Optimizing Energy-Latency Trade-off with Mobile Elements in Wireless Sensor Networks

Lin SHEN, Liqin WANG and Xiangquan SHI

Abstract—Recent research shows that significant energy can be saved in wireless sensor networks by using mobile elements (MEs) capable of carrying data mechanically. Though the use of MEs can reduce the energy consumption at each sensor node, it increases the latency of data transfer from a source node to the base station. To address this issue, we propose a collection-based approach in which a subset of nodes serves as data collection points (CPs) that buffer data originated from sources and transfer data to MEs when they arrive. This will shorten the trajectory of MEs, but will also increase energy consumption of network nodes. There is a trade-off between energy consumption and data delivery latency. In this paper, we introduce a Probabilistic Path Selection (PPS) algorithm to reduce the data collection delay for stochastic event detection scenario. Furthermore, we develop a heuristic algorithm and apply it to situations where CPs are used to enable a flexible trade-off between energy consumption and data delivery latency.

Index Terms—Wireless sensor networks; mobile elements; probabilistic path selection algorithm, data collection points, data delivery latency.

1. INTRODUCTION

The past several years have shown great advances in both the capabilities and miniaturization of wireless sensors. The wireless sensor networks have been deployed in some mission-critical applications such as environmental monitoring [1][2], event detection [3] and security surveillance [4]. These advances herald the development of systems that can gather and harness information in ways previously unexplored. Equipped with sensing and wireless communication capabilities, the nodes can form a network to detect intruders or observe environment in their interesting region. In many data-centric applications, they often produce high-bandwidth sensor data that need to be collected under stringent delay constraints. There are multiple ways in which the sensor readings are transferred from the sensors to a central location. Usually, the readings taken by the sensor nodes are relayed to a base station for processing using the ad-hoc multi-hop network formed by the sensor nodes. While this is surely a feasible technique for data transfer, it creates a bottleneck in the network. The nodes near the base station relay the data from nodes that are farther away. This leads to a non-uniform depletion of network resources and the nodes near the base station are the first to run out of batteries. If these nodes die, then the network is for all practical purposes disconnected.

Several recent works have exploited the use of mobility assisted WSNs in data collection [5]. In this approach, a small number of mobile devices referred to as mobile elements (MEs) roam about sensing fields and collect data from sensors. As a result, significant network energy saving can be achieved by reducing or completely avoiding costly multi-hop wireless transmissions. On the other hand, the energy consumption of MEs is less constrained as they can replenish their energy supplies because of the mobility. However, the primary disadvantage of this approach is the increased latency. For instance, the typical speed of several practical ME systems (e.g., NIMs [6] and Packbot [7]) is about 0.1—1 m/s. As a result, it takes hours for an ME to tour a large sensing field, which cannot meet the delay requirements of many data-intensive applications.

In this paper, we attempt to optimize the energy-latency trade-off. In our approach, a few nodes in a large sensing field serve as data collection points (CPs) that buffer data sent (possibly through multiple hops) from source nodes. MEs periodically visit the CPs, pick up the cached data and carry them back to the base station within the required deadline. The use of CPs enables MEs to collect a large volume of data at a time without traveling a long distance, which achieves high data bandwidth and low communication delay at the same time. The way that CPS affect the data transfer delay and energy consumption. If an ME’s trajectory is short and connected by few CPs, many numbers of multiple forwarding are required which need greater energy due to increased data transfer at each node but the latency is expected to be much shorter. On the other hand, if the ME’s trajectory is long and it visits more CPs, the energy consumption at each node is reduced but the data delivery latency is relatively large. Fig. 1 illustrates a WSN that utilizes such a mechanism. By configuring the number and locations of CPs, the collection-based approach can meet various data intensive applications delay and energy consumption requirements. In this paper, we will present a probabilistic path selection (PPS) algorithm, which selects the traveling path of MEs according to the CPs visit probability. PPS can meet the energy-latency demands for stochastic event detection scenario. We will also introduce a heuristic algorithm for extend energy-latency problem which can select CPs with good ratio of network energy savings to ME travel distance.
In order to reduce the data transmission delay, path selection problem was elaborately studied in these papers. In [7], authors assumed that each sensor generates data at a certain rate and that the data mule needs to collect data before the buffer of each sensor overflows. Gandham et al. [13] used multiple mobile base stations to prolong the lifetime of the sensor networks. Xing et al. [14] presented path selection algorithms for rendezvous based approach. In these works, mobile nodes visit part of static nodes to collect data, but they did not consider the energy consumption and locations of the CPs.

There are also studies on combined data mule and multi-hop forwarding approach. Ho and Fall [15] discussed such approach in the context of Delay Tolerant Networking (DTN) architecture. Burns et al. [16] experimentally showed that controlled mobility can improve performance of routing in a network of randomly mobile nodes.

The most relevant work to our schema is the data mule scheduling (DMS) framework described in [17]. The DMS uses a data mule as an alternative or supplement to multi-hop forwarding in a sensor network, which presented a framework to capture and analyze communication strategies that use combinations of data mule and multi-hop forwarding. So the authors only consider the multi-hop forwarding problem from source node to base station. However, in our work we consider the collection based schemas, which analyze the power consumption among source nodes and CPs.

Also relevant to this paper is the work by Kansal et al. [18], who studied the case in which a data mule periodically travels across the sensor field along a fixed path. They used directed diffusion [19] for collecting data from the nodes outside of the direct communication range of the data mule. Their focus was on designing a robust communication infrastructure that works even in uncertain environments. In our work, we employ a fuzzy mobility control model with delay constraint and also realize a more optimized path selection where every mobile node only needs to visit subset of CPs which makes a shorten traveling path to reduce the data transmitting time.

2. RELATED WORKS

Mobile elements are often used as data collectors, also called Data Mules (DM) or Data Ferries (DF) [4][5][6][8][9][10][11]. Mobile elements traversing the network can collect data from sensor nodes when they come close to them. This naturally avoids multi-hop and removes the relaying overhead of nodes near the base station. In addition, the sensor nodes no longer need to form a connected network (in a wireless sense). Thus a network can be deployed keeping only the sensing aspects in mind. One need not worry about adding nodes, just to make sure that data transfer remains feasible. In addition to increasing the lifetime of the system (by avoiding multi-hop communication), mobility has been shown to increase capacity[8], [9], help in calibration [10], [11], and assist in security[12]. The authors of [4, 5] studied the problem of choosing the path of a data mule that traverses at a constant speed through a sensor field with sensors generating data at a given rate. Their formulation also requires the data mule to visit the exact location of each sensor to collect data. They designed heuristic algorithms (based on EDF scheduling) to find a path that minimizes the buffer overflow at each sensor node. In the Message Ferrying project, R.Pon and M.A.Batalin et al. [6] examined the problem of path and speed optimization of a data mule in a field of stationary nodes.

In order to reduce the data transmission delay, path selection problem was elaborately studied in these papers.
2. The maximum speed of all MEs is $V_m$, which is much smaller than the speed that data travels in the network. In this paper, we assume MEs move at a constant speed of $V_m$.

3. Each source generates a chunk of data synchronously, and the data chunks must be delivered to the BS within the delay constraint.

4. An ME communicates with a node when it is at the location of the node. The energy of MEs is replenishable (e.g., by recharging batteries at the BS).

The delay and energy consumption constraint is required by users. We use the tree structure to represent the topology of the network. Let $T(V, E)$ be the geometric tree with root $B$, in which $V$ is the set of source nodes and $E$ is the set of connection edges. The set of CPs is denoted by $C=\{p_0, \ldots, p_{n-1}\}$, where $p_i$ represents CP $i$. The sub-tree of $p_i$ is denoted by $T_i$, which means all nodes in $T_i$ transmit data to $p_i$.

The objective of optimizing the energy-latency tradeoff is to find a CP selection plan such that the MEs have the relative short total travel time while consume less network energy. Different from the “pure” data mule approach, in which each node sends its data only to the data mule, a node can now forward its data to other neighboring CPs as well. More importantly, if a node decides to forward all data to other CPs, the data mule does not need to collect data directly from this node. Then the data mule can possibly take a shorter path to reduce the travel time.

We present a centralized algorithm based on linear program formulation. Since finding the optimal CP plan that optimize the energy-latency tradeoff is NP-hard, which is proved in theorem 4, we make it an independent problem by changing the objective function.

We minimize the number of sub-nodes of each CP weighted by the time that an ME travels from this CP to the next CP. There are three reasons why this is a reasonable choice as the objective function. First, this function is likely to shorten the path of the ME by forcing the nodes at the edge of network to primarily forward data to CPs. Second, this function allows a smooth transition between the ME approach and multi-hop forwarding. As the data delivery limit grows, more data is forwarded closer to the base station. In a connected network, all the data is eventually forwarded to the base station without using a data mule, which is equivalent to “pure” multi-hop forwarding. Finally, since the function is linear, we can formulate the problem as a linear program as described below.

We assume the location of sensor nodes and the connectivity between them are known. We also assume the following parameters are given:

- $\lambda_i$: Data generation rate of node $i$
- $E_{\text{limit}}$: Energy consumption limit at each node per unit time
- $E_r, E_s$: Energy consumption for receiving and sending unit data

$R$: Bandwidth, i.e., maximum data rate that each node can communicate with other nodes and the data mule

LP formulation:

Suppose $n$ CPs are selected in a trajectory of DM. The distance of every pair of CPs is denoted by $[p_0, p_1], [p_1, p_2], \ldots, [p_{n-2}, p_{n-1}]$. Then we have the following linear program:

Variables Between each pair of $(p_i, p_{i+1})$.

- $w_i$: the number of nodes in the sub-tree of $p_i$
- $z_i$: time that the ME travels between CP$i$ and CP$i+1$
- $x_{ij}$: amount of data sent from node $i$ to $j$ per time unit

Objective Minimize $\sum_{i\in C} w_i z_i$

Constraints

- (Speed) The distance of each $[p_i, p_{i+1}]$ and speed should satisfy
  \[ \frac{p_{i+1} - p_i}{V_m} \leq z_i \] (1)

  For the constant speed model, for the each $[p_{i+1}, p_i]>0$, we have
  \[ z_i = \frac{z_k}{p_{i+1} - p_i} \frac{p_{k+1} - p_k}{p_{i+1} - p_i} \] (2)

  where $k$ is the any value satisfying $[p_{i+1}, p_i]>0$.

- (Connectivity) For $i \neq j$, $x_{ij} \geq 0$, if node $j$ is in the communication range of node $i$. Otherwise $x_{ij} = 0$

- (Flow conservation) $\lambda_i \geq 0$

- (Energy consumption) For each node $i$,
  \[ E_r \sum_j x_{ji} + E_s (\sum_j x_{ji} + \lambda_i) \leq E_{\text{limit}} \] (3)

  where the first term in the left hand side is the amount of energy consumed by receiving data and the second term is that for sending data. All data received from other nodes or generated by itself will be transmitted to MEs or other nodes.

- (Bandwidth) Per unit time, the amount of incoming data is $\sum_j x_{ji}$ and outgoing data is $\sum_j x_{ij} + \lambda_i$. After some manipulations, we obtain
  \[ 2 \sum_j x_{ji} + \lambda_i \leq R. \]

The formulation above is also capable of expressing the case in which each node communicates along the pre-constructed routing tree as in [17]. This is possible by
replacing the connectivity constraint with the following one:

- (Routing tree) For \( i \neq j, x_{ij} \geq 0 \), if node \( j \) is node \( i \)'s parent in the routing tree. Otherwise \( x_{ij} = 0 \).

4. PROBABILISTIC PATH SELECTION

In many WSN applications, especially in stochastic event detection scenarios, the amount and frequency of data generation in sensor nodes varies based on the event occurrence frequency, which is generally a function of the sensor location. Therefore, if an ME goes to a CP to collect data but the CP has no data, the time ME goes to the CP is wasted.

In this section, we will present a Probabilistic Path Selection (PPS) algorithm, in which an ME visit CPs based on certain probability. Our goal is to find a feasible travel policy which can dynamically improve the path of MEs so that they skip some CPs in certain time period. However, this will result in additional data delay for those CPs that have data but are skipped. In the PPS algorithm, a data retransmitting method is proposed in which those CPs transmit data to other CPs that will be visited by MEs in the near future.

A. Soft bound for data transmission delay

A longest data transmission delay (\( D_r \)) is specified by users. It is a random variable and characterized by a cumulative distribution function (cdf). The system should then ensure that the transmission delay at any point is less than \( D_r \). The interval time between the CP's consecutive visits of an ME denoted by \( T_{period} \).

To specify the requirement on data transmission latency, users can simply set the cdf of \( D_r \). The objective of the system then becomes to ensure that the transmission delay of any data is less than \( D_r \). However, we have to address a new critical issue, i.e., how to realize such a soft bound. To address this, we devise a simple yet effective metric, accessibility in one period (AoP).

Definition 1 (AoP): The AoP of one CP (denoted by \( \gamma_p \)) is the probability that MEs visit the CP at least once in a single period.

A CP that is accessed by MEs is also characterized by data transmission delay, denoted by \( D_p \). We derive the cdf of the \( D_p \) which reveals the relationship between \( D_p \) and \( \gamma_p \).

Theorem 1 The cdf of \( D_p \) is given by

\[
F_{D_p}(d) = 1 - (1 - \gamma_p)^k \left(1 - \frac{d - kT_{period}}{T_{period}} \gamma_p\right) \quad (4)
\]

where \( k = \left\lfloor \frac{d}{T_{period}} \right\rfloor \).

Proof. By definition, the cdf of \( D_p \) is

\[
F_{D_p}(d) = \Pr(D_p \leq d) = 1 - \Pr(D_p > d)
\]

This implies that there is no mobile sensor visit the CP in duration of \( d \) since the data is transmitted to it. There are \( k \) full interval and an additional length of \( d - kT_{period} \). The probability that the mobile sensor does not visit the CP is \( 1 - \gamma_p \), and that within the duration of \( d - kT_{period} \) is \( 1 - \gamma_p(d/T_{period} - k) \).

With this metric, it becomes possible to realize the soft bond on data transmission latency. We determine \( \gamma_0 \) that is the minimum AoP whose corresponding \( D_0 \) is less than \( D_r \), i.e.,

\[
D_0 \leq D_r \quad (5)
\]

A higher AoP at a point implies a shorter latency of data transmission at this CP. So we have

\[
D_p \leq D_0, \forall \gamma_p \geq \gamma_0 \quad (6)
\]

By combining (3) and (4), we can conclude that

\[
D_p \leq D_r, \forall p \in F \quad (7)
\]

Thus, by guaranteeing that the AoP of any point is larger than \( \gamma_0 \), we are able to ensure that the data transmission delay is less than the user’s requirement \( D_r \). Note that a more rigid requirement on real-time detection needs a higher \( \gamma_0 \). In the following, we derive the expected value of \( D_0 \).

Theorem 2 The expected value of \( D_0 \) is

\[
E(D_0) = (1 - 0.5 \gamma_0)T_{period}/\gamma_0 \quad (8)
\]

Proof. The expected delay is \( T_{period}/2 \) if the data is transmitted within the first interval. If it is detected in \( j \) period, \( j > 1 \), then the additional latency is introduced. Let \( N \) denote the number of full periods that the data delay undergoes before it is collected.

The probability density function (pdf) of \( N \) is given by
Pr(\(N = k\)) = (1 - \(\gamma_0\))^{k-1} \gamma_0, k \geq 0 \tag{9}

We derive the expect delay by condition on \(N\),

\[E(D_0) = \sum_{i=0}^{\infty} (T_{\text{period}} / 2 \times \text{Pr}(M = i)) = (1 - 0.5\gamma_0)T_{\text{period}} / \gamma_0 \tag{10}\]

For any individual node, the expected delay is a function of \(\gamma_0\), and is inversely proportional to \(\gamma_0\).

B. shortest path selection based on visit probability

The design goals of PPS are: 1) to ensure that the detection latency of any event is statistically bounded by the requirement posed by users; 2) to minimize the transmission time before the mobile sensors and the BS. As discussed previously, the first goal is achieved by ensuring that \(\text{AoP}\) of any point is larger than \(\gamma_0\). Now we focus on the second goal.

Following the probabilistic approach, an ME \(O\) visits a CP in one period with probability \(\beta_{\text{Q}}\), and ignores this point with the probability \(1 - \beta_{\text{Q}}\). The key issue is clearly the determination of the visit probability. To minimize the transmission time, the visit probability should be as small as possible to shorten the length of path. At the same time, however, it ought to be sufficiently large to guarantee the \(\text{AoP}\) of all points in the field.

For an ME to visit CPs, a path connecting all CPs in the field should be constructed. Here we assume the ME should go to the accurate position of CPs to gather data, as shown in Figure 2(a). This assumption does not affect the mobile node moving along the path for the CP must be within its sensing range. The objective is to find a path such that the shortest travel time of mobile nodes by that path is minimized. However, finding a “smooth” path as shown in the figure is computationally expensive. In addition, controlling the mobile node along such a smooth path is often difficult in practice. From these reasons, we have designed a simplified problem to analyze in the following.

As shown in Figure 2(b), we consider a complete graph with vertices at CPs’ locations and assume the ME moves between vertices along a straight line. Each edge is associated with a value which represents the visit probability to this vertex. In this way, while traveling alone an edge, the ME can visit the CP according to the probability and find the shortest path at the same time. We use Euclidean distance as the metric, since we have observed in the experiments that it has a strong positive correlation with the shortest travel time in the induced PPS problem.

\[\begin{align*}
\text{Minimize} & \sum_{v_i, v_j \in \mathcal{H}} (1 - p_{ij}) w_{ij} \\
\text{subject to} & \quad p_{ij} = p_{ji} \\
& \quad 1 - (1 - p_{ij})^m \geq \gamma_0
\end{align*} \tag{11}\]

In (11), \(v_i, v_j\) are two vertexes on \(\mathcal{H}\), \(p_{ij}\) is the visit probability from \(v_i\) to \(v_j\), \(P_{ij}\) means the probability from the base station to vertex \(v_j\), \(m\) is the traveling periods and \(w_{ij}\) is the metric. The objective function (11) requires the visit probability assigned to \(v_j\) is identical. Function (13) ensures the visit probability of CP is larger than the minimum value.

This probabilistic shortest path selection problem by be simply solved by using Dijkstra algorithm.

We focus on the general case that an ME visits the CP with identical probability. To guarantee that the \(\text{AoP}\) of
any point is greater than $\gamma_0$, this schema simply sets the visit probability in each cycle of every sensor to $\gamma_0$. The problem is that when the number of mobile sensor visiting the CP is great, resulting in unnecessary waste of time.

We analyze the data transmission delay achieved by identical probability schema.

Theorem 3 With identical probability schema, the expected detection delay of an event that happens at any point is given by

$$E(D) = \frac{3(1 - \beta)^m - (1 - \beta)^{2m}}{2(1 - (1 - \beta)^m)} T_{period}$$

Proof. Let $N$ denote the number of full periods that elapsed before the data is collected. The pdf of $N$ is

$$Pr(N = i) = (1 - \theta)^{i-1} \theta$$

where $\theta$ is the probability that the CP is accessed within one period. It is obvious that

$$\theta = 1 - (1 - \beta)^m$$

Substituting the value of $\theta$ into (15), we get

$$Pr(N = i) = (1 - (1 - \beta)^m)^{i-1} (1 - (1 - \beta)^m)$$

If an event is detected in the $i$th period, an additional latency of $(i-1)T_{period}$ is introduced. Thus, we compute the expected delay by conditioning on $N$,

$$E(D) = \sum_{i=1}^{\infty} \left( \frac{T_{period}}{2} + (i-1)T_{period} \right) \times Pr(N = i)$$

$$= \frac{3(1 - \beta)^m - (1 - \beta)^{2m}}{2(1 - (1 - \beta)^m)} T_{period}$$

C. Retransmission method to reduce delay

We have mentioned above that in PPS protocol additional data delay will occur if MEs do not visit CPs that have data to transmit. In this section, we introduce the data retransmission method in which those CPs transmit data to other CPs that will be visited by MEs in the near future.

Each CP stores the time that an ME visits it last time. For MEs with the constant speed, the time of traveling in each period is fixed. The maximum waiting time of the CP is $L/v_m$, $L$ is the length of the traveling path of an ME in one period. In this interval, the CP must retransmit data to other CPs while the ME does not visit it.

In the retransmission method, the energy consumption $E$ of the network is

$$E = \sum_i (h_i \times (E_S + E_r) \sum_{j \in T_i} (\lambda_j \times L/v_m))$$

where $i$ denotes the CP that the ME does not visit in the period and $h_i$ denotes the hops between the source CP and the destination CP. $\sum_{j \in T_i} (\lambda_j \times L/v_m)$ is the data amount generated by all nodes in the CP’s sub-tree in one period.

5. COLLECTION POINTS SELECTION

We formally formulate the collection points selection problem as the Min-energy Min-latency (MEML) problem with the following definition:

Definition 2 (MEML): Given a geometric tree $T(V, E)$ rooted at $B$ and a set of source nodes $S = \{s\} \subseteq V$, find a tour $U$ that originates from $B$ and includes collection points (CPs), such that the total length of $U$ is minimized at least not greater than $L$ and

$$U = \arg \min \sum_{s_i \in S} N(CP, s_i)$$

where $N(CP, s_i)$ is the number of source nodes of sub-tree $CP$, means the energy consumption minimized.

We turn attention to the CPs selection on the trajectory of an ME which must satisfy the MEML problem. At first, we present the following theorem regarding the complexity of the MEML problem.

Theorem 4 The MEML problem is NP-hard.

Proof. We show that the decision version of MEML is NP-hard by a reduction from the Traveling Salesman Problem (TSP). A special-case decision version of the MEML problem is to ask if there exists a set of CPs such that the network energy consumption is zero. In order to incur zero network energy consumption, all the sources must be CPs as well. In other words, the ME must visit all the CPs on a tour with the least time. This is exactly the decision version of the TSP problem in which a salesman needs to visit a set of sites on a tour within preset time limit.

Here we present a greedy algorithm referred to as CPUG (utility-based greedy heuristic) in this section. CPUG should select those CPs that save the maximum network energy and the ME visits them with less time. On the other word, CPUG should balance the energy-latency performance under the energy and delay constraints.

![Fig.3. An example of CPUG’s execution](image-url)
energy saved by including it on the ME tour to the length increase of the tour. An important observation is that the utility of a RP varies with the length of ME tour, which is illustrated in Figure 3. If the ME tour is only long enough to cover B and another node, either \( CP_1, CP_2 \) or \( CP_3 \), \( CP_1 \) should be chosen because it saves most energy (there are 5 source nodes in the its sub-tree). However, if the ME tour can cover B, \( CP_2 \) or \( CP_3 \), the utility of \( CP_1 \) becomes zero because all data are picked up by the ME at either \( CP_2 \) or \( CP_3 \) before they reach \( CP_1 \). The iterative structure allows CPUG to dynamically update the utility of nodes when the ME tour is expanded and hence better CP choices can be made.

Fig. 4 shows the pseudo code of CPUG. Initially, the CP list only includes the base station. CPUG then adds source node to the trajectory of ME to decide whether it is CP. CPUG uses procedure TSP(I) to compute the minimum length of a tour that visits all points in set \( L \). TSP(I) can be implemented by a Geometric Traveling Salesman Problem solver. At the beginning of an iteration, CPUG finds all CP candidates (step 2). A node is a CP candidate if it can be visited together with all existing CPs by a tour no longer than \( L \) and the source nodes of all sub-trees become less. The utility of candidate \( x \) is defined as:

\[
u(x) = \frac{\sum T_i \in Q \{N_i(T_i) - N(x)\}}{TSP(Q \cup \{x\}) - TSP(Q)}
\]

where \( u(x) \) is equal to the ratio of the reduction of total distance that data chunks travel along the tree to the tour length increase. CPUG then adds the CP candidate with the greatest utility to the CP list (step 4). Then all the CPs whose utilities become zero are removed from the CP list (step 5), which is necessary since the addition of a new CP may invalidate some existing ones as discussed earlier. Finally, if all source nodes are included in the CP list, CPUG terminates because the ME can visit all of them within the deadline and the total network energy is zero. Otherwise, a new iteration is started to find more CPs.

Input: routing tree \( T(V,E) \), source set \( S = \{s_i\}, L \)

Output: CP list \( Q \)

1. \( Q = \{B\} \).
2. \( W = \{v \in V : (TSP(Q \cup \{v\}) \leq L) \land (N(Q) \cup \{v\} \leq N(Q)) \} \).
   if \( W = \emptyset \) exit.
3. Find \( x \in W \) with maximum \( \mu(x) \) defined by (20). If multiple nodes have the same \( \mu(\bullet) \) value, choose the one with maximum \( TSP(Q \cup \{v\}) - TSP(Q) \).
4. \( Q = Q \cup \{x\} \).
5. if \( \mu(z) = 0 \), \( Q = Q \setminus \{z\} \).
6. if \( S \subseteq Q \), exit, else goto (2).

Fig. 4. UPRG – a utility based greedy CPs selection algorithm

We now discuss an example of CPUG’s execution. Fig. 3 shows the CP lists and the ME tours at the end of three iterations. \( CP_1 \) is included on the tour in the first iteration. In iteration 2, \( CP_2 \) has the greatest utility among all nodes. Although \( CP_2 \) and \( CP_3 \) cause the same length increase of the tour, \( CP_3 \) has a smaller utility as it only has one source node in the sub-tree and \( CP_2 \) has two source nodes in the sub-tree. \( CP_2 \) is added on the tour in iteration 3. \( CP_1 \) is removed from the tour (at step 5 of CPUG) because including \( CP_2 \) and \( CP_3 \) on the tour renders its utility to be zero. As a result, the tour length is shortened and hence more CPs can be included in the following iterations.

6. PERFORMANCE EVALUATION

This section presents the evaluation of CPUG protocol and PPS algorithm. The simulations are written in C++. The radio parameters are set according to the data sheet of the CC1000 radio on Mica2 motes [20]. Radio bandwidth is 40 Kbps and transmission power is 4 dbm with the current consumption of 11.8 mA. The size of each packet is 30 bytes.

Suppose the data sent by the nodes in the network can all be treated as Poisson process. Nodes are randomly distributed in a \( 300 \times 300 \) m region. The BS is located at the top left corner of the region. The number of nodes is 400 unless otherwise indicated. 100 source nodes are randomly chosen. The results are the average of 5 different topologies. During the initialization, a shortest-path routing tree is created to connect all the nodes to the BS. The cost metric of a link is the expected number of transmissions. A source generates and stores a data sample of 2 bytes every second. It sends all the accumulated data (20 K bits) to the BS every 20 minutes (i.e., the deadline is 20 minutes). Each simulation lasts for 100 periods.

We first evaluate the network energy consumption when the speed of an ME varies from 0.2 to 2 m/s. Only radio transmission energy is counted in the simulations. As the baseline, we also plot the total network energy consumption without using the ME, denoted by NET. We also compare our protocol with other existing ME based data collection schema such as RP-CP [14]. Fig. 5 shows that all algorithms yield lower energy consumption when the ME moves faster. This is because the ME is able to collect data on a longer tour within the deadline. When the ME moves along the routing tree, CPUG saves about 30 to 60% more network energy compared to NET. The results of this simulation show that our algorithms can effectively take advantage of
speed increase of the ME, which is particularly important when they are implemented on different ME platforms. In the following simulations, the ME speed is set to be 2 m/s.

Now we focus on the performance of PPS with the fix CPs we selected use CPUG algorithm. We assume there are 20 CPs evenly distributed in the square sensor field.

In the general case, we should find the shortest closed curve that connects all CPs, i.e., the traveling path that the sensor along which visit CPs once is minimized. We get the length of path is 98 meter through solving the TSP problem. So a mobile sensor visits all nodes in one period need 49s.

We now evaluate the algorithms under different network densities. Fig. 9 shows that two algorithms perform better with a higher density since the quality of links among nodes becomes better when each node has more neighbors. However, the performance improvement of RP-CP is not as significant as other algorithms because it uses a fixed ME path independent of node density. In contrast, the other algorithm find the ME path on a geometric tree that better resembles the actual routing tree when the network is denser.

In PPS schema, the ME visits the CPs with different probability according to the events which require various AoP for CPs. ME will select different path based on the visit probability. Figure 7 is the relationship between visit probability of CPs and the average data transmission delay. When the visit probability is low, some of the CPs will not be accessed in this period, the length of ME’s trajectory is short, so the average data transmission delay is corresponding low. From Figure 8 we can see although the transmission delay is reduced with low visit probability, the network energy consumption is high. This is because more data forwarding will occurred for the ME not visiting some CPs.

Figure 9 shows the path selection process when we ME with the visit probability of 80% for all CPs in two period.
7. CONCLUSIONS

We presented a collection-based approach that utilizes mobile elements to collect sensor data in order to optimize the tradeoff between data transmission delay and network energy consumption. We formulated the problem with linear program algorithms. Then we introduced a probabilistic path selection algorithm to minimize the length of mobile elements trajectory and save the network energy at the same time on the certain set of collection points. We designed a collection-based data collection protocol that facilitates reliable data transfer from network nodes to mobile elements. Simulations show that our approach significantly reduces network energy consumption and scale well with network density and mobile element moving speed.

Our future work is to extend the problem formulation and the algorithms to environments with increased uncertainty. For example, we are currently working on relaxing the assumption on communication region. One idea is to employ a “semi-online” algorithm that initially plans the motion offline solely based on the knowledge about small regions around each node.

REFERENCES


Lin Shen is a professor of College of Electrical and Information Engineering at Jiangsu Teachers University of Technology, Changzhou, China. He is currently a Ph.D candidate in telecommunication and information systems at Nanjing University of Science and Technology. His research interests include signal processing, embedded systems, wireless communication and target identification.

Xiangquan Shi is a professor of College of Electro-Optics at Nanjing University of Science and Technology. His research interests include radar systems, target acquisition and identification, random signal theory and technology, and fast digit signal processing. He has published more than 30 papers in journals and conferences. Prof. Shi is a senior member of the Chinese Society of Electronics.