusing on this type of applications, we employ a different data quality metric. Let us assume that application at hand introduces $[Lower\ Upper]$ as the predefined range and is interested to be notified if sensor measurements fall outside this range. We employ definitions listed in Table I to determine whether a prediction is correct.

C. Inbound and outbound regions

Considering the predefined range, sensor node measurements, denoted by R_i can be categorized in three different regions shown in Fig.1:

TABLE I
PROPOSED CONDITIONS TO EVALUATE DATA PREDICTION ACCURACY

Real Data	Prediction	Prediction Result
$R < Lower \ \ R > Upper$	$Lower \leq X \leq Upper$	Not-Accurate
$R < Lower \ \ R > Upper$	$X < Lower \ \ X > Upper$	Accurate
$Lower \leq R \leq Upper$	$Lower \leq X \leq Upper$	Accurate
$Lower \leq R \leq Upper$	$X < Lower \ \ X > Upper$	Not-Accurate

$$\begin{cases} \text{Region B} & R_i > Upper \\ \text{Region A} & Lower \leq R_i \leq Upper \\ \text{Region C} & R_i < Lower \end{cases}$$

Region *A* indicates inbound measurements, while *B* and *C* regions are out of bound regions. Once sensor measurements get closer to the boundaries, it is quite likely that next measurements fall out of the boundary (in case of region *A*) or fall in bound (in case of regions *B* and *C*). Being further from boundaries indicates that sensor measurements are becoming more stable and predictable.

D. Data Sampling

In this work, we define sampling period as the duration in which sampling nodes must collect data from the environment. For collecting data, two different sampling policies for sampling and non-sampling nodes are considered:

1) *Temporal Correlation based Adaptive Sampling*: in which sampling nodes utilize temporal correlation based adaptive sampling approach proposed in our previous work [9] to gather data.

2) *Forced Sampling*: In our approach, only sampling nodes report their data. Therefore, to have a balanced view of the entire network, we introduce periodic forced sampling times in which non-sampling nodes also report their measurements. It is worth mentioning that number of forced samples is negligible in comparison with the maximum number of samples which can be taken in sampling time period.

E. Concepts

In what follows, we will define terms used in the rest of this paper. To be able to measure the closeness of sensor measurements to the boundaries and to assign them some degree of importance, we need a quantity metric. Therefore, we define a constrain that the difference between sensor measurement and predicted data cannot be larger than the maximum difference (Max_Diff) between borders of the predefined data range ($Upper$ or $Lower$) and its mean ($Upper-Middle$ or $Lower-Middle$). As sensor node readings get closer to the boundaries, this difference becomes smaller. We utilize d as a new coefficient parameter to give weight to sensor measurements depending on their closeness to the boundary and to the mean. We define d as:

$$d = \frac{R - Middle}{Upper - Middle} \quad (1)$$

Value of d changes for each measurement depending on in which regions it resides. Value ranges of d in different regions are as follows:

- Region A: This region includes inbound readings and utilizes d parameter to define two sub-regions:

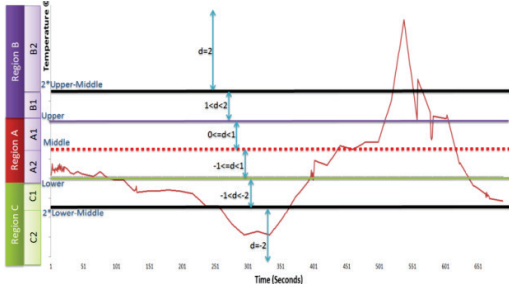


Fig.1. Region definition

- *A1*: whose readings are between *Middle* and *Upper* ($0 < d \leq 1$)
- *A2*: whose readings are between *Middle* and *Lower* ($-1 \leq d < 0$)
- **Region B**: This region includes any value of d bigger than *Upper*. Since we are only interested in measurements that are close to *Upper*, far enough measurements from *Upper* may be ignored. We also ignore any measurement, which falls out of *Upper+Max_Diff*. Considering this upper bound we define two sub-regions:
 - *B1*: in this region, measurements are between *Upper* and ($2 * \text{Upper-Middle}$), therefore d takes values from interval (1,2].
 - *B2*: this sub-region includes measurements whose values are greater than $2 * \text{Upper-Middle}$, for which we fix d to be 2.
- **Region C**: This region includes any value less than *Lower*. The farthest possible measurements for this region fall out of (*lower-Max_Diff*). Considering this lower bound we define two sub-regions:
 - *C1*: in this region, measurements interval is (*Lower*, $2 * \text{Lower-Middle}$), therefore d varies between (-2,-1).
 - *C2*: this sub-region includes measurements whose values are greater than $2 * \text{Lower-Middle}$, for which we fix d to -2.

IV. AN ENERGY EFFICIENT ADAPTIVE SAMPLING APPROACH

Our proposed adaptive sampling mechanism achieves energy efficiency in its data collection through carefully selecting sampling nodes and dynamically changing sampling schedules. Each sampling node tunes its sampling frequency based on the environmental conditions. Sampling nodes are selected using spatial correlation and the closeness of sensor measurements to a pre-defined data range. Temporal correlation among sensor measurements is used to find the best sampling rate. In what follows we explain our approach in details.

A. Sub-clustering

In order to provide a high accurate scheduling, our approach requires to form a number of sub-clusters within each cluster. To do so, cluster head first collects non-periodic samples (in forced sampling times) from all sensor nodes to find out the spatial correlation among sensor nodes' measurements. Cluster head creates sub-clusters following three steps, i.e., data gathering, pre-processing, and K-means clustering.

- **Data Gathering**: The first step involved in creating an accurate sub-clustering is collecting data from all sensor nodes within the cluster. For the data gathering phase, cluster head employs the samples collected during the more recent forced sampling period to generate new set of sub-clusters. We denote the forced samples collected by each sensor node by FD_i , i.e., a vector of consecutive forced samples gathered by node i . At the end of sampling period, sensor nodes calculate mean and standard deviation of their forced samples using the following formulas:

$$\mu_i = \frac{\sum_{j=1}^{NFS} FD_i(j)}{NFS} \quad (2)$$

$$\sigma_i = \sqrt{\frac{1}{NFS} \times \sum_{j=1}^{NFS} (FD_i(j) - \mu_i)^2} \quad (3)$$

where NFS is the number of forced samples. Then they send $\mu_i(t), \sigma_i(t)$ to the cluster head to calculate similarity between sensor node measurements and to perform sub-clustering.

- **Data Preprocessing**: After receiving $\mu_i(t), \sigma_i(t)$ from cluster members, the cluster head defines the inbound and outbound regions and assigns sensor nodes to these regions (as described in Section III) based on their measurements. After that, the cluster head calculates weighted value of sensor node readings. As mentioned previously, measurements closer to the user predefined boundaries have higher importance to be reported than the rest. Therefore, the input to the K-means (being used in the next step) is a combination of similarity between sensor nodes' measurements and the closeness of those measurements to the user pre-defined data boundaries. We call this similarity weighted value (W). The weighted values are defined based on mean of sensor readings and their closeness to the boundaries. According to the Fig.1, for C_2 and B_2 regions, whose readings are too far from *Lower* and *Upper*, respectively, we set a weighted value to be zero. The weighted values for different regions are defined as follows:

$$W_i = \begin{cases} W_i = 0 & \text{If } \mu_i \in B_2 \\ W_i = |\mu_i| \times \frac{(2 \times \text{Upper} - \text{Middle}) - \mu_i}{(\text{Upper} - \text{Middle})} & \text{If } \mu_i \in B_1 \\ W_i = |\mu_i| \times \left(1 - \frac{\text{Upper} - \mu_i}{\text{Upper} - \text{Middle}}\right) & \text{If } \mu_i \in A_1 \\ W_i = |\mu_i| \times \left(1 - \frac{\mu_i - \text{Lower}}{\text{Middle} - \text{Lower}}\right) & \text{If } \mu_i \in A_2 \\ W_i = |\mu_i| \times \left|\frac{\mu_i - (2 \times \text{Lower} - \text{Middle})}{\text{Middle} - \text{Lower}}\right| & \text{If } \mu_i \in C_1 \\ W_i = 0 & \text{If } \mu_i \in C_2 \end{cases} \quad (4)$$

- **K-means Clustering**: After preprocessing and having weighted values, the cluster head executes K-means algorithm for each region periodically (every τ_{SC}) in order to form sub-clusters inside a given cluster. For sensor nodes whose readings are in region *A*, K-means is requested to group sensor nodes in three different sub-clusters whose centers are near to the *Middle*, *Lower* and *Upper*. For sensor nodes whose readings are located in *B* or *C* regions, two sub-clusters are enough.

After sub-clustering, cluster head analyzes defined sub-clusters to determine whether these sub-clusters can be

merged. A key issue here is how two sub-clusters could be examined on merging possibility. First, we need a metric to compare the similarity of two sub-clusters. As it was mentioned in the previous section, going towards data boundaries, less error can be tolerated as error thresholds decrease. The similarity between two sub-clusters is the difference between the mean of these two sub-clusters. If this difference is lower than the minimum distance between the mean of the sub-clusters and the boundaries, then those sub-clusters can be merged. The minimum difference can be calculated by multiplying Max_Diff parameter on minimum value of (d_i, d_j) .

$$|M_i - M_j| \leq [(1 - \max(d_i, d_j)) \times Max_Diff] \quad (5)$$

where M_i and M_j represents the center of a sub-cluster i and j while d_i and d_j shows the closeness of the center of a sub-clusters i and j to the closest boundaries.

B. Selection of Sampling Node

Selection of sampling node procedure chooses a set of sampling nodes in such a way that at least there is one sampling node to monitor the area covered by sub-clusters. After sub-clustering, cluster head broadcasts a small control packet with which informs sensor nodes about their sub-clusters and their centers. Upon receiving this packet, each sensor node decides whether it is an appropriate mode to take the role of sampling node. Since each sub-cluster is selected based on the correlation and similarity among sensor nodes' readings, it is not required for all sensor nodes in this sub-cluster to sample and send data to the cluster head. Therefore, in this step, sensor nodes select the best candidate nodes which can serve as sampling nodes for the next sampling period in a distributed fashion.

In order to select the proper sampling nodes, first, each sensor node exchanges the weighted mean and standard deviation of its last forced samples gathered in last sampling period with other sensor nodes located in its sub-cluster. Upon receiving this data, sensor node finds its interest level IL to other sensor nodes. The IL_{ij} of sensor node i to sensor node j provides a quantity measurement which shows how much sensor node i is interested in selecting sensor node j as a sampling node for its sub-cluster. IL_{ij} (equation 6) consists of two terms; first one is the similarity between two sensor nodes' weighted values. This similarity metric is defined based on the normalized difference between the weighted mean values of two sensor nodes measurements. The second term is the weighted deviation of node i which is normalized by the weighted maximum standard deviation of the given sub-cluster. This term gives an insight about the stability of sensor node readings. Consequently, the interest level IL_{ij} of a sensor node i to select sensor node j as a sampling node is calculated as follows:

$$IL_{ij} = \left(1 - \frac{W_i - W_j}{W_i - W_{Far}}\right) \times \left(1 - \frac{\sigma_j}{\sigma_{max}} \times \frac{d_j}{d_{max}}\right) \quad (6)$$

In order to avoid unproductive computations which may be imposed by either non- or less-correlated sensor nodes, the similarity parameter is only defined for the sensor nodes which satisfy the following conditions:

1. $|\mu_i - \mu_j| \leq (1 - \max(d_i, d_j)) \times Max_err_th$
2. $\begin{cases} (\mu_i \pm \sigma_i) \in A & \text{If } i \in A \\ (\mu_i \pm \sigma_i) \in B & \text{If } i \in B \\ (\mu_i \pm \sigma_i) \in C & \text{If } i \in C \end{cases}$

These conditions implies that readings of two sensor nodes readings are similar if the difference between these readings are less than error threshold and the readings (considering their deviations) is still in the pre-specified regions. When sensor node readings are in the middle of the boundaries, the sensor node can tolerate maximum error which is defined by Max_Diff parameter. The closer to the boundaries, the lower this difference. Therefore, for measuring similarity between sensor node readings, we set the difference between closer reading to the boundary and boundaries (*Lower* or *Upper*) as an error threshold. The second condition implies that sensor nodes readings are similar if considering their deviations, they still stay in the pre-assigned regions. For example if sensor node i is a member of region A , its readings considering its deviation, must stay in A .

Upon calculating ILs , sensor nodes exchange their ILs with other neighboring nodes. Finally, each sensor node has to measure its capability to act as a sampling node for the current sub-cluster. To do so, we introduce sampling degree (SD) which defines a measurement parameter to evaluate the amount of properness of a sensor node to act as sampling node. The appropriateness of sensor node j to act as a sampling node for current sub-cluster depends on different parameters which are defined in equation (7). The first term in this equation shows the sum of the interest levels of sub-cluster members while the second term presents the number of sub-cluster nodes which can be covered by this sampling node. This parameter ($N_{covered}$) is obtained based on the number of sampling nodes whose interest levels are received by the sampling node j . Energy (E_j) is another factor which indicates the available energy of the sensor node j to act as a sampling node. This parameter is derived from the current energy level of the node which is normalized by the maximum energy level of that node. The last term addressed here indicates how well the measurements of the given sampling node can represent the measurements of the sub-cluster. This metric is defined based on the amount of error between the weighted values of sampling node readings and mean of sub-cluster readings. A proper sampling node is such a node whose readings are similar enough to the readings of other cluster members and has enough energy. SD of node j is calculated as follows:

$$SD_j = \sum_{i \in cl_k} IL_i \times N_{covered} \times \frac{E_j}{E_{Max}} \times \frac{1}{(W_j - W_{mean})} \quad (7)$$

When sensor nodes find their SD level, they broadcast it to other sub-cluster members. Thereafter, every sensor node selects the node with the maximum SD value as its sampling node.

C. Data Sampling Period

After sub-clustering and selecting proper sampling nodes, data must be gathered from the environment. During a sampling period, sampling nodes employ temporal correlation

based adaptive sampling mechanism which is proposed in our previous paper [9]. More details about temporal correlation based sampling adjustment can be found in [9].

D. Forced Sampling time

During a sampling period, it is quite likely that correlations among sensor nodes change and sampling node cannot compensate for non-sampling nodes. Therefore, non-sampling and sampling nodes should have the capability to update their correlations, which may change the role of sampling and non-sampling nodes. The updating procedure can be accomplished only in the forced sampling times as all sensor nodes are then awake. At these time stamps, all sensor nodes perform sampling and compare the similarity of their current reading with past readings and based on this similarity will update their status to act as a sampling or non-sampling nodes.

In what follows we explain the situations which can happen:

- If sampling nodes' recent readings fluctuate or if no sensor node is interested in selecting this node as its sampling node, it changes its status to stand-alone sampling node. In this state, sampling node is only responsible for its own readings. Then, it broadcasts a leave message to its neighboring nodes and informs them to change their sampling node. Upon receiving leave message, non-sampling nodes perform distributed sampling selection algorithm to find a new sampling node.
- In case of stability of sampling node readings, sampling node sends a control message containing average and standard deviation of its readings to its neighboring nodes. Upon receiving this message, non-sampling nodes compare their forced samples with readings of the sampling node. If their readings are still correlated and no other sensor sampling node has higher SD values, the non-sampling node does not change its sampling node, otherwise it stays awake and sends sampling selection message.

V. PERFORMANCE EVALUATION

To analyze the effectiveness of our approach, we use a given dataset [5] containing temperature readings collected for a period of two months. The network architecture which is utilized in this simulation consists of several single-hop clusters. We consider only one cluster which consists of five sensor nodes. We consider communication overhead, energy consumption, and data quality as performance evaluation metrics. We compare our approach with two other existing techniques described in [6] and [8]. *ASAP* [8] is an adaptive sampling method which has some similarities with our scheme. In *ASAP*, cluster head or the base station is responsible for selecting sampling nodes, which performs periodic sampling and sends its measurements to the cluster head. Non-sampling nodes only collect data at forced sampling periods. We also combine *ASAP* with the temporal correlation based adaptive sampling presented in [6] and call it a hybrid approach. Fig. 2. shows measurements of node 14, which are close to the boundaries and experience some

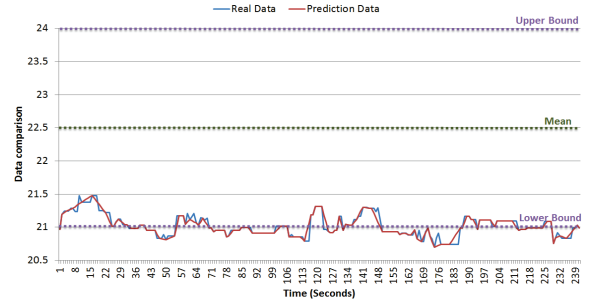


Fig. 2. Comparison between real data and prediction data for sensor node 14 (readings around the lower bound)

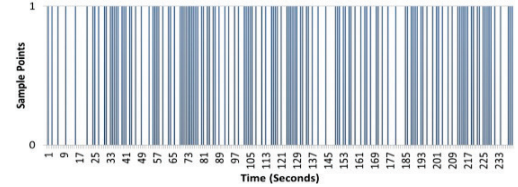


Fig. 3. Sampling points for readings around the boundary

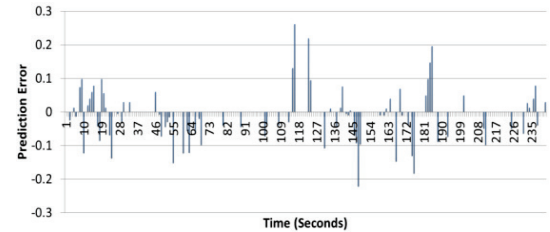


Fig. 4. Prediction Error for readings around the boundary

fluctuations. It is therefore required to sample with high frequency to be able to ensure that it is able to report out of range measurements. As long as all measurement changes are farther than boundaries, our approach are not interested to report changes in measurements which results in high error while being around boundary, it tracks any small changes leading to low error (Fig. 4).

A. Transmission Cost

We define transmission cost as total number of transmitted packets in the network. We consider two types of packets: small control packets and data packets.

Compared with *ASAP* and Hybrid approaches, our approach transmits more control messages as it employs a distributed sampling selection mechanism. This effect can be seen in Fig.5. Despite its higher number of control messages, our approach has the minimum number of data packets transmitted. This is shown in Fig.6. Our approach only reports a message if it cannot accurately detect whether data is in bound or outbound. In *ASAP*, all sensor nodes transfer their data to the cluster head in the forced sampling times while the hybrid approach transfers data only when the prediction error increases. *ASAP*, has the worst data transmission cost.

B. Data Quality

Generally speaking, data quality metric is defined as the average error produced in each cluster. It can clearly be seen from Fig.7. that compared with *ASAP* and Hybrid approaches, our approach produces higher errors. This is due to the fact that we use a different definition for data quality metric, which

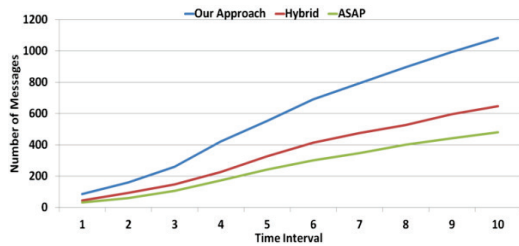


Fig. 5. Comparison of number of transmitted control messages

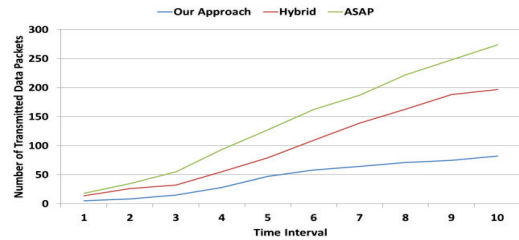


Fig. 6. Comparison of number of transmitted data packets

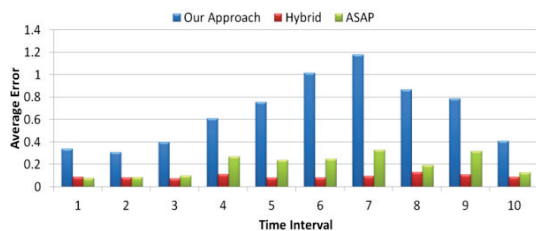


Fig. 7. Produced average error for different algorithms

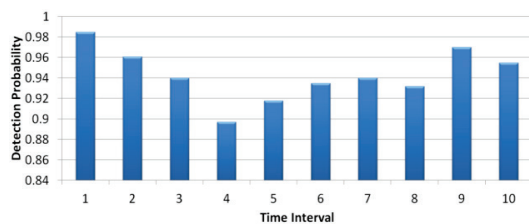


Fig. 8. In bound detection probability

is correctly detecting whether a measurement is in bound. Fig. 8. shows at least 90% detect accuracy to detect whether a sensor measurement is inbound.

C. Energy Cost

Having more data quality in terms of average error results in additional cost in terms of energy consumption. Since ASAP and Hybrid are sensitive to any small changes in the environment, they require more samples and more sampling nodes to monitor the area. This brings about higher energy consumption. Our approach only reports changes when sensor measurements are close to the boundary. At other times, minimum number of sampling nodes with low sampling frequencies is employed to monitor the area. This leads to a low energy consumption as shown in Fig. 9.

VI. CONCLUSION

Current adaptive sampling approaches aim to report every change in sensor measurements to satisfy data quality requirements. However, many monitoring applications are not

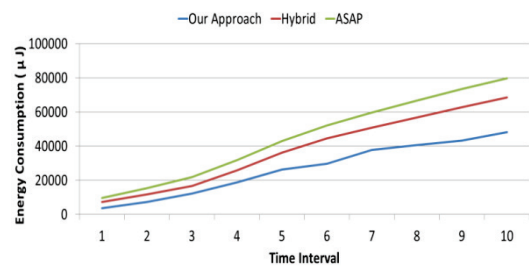


Fig. 9. Energy consumption for different sampling algorithms

interested in all value changes and can tolerate them as long as data measurements falling out of a specific range are reported.

By targeting these applications, in this paper we propose an energy-efficient adaptive sampling approach which leverages the benefit of spatio-temporal correlation that exists in sensor data to report changes in sensor measurements falling outside a given data range. In other hand, this approach carefully selects a dynamically changing subset of sensor nodes to sample and transmit their data. Our simulation results using a given dataset shows the energy gain of our approach by reducing number of transmitted sensor measurements and lowering down number of sampling nodes.

ACKNOWLEDGEMENT

This work is supported by IST FP7 STREP GENESI and FREE projects.

REFERENCES

- [1] A. Deligiannakis, and Y. Kotidis; "Exploiting Spatio-temporal Correlations for Data Processing in Sensor Networks" Springer-Verlag Berlin Heidelberg, LNCS 4540, pp. 45–65, 2008.
- [2] P. Padhy, R. K. Dash, K. Martinez, N. R. Jennings, "A utility-based adaptive sensing and multihop communication protocol for wireless sensor networks", ACM Transactions on Sensor Networks (TOSN), Volume 6 Issue 3, June 2010 No. 27 ACM New York, NY, USA
- [3] K. Deng, A. Moore, and M. Nechyba, "Learning to Recognize Time Series: Combining ARMA models with Memory-based Learning" IEEE. Symp. on Computational Intelligence in Robotics and Automation, 2004, pp. 246–250
- [4] K-means Clustering ; http://en.wikipedia.org/wiki/K-means_clustering
- [5] Aiello G, Scalia G.L, Micale R "Simulation analysis of cold chain performance based on time-temperature data", Production Planning & Control, DOI:10.1080/09537287.2011.564219
- [6] S. Chatterjea, and P.J.M Havinga, "An Adaptive and Autonomous Sensor Sampling Frequency Control Scheme for Energy-Efficient Data Acquisition in Wireless Sensor Networks. In: 4th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), 11-14 Jun 2008, Santorini, Greece. pp. 60-78.
- [7] A.L. Wood, G. V. Merrett, S. R. Gunn, B.M. Al-Hashimi, N.R. Shadbolt, and W. Hall, "Adaptive sampling in context-aware systems: a machine learning approach." IET Wireless Sensor Systems 2012, London, Jun 2012.
- [8] B. Gedik, L. Liu, P. S. Yu, "ASAP: An Adaptive Sampling Approach to Data Collection in Sensor Networks", IEEE Trans. Parallel Distributed Systems, Vol. 18, N. 12, December 2007
- [9] A. Masoum, N. Meratnia, P.J.M Havinga, "Quality Aware Decentralized Adaptive Sampling Strategy in Wireless Sensor Networks", The 9th IEEE International Conference on Ubiquitous Intelligence and Computing, 2012, Fukuoka, Japan, September 04-07, 2
- [10] D.N. Nkwogu and A.R. Allen "Adaptive Sampling for WSN Control Applications Using Artificial Neural Networks" Journal of Sensor and Actuator Networks 2012, 1(3), 299-320