Cloud-Assisted Data Fusion and Sensor Selection for Internet-of-Things

Praveen Jayasankar, Department of Computer Science, Meenakshi Sundararajan Engineering College, praveenjayasankar@gmail.com, +91-9176559214

Rachel Hannah, Department of Computer Science, St. Joseph's College of Engineering, rachellannah007@gmail.com, +91-9790925384

Abstract—The Internet of Things (IoT) is connecting people and smart devices on a scale that was once unimaginable. One major challenge for the IoT is to handle vast amount of sensing data generated from the smart devices that are resource-limited and subject to missing data due to link or node failures. By exploring cloud computing with the IoT, we present a cloud-based solution that takes into account the link quality and spatio-temporal correlation of data to minimise energy consumption by selecting sensors for sampling and relaying data. 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 formulate the sensor selection problem and solve it using both constraint programming (CP) and greedy search. We then use our message passing algorithm (belief propagation) for performing inference to reconstruct the missing sensing data. ASBP is evaluated based on the data collected from real sensors. The results show that while maintaining a satisfactory level of data quality and prediction accuracy, ASBP can provide load balancing among sensors successfully and preserves 80% more energy compared with the case where all sensor nodes are actively involved.

I. INTRODUCTION

The Internet has enabled an explosive growth of information sharing. With the advent of embedded and sensing technology, the number of smart devices including sensors, mobile phones, RFIDs, and smart grids, has grown rapidly in recent years. Ericsson and Cisco predicted that 50 billion small embedded sensors and actuators will be connected to the Internet by 2020 [1] forming a new Internet paradigm called Internet-of-Things (IoT). IoT can support a wide range of applications in different domains, such as health care, smart cities, pollution monitoring, transportation and logistics, factory process optimisation, home safety and security [2], [3].

In the past decade, many studies have contributed to the hardware, software, and protocol design of the smart devices, such as wireless sensor networks (WSNs) [4]–[6]. Machine to machine automation with wireless sensors is being widely deployed, but usually in islands of disparate systems. The evolution of IoT attempts to connect these existing systems to the cloud, which enables advanced data fusion, storage, and coordination capability for achieving higher data quality and energy efficiency. The upcoming challenge of IoT lies in handling volumes of data generated from enormous amount of devices, which is known as the big data problem.

The wireless sensors in many IoT applications are battery powered, resulting in extreme energy constraints on their operations, such as sampling, data processing and radio communications. To conserve energy and achieve longer network lifetime, the costs of sensor sampling, processing, and radio communications have to be minimised. It is often the case that sensor readings in the same spatial regions are highly correlated. Depending on the application, the sensor readings are temporally correlated as well. By leveraging the computation capability of the cloud, data fusion can be performed to increase the data quality by exploring the spatial and temporal correlation of data. The wireless sensors can be coordinated by the cloud to be on and off according to the change of the environment. In this paper, we explore a seamless solution by integrating cloud and IoT to provide comprehensive data fusion and coordination of sensors to improve data quality and reduce energy consumption.

Belief propagation (BP) [7]–[9] is a technique for solving inference problems. In the IoT context, the belief of a sensor node is the data measurement of an event in the environment, and BP provides an iterative algorithm (also called the sum-product algorithm) to infer the measurements of the sensor nodes, especially in cases where the data is missing, because of packet losses or because there is no data available at some selectively disabled sensor nodes (mainly to conserve energy and reduce radio inference). In BP each sensor node determines its belief by incorporating its local measurement with the beliefs of its neighbour sensor nodes (spatial cooperation), and its beliefs obtained in the past (temporal cooperation). In such inference problems, the assumption that the data is spatial-temporally correlated significantly improves the accuracy of data inference using BP in WSNs.

In monitoring applications for the IoT, the data is collected and put in an environment matrix (EM) [10], where the data readings for each sensor node are stored in one row of the matrix and each column index represents a timestamp for the interval at which the data was sampled. Hence, an EM is a matrix of size N x T where N is the number of sensor nodes and T the number of time intervals, and the time dimension T is expanding as more data is collected. BP performs the
inference iteratively from the stream of data that are stored in EM based on the current and past data. Therefore, unlike the compressed sensing (CS) [11] approach, BP does not require a complete EM for the whole duration of the time interval to perform inference.

In this paper, we explore cloud-assisted adaptive sensing and data fusion to reduce energy consumption and improve data quality for the IoT. 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 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. If the selected active sensor nodes are located on a path with greater link quality, then the value of the data utility increases. Likewise, if the selected active sensor nodes result in a lower data correlation, then the data utility is increased. In each round of ASBP, the minimum number of selected active sensor nodes (which is a parameter of our sensor selection optimisation problem) is adaptively tuned based on the performance of the BP inference (data prediction accuracy) throughout the previous round. 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. To model the sensor selection problem effectively, we use both constraint programming (CP) [12] and heuristic-based greedy algorithm. Constraint programming is a powerful framework to model and solve combinatorial problems. A constraint programming model consists of variables, variable domains, and constraints, and objective function (if required), in which the constraints express the relation between the variables. The core concept in CP is constraint propagation. Constraint propagation performs reasoning on a subset of variables, variable domains, and constraints to infer more restrictive variable domains, such that the restricted domains still contain all solutions to the problem. CP combines constraint propagation with search procedure to find a local or global optimum (using branch-and-bound search space exploration) to an optimisation problem.

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 belief propagation (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, namely constraint programming (CP) and a heuristic-based greedy algorithm to solve the problem. The 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 (for both CP and heuristic-based algorithm) 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 on sensor selection, and we solve it using two approaches (CP and heuristic-based greedy algorithm). In Section VI, we conduct simulations to evaluate our solutions based on a real deployment of a wireless sensor network. Finally, we summarise and conclude the paper in Section VII.

II. RELATED WORK

The information industry benefits greatly from the technological advancements brought by the IoT [13], [14]. The IoT creates a bridge between many available and recent technologies, such as wireless sensor networks (WSNs), cloud computing, and information sensing [14]–[16]. In monitoring and data acquisition IoT-based systems, it is necessary to collect data effectively and efficiently [14], [15], [17], [18]. The IoT provides a platform for WSNs to connect to internet and benefit from the power of cloud computing and data fusion. Therefore, it is necessary to study data collection schemes that can seamlessly integrate with the cloud and IoT systems. Data collection has been widely studied for stationary wireless sensor networks. Gnawali et al. [19] present the state-of-the-art routing protocol for a sensor network where the nodes are forwarding data directly to a sink. They consider stationary WSNs that have static routes from the wireless sensors to the sink. Madden et al. [20] introduced a distributed query processing paradigm called acquisitional query processing (ACQP) for sensor network data collection. The goal was to ensure a flexible tasking of motes via a relational query interface, while providing lifetime constraints, data prioritisation, event batching, and rate adaptation.

Prediction-based energy-efficient approaches aim at predicting the data to minimise the number of transmissions. Chou et al. [21] proposed a distributed compression based on source coding, which highly relies on the correlation of the data, and it compresses the sensor readings with respect to the sensor past readings, and the reading measured by the other sensor nodes. 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 addressed the use of compressive sensing [11]. The authors use compressive sensing to exploit the temporal stability, spatial correlation, and the low-rank structure of the environment matrix (EM). They propose an environ-mental space time improved compressive sensing (ESTI-CS) algorithm to improve the missing data estimation. Although compressive sensing achieved good accuracy on the estimation of the missing data, it does only consider implicit spatio-temporal correlation in the data. Furthermore, compressive sensing approaches rely on the construction of a data matrix and thus require the synchronization of the sensors on the data collection. However, in our work we present a belief propagation approach for the prediction of missing data, where the spatio-temporal correlation is explicitly enforced and the inference is performed online and iteratively as the data is...
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 IoT application, stationary sensor nodes collect environmental data, such as temperature, humidity, light intensity, noise level, etc. Figure 1 shows the network architecture of our data collection in IoT applications. We support heterogenous networks, where data can be collected from various devices. The network supports multi-hop routing and the gateways collect the data and forward the data to the cloud, where the data fusion is performed to further analyse the data, predict missing data, and store the data in the data centres.

The highly distributed nature of the cloud systems reduces the latency between the gateways and the cloud servers, and it also provides load balancing for the computation overhead and the data storage. The computation power of the servers in the cloud is used to improve data quality and save energy of the sensor nodes using our ASBP protocol (to be discussed further in Section III-B). Our ASBP protocol requires both the computational power and the media delivery of the cloud services. It is crucial for the ASBP protocol to leverage the computational power of the cloud to perform the offline pre-processing and the online sensor selection. However, the media delivery in our protocol (i.e., monitoring, analysis, and visualisation of the data) can also be performed using content delivery networks (CDNs) [22].

The sensor nodes periodically sample data, which is forwarded to the cloud using a multi-hop routing protocol (the acquisitional query processing system in TinyDB [20], or the collection tree protocol [19]). In this work, we use the real data collected at the Intel Berkeley research lab [23]. Figure 2 shows the map of the Intel Berkeley research lab, and the location of the deployed sensor nodes, which are marked with hexagon shapes, and the sensor id. The link thickness between the sensor nodes represents the value of the link quality aggregated throughout the experiment.

The data is collected at the cloud using the gateways associated with different applications of IoT. The gateway only relays the data to the servers in the cloud, and it is at least aware of the routing tables of the sensor nodes. In this paper, we refer to the gateway and the base station as the same entity, however the actual computations (the CP solver and greedy algorithm in Section IV-A and IV-B) are performed on the cloud, and all coordinations are relayed by the gateway.

B. Protocol Design

In our setup, the sensor nodes collect and report the data periodically (typically every 30 seconds). Our protocol operates in several rounds (a round is a time interval where the network repeats the same behaviour), and each round includes two phases. The first phase is used to collect the minimum required information, 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 is generally acting as a gateway between the sensor network and the cloud, and it also forwards the computations to be performed on the cloud as shown in Figure 1. The cloud then computes the correlation coefficient matrix from the sensor data, and also uses the routing tables to compute 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) and select a subset of sensor nodes to be active during the second phase. The active sensor nodes are the only sensor nodes in the network that are participating in the data.

Fig. 1: The network architecture, where the nodes in an IoT application forward the data to the cloud. The servers perform node coordination to improve data quality and save energy, while the data centres stores the collected data as the data fusion and the data loss prediction is performed.

Fig. 2: The map of the Intel Berkeley research lab, with the hexagon shape nodes indicating the locations and the ids of the sensor nodes, which are deployed to monitor temperature, humidity, and light intensity. The value of the aggregated link quality is represented with the thickness of the link between the sensor nodes.
collection and relaying the data to the base station.

The sensor selection problem is solved on the cloud us-ing 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) [8], [9] to infer incrementally the missing data due to the inactive sensor nodes and packet losses (further explained in Section V). BP captures the high spatio-temporal correlation in the data using a graphical model, which is taken into account in modelling our sensor selection optimisation problem. 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 finds a global optimum solution to the problem, whereas our heuristic-based algorithm finds a good quality local optimum solution. The CP model selects the routes for packet relays dynamically. Dynamic routing is essential for networks with multiple shortest paths to the base station, large varieties in the link quality, and energy-efficiency concerns. However, a heuristic-based greedy algorithm with good quality solutions is well suited for networks where the sensor selection problem cannot be solved in a centralised way, and data accuracy is of less concern.

A. Constraint Programming Model with Dynamic Routing

As we mentioned in Section III-B, throughout phase one of ASBP the data is collected in the environment matrix (EM), which is used to compute the correlation coefficient matrix. We also estimate the link quality from the packet reception rate during phase one. We then have the following constants in our sensor selection model:

Let S be the set of WSN sensor nodes, with $|S| = N$. Let $L[s \, \models \, s_1; s_2]$ 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 $s_1; s_2 \in \mathbb{S}$. If $s_1$ is not the direct neighbour of $s_2$, then $L[s \, \models \, s_1; s_2] = 0$.

Let $B[s]$ be the link quality between the base station and a direct neighbour sensor node $s$, and otherwise $B[s] = 0$.

Let $s; s_j$ be sensor $i$ and sensor $j$ with $s; s_j \in \mathbb{S}$.

Link quality between neighbour sensor nodes $s_1$ and $s_2$.

$B[s]$ Link quality between the base station and a direct neighbour sensor node $s$, and otherwise $B[s] = 0$.

For the following constants in our sensor selection model:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N$</td>
<td>Total number of sensor nodes.</td>
</tr>
<tr>
<td>$S$</td>
<td>Set of all sensor nodes with $</td>
</tr>
<tr>
<td>$s; s_j$</td>
<td>Sensor $i$ and sensor $j$ with $s; s_j \in \mathbb{S}$.</td>
</tr>
<tr>
<td>$L[s , \models , s_1; s_2]$</td>
<td>Link quality between sensor nodes $s_1$ and $s_2$.</td>
</tr>
<tr>
<td>$B[s]$</td>
<td>Absolute value of the correlation of the data between sensor nodes $s_1$ and $s_2$, with $C[s_1; s_2] = 2 \cdot [0; 1]$.</td>
</tr>
<tr>
<td>$p$</td>
<td>Path $p$ is denoted by $p : h(s; s_1); (s_1; s_2); (s_2; s_3); \ldots; (s_n; s_1)$ from sensor $s_1$ to $s_n$ with length $n$ and $s_n$ is directly linked to the base station.</td>
</tr>
<tr>
<td>$P[s]$</td>
<td>Set of all shortest paths $p$ from the sensor $s$ to the base station.</td>
</tr>
<tr>
<td>$E[s]$</td>
<td>Residual energy of the sensor $s$.</td>
</tr>
<tr>
<td>$x[s]$</td>
<td>Boolean variable with value 1 if sensor node $s$ is selected for the data collection, and 0 otherwise.</td>
</tr>
<tr>
<td>$q[s]$</td>
<td>Maximum achievable path quality among all possible shortest paths from sensor $s$ to the base station.</td>
</tr>
</tbody>
</table>

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 $C[s_1; s_2] = 2 \cdot [0; 1]$.

Let $P[s]$ be the set of all shortest paths from the sensor $s$ to the base station, where a path $p 2 P[s]$ of length $n$ is denoted by $p : h(s; s_1); (s_1; s_2); (s_2; s_3); \ldots; (s_n; s_1)$ with $s_1 = s$ and $s_n$ is directly linked to the base station.

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. A summary of the introduced notations are given in Table I.

A summary of the introduced notations in our model.

We require the maximisation of the path quality:

$$\max_{s \in S \, x[s]} \quad s_2 S \quad x[s] \quad q[s] \quad (1)$$

A second objective is to minimise the correlation of the data between the selected sensors. This objective implies that data from the inactive sensors is more likely to have a high correlation with the enabled sensors, hence improving the accuracy of the missing data construction:

$$\min_{s; s_j \in S} \quad x[s] \quad x[s_j] \quad C[s, s_j] \quad (2)$$

We define the data utility $u[s]$ to be the weighted linear sum of the two objective terms in (1) and (2) for sensor $s$:

$$8s 2 S \quad u[s] = \quad \lambda_1 x[s] \quad q[s] \quad (3)$$

$$\quad \lambda_2 \quad \max_{s \in S \, s_j \in S \, x[s]} \quad x[s_j] \quad C[s, s_j] \quad (3)$$

where $\lambda_1$ and $\lambda_2$ are non-negative weight coefficients used to normalise and allow preference adjustment between the path quality and the aggregated correlation of the data for sensor $s$ versus all the other sensors in the network.
The combined objective considering the data utility \( u[s] \) and the residual energy \( E[s] \) of the sensor nodes becomes:

\[
\begin{align*}
\text{maximise} & \quad \sum_{s \in S} E[s] u[s] \\
\text{subject to} & \quad \text{routing constraints} (7),
\end{align*}
\]  

(4)

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

The path quality constraint enforces that the path quality \( q[s] \) from a selected sensor to the base station must exceed a given threshold:

\[
8s \in S; \quad q[s] \geq x[s]
\]  

(5)

where the threshold is adjusted according to the link quality to provide a consistent packet delivery on a path to the base station (typically \( 2 \leq 0.3 \leq 0.7 \)).

The routing constraint enforces that a path with higher quality is preferred in selecting the active sensors and all sensors on such a path must be active. For example, Figure 3 shows two paths \( p_1 \) and \( p_2 \) from sensor node 1 to the base station B. We assume that the link quality between sensor nodes on a path to the base station is an independent random variable. Therefore, path quality is the joint probability of the link quality probabilities along a path to the base station:

\[
q[p_1] = L[1; 5] L[5; 6] L[6; 4] B[4] = 0.78 \times 0.88 \times 0.65 \times 0.91 = 0.406
\]

\[
q[p_2] = L[1; 5] L[5; 6] L[6; 4] B[4] = 0.86 \times 0.75 \times 0.66 \times 0.91 = 0.387
\]

where \( q[p_1] \) denotes the path quality of the path \( p_1 \) originated at sensor s. The path \( p_1 \) has a higher path quality \( q[p_1] > q[p_2] \). Hence, when maximising the path quality \( q[p_1] \) the path \( p_1 \) is preferred for routing the data, and to enforce that all sensors on the path must be selected, the path quality \( q[p_1] \) is constructed as follows:

\[
\]  

(6)

Assuming sensor 1 is selected \( x[1] = 1 \), the path quality constraint (5) requires that \( q(1) > 0 \), and according to (6), either all sensors on path \( p_1 \) or \( p_2 \) must be active \( x[5] = x[6] = x[4] = 1 \) or \( x[2] = x[3] = x[4] = 1 \). Note that the routing constraint (6) must be enforced only if the origin sensor 1 is selected \( x[1] = 1 \), and otherwise the value of the path link should not be included in the objective function (4). Therefore, the non-linear term \( x[s] q[s] \) is used in the construction of data utility (3). In general, the routing constraint becomes \( 8s \in S \):

\[
q[s] = \sum_{p \in P} x[p] B[p] \prod_{i=1}^{k} x[i] L[i; s] C[s; s^0] x[s^0] L[s^0; s^0] L[s^0; s^0]
\]  

(7)

where \( n_p \) is the last sensor on the path \( p \), and \( s^0, s^0 \) are two adjacent sensors on the path \( p \). For example, in Figure 3 for the paths \( p_1 \) and \( p_2 \) we have \( n_p = n_{p_2} = 4 \). The \( n \)-ary constraint max is essential in our CP implementation of the routing constraints (7).

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

\[
\sum_{s \in S} x[s] \geq \max
\]

(8)

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

In summary, our constraint programming model for the sensor selection problem is defined as:

**Inputs:**
- \( L \): Link quality estimations.
- \( B \): Base station link quality estimations.
- \( C \): Correlation coefficient matrix.
- \( E \): Residual energy.

**Outputs:**
- \( x \): Selected sensors with \( x[s] = 1 \) iff sensor \( s \) is selected for data collection and \( x[s] = 0 \) otherwise.
- \( u[s] \): Data utility of sensor node \( s \).
- \( q[s] \): Path quality achieved in the routing of data from sensor node \( s \) to the base station.

**Objective:**

\[
\text{maximise} \quad \sum_{s \in S} E[s] u[s] x[s]
\]  

\[
\text{subject to} \quad q[s] \geq x[s]
\]  

\[
\]  

such that:

\[
8s \in S; \quad u[s] \neq 0 \quad \land \quad x[s] \geq 0
\]

(9)

This CP model is directly expressed and solved in our chosen CP solver without further transformation to the formulation. For our CP implementation of this model, we derive implied constraints from the routing constraints (7), to reduce the search effort needed to solve the problem. We observe that some sensor nodes are often shared along the shortest paths from the origin sensor node \( s \) to the base station. For example, in Figure 3 sensor node 4 is shared by both paths \( p_1 \) and \( p_2 \). If
sensor node 1 is selected it implies that sensor 4 must be also selected regardless of which path is used in forwarding the data to the base station. We incorporate these implied constraints in our model to help improve the performance of the solver:

\[ 8s 2 S; (x[s] = 1) \rightarrow x[s] = 1 \forall s \in 2P [s] x[s^0] = 1 \] (9)

where \( P[s] \) is the intersection set of all sensor nodes on the paths from \( s \) to the base station. The implication (9) states that if the sensor node \( s \) is selected \((x[s] = 1)\), then the conjunction of all shared sensor nodes on the paths from \( s \) to the base station must be 1, enforcing that all the shared sensor nodes are part of the solution \((x[s^0] = 1; s^0 \in 2P[s])\).

Our custom search procedure branches on the \( x[s] \) decision variables. It selects a sensor with the largest mid value in the domain of the data utility \( u[s] \). The mid value is often a better choice when the domain range is large, which is the case at the beginning of the search. The search procedure breaks ties by selecting the closest sensor to the base station with hop-count as the metric. We then set the value of \( x[s] \) to 1 on the left branch and 0 on the right branch.

B. Heuristic-based Greedy Algorithm

Instead of using CP to solve the sensor selection problem optimally as described above, we also designed a heuristic-based algorithm built upon a simple greedy search strategy. The intuition behind 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 base station, 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. While simple, the heuristic algorithm may only find a local optimum to the sensor selection problem, which might be far from the global optimum.

Our heuristic algorithm returns a set \( \text{idSelected} \) of sensor nodes to be selected during the phase two of each round in ASBP protocol. The algorithm takes the constant set of sensor nodes \( S \), link quality \( L \), base station link quality \( B \), correlation \( C \), and initial energy \( E \) as an input, in addition to the parameters and representing the minimum threshold on the number of selected sensors and the link quality, respectively. Our heuristic-based algorithm is listed in Algorithm 1. In our algorithm, the identifier of a variable is written with italic font, and the identifier of a function is written with typewriter font. Here, the variables are imperative programming variables as opposed to the CP decision variables of Section IV-A.

The heuristic algorithm creates a set of selected sensors \( \text{idSelected} \) (line 1), and initialise it with all the possible sensor ids. The function \( \text{BestShortestPath} \) (line 2) takes the link quality matrix \( L \) and base station link quality array \( B \) as an input, and returns an array \( q \) of path quality values for the shortest path from each sensor node to the base station. The implementation of \( \text{BestShortestPath} \) is trivial, as it uses Dijkstra’s algorithm [24] to compute the shortest paths, while respecting the path quality constraint (5).

The heuristic algorithm maintains a set \( \text{idNonReachable} \) of sensor nodes that are not able to reach the base station due to the violation of the path quality constraints (5) (line 3). Before entering the main loop of the algorithm, any sensor nodes in the set \( \text{idNonReachable} \) are removed from the set of selected sensor nodes (line 4), and the values of link quality, base station link quality, and correlation for those sensor nodes in \( \text{idNonReachable} \) are set to 0 from the corresponding data using the function \( \text{SetZero} \) (lines 5–7). The function \( \text{SetZero} \) takes an \( n \times n \) 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 \((A(i; j) = 0 \text{ } A(i; j) = 0; 1 \leq j \leq n)\), and if \( A \) is a one dimensional array, then it only sets \( A[i] = 0 \). In other words, \( \text{SetZero} \) reflects the unreachability of the sensor nodes in \( \text{idNonReachable} \) into the network data structures (link quality and correlation).

The main loop of the algorithm (line 9) iteratively selects a sensor node that contributes the least value to the objective (4) (equivalent to a sensor node with the lowest data utility weighted by the initial energy), and performs a lookahead move (lines 10–20) to detect if removing this sensor node violates any of the constraints. The set \( \text{idFeasible} \) of feasible sensor nodes is initialised with the set of selected sensor nodes \( \text{idSelected} \) (line 8). The set \( \text{idFeasible} \) is used to keep the track of the sensor nodes that are potentially removable from the set of selected sensor nodes \( \text{idSelected} \). A lookahead move is performed, by first creating copies \( L^0 \) and \( B^0 \) from \( L \) and \( B \), respectively (lines 10–11). We then select a least contributing sensor \( \text{fidMing} \) that minimises the value of the objective function (lines 12). To perform the lookahead move, the link quality data for the sensor \( \text{fidMing} \) is set to zero (lines 13–14), and then the path quality \( q \) is updated (line 15) to discover the non-reachable sensor nodes \( \text{idNonReachable} \) (line 16).

If removing the non-reachable sensor nodes in the set \( \text{idNonReachable} \) from the set of selected sensor nodes \( \text{idSelected} \) (line 17) causes the violation of the active sensor constraint (line 18), then the sensor node \( \text{fidMing} \) is removed from the set \( \text{idFeasible} \) of feasible sensor nodes (line 19), and we skip to the next iteration (line 20). If the lookahead move does not violate the active sensor constraint, then we replace the set of selected sensors \( \text{idSelected} \) and the set of feasible sensor nodes \( \text{idFeasible} \) with the potential sensor set \( \text{idPotential} \), and we set the values of link quality and correlation for the non-reachable sensor nodes \( \text{idNonReachable} \) to zero using the function \( \text{SetZero} \) (lines 21–25). The algorithm ends if there are no more feasible sensor nodes (\( \text{idFeasible} = \phi \)) or the active sensor constraint is violated.

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 [8], [9]. In general, it
Algorithm 1: The heuristic-based greedy algorithm with dynamic routing

```plaintext
input : S; L; B; C; E; ;
output: idSelected
1 idSelected = S
2 q = BestShortestPath(L; B)
3 idNonReachable = {2 | s j q[s] < g}
4 idSelected = idSelected \ idNonReachable
5 SetZero(L; idNonReachable)
6 SetZero(B; idNonReachable)
7 SetZero(C; idNonReachable)
8 idFeasible = idSelected
9 while idFeasible 6=; ^jidSelectedj > do
10 L = L
11 B = B
12 idMin = min arg min (E[s] u[s])
13 SetZero(L; fidMing)
14 SetZero(B; fidMing)
15 q = BestShortestPath(L; B)
16 idNonReachable = idSelected \ idNonReachable
17 idPotential = idSelected \ idNonReachable
18 if jidPotentialj < g then
19 idFeasible = idFeasible \ fidMing
20 continue
21 idSelected = idPotential
22 idFeasible = idPotential
23 SetZero(L; idNonReachable)
24 SetZero(B; idNonReachable)
25 SetZero(C; idNonReachable)
26 return idSelected
```

assumes 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, $i = 1; : : : ; N$. For the application of IoT, $y_i$ 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 $i(x_i; y_i)$. Very often, we consider the observation $y_i$ to be fixed and write $i(x_i)$ as a short-hand of $i(x_i; y_i)$. Further, there are also statistical dependencies between the several underlying events $x_i$, encoded in a so-called potential function $j(x_i; x_j)$. In IoT, 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:

$$b(x_i g N_i = f) = i(x_i; x_j) i(x_i)$$

A graphical depiction of this model is shown in Figure 4. The rectangles are the observation nodes $y_i$ and the circles represent the underlying events $x_i$. The potential functions

![Fig. 4: An example of a graphical model.](image)

Fig. 5: 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 $j$ 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:

$$m_{ij}(x_j) = x_i i(x_i; x_j) m_i(x_i)$$

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 $i$. Intuitively, such a message models how likely it is at node $i$ that node $j$ will be in the state of $x_j$ when node $i$ is in the state $x_i$. Thus, BP performs message passing between nodes until reached convergence, and the inference is done by maximising the belief at each node, which is to gather all incoming messages and the local belief, i.e.:

$$b(x_i) = \prod_{j \in N(i)} m_{ij}(x_j)$$

The message passing process in BP is illustrated in Figure 5.

BP is well established in both theory and practice. For example, while it is known that BP is only guaranteed to converge on tree graphs, loopy belief propagation has been shown to work well in most cases for graphs with loops [25]. In addition, there are two general BP variations which are sum-product and max-product BP respectively [26]. The latter is adopted in this paper because of its efficiency.

B. BP for Inference on IoT

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

- **Spatial correlation**: Data from different sensors may be correlated with each other. Note that we do not assume that strong correlations always exist between data from
nearby sensors. Instead, we compute the correlation co-efficients 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.

Temporal correlation: Data from the same sensor may be correlated over time. Here, we simply 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 6 where \( x_i \) denote the true reading of sensor \( i \) at time \( t \). The link between \( x_i^t \) and \( x_j^t \) represents the temporal correlations, with a temporal potential function defined as:

\[
i(x_i^t; x_j^t) = \exp \left( \frac{(x_i^t - x_j^t)^2}{2} \right)
\]

Similarly, the link between \( x_i \) and \( x_i \) represents the spatial correlations, with a spatial potential function associated and defined as:

\[
s(x_i^t; x_i^t) = \exp \left( \frac{(x_i^t - x_i^t)^2}{2} \right)
\]

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 \( x_i \) and \( y_i \) is defined as:

\[
e(x_i^t; y_i) = \exp \left( \frac{(x_i^t - y_i)^2}{2} \right)
\]

\( y_i^t \) can be missing for two reasons: either sensor \( i \) is in the sleep mode or the packet failed to reach the base station. When it is missing, we turn the evidence function into a constant, i.e., \( e(x_i^t; y_i) = 1 \), for all possible values of \( x_i^t \). Such a constant evidence function essentially treats everything as equally possible. Intuitively, BP handles missing sensor readings by reasoning from the past data and the sensor nodes with correlated data. Note that \( i \) and \( j \) are parameters that can be learned from some training data [27].

In comparison with approaches such as the compressed sensing based approach in [11], BP based on the graph in Figure 6 is advantageous for several reasons:

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 [11] 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 [28]. Quantisation of network data saves storage as it encodes the data into fewer bits. It requires fewer number of transmissions, and smaller packet size. In many applications a quantised measure is informative enough to represent aspects of the network. For example, many heating, ventilation, and air conditioning (HVAC) sensors only react if temperature or humidity falls within certain thresholds. 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 \([\text{min}, \text{max}]\), and values outside this interval are mapped either to \( \text{min} \) or \( \text{max} \). The quantisation is done by partitioning the interval into \( R \) bins using \( R \) thresholds, denoted by \( r = 0, 1, \ldots, R \) \( r \) \( T \). 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:

\[
Q(x) = b_i, \quad \text{if} \quad i < x \leq i + 1, \quad i = 0, \ldots, R:
\]

where \( 0 = \text{min} \) and \( R = \text{max} \). The bin index values \( b_0, \ldots, b_R \) are stored in a codebook, and a metric value can then be represented by a bin index that is encoded into few bits. For example, Figure 7 shows six data values \( x_1, \ldots, x_6 \) quantised into four bins with 2-bit binary indices \( b_1 = (00) \leq 0, \ldots, b_4 = (11) \leq 3 \) according to (10).

The length of each partition \( i \) is either uniform with \( i = \frac{\text{max} - \text{min}}{R} \), or non-uniform. In general, the thresholds are chosen according to the requirements of the application, adaptively adjusted, or learned from a set of training data. For example, consider an indoor temperature monitoring, where the temperature varies at most between 0 and 50 degrees. Given 0.2 degrees temperature accuracy requirement of the application, the minimum number of quantisation level is \( \text{min} = 0.2 = 50 = 0.2 = 250 \), which implies that at least 8-bit quantisation resolution \( 2^8 = 256 \) bins is necessary in order to satisfy the requirement of the application.
The minimum and sensor inference matrix represented with Lab [23]. The ion correlated. Our experiments shows that 8 is at least we set ! for the Mica2Dot mote [29], under Mac OS X 10.9.2 64 bit. The data correlation. However, it does not affect the BP prediction data is also highlighted. Since the data is highly correlated, then the uniform quantised coefficient matrix using the jet colour map. We expect that values outside the interval are mapped to the closest bin is represented with an 8 bit value in the codebook. The error is upper bounded by the bin length, given by: 

\[(x) < \frac{x}{R} \]

The quantisation error is inversely proportional to R, whereby a smaller R leads to a larger \[Q(x)\]. When R is as large as \[\max \min\] the quantisation becomes equivalent to the rounding of the real value, which is almost lossless.

**VI. EXPERIMENTS**

**A. Experimental Setup**

We experiment with the real data collected from 54 sensor nodes deployed in the Intel Berkeley Research Lab [23]. 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 connectivity data, representing the link quality between any two sensor nodes, and between sensor nodes and the base station. In our simulations of the ASBP protocol, we selected a time interval of 10 hours, consisting of 5 rounds of two hours, such that at least 30% of data is transmitted successfully to the base station.

We apply a uniform quantisation on the temperature data between 10th and 90th percentile into 256 bins, where each bin is represented with an 8-bit value in the codebook. The values outside the interval are mapped to the minimum and maximum of the interval accordingly.

Figure 8 demonstrate the absolute values for the correlation coefficient matrix using the jet colour-map. We expect that since the data is highly correlated, then the uniform quantised data is also highly correlated. Our experiments shows that 8-bit quantisation resolution introduces at most 15% error in the data correlation. However, it does not affect the BP prediction results, as the data is already quantised when received at the base station.

In our energy consumption evaluations, we consider 14mA transmission cost, as reported for the Mica2Dot mote [29], used in the Intel Berkeley research lab deployment. It is worth noting that our estimation of the energy cost provides a lower bound on the total transmission energy spent by the network compared to a deployed network. Implementation and testing on a deployed network will be done in our future work.

In our simulations, each round is two hours, where phase one of a round ends if at least 20 data readings are collected at the base station from all the sensor nodes. The weights \(\|1\|\) and \(\|2\|\) in data utility (3) are chosen to normalise the path quality and correlation. We expect that at least sensor nodes are selected, hence the path quality is scaled by the number of sensor nodes \(\|7\|\) of sensor nodes \((\|1\| = \), because the sum of the correlation is at least we set \(\|2\| = 1\). The threshold of (5) is set to 0:7. The base station then solves the sensor selection optimisation problem and initiates the second phase of the ASBP protocol.

**B. Results and Analysis**

We evaluate the performance of our ASBP in terms of data utility, energy efficiency, and data prediction accuracy. We compare the data prediction error of the results of our constraint programming model, heuristic-based algorithm, and a random sensor selection. On the inference accuracy, we compare with the CS-based approach in [11] which we consider as the state-of-the-art. Our simulation of the ASBP protocol is implemented in C++, and the CP model is implemented using the constraint programming solver Gecode [30] (revised 4.2.1), and runs under Mac OS X 10.9.2 64 bit on an Intel Core i5 2.6GHz with 3MB L2 cache and 8GB RAM.

Figures 9a, and 9b 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 in (8) to control the total number of selected sensor nodes for data collection. The increase in the minimum base station link quality to 70% affects the routing of the data in the multi-hop data collection. It increases the size of the data collection 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.

In our results the CP sensor selection achieves the optimum data utility, and the greedy heuristic-based algorithm manages to find a satisfactory local optimum. 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. In all cases, the solution of the sensor selection problem for CP and the heuristic-based algorithm were found in less than one minute. For larger networks (more than 100 nodes), we advocate using our greedy heuristic-based method, as it is at least two orders of magnitude faster than the CP approach, and still achieves near optimum solution.

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 (3) compared 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 9b fails to find a solution for more than 30 selected sensor nodes, because our requirement for reaching the base station is limited to at least 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 (in terms of the number of transmission for data collection and node coordination) for the data transmissions with both settings 30% and 70% on the minimum base station link quality is shown in Figure 9c. 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 10a shows the BP results with the CP model, heuristic-based algorithm, and random sensor selection strategies, upon varying the minimum number of active nodes.

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

Fig. 9: 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 9a) and 70% (Figure 9b) were used for the base station link quality, upon varying the minimum number of selected sensor nodes.

(a) The prediction error of BP with CP, Heuristic and Random (b) The prediction error of BP versus CS using tic, and random node selection (c) The prediction error of BP versus CS using tic, and random node selection, our heuristic-based node selection algorithm. random node selection.

Fig. 10: The prediction mean square error of our BP-based approach and the CS-based approach in [11] using the CP model, the heuristic-based algorithm, and the random sensor selection strategies, upon varying the minimum number of active nodes.
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 10a. 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 $\theta = 10$ has an average error of about 5%, which indicates that in the temporal domain in average the prediction of the belief propagation deviates 5% from the ground truth. At the same data point, the standard deviation (SD) is about 12%, and increasing the number of selected sensor nodes 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 9c 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 5% 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).

On the inference accuracy, we compared our BP-based approach with the CS-based approach in [11]. In particular, [11] modelled the estimation of the lost data as a problem of matrix completion, where an EM matrix is constructed by recording the data reading of a particular sensor at a particular time. The EM matrix is incomplete because some data is lost during transmission and some sensors are inactive, i.e. not selected, during some time periods. By applying the matrix completion techniques developed in CS, the missing data in the EM matrix can also be estimated. While interesting, a drawback of the matrix completion formulation in [11] is that in order to construct the EM matrix, data must be collected in different sensors regularly and in a synchronised way so that the data in the time dimension is consistent. In contrast, our BP-based approach makes no such assumption and allows the sensors to collect data at irregular frequencies or even randomly. This is possible due to the explicit modelling of the data correlations in time and in space in the potential functions [9].

Figure 10b and 10c show the comparisons between our BP-based approach and the CS-based approach in [11] using the heuristic-based and random node selection respectively. It can be seen that on the heuristic-based node selection, BP is strictly better than CS. For example, BP achieves 16% lower prediction error compared to CS when $\theta = 10$. On the random node selection, the two perform similarly. Note that the results on random node selection is the average of 10 runs. Such results reveal the advantage of BP that the spatio-temporal correlations is explicitly encoded in the graph structure and in the potential functions which leads to the better accuracy in Figure 10b. On the other hand, in Figure 10c, BP builds the graph and learns the potential functions on randomly selected nodes without considering the correlations, whereas CS assumes the random sampling of the data which holds here.

Even in such scenarios, BP still achieves a similar performance as CS.

VII. CONCLUSION

By exploring cloud computing with the IoT, we present a cloud-based solution that takes into account the link quality and spatio-temporal correlation of data to minimise energy consumption by selecting sensors for sampling and relaying data. We have presented a novel cloud based adaptive sensing belief propagation (ASBP) protocol with energy-efficient data collection for the internet of things (IoT) applications. ASBP solves an optimisation problem to select an optimal set of active sensor nodes that maximises the data utility and achieves energy load balancing. In our protocol, belief propagation (BP) iteratively infers the values of the missing data from the stream of active sensor readings. We have also compared our BP prediction results with the widely used compressive sensing technique [11], and show that our BP algorithm significantly outperforms compressive sensing. 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 the performance of our ASBP protocol by extensive simulations using real data collected at the Intel Berkeley research lab sensor deployment and their link quality estimates. 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 5% 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 partially sponsored by the Vinnova for the project GreenIoT, the Swedish Foundation for Strategic Research (SSF) under research grant RIT08-0065 for the project ProFuN: A Programming Platform for Future Wireless Sensor Networks, the German academic exchange service (DAAD) U4 Network, and the Simulation Centre, sponsored by the Lower Saxony State and Volkswagen Foundation. Special thanks to the reviewers for pointing out ways to improve the presentation of this paper.

REFERENCES


