Energy-Efficient Data Preservation in Intermittently Connected Sensor Networks

Masaaki Takahashi, Bin Tang, and Neeraj Jaggi
Department of Electrical Engineering and Computer Science
Wichita State University, Wichita, KS 67260
mxtakahashi@wichita.edu, bintang@cs.wichita.edu, neeraj.jaggi@wichita.edu

Abstract— **We study data preservation in intermittently connected sensor networks, wherein the sensor nodes do not always have connected paths to the base station. In such networks, the generated data is first stored inside the network before being uploaded to the base station when uploading opportunity arises. How to preserve the data inside the network is therefore an important problem. The problem becomes more challenging when sensor nodes have finite and unreplenishable battery energy. In this paper, we identify, formulate and study the data preservation problem in the intermittently connected sensor networks under energy constraints at sensor nodes. The problem aims to preserve the data inside the network for maximum possible time, by distributing the data items from low energy nodes to high energy nodes. We first show that this problem is NP-hard. We then design a centralized greedy heuristic and a distributed data distribution algorithm, and compare their performances using simulations.**

Keywords –Data Preservation, Algorithms, Intermittently connected Sensor Networks

I. Background and Motivation

Data gathering is one of the important functionalities of sensor networks. In many data gathering applications such as object tracking [11] and intrusion detection [19], data is time-sensitive and needs to be transmitted back to the base station in near-real-time fashion. However, there are many applications that do not need real-time data transmission and access, such as acoustic sensor networks [14], underwater or ocean sensor networks [10, 22], and environmental monitoring [13, 15]. They are mainly used in scientific applications by domain scientists to collect scientific data for further analysis. For example, environmental scientists deploy a sensor net to study light variations on the forest floor due to canopy closure, and only need to collect the data months later when the experiment is over [15]. Another example is EnviroMic [14], a large-scale and long-term audio sensor network deployment to collect data for bird vocalization monitoring and recording.

In such applications, there is no longer a need to maintain base stations in the sensor field to collect real-time data. Data generated inside the network is first stored in the network and then uploaded to the faraway base station via different means. These uploading opportunities could be periodic visit by human operator or data mule [6, 7], or transmission to the base station through wireless communication such as a low rate satellite link [16]. We refer to such sensor networks *intermittently connected sensor networks*[1]. The main function of the intermittently connected sensor networks in these applications is to collect and store the data in the network before the next uploading opportunity arises.

There are three main factors contributing to the data loss in such sensor networks: energy depletion of sensor nodes, storage depletion of sensor nodes, and sensor node hardware failure. Overcoming the obstacle of data loss and preserving data in-network until upload opportunities arise is a new challenge. We have addressed the storage depletion induced data loss in our previous research [20]. In this paper, we focus on the energy depletion induced data loss. Our goal is to preserve the complete set of data inside the sensor network for the maximum amount of time, considering that each sensor node has limited battery energy.

In our model, sensory data is generated and initially stored at some sensor nodes (referred to as *source nodes*). The generated data items should then be distributed to high energy sensor nodes (referred to as *destination nodes*) for the purpose of data preservation. Ideally all the data items should be distributed from their source nodes to destination nodes that have the highest energy level in the network (when the source nodes are among the highest energy level nodes, their data need not be distributed). However, there are several challenges to achieve that. First, due to the non-uniform energy consumption during data distribution, the highest energy level nodes before distribution are not necessarily the highest energy level nodes after distribution. Second, for each data item, deciding where it is distributed to and the path along which the data is distributed is a complex decision. When each node has limited energy, the research challenge is not to minimize the total energy consumption during the data distribution process, but to load balance individual energy consumption of different sensor nodes. Third, such data distribution process, if not managed well, could be a serious energy drain to all the nodes involved in the process, further expediting the energy-depletion induced data loss. Therefore, it is important to find energy-efficient data distribution algorithm so that data can be preserved for maximum amount of time.

[1]Note that intermittently connected sensor networks are different from delay tolerant network (DTN) [3]. In DTN, mobile nodes are intermittently connected with each other due to their mobility and low density, and data is opportunistically forwarded by relay nodes to destination nodes. In the intermittently connected sensor network, since all the sensors are disconnected from base stations for substantially large period of time, data is uploaded to the base station only when uploading opportunities arise.

To address above challenges, we formulate the data preservation problem in intermittently connected sensor networks, with the objective to preserve the data inside the network for maximum amount of time. We study it as a graph-theoretic problem and show that it is NP-hard. We design a centralized greedy algorithm to distribute and preserve the data for maximum amount of time. The centralized nature of algorithm is unsuitable for large scale distributed sensor network. Therefore, we also design a distributed algorithm and compare its performance with the centralized algorithm.

II. **Related Work**

Most of the sensor network research assumes a multiple-to-one communication pattern, i.e., data sensed at different sensor nodes is directly communicated back to one or more base stations. The data preservation in a intermittently connected sensor network studied in this paper is a dramatic shift from the current sensor networking paradigm, and has not attracted much attention from the sensor network research community. Consequently, the important problem of maximizing data preservation time in sensor network has not been studied before. The most related work to ours is by Tilak et al. [21]. They propose to store the data in the network and propose collaborative storage techniques to efficiently manage data in storage constrained sensor networks. However, they do not address storing data under energy constraints. In this paper, we consider the energy constraint at individual node and focus on preserving data using data distribution technique, with the goal of maximizing data preservation time. The data preservation time maximization is related to network lifetime maximization. Below we provide a brief survey on the current research on data distribution and network lifetime maximization in sensor networks.

Luo et al. [15] were the first to study data distribution for disconnected operations in sensor networks. They present a cooperative storage system for sensor networks called EnviroStore, to improve the utilization of the network's data storage capacity. Tang et al. [20] further formalize this problem and show that it is equivalent to the minimum cost flow problem, which can be solved optimally. They also design an energy-efficient distributed data redistribution algorithm. Both work focus on storage depletion (storage overflow) induced data redistribution and are not concerned with maximizing data preservation time due to energy depletion of sensor nodes, which is the topic of this paper.

Maximizing network lifetime has been a very active research area in sensor networks. Different research work define network lifetime differently. Chang and Tassiulas [2] propose a shortest cost path routing algorithm for maximizing network lifetime based on link costs that reflect both the communication energy consumption rates and the residual energy levels at the two end nodes. They define lifetime as the time when network partition takes place. Park and Sahni [18] study a sequence of routing requests, each of which is for a source and destination pair. They define lifetime as when the first routing request can not be successfully routed. Xue et al. [24] and Kalpakis et al. [9] study the maximum lifetime data gathering problem considering data aggregation. Recently, Xiong et al. [23] propose polynomial-time and near optimal integer program-based algorithms; Zhang and Shen [25] maximize network lifetime through balancing energy consumption for uniformly deployed data-gathering sensor networks. In [9, 23–25], the network lifetime is defined as the time when the first node depletes its energy.

Our work differs from previous work in the following aspects. Unlike previous research which almost always assumes that data is transmitted immediately to the base station, our work involves moving the data inside the network in order to preserve it for as long as possible, and hence does not involve transmission scheduling. Consequently, the data preservation time in our work is defined as the time when the preservation of any data item can no longer be satisfied. Our definition is more general since both energy depletion of the first sensor and network partition do not necessarily cause the data loss and the violation of data preservation.

III. **Problem Formulation**

Network Model. The sensor network is represented as a general graph $G(V, E)$, where $V = \{1, 2, ..., N\}$ is the set of N uniformly distributed nodes, and E is the set of edges. Two nodes are connected by an edge if they are within the transmission range of each other, thus can communicate directly. Let d_{ij} denote the shortest path distance (in terms of number of hops) between nodes i and j.

There are a set of p data items $D = \{D_1, D_2, ..., D_p\}$, each of which has unit size[2] and is initially generated and stored at some sensor node, called its *source node*. We assume that each data item is stored in a distinct source node, therefore there are p source nodes in the network. We assume that the storage capacity of each node is one unit i.e. each sensor (including source nodes) can only store one data item. Let V_s denote the set of source nodes. Without loss of generality, let $V_s = \{1, 2, ..., p\}$, and let data item D_i be stored at node i.

Energy Model. Each sensor node (including the source node) i has a finite and unreplenishable initial energy E_i. Energy consumption in data preservation can be expressed as the number of messages transmitted during the data distribution process. Since number of messages transmitted from any sender to any receiver equals the number of hops between them, we use the number of hops to measure the energy consumption of transmitting the data item.[3] More specifically, for each node, sending or receiving a data item costs 0.5 units of its energy. Therefore, if a node is the sender or the receiver of a data item, it incurs 0.5 units of energy; if a node is an intermediate node relaying the data item, it incurs 1 unit of its energy (by both sending and receiving it). This assumption is consistent with the model that energy consumption of sending

[2] Although, in this paper, we assume all the data items have the same unit size, our work can easily be extended to the case when data items have different sizes.

[3] We are aware that the first order radio model [5] is a more realistic energy model, wherein the energy consumption depends on the distance between nodes. For uniformly deployed sensor nodes, the number of hops is a good approximation of the energy consumption. Liu et al. [12] and Nuggehalli et al. [17] also assume that transmitting one packet (or data item) of unit size over one hop consumes one unit of energy.

a data item equals the number of hops between the sender and the receiver. Therefore total energy consumption in the entire network is the total number of hops all the data items traverse in the data distribution process. We also assume that there exists a contention-free MAC protocol (e.g. [1]) that provides channel access to the nodes.

Data Preservation Time. We assume that the *energy draining rate* of each sensor is a constant c, meaning that each node depletes c units of energy per round, in addition to the energy consumption if it participates in the data preservation. The constant energy depletion rate has been demonstrated by Jurdak et al. [8]. If a node's energy is depleted, its stored data gets lost. The *data preservation time* of the sensor network is defined as the time when the first data loss occurs due to the energy depletion of the sensor node that stores it. Obviously, if there is no data distribution taking place, the data preservation time of the network equals the minimum energy among all the source nodes divided by c. To prolong the data preservation time, it is necessary to move data from nodes with low energy level to nodes with high energy level. Before formulating the data preservation problem, we first present our assumptions and notations below.

Assumptions and Notations. The data distribution process takes place in rounds, starting from round 1. In each round,[4] the p data items are distributed from one node to another for the purpose of data preservation. We assume that the data distribution in any round starts at the beginning of that round and finishes before the end of that round.

Table I lists all the notations. S_j^t denotes the node storing D_j at the beginning of round t. Since S_j^{t+1} is also the node storing D_j at the end of round t (assuming energy of S_j^{t+1} is not depleted), we call S_j^t *source node* and S_j^{t+1} *destination node* of D_j in round t, meaning that D_j is distributed from S_j^t to S_j^{t+1} in round t. It is possible though, that $S_j^t = S_j^{t+1}$, meaning that D_j is not distributed in round t. $V_s^t = \{S_1^t, S_2^t, ..., S_p^t\}$ is the set of source nodes at the beginning of round t, we have $V_s^1 = V_s = \{1, 2, ..., p\}$. Therefore, V_s^{t+1} is also the set of the nodes storing data items at the end of round t. The *distribution path* of D_j in round t, denoted as $P_j^t : S_j^t, ..., S_j^{t+1}$, is the sequence of distinct nodes along which D_j is distributed from S_j^t to S_j^{t+1} in round t. The *data preservation strategy* up until round t is the combination of all the distribution paths for all the data items, in all the rounds up until round t.

In each round, the energy consumption of a node consists of two parts: one due to energy draining rate and the other due to the node's involvement in the data distribution. The remaining energy level of node i at the end of round t is:

$$E_i^t = E_i^{t-1} - c - \sum_{j=1}^{p} x_{ij}^t,$$

where x_{ij}^t is the energy cost of node i in distributing D_j in round t. $x_{ij}^t = 0.5$ if in round t, node i is either source or destination node of D_j, but not both, that is, $i = S_j^t \neq S_j^{t+1}$ or $i = S_j^{t+1} \neq S_j^t$; $x_{ij} = 1$ if node i is an intermediate node relaying D_j in round t, that is, $i \in P_j^t$ and $i \notin \{S_j^t, S_j^{t+1}\}$; and $x_{ij} = 0$ if i is not involved in the data distribution of D_j in round t, that is, $i \notin P_j^t$.

[4]We assume that the duration of each round is long enough such that data items can be distributed via multi-hop paths from one node to another.

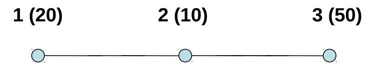

Fig. 1. Illustration of Observation 1 using a linear sensor network with three nodes. The number in the parenthesis indicates the initial energy level of each node. Node 1 is the only source node, with one data item. If node 1 waits before its energy depletes to zero to distribute its data item, node 2 already depletes its energy and causes network partition.

Data preservation time, denoted as T, is the round by the end of which the first data loss occurs, that is, $|V_s^T| = p$ and $|V_s^{T+1}| < p$. The goal of the problem is to find an optimal data preservation strategy such that T is maximized.

TABLE I
NOTATION SUMMARY

Notation	Explanation
S_j^t	The node storing D_j at the beginning of round t
P_j^t	The distribution path of D_j in round t
x_{ij}^t	The energy cost of node i in distributing D_j in round t
E_i^t	The remaining energy level of node i at the end of round t
V_s^t	The set of nodes storing data items at beginning of round t
T	The data preservation time: $\lvert V_s^T\rvert = p$ and $\lvert V_s^{T+1}\rvert < p$

Observation 1: To maximize T, the source nodes V_s should not wait to distribute their data right before they deplete their energy. That is, node $i \in V_s$ should not wait until round $\lceil \frac{E_i}{c} \rceil$ to distribute its data D_i.

This can be illustrated by a simple example of three node sensor network, shown in Fig.1. Each edge is of unit hop. Node 1 is the only source node, with one data item. The number in the parenthesis indicates the initial energy level of each node. The energy draining rate of each sensor is given by $c = 1$. Before node 1 waits until its energy depletes to zero, node 2 already depletes its energy and causes network partition, thus preventing node 1 from distributing data to node 3. Therefore, the data distribution decision of each source node should be made earlier than when its energy is depleted. In this example, if the energy level of node 2 equals 1, the data should be distributed immediately in round 1.

Lemma 1: There exists an optimal distribution strategy which finishes distributing all data items in round 1.

Proof: Let T^{OPT} denote the maximum T achieved by an optimal data preservation strategy. In this optimal strategy, for each data item D_j, denote its destination node in round t as $(S_j^{t+1})^{OPT}$ and the distribution path in round t as $(P_j^t)^{OPT}$, where $1 \leq t \leq T^{OPT}$. Now we design a new data preservation strategy: in round 1, set the destination node of data item D_j as $(S_j^{T^{OPT}})^{OPT}$, and D_j's distribution path as $(P_j^1)^{OPT}, (P_j^2)^{OPT}, ..., (P_j^{T^{OPT}})^{OPT}$; for rounds greater than 1, D_j stays at $(S_j^{T^{OPT}})^{OPT}$ and is not distributed.[5]

[5]Note that if a relay node v is visited multiple times in D_j's distribution process, then all the distribution paths between the first and last visit of v are redundant and can be omitted.

We claim that the above data preservation strategy achieves T^{OPT}. This is because if there exists an optimal solution wherein some data item is distributed in multiple rounds and finally reaches its final destination node, it can be distributed in round 1 from its source node to the final destination node, following the same sequence of distribution paths. Such a strategy retains the energy level at each node in the network at the end of round T^{OPT}. Therefore, this data preservation strategy, which only takes place in round 1, is also an optimal data preservation strategy and achieves T^{OPT}. ∎

A *destination node post distribution* is defined as the node which finally stores the data item, and the data item is not distributed again out of this node. Let E_{min} be the minimum energy level among all destination nodes, post distribution.

Theorem 1: Maximizing the data preservation time in the network is equivalent to maximizing E_{min}.

Proof: Observation 1 and Lemma 1 essentially indicate that the data preservation problem is a static problem wherein the optimal data preservation strategy is to distribute all the data to final destination nodes in the first round itself. By way of contradiction, let's assume that for an optimal data preservation strategy OPT, the minimum energy among all the destination nodes post distribution is not the maximum among all the data preservation strategies. That is, it is less than the minimum energy of destination nodes post distribution in another data preservation strategy $OPT^{'}$. In this case, the minimum energy destination node in $OPT^{'}$ depletes its energy later than that of the minimum energy destination node in OPT, yielding a longer data preservation time than that of OPT. This contradicts with the assumption that OPT yields maximum data preservation time. ∎

From Theorem 1, the following algorithm constitutes the optimal data preservation strategy: *Find each data item a destination node and a path to distribute the data from its source node to its destination node under the energy constraint at each node, such that the minimum energy among all the destination nodes is maximized, post distribution*. Here if a data item's destination node is its source node, it is not distributed. Theorem 1 essentially says that the optimal data preservation strategy is independent of the energy draining rate c, that is, the optimal data preservation strategy remains unchanged for different c. However, the maximum data preservation time does depend on c, and equals the minimum energy among all the destination nodes post distribution divided by c. Below, we formulate a static problem without using draining rate c and round t, which is equivalent to the data preservation problem. We call it *static data preservation problem.*

Problem Formulation. A *distribution function* is defined as $r : D \rightarrow V$, indicating that data item $D_i \in D$ is distributed from node i (note that node i is the source node of D_i) to node $r(i) \in V$. Let $P_i : i, ..., r(i)$ be the *distribution path* of D_i, denoting the sequence of distinct sensor nodes along which D_i is distributed from i to $r(i)$ ($i = r(i)$ indicates that node i is also D_i's destination node, which means D_i is kept in its source node i and need not be distributed). Let V_d be the set of the destination nodes of the data distribution, i.e. $V_d = \{r(1), r(2), ..., r(p)\}$. Let $E_i^{'}$ denote node i's energy level after the distribution of all data items is done, and let x_{ij} be the energy cost incurred by node i in distributing the data item D_j from node j to $r(j)$. Then,

$$E_i^{'} = E_i - \sum_{j=1}^{p} x_{ij}, \forall i \in V \quad (1)$$

where $x_{ij} = 0.5$ if either $i = j \neq r(j)$ or $i = r(j) \neq j$, $x_{ij} = 1$ if $i \in P_j$ and $i \notin \{j, r(j)\}$, and $x_{ij} = 0$ otherwise. Here, i could be either the source node of D_j or the destination node of D_j, but not both (with cost 0.5), or an intermediate relaying node (with cost 1), or not involved with the distribution (with zero cost).

The objective of the static data preservation problem is to find a distribution function r and a set of paths $\mathcal{P} = \{P_1, P_2, ..., P_p\}$, to distribute each of the p data items, such that the minimum energy among all the destination nodes V_d is maximized post distribution, i.e.

$$\max_{r,\mathcal{P}} \min_{1 \leq i \leq p} E_{r(i)}^{'}, \quad (2)$$

under the energy constraint that

$$E_i^{'} \geq 0, \forall i \in V, \quad (3)$$

which implies that any node can not spend more energy than its initial energy level. The maximum data preservation time of the entire network is therefore $\max_{r,\mathcal{P}} \min_{1 \leq i \leq p} E_{r(i)}^{'}/c$.

Theorem 2: The static data preservation problem is NP-hard.

Proof: We show that the disjoint connecting paths (DCP) problem [4], which is known to be NP-hard, is a special case of the decision version of our problem. The DCP problem is as follows. Given a graph $G(V, E)$ and a set of p disjoint source and destination vertex pairs (s_i, t_i), where $s_i, t_i \in V$ for $1 \leq i \leq p$, the goal is to find whether there are p vertex-disjoint paths $P(s_1, t_1), P(s_2, t_2), ..., P(s_p, t_p)$ in G.

In our static data preservation problem, for all the nodes in V, let the p source nodes be $S = \{s_1, s_2, ..., s_p\}$, and let $T = \{t_1, t_2, ..., t_p\}$ be another p nodes, and $T \cap S = \phi$ (empty set). Then assign the energy level of each node in T as $E \gg 1$, and the energy level for other nodes in $(V - T)$ (including S) as 1. We claim that to determine whether the maximum data preservation time of the network is $(E - 0.5)/c$ is the same as solving DCP problem, whether there exist p vertex-disjoint paths connecting the source and destination vertex pairs.

On one hand, if the maximum data preservation time equals $(E - 0.5)/c$, it must be the case that each data item in one of the nodes in S is distributed to one of the nodes in T. Since sending and receiving any data item for each node costs 0.5 units of energy, and all the nodes in $(V - T)$ have energy level of 1, the p redistribution paths $\{P(i, r(i))\}$, $1 \leq i \leq p$, must be mutually vertex-disjoint. On the other hand, if there exist p vertex-disjoint paths connecting p source and destination vertex pairs, these p paths can be used to distribute the p data items. In this case, the energy level of each node in T is $E - 0.5$, therefore maximum data preservation time equals $(E - 0.5)/c$. ∎

IV. Data Distribution Algorithms

Centralized Data Distribution Algorithm (CDA). Since the static data preservation problem is NP-hard, we propose a centralized heuristic as follows. The key idea is to distribute data to the nodes with highest energy level at the moment of distribution, while not using these nodes as intermediate relaying nodes for data distribution.

Algorithm 1: Centralized Data Distribution Algorithm

Initiation: all the data items are marked as not distributed, all the nodes with data items are marked as source nodes, all the nodes (including the source nodes) are marked as non-destination nodes initially.

BEGIN

while (there is still data item not yet distributed)
  Find the non-destination node that has the maximum energy level;
  if (this node is a source node)
    Mark its data item as distributed (even though this data item is not really moved), mark this node as a non-source node as well as a destination node;
  else //this node has one free storage
    Mark it as a destination node;
    Find the source node that is closest to this destination node (in term of number of hops), distribute its data item to this destination node along the shortest path that has least number of destination nodes; if all the shortest paths have the same number of destination nodes, choose the shortest path with the minimum energy of its included destination nodes the highest among all the shortest paths;
    Mark this data item as distributed;
    Mark this source node as non-source node;
    Update the remaining energy level of all the nodes on this shortest path;
end while;
RETURN minimum energy among destination nodes.
END. ◊

Distributed Data Distribution Algorithm (DDA). Each source node X performs the following:

1. Node X broadcasts an "offload" message to all its neighbors (limited to one hop neighbors only) with its remaining energy level.
2. A neighbor node Y upon receiving this "offload" message, checks if (its energy level $\geq$ energy level of X).[6]
   i. If No, neighbor node Y discards the message.
   ii. If Yes, neighbor node Y checks if (Y has storage space available).
      a. If No, neighbor node Y discards the message.
      b. If Yes, neighbor node Y replies back to X an ACK message with its remaining energy level.
3. After node X receives all the ACKs from its neighboring nodes (within a timeout interval), node X sends its data item to the neighbor with the highest energy level, and deletes the data item from its storage.
4. If node X does not receive any ACK after the timeout interval, it keeps the data item.

The neighbor node that obtains the data item from source node X is now a source node, and follows above algorithm to distribute the data item.

[6]Note that if Y receives multiple "offload" messages, it processes them one by one.

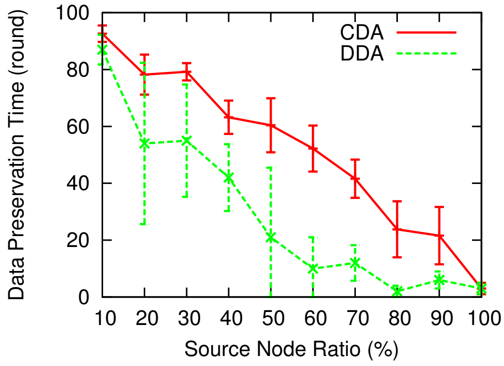

Fig. 2. Data preservation time with respect to the source nodes ratio. Initial energy is between 1 and 100. Energy draining rate c is 1 unit/round.

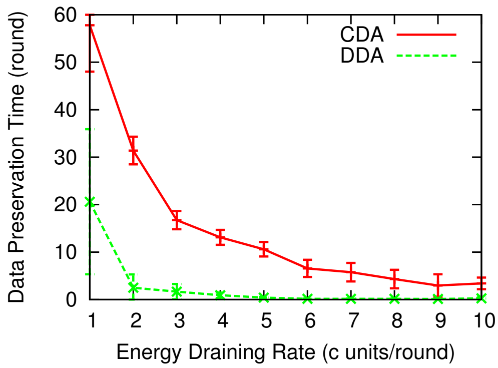

Fig. 3. Data preservation time with respect to the energy draining rate. Initial energy is between 1 and 100. Source node ratio is 50%.

V. Performance Evaluation

In this section, we present the simulation results and discussions. We adopt a grid-like topology to represent the sensor network (note that our proposed algorithms are applicable to other topologies). In all cases, the transmission range of the sensor is one unit, the length of each grid edge. The network size is 5×5. We randomly choose the source nodes in the network and vary the number of source nodes as a fraction of the network size, from 10%, 20%, ..., to 90%, 100%. Each source node initially has one data item. The storage capacity of each node is 1 unit. Each data point is an average over five runs. In all plots, we show error bars indicating the 90% confidence interval.

Fig. 2 and Fig. 3 show the data preservation time (in terms of number of rounds) as a function of source node ratio and energy draining rate, respectively, with the initial energy level of each node randomly chosen between 1 and 100. Fig. 2 shows that CDA achieves data preservation time larger than

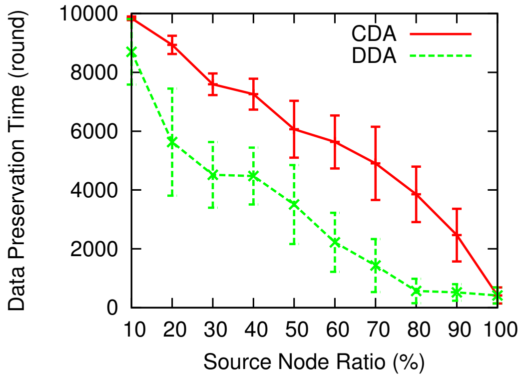

Fig. 4. Data preservation time with respect to the source nodes ratio. Initial energy is between 1 and 10000. Energy draining rate c is 1 unit/round.

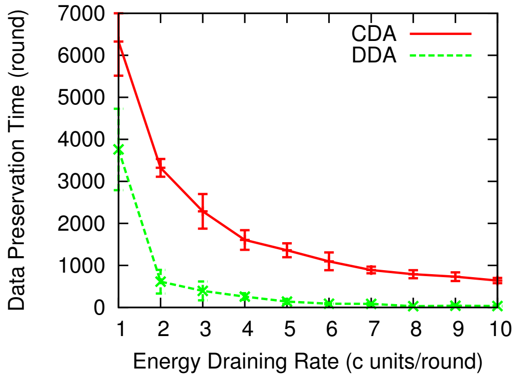

Fig. 5. Data preservation time with respect to the energy draining rate. Initial energy is between 1 and 10000. Source node ratio is 50%.

60 rounds when source node ratios are less than 50%. It also shows that with the increase of the source nodes in the network, data preservation time decreases for both CDA and DDA. This is because the more source nodes, the more data items need to be distributed, which costs more energy and therefore decreases the data preservation time. When every node is a source node, CDA and DDA perform the same since all the data stay with its source node. For most of the source node ratio range, DDA performs worse than CDA due to its localized behavior and the number of overhead messages in DDA. Fig. 3 shows that the performance of CDA and DDA for different energy draining rates. Fig. 4 and Fig. 5 show the same comparison, with the initial energy level of each node randomly chosen between 1 and 10000.

VI. Conclusion and Future Work

We study data preservation in intermittently connected sensor networks and formulate it as a graph-theoretic problem. We show that this problem is NP-hard and design a centralized greedy heuristic and a distributed data distribution algorithm. We empirically show that the centralized greedy algorithm performs close to the optimal. In future, we plan to incorporate storage constraint into our problem, and explore the tradeoff between storage and energy resources towards maximizing data preservation.

References

[1] Costas Busch, Malik Magdon-Ismail, Fikret Sivrikaya, and Bulent Yener. Contention-free mac protocols for wireless sensor networks. In *Proc. of DISC*, pages 245–259, 2004.

[2] J.-H. Chang and L. Tassiulas. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Trans. Networking*, 12:609–619, 2004.

[3] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *Proc. of SIGCOMM*, 2003.

[4] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[5] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. of HICSS 2000*.

[6] S. Jain, R. Shah, W. Brunette, G. Borriello, and S. Roy. Exploiting mobility for energy efficient data collection in wireless sensor networks. *MONET*, 11(3):327–339, 2006.

[7] D. Jea, A. A. Somasundara, and M. B. Srivastava. Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *Proc. of the IEEE DCOSS*, 2005.

[8] Raja Jurdak, Antonio G. Ruzzelli, and Gregory M.P. O'Hare. Radio sleep mode optimization in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 9(7):955–968, 2010.

[9] K. Kalpakis, K. Dasgupta, and P. Namjoshi. Maximum lifetime data gathering and aggregation in wireless sensor networks. In *Proc. IEEE Intl Conf. Networking (NETWORKS)*, pages 685–696, 2002.

[10] S. Li, Y. Liu, and X. Li. Capacity of large scale wireless networks under gaussian channel model. In *Proc. of MOBICOM 2008*.

[11] Chih-Yu Lin, Wen-Chih Peng, and Yu-Chee Tseng. Efficient in-network moving object tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 5(8):1044–1056, 2006.

[12] Changlei Liu and Guohong Cao. Distributed monitoring and aggregation in wireless sensor networks. In *Proc. of Infocom 2010*.

[13] T. Liu, C. M. Sadler, P. Zhang, and M. Martonosi. Implementing software on resource-constrained mobile sensors: experiences with impala and zebranet. In *Proc. of the IEEE International Conference on Mobile Systems, Applications, and Services(MobiSys 2004)*, 2004.

[14] L. Luo, Q. Cao, C. Huang, L. Wang, T. Abdelzaher, and J. Stankovic. Design, implementation, and evaluation of enviromic: A storage-centric audio sensor network. *ACM Transactions on Sensor Networks*, 5(3):1–35, 2009.

[15] L. Luo, C. Huang, T. Abdelzaher, and J. Stankovic. Envirostore: A cooperative storage system for disconnected operation in sensor networks. In *Proc. of INFOCOM 2007*.

[16] Ioannis Mathioudakis, Neil M. White, and Nick R. Harris. Wireless sensor networks: Applications utilizing satellite links. In *Proc. of the IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007)*, pages 1–5, 2007.

[17] Pavan Nuggehalli, Vikram Srinivasan, and Carla-Fabiana Chiasserini. Energy-efficient caching strategies in ad hoc wireless networks. In *Proc. of MOBICOM 2003*.

[18] Joongseok Park and Sartaj Sahni. An online heuristic for maximum lifetime routing in wireless sensor networks. *IEEE Transactions on Computer*, 55(8):1048–1056, 2006.

[19] Adrian Perrig, John Stankovic, and David Wagner. Security in wireless sensor networks. *Commun. ACM*, 47:53–57, June 2004.

[20] Bin Tang, Neeraj Jaggi, Haijie Wu, and Rohini Kurkal. Energy efficient data redistribution in sensor networks. In *Proc. of IEEE MASS 2010*.

[21] Sameer Tilak, Nael B. Abu-Ghazaleh, and Wendi Heinzelman. Collaborative storage management in sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 1(1/2):47–58, 2005.

[22] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data collection, storage, and retrieval with an underwater sensor network. In *Proc. of SenSys 2005*.

[23] S. Xiong, J. Li, and L. Yu. Maximizing the lifetime of a data-gathering wireless sensor networks. In *IEEE SECON 2009*.

[24] Y. Xue, Y. Cui, and K. Nahrstedt. Maximizing lifetime for data aggregation in wireless sensor networks. *Mobile Networks and Applications*, 10:853–864, 2005.

[25] Haibo Zhang and Hong Shen. Balancing energy consumption to maximize network lifetime in data-gathering sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 20:1526–1539, 2009.