

Reward and Punishment based Cooperative Adaptive Sampling in Wireless Sensor Networks

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

Pervasive Systems, Department of Computer Science

University of Twente, The Netherlands

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

Abstract— Energy conservation is one of the main concerns in wireless sensor networks. One of the mechanisms to better manage energy in wireless sensor networks is adaptive sampling, by which instead of using a fixed frequency interval for sensing and data transmission, the wireless sensor network employs a dynamic scheme based on how frequent pattern of sensed data changes. Selecting an appropriate sampling rate for wireless sensor networks to ensure both long network life-time and high data quality is challenging. Lack of cooperation between sensor nodes to enable them to adapt their sampling rates while having an eye on the overall energy use is one of the main drawbacks of the current data gathering techniques in wireless sensor networks. Through cooperation, sensor nodes can obtain enough knowledge about resources available in the network and environmental conditions they observe. This information can help them to better and more intelligently select their own sampling rates. In this paper, we propose a cooperative adaptive sampling mechanism based on the award and punishment concept to motivate sensor nodes to cooperate with each other. We define a utility function for every sensor node, which aims at finding a good balance between its data prediction error and remaining energy. When some sensor nodes in a neighbourhood experience frequent environmental changes, other nodes lower down their sampling rates to enable them to increase their sampling rate to keep the overall network data quality high and energy consumption low.

I. INTRODUCTION

Wireless sensor networks (WSNs) are composed of many low-power nodes that are distributed in an environment for monitoring and control purposes. Example applications of WSNs include habitat monitoring, structure monitoring, health monitoring and feedback, as well as industrial safety, to name but a few. Energy consumption is one of the most important challenges in these networks [1]. To this end, many efforts have been directed towards optimization of energy consumption and increase of network life-time. One of the mechanisms to better manage energy consumption in WSNs is concerned with the data collection.

Observation of environmental phenomena is done through sampling techniques with specific spatial and temporal resolutions. Choice of sampling frequency, i.e., the rate at which data is collected, depends on application, its requirements, and goals. Since high frequency sampling has high impact on network life-time, various energy efficient sensing and transmission techniques have been previously proposed. One of these techniques is related to adaptive

sampling, by which instead of using a fixed frequency interval for sensing and transmission, wireless sensor network employs a dynamic scheme based on dynamicity of data and network resources. The main concept behind adaptive sampling is that when data characteristics remain constant for a long time, sensor nodes will decrease their sampling frequency and in turn their transmission rate. Due to the fact that data characteristics remain unchanged in this period, decreasing sensing and transmission rates will not lead to decrease of data quality. Once collected data presents frequent changes, sensor nodes will increase their sampling frequency to keep quality of collected data in an acceptable level.

While high density can offer the advantage of providing redundant data and can therefore ensure the data quality in case of hardware or link failure, it can also lead to extreme communication overhead and high resource consumption in terms of extensive use of bandwidth, memory, and energy. Finding a balance between network resource consumption and data quality is important for adaptive sampling.

Adaptive sampling techniques can be categorized in different ways, for example based on activity and energy parameters [2]. Most of adaptive sampling models try to adapt the sampling frequency to the environmental changes by predicting future samples using centralized or decentralized models, which may benefit from spatial, temporal, and spatio-temporal correlation between sensor data. The decision on which sampling frequency to employ can be made centrally in a gateway or locally by each individual node. Regardless of where sampling frequency is set and which model is used for data prediction, current adaptive sampling techniques do not make the right balance between network life-time and data quality, as these two requirements are tackled independently.

In this paper, we propose a reward and punishment based cooperative adaptive sampling to satisfy both network life-time and data quality requirements. We assume clustered sensor nodes, in which sensor nodes have a direct access to the cluster head. In other words, we assume a star topology. The idea behind our adaptive sampling technique is that when a node detects frequent environmental changes, it increases its sampling rate while other nodes decrease their sampling rates to keep the overall network data quality high and energy consumption low. We define a network utility function as sum of utility functions of all sensor nodes. Using this utility function, our ultimate goal is to minimize network-wide data prediction error and energy consumption ratio. For motivating

sensor nodes to cooperate with each other we will use award and punishment concept.

The rest of the paper is organized as follows. Section II provides an overview of related work, while Section III presents our cooperative adaptive sampling method. Simulation setup and results will be presented in Section IV. Conclusions and future work will end the paper in Section V.

II. RELATED WORK

Many of current adaptive sampling techniques rely on modelling and prediction of data to reduce number of actual measurements. There are a number of issues related to modelling, for instance whether it is centralized (performed in a sink node or cluster head) or decentralized (performed locally in each node), or whether spatial and temporal correlations are used. Temporal correlation is related to the fact that data collected by a single node may present similarities at different time instances, while spatial correlation is related to the fact that data collected at different nodes may present similarities at a single time instance. Previous work such as the ones reported in [4] is example of centralized adaptive sampling based on temporal correlation. Another important issue in here is the model itself, which can be for example based on probability [6, 7] or Gaussian [8] distributions.

A MAC protocol based on traffic adaptive periodic data collection is presented in [9], in which each node has a scheduling scheme based on its sampling rate. For supporting this scheme, a route partitioning mechanism is used, for which sensor nodes send route discovery packets to the sink node. Each node that is in the route to the sink puts its sampling frequency and address in this packet. After receiving all route discovery packets, the sink node partitions network routes based on nodes sampling frequencies. In case of having overlapping sampling frequencies, nodes adjust their frequencies to the longest duty cycle. Willet et al. consider spatial correlation-based adaptive sampling and propose a two steps algorithm [10]. In the first step, called the preview step, only a subset of nodes is turned on and sends data to the sink. In the second step, called refinement, the sink may activate additional sensors in those locations where the spatial correlation is low. This is accomplished using a “backcast” procedure, in which the sink sends an activation message to those cluster-heads residing in the smallest square areas generated in the preview step. For reliable data gathering in mobile sensor networks, Dang et al. [11] models the observation area as a set of grid points. At every round, the sensor node having the least data uncertainty covariance matrix is the node to sample. This mechanism is not scalable, as when a new node is added to the network, computation time of finding next sampling node will increase exponentially.

In [20], each sensor is considered as an autonomous agent and decides on how and when it should sample data based on its individual task and local knowledge. The idea is each node should vary its sampling rate to maximize the value of information it produces.

There are also some adaptive sampling techniques based on spatio-temporal correlation [13]. The adaptive sampling approach of Gedik et al. [14] elects cluster heads and assigns nodes to the clusters based on sensor readings similarity and the hop counts. Cluster construction algorithm is executed periodically based on energy level and change of data behaviour in individual nodes. Osborne et al. [15] use multi output Gaussian model to determine when and which node must sense and send its readings. While the aforementioned techniques are all centralized, in what comes hereafter we consider decentralized adaptive sampling methods.

Padhy et al. consider temporal correlation while defining a utility-based sensing and communication protocol [16]. 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. Use of time series forecasting methods to predict sampling and transmission rate is reported in [18]. Using cross layer information, this method adapts to the topology changes. Makarenko et al. propose a negotiation-based Bayesian method, in which each node uses a local Bayesian non-linear filter and selects a new sampling frequency based on its collected data and filter output [19].

An adaptive, lazy, energy-efficient protocol (ALEP) is also proposed in [5], which adjusts the sampling frequency based on availability of resources and change frequency of data consistency model.

III. REWARD AND PUNISHMENT BASED COOPERATIVE ADAPTIVE SAMPLING

Looking at the previous works, we can conclude that none of them uses cooperation among the nodes to compute a suitable sampling frequency for the entire network. Instead, each node only decides on its own its sampling frequency based on its limited observation. This although may lead to local energy conservation at the node level, it can result in a global data quality loss. Our proposed cooperative adaptive sampling method considers cooperation between nodes to find a balance between global energy conservation and global data quality. To do so, we aim to find the best sampling frequency for each node based on the frequency at which data behavioural pattern changes as well as the network energy level. In other side, each sensor node can decide to cooperate with network or act selfish. We consider award and punishment concept to motivate sensor nodes to cooperatively find a good sampling rate satisfying long network life-time and high data quality requirements. At first, we explain our assumptions.

First of all, we consider three states for the data behavioural pattern: Stable-state denoted as A, Transition-state, denoted as B, and Unstable-state, denoted as C. To be able to determine state of data behavioural pattern indicating the environment states, we define two thresholds θ_1 and θ_2 , against which we compare node’s current observation and node’s prediction.

In the A state (stable state), data behavioural pattern rarely changes so sensor nodes can sample with a low frequency while data prediction error remains low (smaller than θ_1). In

the B state (transition state), data behavioural pattern starts to present some abnormal behaviour and more frequent changes occur (data prediction error is between θ_1 and θ_2). In the C state, many changes occur in the data behavioural pattern, which may indicate that an event may occur (data prediction error is more than θ_2). In this case, sensor nodes must use higher sampling frequency. Our design principle is to help nodes in the C state to increase their sampling frequency while nodes in the A or B states decrease their sampling frequency and by doing so they will keep the network-wide energy consumption low and data quality high.

To keep an eye on how much error is generated at the cluster head when sampling frequency is reduced, we assume that nodes use a linear regression model for predicting future observations. The same model is also used at the cluster head.

Sensor nodes must know whether they need to send their readings to the cluster head when they change their sampling frequency. Therefore, we define the following rules:

- If the new sampling frequency changes sensor node's state from A to B or if it does not cause any state change, sensor node does not need to send its readings to the cluster head.
- If the new sampling frequency changes sensor node's state from B to A, sensor node needs to send its new readings to the cluster head and to update its prediction model.
- If sensor node enters the C state, it must send its new readings to the cluster head.

Our algorithm has four main phases, i.e., initialization, finding a new sampling frequency for the whole network, predicting own sampling frequency, deciding whether to change the sampling frequency at the node level.

A. Initialization

At this phase, we define a set of possible sampling rates as:

$$SRSet_i = \{SR_{min}, SR_1, \dots, SR_{max}\} \quad (1)$$

To have an insight about actual data behavioural change, at the initialization phase, we allow every node to sample based on its maximum sampling frequency for one full day. We take this dataset as the ground truth. At the end of the first day and using the regression model, every node predicts its observations for all possible frequencies in the SRSet. Based on these predications, each node is able to calculate a data prediction error, which is calculated as:

$$Err_{ij} = Err(SR_i, TP_j) = Err(SR_i) - Err(SR_{max}) \quad (2)$$

In addition to this error measurement, we use number of data changes during each time period as an indication of degree of data dynamicity (data change). We utilize the following equation to calculate degree of dynamicity (DYN) for each time period and each sampling frequency.

$$DYN_{ij} = DYN(SR_i, TP_j) = \frac{\text{Number of data changes}}{\text{Number of readings}} \quad (3)$$

Based on these calculations, we create a table called environmental changing table (ECT), as shown in Table I. Each entity (i,j) in this table consists of two parameter, i.e., Err and DYN, where Err is the difference between node's

observation at the maximum sampling frequency and the designated sampling frequency at every time period and DYN is the degree of dynamicity for the designated sampling rate at every time period. To be able to relate data behavioural pattern with error and identify which state nodes are in, we need to create a new table called SRDY with degree of dynamicity (DYN) as its columns and possible sampling rates of SRSet as its rows, as shown in Table II. Every cell in this table represents the error measurement corresponding to a certain degree of dynamicity and sampling rate.

TABLE I. ECT TABLE

Sampling Rates	Time Periods		
	TP_1	TP_N
SR_1	(Err_{11}, Dyn_{11})	(Err_{1N}, Dyn_{1N})
SR_2
....
SR_S	(Err_{S1}, Dyn_{S1})	(Err_{SN}, Dyn_{SN})

TABLE II. SRDY TABLE

Sampling Rates	Degree of Dynamicity		
	Dyn_{11}	Dyn_{1N}
SR_1	Err_{11}	Err_{1N}
SR_2
....
SR_S	Err_{S1}	Err_{SN}

One should note that it is possible to have similar degrees of dynamicity and different error for different time periods and one sampling frequency. For instance, at time periods 1 and 3, we can have similar dynamic degree $Dyn_{11}=Dyn_{13}$ and different errors Err_{11} and Err_{13} . In this case, we use mean value of the errors as the average error corresponding to that specific degree of dynamicity.

These calculations and tables creations are performed once and offline and therefore has no burden on the resources of the wireless sensor network. Once SRDY tables are created, each node sends its SRDY table to other nodes.

During the initialization phase, cluster head will assign to each node a maximum energy that is allowed to be used at each time period. We call this (TEg). If the maximum amount of energy for a specific time period is not used, it can be transferred to the next time period. The cluster head also assigns one of the sensor nodes as the decision making node. The decision node (DNode) is responsible to calculate and suggest new sampling frequency for each node based on the procedure described in Section B. Selection of the decision making node is always based on nodes energy level. For calculating remaining energy at each time period, we will use this formula [17]:

$$RsdEg_i = E_{Last} - (a * E_{Sens} + b * E_{Trans} + E_{Process}) \quad (4)$$

, where E_{Last} is the last residual energy which is calculated in the last time period; E_{Sens} is the amount of energy required for sensing one bit; E_{Trans} is the amount of energy required for transmitting one bit; $E_{Process}$ is the amount of energy required for processing; a and b are the number of sensed and transferred data in bit.

To motivate sensor nodes cooperation in defining their sampling frequency, we define two parameters, i.e., cooperation parameter (CP) and prediction parameter (PP).

The former represents node's cooperation level, while the latter indicates node's data prediction accuracy. If a sensor node cooperates with other nodes and accepts suggested sampling frequency of the decision node, its CP parameter will be increased at the beginning of each time period. Similarly, if degree of dynamicity of the new prediction of a sensor node is in line with its current degree of dynamicity, its PP parameter will be increase. We use CP and PP parameters to reward or punish the sensor node.

We compare cooperation and prediction parameters against two thresholds, i.e., TH_{CP} and TH_{PP} :

- If ($CP > TH_{CP}$ and $PP > TH_{PP}$): the sensor node has cooperated with the decision node and its degree of dynamicity prediction has been correct. Therefore, we consider this node as a trusted node.
- If ($CP > TH_{CP}$ and $PP < TH_{PP}$): the sensor node has cooperated with network but most of the times its degree of dynamicity predictions has been wrong. Therefore, we consider this node as a normal node.
- If ($CP < TH_{CP}$): the sensor node has not been cooperative and we consider this node selfish.

B. Finding a New Suitable Sampling Frequency by the Decision Node

The decision node (DNode) will receive current sampling frequency, prediction error, residual energy, cooperation and prediction parameters of each node and will use the Multiple Choice Knapsack Problem (MCKP) optimization technique [3] to find a new suitable sampling frequency for each node.

There are some conditions that the optimization technique must consider while finding the best sampling frequency for each node:

- The amount of prediction error must be kept lower than the predefined threshold θ_2 . This means that sampling frequencies causing errors more than θ_2 will be ignored.

$$Err_i \leq \theta_2, \quad \forall i \in [1 \dots N] \quad (5)$$

- For each node, the required energy ($RqdEg_i^j$) for sensing and transmitting data for next time period must be lower than the maximum available energy for each time period (TEg_i^j) plus the current residual energy ($RsdEg_i^{j-1}$).
- $RqdEg_i^j \leq TEg_i^j + RsdEg_i^{j-1}, \forall i, j \in [1 \dots N]$ (6)

In fact, the optimization algorithm minimizes the ratio between data prediction error and energy consuming for each time period, i.e.,

$$Min \sum_{i=1}^N \frac{Err_i}{RsdEg_i} \quad (7)$$

The decision node runs the optimization algorithm with the last sampling frequency and prediction error of each node as its input and finds degree of dynamicity for each node. It then finds the best sampling frequency for different nodes by using SRDY table and degree of dynamicity of nodes as well as energy calculation

After calculating the available energy for each node in the next time period, the decision node finally assigns the next decision node based on maximum available energy and prediction and cooperation parameters.

C. Predicting Own Sampling Frequency

After receiving new sampling frequency, prediction error, and residual energy from the decision node, each sensor node will decide whether to use decision node's suggested sampling frequency or its predicted sampling frequency.

To predict its own sampling frequency, the sensor node stores 3 recent readings and degree of dynamicity parameters. Then it uses regression model and SRDY table to predict new DYN for the current time period and its current sampling frequency. After predicting its new sampling frequency, it can decide whether to cooperate with the network and use decision node's suggested sampling frequency or act selfish and uses its own prediction. If it uses decision node's suggestion, we award it by increasing its CP parameters. Otherwise, its CP parameter will be decreased. Also, at first of each time period, sensor node predicts DYN for current time period and stores it. At the end of time period, it compares this prediction with current DYN. If its prediction was true, its PP will be increased, otherwise it will be decreased.

D. Deciding on Whether to Change the Sampling Frequency

Decision on whether to use the decision node's suggested sampling frequency will be made based on the following rules:

- Trusted nodes will use their own predicated sampling frequency.
- Normal nodes will be given the opportunity to choose between their prediction and decision node's suggestion. If they cooperate and select decision node's suggested sampling frequency, their CP parameter will be increased and the other way around.
- Selfish nodes will not be allowed to act as decision node for a certain time periods. This is Ppart of the punishment procedure. Also during this time periods, selfish nodes have to use decision node's suggested sampling frequency and cannot update their CP parameter. This means that during the punishment period, selfish nodes have to cooperate with the network but will not receive any award for their cooperation.

During each time period if prediction error increases and puts the sensor node in the C state, we apply the following rules:

- Trusted nodes will continue predicting their new sampling rates and use them. They will increase their PP parameter if their prediction is in line with the current degree of dynamicity. Otherwise they will decrease it.
- Non-trusted nodes will use the maximum available sampling frequency of the SRSet matching their current residual energy level.

At the end of each time period, each sensor node applies the following rules:

- If the previous and current degrees of dynamicity are in line then sensor node will increase its PP parameters. Otherwise, it will decrease it.
- If $CP > TH_{CP}$ and $PP > TH_{PP}$ then sensor node is identified as a trusted node and will use the regression model and previous reading to predict its sampling frequency
- It will use equation (4) to update its residual energy.

- It will send its current sampling frequency, prediction error, residual energy, and predicted sampling frequency to the decision node.

Pseudocode of our technique can be found below.

Initialization

1. Define Sampling rate set $SRSet_i = \{SR_{min}, SR_1, \dots, SR_{max}\}$
2. For one day, Do sample with SR_{max} and complete data set for different time periods
3. Predict observation for different sampling rates
4. For each SR_i in Each TP_j , Find these difference:

$$Err_{ij} = Err(SR_i, TP_j) = Err(SR_i) - Err(SR_{max})$$
5. For each SR_i in Each TP_j , find Dynamicity Parameter:

$$DYN_{ij} = DYN(SR_i, TP_j) = \frac{\text{Number of data changes}}{\text{Number of readings}}$$
6. Make ECT table
7. Make SRDY table
8. Send SRDY table for other nodes
9. Assign max available energy for each time period for each node
10. Set CP and PP parameters

Finding a New Suitable Sampling Frequency by the Decision Node

1. For each node:
 - 1.1 Receive ($SR_i, Err_i, RsdEg_i, CP_i, PP_i, SR_{Sugg}$)
 - 1.2 If ($CP > TH_{CP}$ and $PP > TH_{PP}$) { Consider SR_{Sugg} as a fixed SR in KCMP;
Send SR_{Sugg} to KCMP as a fixed SR}

Else send (SR_i, Err_i) to the KCMP as an input.
2. Run KCMP and get Different new ($NewSR_i, NewErr_i$) for different nodes
3. For each Node, If ($CP \leq TH_{CP}$)
{ If ($TStamp == 0$) then { $TStamp = X * Tperiod$; $K_i = 1$ }
Else { $TStamp--$; If ($TStamp == 0$) then $K_i = 0$; }
4. Find New energy level
5. Find a node with max available energy level and with ($CP > TH_{CP}$) and select it as a Next DNode
6. Send ($NewSR_i, NewErr_i, NewRsdEg_i, K_i, NewDNode$)

Predicting Own Sampling Frequency

1. Receive ($SR_i, Err_i, RsdEg_i, Next DNode$)
2. Predict Dynamicity for current TP (DYN_{pre})
3. use regression model and SRDY table to predict current sampling frequency.

Deciding on Whether to Change the Sampling Frequency

1. If ($CP > TH_{HCP}$ and $PP > TH_{HPP}$) {use SR_N ;}
Else if ($CP > TH_{CP}$) { If it select SR_N then $CP = CP - 1$; }

Else If it select SR_{DN} ; $CP = CP + 1$; }

Else If ($CP < TH_{CP}$) { If ($K_i == 1$) then use SR_{DN}
Else {Use SR_{DN} ; $CP = CP + 1$; }

During Time period if Error increases and puts Sensor in State C:

2. If ($CP > TH_{CP}$ and $PP > TH_{PP}$) {Use SR_N ; $PP = PP - 1$; }
Else use SR_{max} ; **At the end of Time Period:**
3. If ($DYN_{cur} == DYN_{pre}$) then $PP = PP + 1$;
4. Else $PP = PP - 1$;
5. Update energy model
6. If ($CP > TH_{CP}$ and $PP > TH_{PP}$) then $H_i = 1$
7. Else $H_i = 0$;
8. Using regression model and previous reading find SR_{Sug}
9. Send ($SR_i, Err_i, RsdEg_i, H_i, K_i, SR_{Sugg}$) to DNode

IV. SIMULATION RESULT

In this section, we show the trade-off between energy consumption and data quality achieved by our reward and punishment based cooperative adaptive sampling technique. To test performance of our technique, we compare it with the

decentralized adaptive sampling technique presented in [20]. We consider a network consisting of 20 nodes, which present no spatial correlation. The network topology is star and sensor nodes monitor the temperature. We use the dataset of [12] as the ground truth, i.e., data obtained from the highest sampling rate. We assume $\theta_1 = 3$ and θ_2 will change from 4 to 9. We repeat the simulation 20 times and show the mean results in the following figures. We also assume that the acceptable error that the network can tolerate is 25%. Table III represents our simulation parameters.

TABLE III. SIMULATION PARAMETERS

Parameter	Value of Parameters
	Value
SRSet	{5,10,15,20,30,60}
θ_1	3
θ_2	[4,9]
$\sum_{i \in n} Err_i$	≤ 0.25

Figure 1 and 2 show the effect of threshold θ_2 on the prediction error and energy consumption. As it can be seen, compared with work of [20], our technique saves more energy while it has a similar prediction error. Smaller values of θ_2 lead to more data transmission and consequently more energy consumption. When this threshold increases, sensor nodes can tolerate more changes and do not need to frequently send data. However, due to lack of this threshold and not tolerating prediction errors, the decentralized method [20] consumes more energy.

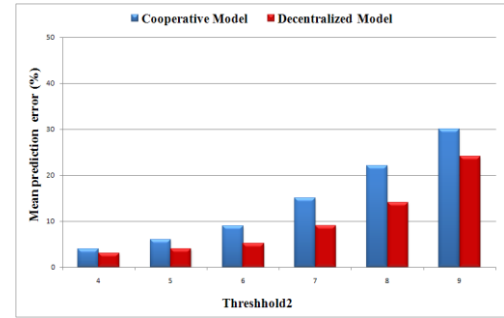


Fig. 1. Mean prediction error versus θ_2

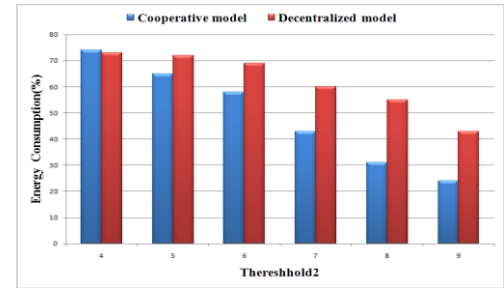


Fig. 2. Energy consumption versus θ_2

Figure 3 shows the effect of different states on number of samples. As it can be seen, in our cooperative technique number of samples increases as node detects data behavioural change in the B state. Due to lack of states in the decentralized approach, number of samples does not change in A or B states. This also explains why the decentralized approach consumes more energy than our technique.

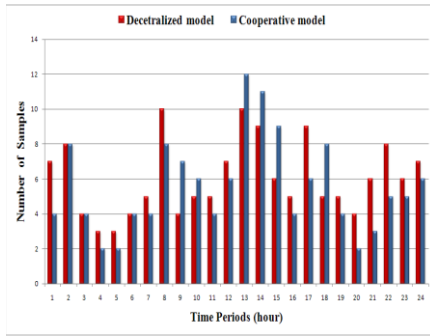


Fig. 3. Number of samples for one day

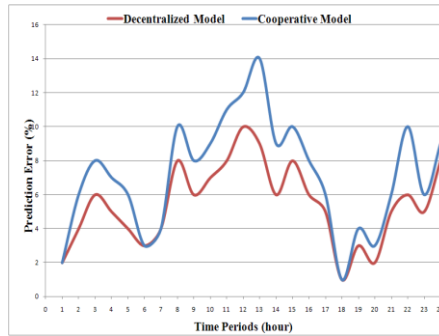


Fig. 4. Prediction error versus time period

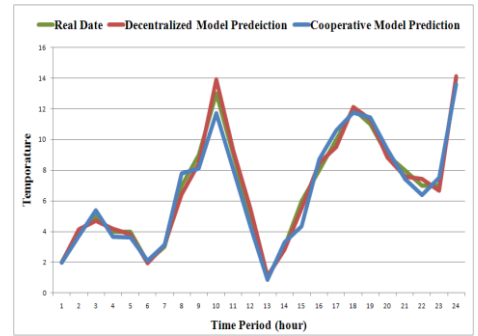


Fig. 5. Comparing real data with predictions in one day

Figure 4 shows the effect of different states on prediction error. As it can be seen, in our cooperative technique prediction error increases as node detects data behavioural change in the B state. Therefore, our technique generates more prediction error than the decentralized approach [20] only in the B state. However, the generated error of our approach remains within the tolerable error of 25%. In the A and C states, both techniques generate similar prediction errors.

Figure 5 compares the real data with predicted data. It shows that for our method, the prediction error increases as node detects data behavioural change in the B state.

V. CONCLUSION

In this paper, we propose an aware and punishment based cooperative adaptive sampling technique to satisfy both network life-time and data quality requirements. Instead of changing the sampling frequency only based on energy level and data quality of a single node, we propose to change nodes sampling frequency based on network-wide energy level and global data behavioral pattern change. Our technique outperforms a previous decentralized approach in terms of energy consumption while maintaining similar prediction error.

ACKNOWLEDGMENT

The work presented in this paper is supported by FREE project that is funded as part of the Innovation programmed PointOne organized by SenterNovem, the Netherlands.

REFERENCES

- [1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey." *Computer Networks* 38 (2002)
- [2] C. Alippi, G. Anastasi, M. Di Francesco, M. Roveri: "Energy Management in Wireless Sensor Networks with Energy-Hungry Sensors", *IEEE-Instrumentation and Measurement magazine*, Vol. 2, No. 2, pp 16-23, 2009.
- [3] R.D. Armstrong, P. Sinha, A.A. Zoltners, : "The multiple-choice nested knapsack problem." *Manag. Sci.* 28, 34-43 (1982)
- [4] J. Zhou, D. De Roure, : "FloodNet: Coupling Adaptive Sampling with Energy Aware Routing in a Flood Warning System", *Journal of Computer Science and Technology*, Vol. 22, N. 1, pp. 121-130, January 2007
- [5] Sh. Kewei and Sh. Weisong, "Consistency-driven data quality management of networked sensor systems", *Journal of Parallel and Distributed Computing* Volume 68 , Issue 9 (September 2008) Pages: 1207-1221 Year of Publication: 2008 ISSN:0743-7315

- [6] B.R. Friden, "Science from Fisher Information: A Unification" ISBN 0521810795, 2004
- [7] F. Zhao and L.J. Guibas "Wireless sensor networks: an information processing approach" ISBN: 1558609148, Boston: Elsevier-Morgan Kaufmann; 2004.
- [8] A. Krause, C. Guestrin, A. Gupta and J. Kleinberg, "Near-optimal sensor placements: maximizing information while minimizing communication cost" *Information Processing In Sensor Networks Proceedings of the 5th international conference on Information processing in sensor networks Nashville, Tennessee, USA* Pages: 2 - 10, Year of Publication: 2006 ISBN: 1-59593-334-4
- [9] M. Vahabi, M. F. A. Rasid, R. S. A. R. Abdullah, and M. H. F. Ghazvini "Adaptive Data Collection Algorithm for Wireless Sensor Networks" *IJCSNS International Journal of Computer Science and Network Security*, VOL.8 No.6, June 2008
- [10] R. Willett, A. Martin, R. Nowak, "Backcasting: Adaptive Sampling for Sensor Networks", *Proc. International Symposium on Information Processing in Sensor Networks (IPSN 2004)*, pp. 124-133, 26-27 2004.
- [11] T. Dang, N. Bulusu, W. Feng, S. Frolov and A. Baptista, "Adaptive sampling in the Columbia river observation network", *5th ACM Conference on Embedded Networked Sensor Systems (Sensys)*. 429-430, 2007.
- [12] http://sensorscope.epfl.ch/index.php/Main_Page
- [13] T. Arici and Y. Altunbasak, "Adaptive sensing for environment monitoring using wireless sensor networks," in *Wireless Communications and Networking Conference*, 2004, pp. 21-25.
- [14] 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
- [15] M.A. Osborne, A. Rogers, S. Ramchurn, S.J. Roberts and N.R. Jennings, "Towards realtime information processing of sensor network data using computationally efficient multi-output Gaussian processes." In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*. 109-120, 2008.
- [16] P. Padhy, R. K. Dash, K. Martinez, N. R. Jennings, "A Utility-Based Sensing and Communication Model for a Glacial Sensor Network", *Proc. International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, Hakodate, Japan, 2006.
- [17] M. N. Halgamuge, M. Zukerman, K. Ramamohanarao, and H. L. Vu, "An estimation of sensor energy consumption," *Progress In Electromagnetics Research B*, Vol. 12, 259-295, 2009.
- [18] S. Chatterjea and P. Havinga, "An Adaptive and Autonomous Sensor Sampling Frequency Control Scheme for Energy-Efficient Data Acquisition in Wireless Sensor Networks" *BookChapter, Springer Berlin / Heidelberg* ISSN0302-9743 (Print) 1611-3349 (Online) Volume 5067/2008 *Book Distributed Computing in Sensor Systems* Pages 60-78 *Subject Collection Computer Science*.
- [19] G. Mainland, D.C. Parkes and M. Welsh, "Decentralized, adaptive resource allocation for sensor networks" *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2* Pages: 315 - 328, 2005
- [20] J. Kho, A. Rogers, and N.R. Jennings, "Decentralized Control of Adaptive Sampling in Wireless Sensor" *ACM Transactions on Sensor Networks (TOSN)* Volume 5 , 2009