Energy-Efficient Sensor Selection for Data Quality and Load Balancing in Wireless Sensor Networks

Farshid Hassani Bijarbooneh*, Wei Du†, Edith Ngai*, Xiaoming Fu‡
*Uppsala University, Department of Information Technology, Box 337, SE–75105 Uppsala, Sweden
Email: {Farshid.Hassani, Edith.Ngai}@it.uu.se
†Research Unit in Networking (RUN), University of Liège, Belgium
Email: weidu@montefiore.ulg.ac.be
‡Institute of Computer Science, University of Göttingen, Göttingen, Germany
Email: fu@cs.uni-goettingen.de

***Abstract*—It is common to deploy stationary sensors in large geographical environments for monitoring purposes. In such cases, the monitored data are subject to data loss due to poor link quality or node failures. Fortunately, the sensing data are highly correlated both spatially and temporally. In this paper, we consider such networks in general, and jointly take into account the link quality estimates, and the spatio-temporal correlation of the data to minimise energy consumption by selecting sensors for sampling and relaying data. In particular, we propose a multi-phase adaptive sensing algorithm with belief propagation protocol (ASBP), which can provide high data quality and reduce energy consumption by turning on only a small number of nodes in the network. We explore the correlation of data, formulate the sensor selection problem and solve it using constraint programming (CP) and greedy search. Bayesian inference technique is used to reconstruct the missing sensing data. We show that while maintaining a satisfactory level of data quality and prediction accuracy, ASBP successfully provides load balancing among sensors and preserves 80% more energy compared to the case where all sensor nodes are actively involved.**

I. Introduction

Wireless sensor networks (WSNs) consists of a large number of embedded devices capable of sensing, processing, and communicating the habitat and environmental data [1].

It is often the case that sensor readings in the same spatial regions are highly correlated, and depending on the application the readings are temporally correlated as well. Exploiting this aspect significantly improves energy consumption in WSNs.

Belief propagation (BP) [2]–[4] is a technique for solving inference problems. In the WSN context, the belief of a sensor node is the data measurement of an event in the environment, and BP provides an iterative algorithm to infer the measurements of the sensor nodes, especially in cases where the data are missing, because of packed losses or because there is no data available at some selectively disabled sensor nodes. In such inference problems, the assumption that the data are spatio-temporally correlated significantly improves the accuracy of data inference using BP in WSNs.

In this paper, we propose an adaptive sensing belief propagation protocol (ASBP), where the data is collected in several *rounds* (a round is a fixed time interval where the network repeats the same behaviour) by active sensors (sensors that are collecting data in a each round). We formulate and solve an optimisation problem that selects the active sensors in each round, by maximising the data utility while maintaining energy load balancing. We define *data utility* as a metric computing the sum of the qualities of the path links from the selected active sensor nodes to the base station, subtracted by the sum of the correlations of the selected active sensors. In addition to BP, we also use data quantisation to further compress the data and reduce the transmission costs. In our active sensor selection formulation, we consider non-linear multi-hop routing protocol constraints. We use constraint programming (CP) [5] and compare it with our heuristic-based greedy algorithm.

The contributions of this paper are as follows. (1) We present a novel data collection scheme (ASBP) that utilises highly correlated spatio-temporal data in the network and uses BP to reconstruct the missing data due to packet losses and the sensor selection strategy. (2) We formulate the active sensor selection optimisation problem, and propose two approaches to solve the problem. Our CP approach solves the problem to optimality. (3) We conduct extensive simulation with a real deployment of a sensor network and the collected data to evaluate the impact of our proposed solution on the overall energy consumption, data utility, and accuracy (error prediction of the missing data).

The remainder of this paper is organised as follows. In Section II we discuss the related work. In Section III, we give the system overview. In Section IV, we describe the formulation of our optimisation problem, and we solve it using CP and heuristic-based greedy algorithm. In Section VI, we evaluate the performance of our ASBP protocol on a real deployment of a wireless sensor network. Finally, we summarise and conclude the paper in Section VII.

II. Related Work

There is now a substantial body of work on energy-efficiency to prolong WSNs lifetime. Many of the approaches that are aimed at minimising the energy consumption can be categorised into cluster-based and prediction-based. For instance, see LEACH clustering protocol [6] and the references therein. LEACH is an application-specific clustering protocol, which shows significant improvements on the network lifetime. However, it does not guarantee a good cluster head distribution and assumes uniform energy consumption for the cluster heads.

Prediction-based energy-efficient approaches aim at predicting the data to minimise the number of transmissions. Chou *et al.* [7] proposed a distributed compression based on source coding, which highly relies on the correlation of the data. They used adaptive prediction to track the correlation of the data, which is used to estimate the number of bits needed in source coding for data compression. Recent work in WSN

978-1-4799-4852-9/14/$31.00

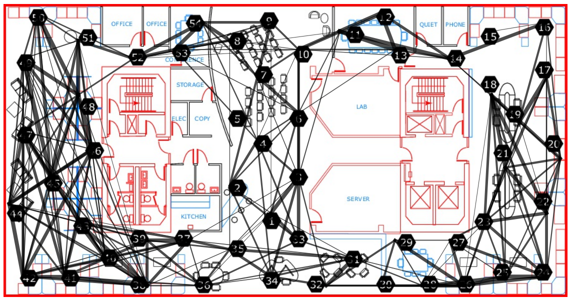


Fig. 1: The map of the Intel lab, with the diamond shape nodes indicating the locations and the ids of the sensor nodes.

addressed the use of compressive sensing [8]. The authors use compressive sensing to exploit the temporal stability, spatial correlation, and the low-rank structure of the environment matrix (EM). Although compressive sensing shows improved missing data estimation, it does only consider implicit spatio-temporal correlation. Furthermore, compressive sensing approaches rely on a complete EM matrix to be available for the prediction of the missing data. However, in our work we present a belief propagation approach for the data prediction, where the spatio-temporal correlation is explicitly enforced, and the inference for predicting the missing data is performed iteratively as the data are received at the base station. In addition to the above, to the best of our knowledge, there has been no work addressing a CP approach for energy-efficient sensor selection with dynamic routing, while considering the link quality and correlation of the data.

III. System Overview

A. Network Model

In our application, stationary sensor nodes are deployed in a sensor field to collect environmental data. The sensor nodes periodically sample data, which is forwarded to the sink using a multi-hop routing protocol. In this work, we use the real data collected at the Intel lab [9]. Figure 1 shows the map of the Intel lab, and the location of the deployed sensor nodes, which are marked with diamond shapes, and the sensor id. The link thickness between the sensor nodes represents the value of the link quality aggregated throughout the experiment.

The sink is a base station with high computation capacity and memory. It collects the sensor node readings and performs analysis and computations on the data. The base station communicates with only the neighbour sensor nodes in the communication range, and is assumed to be aware of the network topology and the routing tables of the deployed sensor nodes. In this paper, we refer to the sink and the base station as the same entity.

B. Protocol Design

In our setup, the sensor nodes collect and report the data periodically (typically every 30 seconds). In our protocol each round includes two phases. The first phase is used to collect the link quality and sensor readings from all sensors, which is used in the second phase to improve energy-efficiency, energy load balancing, and the data quality. The two phases in each round are as follows:

Phase 1: Phase one begins as all sensor nodes become active, and starts collecting and forwarding a fixed number of quantised data to the base station (typically 20 sensor readings). Throughout this phase the routing protocol estimates the link quality for the shortest routes between the sensor nodes and the base station. The base station then computes the correlation coefficient matrix from the sensor data, and computes all the shortest paths from the sensor nodes to the base station. These data (link quality, correlation, and shortest routes) are then used as an input to solve our sensor selection optimisation problem (further explained in Section IV) The sensor selection problem is solved using either constraint programming (CP) or a heuristic-based greedy algorithm to select a set of active sensor nodes, such that it maximises the spatio-temporal correlation with the inactive sensor nodes, while considering link quality and the dynamic routing.

Phase 2: The base station broadcast a message that informs a subset of the sensor nodes to become inactive (sleep mode with no radio activity) for a given period of time (typically 2 hours). In this phase, the base station performs the belief propagation algorithm (BP) [3], [4] to infer incrementally the missing data due to the inactive sensor nodes and packet losses (further explained in Section V). As the second phase is completed, the base station continues to use BP during the first phase of the next round. This allows us to compare the inference results during the first phase with the ground truth, and to compute the error in prediction. This error is then used by our protocol to give feedback (on the minimum number of selected sensor nodes) to the sensor selection optimisation problem of the next round. This allows a dynamic control over the accuracy of the data prediction in phase two. Throughout this paper, we say adaptive sensing with belief propagation protocol (ASBP) to refer to the protocol design above.

IV. Problem Formulation

We present our constraint programming (CP) model for the sensor selection problem, followed by our heuristic-based greedy algorithm. The CP model guarantees a global optimum solution to the problem, whereas the greedy search finds a good quality local optimum solution.

A. Constraint Programming Model with Dynamic Routing

Let S be the set of WSN sensor nodes, with $|S| = N$. Let $L[s_1, s_2]$ be the link quality between neighbour sensor nodes s_1 and s_2, indicating the probability of receiving a packet sent from s_1 to s_2, with $s_1, s_2 \in S$. Let $B[s]$ be the link quality between the sensor node s and the base station. Let $C[s_1, s_2]$ be the absolute value of the correlation of the data between sensor nodes s_1 and s_2, with $C[s_1, s_2] \in [0, 1]$. Let $P[s]$ be the set of all shortest paths from the sensor s to the base station, where a path $p \in P[s]$ of length n is denoted by $p : \langle (s_1, s_2), (s_2, s_3), \cdots, (s_{n-1}, s_n) \rangle$ with $s_1 = s$. Let $E[s]$ be the residual energy of the sensor s at the end of the first phase in ASBP protocol. Let $x[s]$ be a Boolean variable with value 1 if the sensor node s is selected for the data collection, and 0 otherwise. Let $q[s]$ represent the maximum achievable path quality among all possible shortest paths from sensor s to the base station, in a solution to the sensor selection problem. We define the data utility $u[s]$ for sensor s to be the path quality of s subtracted by the correlation of the data between s and all other sensor nodes ($\forall s \in S$):

$$u[s] = w_1 \cdot x[s] \cdot q[s] - w_2 \cdot \sum_{\substack{s' \in S, \\ s' \neq s}} x[s] \cdot x[s'] \cdot C[s, s'] \quad (1)$$

where w_1 and w_2 are non-negative weight coefficients used to normalise the two terms. The combined objective considering the data utility $u[s]$ and the residual energy $E[s]$ of the sensor nodes becomes:

$$\texttt{maximise} \quad \textstyle\sum_{s \in S} E[s]^{\alpha} \cdot u[s] \tag{2}$$

where α is a parameter to adjust the weight of the energy coefficient on the data utility (typically α is set to 0.5).

The *routing constraint* enforces that if sensor s is selected then the path quality $q[s]$ must be maximum among the shortest paths between s and the base station ($\forall s \in S$):

$$q[s] = \max_{p \in P[s]} \left(B[n_p] \cdot \prod_{(s',s'') \in p} \Big(x[s''] \cdot L[s', s''] \Big) \right) \tag{3}$$

where n_p is the nearest neighbour sensor to the base station on the path p, and s', s'' are two adjacent sensors on the path p.

The *path quality* constraint enforces that the path quality $q[s]$ from a selected sensor to the sink must exceed a given threshold t:

$$\forall s \in S, \quad q[s] \geq x[s] \cdot t \tag{4}$$

where the threshold t is adjusted to provide a consistent packet delivery on a path to the sink (typically $t \in [0.3, 0, 7]$).

The *active sensor* constraint enforces that the minimum number of active sensors is at least m:

$$\textstyle\sum_{s \in S} x[s] \geq m \tag{5}$$

where m provides a trade-off between energy efficiency and data quality (belief propagation inference error).

In summary, our constraint programming model consists of the objective (2) and constraints (1), (3), (4), and (5).

Our CP model also benefits from implied constraints and customised search procedure, which is not presented in this paper due to space reasons.

Algorithm 1: The heuristic-based greedy algorithm

input : S, L, B, C, E, m, t
output: $\mathit{idSelected}$

1 $\mathit{idSelected} \leftarrow S$
2 $q \leftarrow \texttt{BestShortestPath}(L, B)$
3 $\mathit{idNonReachable} \leftarrow \{s \in S \mid q[s] < t\}$
4 **if** $\mathit{idNonReachable} \neq \emptyset$ **then**
5 $\quad \mathit{idSelected} \leftarrow \mathit{idSelected} - \mathit{idNonReachable}$
6 $\quad \texttt{SetZero}(L, \mathit{idNonReachable})$
7 $\quad \texttt{SetZero}(B, \mathit{idNonReachable})$
8 $\quad \texttt{SetZero}(C, \mathit{idNonReachable})$
9 $\mathit{idFeasible} \leftarrow \mathit{idSelected}$
10 **while** $\mathit{idFeasible} \neq \emptyset \ \wedge |\mathit{idSelected}| > m$ **do**
11 $\quad L' \leftarrow L$
12 $\quad B' \leftarrow B$
13 $\quad \mathit{idMin} \leftarrow \min\left(\arg\min_{s \in \mathit{idFeasible}} \left(E[s]^{\alpha} \cdot u[s]\right)\right)$
14 $\quad \texttt{SetZero}(L', \mathit{idMin})$
15 $\quad \texttt{SetZero}(B', \mathit{idMin})$
16 $\quad q \leftarrow \texttt{BestShortestPath}(L')$
17 $\quad \mathit{idNonReachable} \leftarrow \mathit{idSelected} \cap \{s \in S \mid q[s] < t\}$
18 $\quad \mathit{idPotential} \leftarrow \mathit{idSelected} - \mathit{idNonReachable}$
19 $\quad$ **if** $|\mathit{idPotential}| < m$ **then**
20 $\qquad \mathit{idFeasible} \leftarrow \mathit{idFeasible} - \mathit{idMin}$
21 $\qquad$ **continue**
22 $\quad \mathit{idSelected} \leftarrow \mathit{idPotential}$
23 $\quad \mathit{idFeasible} \leftarrow \mathit{idSelected}$
24 $\quad \texttt{SetZero}(L, \mathit{idNonReachable})$
25 $\quad \texttt{SetZero}(B, \mathit{idNonReachable})$
26 $\quad \texttt{SetZero}(C, \mathit{idNonReachable})$
27 **return** $\mathit{idSelected}$

B. The Greedy Search Algorithm

Our heuristic-based algorithm is listed in Algorithm 1. The intuition behind this algorithm is that we should remove a sensor if 1) the data from the sensor are strongly correlated with the others, meaning that we can predict fairly accurately the reading from that sensor; 2) the sensor is already over-used, meaning that the sensor has a low energy; 3) the sensor has a poor connection to the sink node, meaning that the data transmission from that sensor has a high risk to fail. Thus, we do a greedy selection by taking all three aspects into consideration and remove sensors one by one until we are left with the required number of sensors.

Our heuristic algorithm returns a set $\mathit{idSelected}$ of selected sensors. The algorithm takes the same input constants S, L, B, C, E, and variables m, and t as explained in Section IV-A.

The heuristic algorithm creates a set of selected sensors $\mathit{idSelected}$ (line 1), and initialise it with all the possible sensor ids. The function `BestShortestPath` (line 2) returns an array q of path quality values from each sensor to the base station, while respecting the *path quality* constraint (4).

The heuristic algorithm maintains a set $\mathit{idNonReachable}$ of sensor nodes that are not able to reach the base station due to the violation of the *path quality* constraints (4) (line 3). Sensor nodes in the set $\mathit{idNonReachable}$ (line 4) are removed from the set of selected sensor nodes (line 5), and the values of link quality, base station link quality, and correlation for those sensor nodes in $\mathit{idNonReachable}$ are set to 0 from the corresponding data using the function `SetZero` (lines 6–8). The function $\texttt{SetZero}(A, \mathit{Ids})$ takes a $n \times n$ matrix A, and a set of indices Ids, and for each index i in Ids sets the value of every possible pair of (i, j) $1 \leq j \leq n$ in A to zero ($A(i,j) = 0 \wedge A(j,i) = 0, \ 1 \leq j \leq n$), and if A is a one dimensional array, then it only sets $A(i) = 0$.

The main loop of the algorithm (line 10) iteratively selects a sensor node that contributes the least value to the objective (2) (equivalent to a sensor node with the lowest data utility weighted by the initial energy), and performs a lookahead move (lines 11–21) to detect if removing this sensor node violates any of the constraints. To perform the lookahead move, the link quality data for the sensor idMin is set to zero (lines 14–15), and then the path quality q is updated to discover the non-reachable sensor nodes $\mathit{idNonReachable}$ (line 16).

if *active sensor* constraint (5) (line 19) is violated, then we skip to the next iteration (line 21). Otherwise, we update the set $\mathit{idFeasible}$ of feasible sensors and continue to the next iteration until no more feasible sensor node is remained, or m sensor nodes are selected.

V. Bayesian Inference and Data Quantization

This section describes how to use belief propagation to infer the missing data because of the inactive sensor nodes and the data transmission losses of the active sensor nodes throughout the second phase of our ASBP protocol.

A. Introduction to Belief Propagation

Belief propagation (BP) is a classic algorithm for performing inference on graphical models [3], [4]. In general, it assumes

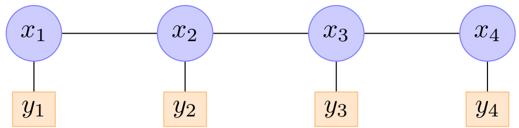


Fig. 2: An example of a graphical model.

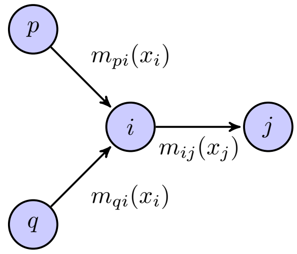


Fig. 3: A graphical depiction of message passing from nodes p and q to the node i in belief propagation (BP). The updated message $m_{ij}(x_j)$ is then sent to the node j.

that some observations are made and the task is to infer the underlying events behind these observations. Denote y_i the observation at node i and x_i the underlying event, $i = 1, \dots, N$. For the application of WSNs, y_i is the reading of sensor i about some phenomenon that is being monitored, such as the temperature, and x_i is the true reading of the phenomenon. Clearly, there are some statistical dependencies between y_i and x_i, encoded in a so-called evidence function $\phi_i(x_i, y_i)$. Very often, we consider the observation y_i to be fixed and write $\phi_i(x_i)$ as a short-hand of $\phi_i(x_i, y_i)$. Furthermore, there are also statistical dependencies between the several underlying events x_i, encoded in a so-called potential function $\phi_{ij}(x_i, x_j)$. For WSNs, the potential function captures spatial correlations between the readings at nearby sensors.

Given the above notation, the inference of the x_i can be formulated as the maximisation of the following belief function:

$$b(\{x_i\}_{i=1}^N) = \prod_{ij} \phi_{ij}(x_i, x_j) \prod_i \phi_i(x_i). \tag{6}$$

A graphical depiction of this model is shown in Figure 2. The rectangles are the observation nodes y_i and the circles represent the underlying events x_i. The potential functions are associated with the links between x_i and the evidence functions are associated with the links between y_i and x_i.

BP performs inference by passing messages between nodes in the graph. The message from i to j is defined as:

$$m_{ij}(x_j) = \sum_{x_i} \phi_i(x_i)\phi_{ij}(x_i, x_j) \prod_{k \in N(i), k \neq j} m_{ki}(x_i), \tag{7}$$

where $N(i)$ denotes the neighbours of node i. The message essentially integrates all messages from the neighbours of i, except j, as well as the local evidence seen at i. The BP inference is done by maximising the belief at each node:

$$b_i(x_i) = \phi_i(x_i) \prod_{j \in N(i)} m_{ji}(x_i). \tag{8}$$

The message passing process in BP is illustrated in Figure 3. In this paper, we adopt the max-product variation of BP [10].

B. BP for Inference on WSNs

In using BP for inferring the missing data on WSNs, we need to construct a graph to model the correlations between sensor readings. There are two types of correlations on WSNs:

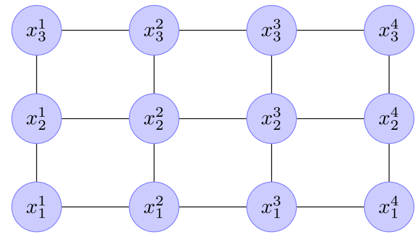


Fig. 4: A depiction of the graphical model built for WSNs.

a) Spatial correlation: Data from different sensors may be correlated. Note that we do not assume that strong correlations always exist between data from nearby sensors. Instead, we compute the correlation coefficients between each pair of sensor nodes from the observed data. We claim spatial correlations only when we see large correlation coefficients, regardless of the spatial distance between two sensors.

b) Temporal correlation: Data from the same sensor may be correlated over time. Here, we assume that the sensor reading at time t is strongly correlated with that at time $t-1$.

Thus, we built our graph as illustrated in Figure 4 where x_i^t denote the true reading of sensor i at time t. The link between x_i^t and x_i^{t-1} represents the temporal correlations, with a temporal potential function defined $\phi_i^t(x_i^t, x_i^{t-1}) = \exp(-\frac{(x_i^t - x_i^{t-1})^2}{\sigma_i^2})$. Similarly, the link between x_i^t and x_j^t represents the spatial correlations, with a spatial potential function defined $\phi_{ij}^s(x_i^t, x_i^{t-1}) = \exp(-\frac{(x_i^t - x_j^t)^2}{\sigma_{ij}^2})$. Note that the noisy sensor reading y_i^t is omitted from the graph for the purpose of simplification, and the evidence function associated with the link between x_i^t and y_i^t is defined $\phi_i^e(x_i^t, y_i^t) = \exp(-\frac{(x_i^t - y_i^t)^2}{\sigma_i^2})$. y_i^t can be missing for two reasons: either sensor i is in the sleep mode or the packet failed to reach the sink node. When it is missing, we turn the evidence function into a constant, i.e., $\phi_i^e(x_i^t, y_i^t) = 1$, for all possible values of x_i^t. Note that σ_i and σ_{ij} are parameters that can be learned from some training data [11].

Comparing to alternative approaches such as the *compressed sensing* based approach in [8], BP based on the graph in Figure 4 is advantageous for two reasons: 1) BP captures the spatial and temporal correlations between sensors explicitly via a graphical model which is updated over time. For example, we can disconnect the sensor nodes when the correlation coefficients drop below some threshold. 2) BP allows the incremental inference that infers the missing data at time t from the available data at just time t and $t-1$. In contrast, the CS-based approach in [8] takes as input a data matrix with missing entries, and thus can only perform inference in a batch mode for a time interval.

We will demonstrate these advantages and the inference accuracy in Section VI.

C. Data Quantization

Quantisation is a classic technique in signal processing that has been widely used for data compression [12]. In summary, quantised measures are less fine-grained and lossy, however there are many advantages in using a quantised measure:

- A quantised measure is informative enough for describing the correlation between the data.
- A quantised measure can be encoded into a few bits,

saving storage and transmission costs.

- A quantised measure is coarse and thus cheaper to obtain. It is also stable and highly adjustable to match the needs of the network application.

Let the metric to be quantised take on values in the range $[r_{\min}, r_{\max}]$, and values outside this interval are mapped either to $r_{\min}$ or $r_{\max}$. The quantisation is done by partitioning the interval into R bins using $R-1$ thresholds, denoted by $\tau = \{\tau_1, \ldots, \tau_{R-1}\}$. Each bin is represented by a value within the range of the bin, e.g., the centroid point of the bin's range. Let the value b_i represent the i^{th} bin. A table look-up is used to map the metric value to b_i according to the bin threshold:

$$Q(x) = b_i, \quad \text{if } \tau_{i-1} < x \leqslant \tau_i,\ i = 1, \ldots, R. \tag{9}$$

where $\tau_0 = r_{\min}$ and $\tau_R = r_{\max}$. The bin indices values $\{b_1, \ldots, b_R\}$ are stored in a codebook, and a metric value is represented by a bin index, that is encoded into few bits.

In general, the thresholds τ are chosen according to the requirements of the application, adaptively adjusted, or learned from a set of training data.

VI. EXPERIMENTS

A. Experiment Setup

We experiment with the real data collected from 54 sensor nodes deployed in the Intel Berkeley Research Lab [9]. The data is collected by a base station, and includes temperature, humidity, light intensity, and voltage values once every 30 seconds, throughout a time span of 36 days. The data set also includes aggregated link quality data. In our simulations, we selected a time interval of 10 hours, consisting of 5 rounds of two hours.

We apply a uniform quantisation on the temperature data in the range of $[0, 50]$ into 256 bins. The values outside the interval are mapped to the minimum and maximum of the interval accordingly.

In our energy consumption evaluations, we consider 14mA transmission cost, as reported for the Mica2Dot mote [13], used in the Intel lab deployment.

In our simulations, phase one of a round ends when at least 20 data readings are collected from all the sensor nodes. The weights w_1 and w_2 in data utility (1) are chosen to normalise the path quality and correlation. We expect that at least m sensor nodes are selected, hence the path quality is scaled by the minimum number m (5) of sensor nodes ($w_1 = m$), because the sum of the correlation is at least m we set $w_2 = 1$.

B. Results and Analysis

We evaluate the performance of our ASBIP in terms of data utility, energy efficiency, and data prediction accuracy. We compare the results of our constraint programming model, heuristic-based algorithm, and a random sensor selection. Our simulation of the ASBP protocol is implemented in C++, and the CP model is implemented using the constraint programming solver *Gecode* [14] (revision 4.2.1).

Figures 5a, and 5b compare the total data utility and energy consumption achieved in one round by the ASBP protocol using CP, our heuristic-based algorithm, and random sensor selection, with a minimum of 30% and 70% for the base station link quality, respectively. For each result, we vary the parameter m in (5) to control the total number of selected sensor nodes for data collection. Note that we only increase the base station minimum threshold, whereas the minimum

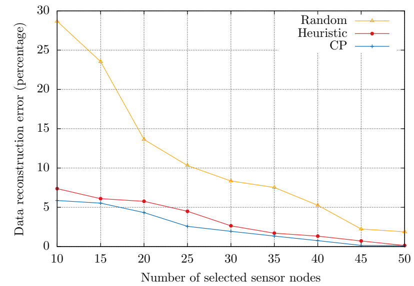


Fig. 6: The prediction mean square error of BP with the CP model, our heuristic-based algorithm, and random sensor selection strategies, upon varying the minimum number m of selected sensor nodes.

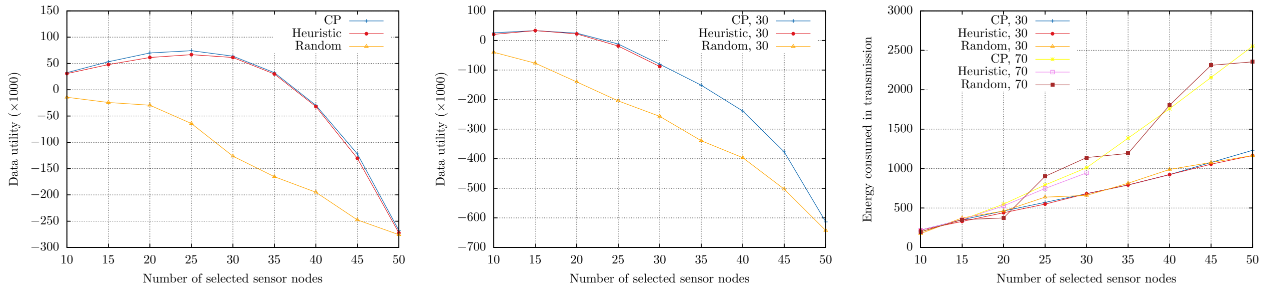
link quality between the sensor nodes is 30%. The increase in the minimum base station link quality to 70% affects the multi-hop routing. It increases the path size to 5 hops, which requires the sensor nodes closer to the base station to relay also the data for the nodes further away. Hence, the path quality $q[s]$ is decreased, and the total data utility is reduced.

The results show that the general traditional random approach does perform very poorly compared to the global optimum. The results for the random sensor selection are computed by taking the mean of the data utility and energy consumption for 10 random sensor selections. We observe that the data utility increases up to 25 selected nodes and then decreases. This is because of the trade-off between the path quality and the correlation. As the number of selected sensors increases the sum of the data correlation between a selected sensor node and all the other sensor nodes becomes a larger factor in the data utility term (1) compare to the path quality term, hence the data utility decreases. We conclude that an efficient sensor selection strategy should select 25 sensor nodes in order to maintain a balance between the path quality and the data correlation.

The heuristic-based strategy in Figure 5b fails to find a solution for more than 30 selected sensor nodes, because of the more limited requirement of 70% link quality, and without backtracking the greedy algorithm fails at maintaining a route to the base station for all selected sensor nodes.

The total energy consumption for the data transmissions with both settings 30% and 70% on the minimum base station link quality is shown in Figure 5c. The minimum base station link quality is denoted in the legend of the plot. We observe that at the same threshold on the base station link quality, the energy consumption is almost independent of the sensor selection strategy. However, the energy consumption is almost doubled as the base station link quality threshold is increased to 70%, which is due to the additional multi-hop relay of the data required to reach the base station.

Figure 6 shows the BP results with the CP model, heuristic-based algorithm, and random sensor selection strategies, upon varying the minimum number m of selected sensor nodes. We first compute the mean square error (MSE) of the predicted data versus the ground truth for each sensor node in the temporal domain. The result is an array of 54 MSE values on the sensor node predicted data. We then plot the mean of the MSE error in Figure 6. The results for the random sensor selection are computed by taking the average of 10 runs. The standard deviation of CP and the heuristic-based algorithm is at most 12%. The CP model with $m = 10$ has an average error



(a) Minimum 30% base station link quality (b) Minimum 70% base station link quality (c) Energy consumption

Fig. 5: Data utility and energy consumption for data transmission obtained by simulating the ASBP protocol in one round and solving the sensor selection problem with the CP model, our heuristic-based algorithm, and random sensor selection. Minimum thresholds of 30% (Figure 5a) and 70% (Figure 5b) were used for the base station link quality, upon varying the minimum number m of selected sensor nodes.

of about 6%, which indicates that in the temporal domain in average the prediction of the belief propagation deviates 6% from the ground truth. At the same data point, the standard deviation (SD) is about 12%, and increasing the number of selected sensor nodes m always drops the value of SD. As we expected, the best sensor selection (by CP) achieves the minimum error, whereas the random sensor selection does not consider the correlation of the data, and as a result has a higher prediction error.

The results compared with the energy consumption in Figure 5c show that we can save up to 80% energy by selecting only 10 sensor nodes to be active for the data collection in each round, while maintaining at most the satisfactory average error of 6% with an SD of 12% in the prediction accuracy. In our approach, depending on the application and the required accuracy, we can adjust the selected number of sensor nodes as a trade-off between the energy consumption and data quality (accuracy of the belief propagation).

VII. Conclusion

We have presented a novel adaptive sensing belief propagation (ASBP) protocol for energy-efficient data collection in wireless sensor networks. The ASBP protocol is designed to take advantage of data quantisation and spatio-temporal correlation of the data in order to prolong the lifetime of the network. ASBP solves an optimisation problem to select an optimal set of active sensor nodes that maximises the data utility and achieves energy load balancing. The data utility measures the path quality (the packet reception probability over a path in the network), and correlation of the data for the active sensor nodes. In our protocol, belief propagation (BP) iteratively infers the values of the missing data from the stream of active sensor readings. The accuracy of the BP is adjustable based on the minimum number of active sensors. We formulate and solve the active sensor selection optimisation problem using constraint programming (CP), and compare it with our heuristic-based greedy algorithm.

We have evaluated our ASBP protocol using real data collected at the Intel lab sensor deployment. The simulation results show that our ASBP protocol can greatly improve energy-efficiency up to 80%, with the optimal CP active sensor selection, while maintaining in average 6% error in the BP data inference.

As future work, we plan to extend our ASBP protocol to a fully distributed implementation for real deployment, and compare versus our current optimal results. We are also interested to integrate adaptive sampling rate into our current results, as well as investigating multi-sink scenarios.

Acknowledgements

This research is in part sponsored by the Swedish Foundation for Strategic Research (SSF) under research grant RIT08-0065 for the project *ProFuN: A Programming Platform for Future Wireless Sensor Networks*, and sponsored by the U4 Strategic University Network DAAD Programme Strategic Partnership and Thematic Networks. Special thanks to the reviewers for pointing out ways to improve the presentation of this paper.

References

[1] J. Amaro, F. J. T. E. Ferreira, R. Cortesao, N. Vinagre, and R. Bras, "Low cost wireless sensor network for in-field operation monitoring of induction motors," in *ICIT*, March 2010, pp. 1044–1049.

[2] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[3] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Exploring artificial intelligence in the new millennium," G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann Publishers Inc., 2003, pp. 239–269.

[4] F. V. Jensen, *Introduction to Bayesian Networks*, 1st ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1996.

[5] F. Rossi, P. van Beek, and T. Walsh, Eds., *Handbook of Constraint Programming*. Elsevier, 2006.

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *HICSS*, vol. 8, 2000, p. 8020.

[7] J. Chou, D. Petrovic, and K. Ramachandran, "A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks," in *INFOCOM*, vol. 2, 2003, pp. 1054–1062.

[8] L. Kong, M. Xia, X.-Y. Liu, M.-Y. Wu, and X. Liu, "Data loss and reconstruction in sensor networks," in *IEEE INFOCOM*, 2013.

[9] S. Madden, "Intel lab data, 2004," Available from http://www.select.cs.cmu.edu/data/labapp3/index.html, 2014.

[10] Y. Weiss and W. T. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Transactions on Information Theory*, vol. 47, pp. 736–744, Sep. 2006.

[11] J. Su, H. Zhang, C. X. Ling, and S. Matwin, "Discriminative parameter learning for bayesian networks," in *Proceedings of the 25th ICML*. ACM, 2008, pp. 1016–1023.

[12] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.

[13] G. Anastasi, A. Falchi, A. Passarella, M. Conti, and E. Gregori, "Performance measurements of motes sensor networks," in *Proceedings of the 7th MSWiM*. ACM, 2004, pp. 174–181.

[14] Gecode Team, "Gecode: A generic constraint development environment," 2006, available from http://www.gecode.org/.