Maximizing Data Preservation in Intermittently Connected Sensor Networks

Xiang Hou[1], Zane Sumpter[1], Lucas Burson[1], Xinyu Xue[1], Bin Tang[2]
[1]Department of Electrical Engineering and Computer Science, Wichita State University, USA
[2]Department of Computer Science, Azusa Pacific University, USA
Email: xxhou, zpsumpter, ljburson, xxxue2@wichita.edu, btang@apu.edu

Abstract— **In intermittently connected sensor networks, wherein sensor nodes do not always have connected paths to the base station, preserving generated data inside the network is a new and challenging problem. We propose to preserve the data items by distributing them from storage-depleted data generating nodes to sensor nodes with available storage space and high battery energy, under the constraints that each node has limited storage capacity and battery power. The goal is to maximize the minimum remaining energy among the nodes storing the data items, in order to preserve them for maximum amount of time until next uploading opportunity arises. We first give feasibility condition of this problem by proposing and applying a Modified Edmonds-Karp Algorithm (MEA) on an appropriately transformed flow network. We then show that when feasible solutions exist, finding the optimal solution is NP-hard. We develop a sufficient condition to solve the problem optimally. We then design a centralized greedy heuristic with less time complexity than that of the optimal, which also works when feasibility can not be satisfied and network partitions arise. Via extensive simulations, we show that the heuristic performs comparably to optimal.**

Keywords – Data Preservation, Intermittently Connected Sensor Networks, Network Flow-Based Algorithms

I. Background and Motivation

Many of the modern sensor network applications, such as solar-powered sensor networks [12], underwater or ocean sensor networks [4], and sensor networks monitoring volcano eruption and glacial melting [7, 13], are deployed in remote areas and challenging environments, where the deployed sensor network must operate without a closeby base station for a long period of time. In these applications, a large volume of generated data is first stored inside the network, and then uploaded to the distant base station via low rate satellite link [8], or periodic visit by data mules [3]. In a challenging environment, however, such uploading opportunities would be unpredictable and rare, making network connectivity to the distant base station inherently intermittent. We refer to such sensor networks as *intermittently connected sensor networks*.

When events of interest take place, sensors close to them may collect data more frequently than nodes far away, therefore run out their storage space more quickly than others and can not store newly generated data. The overflow data thus must be distributed/offloaded to other sensor nodes with available storage space to avoid getting lost. Besides, all the sensor nodes have finite and unreplenishable battery power, and are awake all the time monitoring while waiting for the data uploading opportunities, draining their battery energy constantly (Duty-cycling, in which individual sensor node activates very briefly for sensing and communication and stays in dormant state for a long period of time, is beyond the scope of this paper). It is therefore preferable that data is offloaded to sensor nodes with not only free storage space, but also high battery energy, to be preserved for a longer time. In this paper, we aim to preserve the overflow data for maximum amount of time by distributing it from storage-depleted data generating nodes (referred to as *data generators*) to sensor nodes with available storage space and high battery power (referred to as *destination nodes*), while considering that each sensor node has limited battery power and storage capacity. Note that there could be data generating nodes whose storage is not depleted yet and therefore can store more data - they are not considered as data generators in this paper. We refer to this problem as *storage-depletion induced data preservation problem (SDP)*. The SDP is naturally divided into two sub-problems:

- **Feasibility of Data Preservation.** Due to energy constraint at sensor nodes and in the event of network partitions, it is possible that not *all* the overflow data items can be offloaded. In this case, we say that the data preservation is *infeasible*. For any instance of SDP, we must decide whether the data preservation is feasible. If not, we endeavor to offload as many data items as possible.
- **Data Preservation Maximization.** When feasible solutions exist, it is important to achieve energy balancing among nodes that store data, to preserve all the data items. We assume that data preservation fails when the first data loss occurs, and are interested in maximizing the minimum remaining energy of destination nodes, after all the data items are successfully distributed. The challenge is that we not only need to find destination nodes to store the data items, but also need to find the paths along which each data is distributed from its data generator to destination node.

II. Problem Formulation of Data Preservation Maximization

Network Model. The sensor network is represented as an undirected connected graph $G(V, E)$, where $V = \{1, 2, ..., N\}$ is the set of N nodes, and E is the set of edges. There are p storage-depleted data generators, denoted as V_s. Without loss of generality, let $V_s = \{1, 2, ..., p\}$. Data generator i is referred to as DG i. The sensory data are modeled as a sequence of raw data items, each of which has the same unit size. Let s_i denote the number of data items DG i needs to distribute (that is, the amount of overflow data at DG i). Let $q = \sum_{i=1}^{p} s_i$ be the total number of data items to be distributed in the network. Let m_i be the available free storage space (in terms of number of data items) at non-DG sensor node $i \in V - V_s$.

Energy Model. Sensor node i (including DGs) has a finite and unreplenishable initial energy E_i, which is an integer number. In our energy model, for each node, sending a data item costs 0.5 unit of energy and receiving a data item costs 0.5 unit of energy. If a node is the DG offloading its data item or a destination node receiving the data item, it costs 0.5 unit of energy; if a node is an intermediate node relaying the data item, it costs one unit of energy (0.5 receiving and 0.5 sending).

Problem Formulation. Let $D = \{D_1, D_2, ..., D_q\}$ denote the set of q data items to be distributed in the entire network. Let $S(i) \in V_s$, where $1 \le i \le q$, denote the DG of data item D_i. A *distribution function* is defined as $r : D \to V - V_s$, indicating that data item $D_j \in D$ is distributed from $S(j)$ to its *destination node* $r(j) \in V - V_s$. Let V_d denote the set of destination nodes, i.e., $V_d = \{r(j) | 1 \le j \le q\} \subseteq V - V_s$. Let $P_j : S(j), ..., r(j)$, referred to as the *distribution path* of D_j, be the simple path (i.e., a set of distinct sensor nodes) along which D_j is distributed from $S(j)$ to $r(j)$ (note that $r(j) \neq S(j)$, since each item must be offloaded from its storage-depleted DG). Let x_{ij} be the energy cost incurred by sensor node i in the process of distributing data item D_j from $S(j)$ to $r(j)$, and let E'_i denote i's energy level after all the q data items are distributed. Then, $E'_i = E_i - \sum_{j=1}^{q} x_{ij}, \forall\, i \in V$, where $x_{ij} = 1$ if $i \in P_j - \{S(j), r(j)\}$, $x_{ij} = 0.5$ if $i \in \{S(j), r(j)\}$, and $x_{ij} = 0$ otherwise.

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

$$\max_{r,\mathcal{P}} \min_{1 \le i \le q} E'_{r(i)}, \quad (1)$$

under the energy constraint that each node can not spend more energy than its initial energy level, $E'_i \ge 0, \forall\, i \in V$, and the storage capacity constraint that the number of data items offloaded to node i is less than or equal to node i's storage capacity, $|\{j \mid r(j) = i, 1 \le j \le q\}| \le m_i, \forall\, i \in V$.

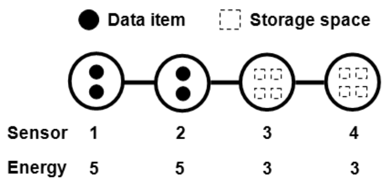


Fig. 1. Illustration of the SDP problem.

EXAMPLE 1: Fig. 1 gives an example of the SDP in a small linear sensor network with four sensor nodes. The initial energy level of each node is also indicated. Nodes 1 and 2 are DGs, with 2 and 2 data items to offload respectively. Nodes 3 and 4 are non-DGs, with 4 and 4 available storage spaces, respectively. The optimal solution is that all the data items are offloaded to node 3, resulting in the minimum remaining energy of destination nodes as 1 (node 3's remaining energy post distribution). Other solutions that one or more data items are offloaded to node 4 are not optimal. □

Theorem 1: The data preservation maximization problem is NP-hard.

Proof: We prove it by reducing the maximum 3-Dimensional matching problem [2] to data preservation maximization problem. The proof is omitted due to space constraints. ∎

Related Work. Tang et al. [10] have studied energy-efficient data redistribution problem in data-intensive sensor networks. Valero et al. [11] combine both data redistribution and retrieval into a single problem and propose an energy efficient approach to preserve the data in cases where communications with the sink are disrupted. However, both work implicitly assume that each sensor node has infinite energy level. Consequently, their objectives are to minimize the total energy consumption in data redistribution (and retrieval), which are shown to be solvable optimally in polynomial time. In this paper, we consider that each node has finite and unreplenishable energy. To preserve data for longest time, it is more desirable to maximize the minimum remaining energy of the destination nodes, which is shown to be NP-hard.

Takahashi et al. [9] propose data preservation heuristics in intermittently connected sensor networks. However, it assumes that each data generator has only one data item and each sensor node has one storage capacity, and presents only heuristic algorithms. Our work generalizes their work by permitting each data generator to have an arbitrary number of data items and each node to have arbitrary storage capacity. Furthermore, we present an optimal solution under some conditions.

Our work was inspired by a sequence of system research in disconnection-tolerant storage networks (EnviroStore [6], EnviroMic [5], SolarStore [14], and AdaptSens [12]). EnviroStore and EnviroMic are cooperative distributed storage systems designed for disconnected operations of sensor networks, to improve the utilization of the network's data storage capacity. EnviroStore provides more general storage balancing solutions, while EnviroMic focuses on acoustic monitoring, storage and trace retrieval. SolarStore and AdaptSens extend both above storage systems by considering solar energy and data reliability, and present an adaptive data collection, replication, and storage service for solar-powered sensor networks. In contrast, our work models the data preservation, coupled with storage and energy dynamics and balancing, as graph-theoretic problems. More specifically, our work has the theoretical roots in network flows, while addressing a specific sensor network application with storage/energy constraints of sensor nodes. We focuses on the hardness of the problems and try to achieve the optimality for the algorithm.

All of the above works do not identify and solve the feasibility problem in data preservation. As a result, their proposed techniques are not applicable for data preservation under challenging environment, wherein network partitions and energy-depleted sensor nodes are common.

III. Feasibility of Data Preservation

To find if any given instance of SDP is feasible, we first transform undirected graph $G(V, E)$ into a new directed graph $G'(V', E')$ as follows:

1. Replace each undirected edge $(i, j) \in E$ with two directed edges (i, j) and (j, i). Set the capacities of all the directed edges as infinite.
2. Split node $i \in V$ into two nodes: *in-node* i' and *out-node* i''. Add a directed edge (i', i'') with capacity of E_i, the initial energy level of node i. All the incoming directed edges of node i is incident on i' and all the outgoing directed edges of node i emanate from i''.
3. Add a new source node s, connect it to in-node i' of DG $i \in V_s$ with an edge of capacity s_i. Add a sink node t, and connect out-node j'' of non-DG node $j \in V - V_s$ to t with an edge of capacity m_j.

Figure 2 (a) shows the transformed network graph corresponding to the linear sensor network in Figure 1.

Modified Edmonds-Karp Algorithm (MEA). Edmonds-Karp algorithm [1] is an efficient maximum flow algorithm, wherein it always finds a shortest path between source and sink in the residual graph, and uses it as the next augmenting path. Next we present a modified Edmonds-Karp algorithm, called MEA (Algorithm 1), and demonstrate that it can be applied to above transformed G' to test the feasibility of any given instance of SDP. We first give below definitions.

Definition 1: (Sending Edge and Receiving Edge in a residual graph of $G'(V', E')$) In any residual graph of $G'(V', E')$ with some flow, for an *s-t* augmenting path, the second edge and the penultimate edge are defined as sending edge and receiving edge, respectively. The capacities of the sending and receiving edges represent the current energy level of the corresponding DG and non-DG nodes. □

Algorithm 1: Modified Edmonds-Karp Algorithm (MEA).
Input: $G'(V', E')$
Output: flow f

0. **Notations**:
   f: current flow from s to t
   $G_f^{'}$: residual graph of G' with flow f
   $c_f(u, v)$: residual capacity of edge (u, v)
1. $f = 0$; $G_f^{'} = G'$
2. **while** ($G_f^{'}$ contains an *s-t* path P using BFS)
3.   **for** each edge (u, v) in P
4.   **if** (u, v) is the sending edge or the receiving edge
5. $c_f(u, v) = 2 \times c_f(u, v)$
6.   Let $c_f(P) = \min\{c_f(u, v) : (u, v) \in P\}$
7.   Augment flow f along P
8.   **for** each edge (u, v) in P
9.   **if** (u, v) is an sending edge or the receiving edge
10. $c_f(u, v) = 0.5 \times c_f(u, v) - 0.5 \times c_f(P)$
11. $c_f(v, u) = c_f(v, u) + 0.5 \times c_f(P)$
12.   **else**
13. $c_f(u, v) = c_f(u, v) - c_f(P)$
14. $c_f(v, u) = c_f(v, u) + c_f(P)$
15.   **end while;**
16.   **RETURN** f

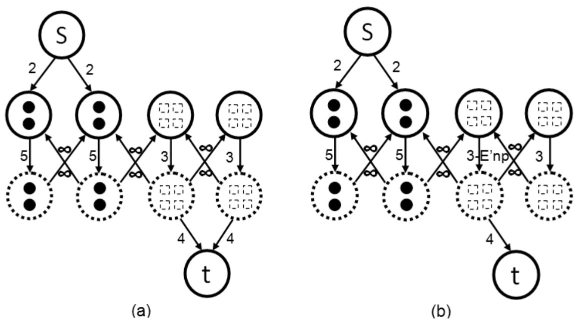


Fig. 2. (a) The transformed graph $G'(V', E')$ of the linear sensor network $G(V, E)$ in Figure 1 for feasibility problem. Whether there exists a maximum flow of 4 in the transformed graph indicates whether there exists a feasible data preservation strategy in the original linear sensor network. (b) The transformed graph $G''(V'', E'')$ of the linear sensor network $G(V, E)$ in Figure 1, given that the optimal destination nodes (node 3), and the minimum energy destination node n_p (node 3) with its energy level post distribution E'_{n_p} are all known.

Discussion of MEA. The time complexity of MEA is the same as Edmonds-Karp, which is $O(|V'||E'|^2) = O(N^5)$ [1]. However, there are several significant differences between MEA and Edmonds-Karp algorithm. First, in MEA, to find an *s-t* path in $G_f^{'}$ using BFS (line 2), all sending and receiving edges in $G_f^{'}$ with residual capacity greater than zero are considered (therefore an sending or receiving edge with capacity 0.5 will still be a valid edge in any *s-t* path). Second, to find residual capacity of an augmenting path P (line 6), the capacities of sending and receiving edges are doubled (lines 3-5). This is because that residual capacity of any augmenting path should be a positive integer while residual capacities of sending or receiving edges could be multiples of 0.5. Therefore, the residual capacities of sending and receiving edges are doubled first in order to find the amount of flow (data items) that can be sent along P. Third, in line 10, the residual capacities of the sending and receiving edge in P are first halved, to bring back to their correct values; then reduced by half of the residual capacity of path P, since it costs 0.5 unit of energy for the DG node (resp. destination node) to send (resp. receive) one data item.

Theorem 2: For any instance of the SDP, it is feasible to distribute all the q data items from DGs to other nodes if and only if that there is a maximum s-t flow of value q generated by the MEA in $G'(V', E')$. MEA also gives the distribution path for each data items.

Proof: The proof is omitted due to space constraints. ■

Below corollary immediately follows.

Corollary 1: When not all the data items can be offloaded due to energy constraint of sensor nodes, the MEA gives maximum number of data items that can be offloaded and each data item's distribution path.

IV. Centralized Algorithms for Data Preservation Maximization

A. *Optimal Data Preservation Algorithm When V_d is Known*

Theorem 3: When feasibility satisfies for a sensor network $G(V, E)$, in the optimal solution, if the set of destination nodes V_d, the minimum energy destination node (n_p) and its energy post distribution ($E_{n_p}^{'}$) are known, then finding the q corresponding distribution paths is equivalent to finding the maximum flow of value q on an appropriately transformed graph $G''(V'', E'')$ using MEA.

Proof: First we transform undirected graph $G(V, E)$ into a new directed graph $G''(V'', E'')$.

1. Replace each undirected edge $(i, j) \in E$ with two directed edges (i, j) and (j, i). Set the capacities of all the directed edges as infinity.
2. Split node $i \in V$ into two nodes: *in-node* $i^{'}$ and *out-node* $i^{''}$. Add a directed edge $(i^{'}, i^{''})$: if $i \in V_d$, set the edge capacity as $E_i - E_{n_p}^{'}$ (note that $E_i - E_{n_p}^{'}$ is always greater than or equal to zero); otherwise, set the edge capacity as E_i, the initial energy of node i. All the incoming directed edges of node i is incident on $i^{'}$ and all the outgoing directed edges of node i emanate from $i^{''}$.
3. Add a new soruce node s and connect it to in-node i' of DG $i \in V_s$, and set the edge capacity as s_i. Add a new sink node t, and connect out-node j'' of destination node $j \in V_d$ to t with an edge of capacity m_j.

The rest of the proof is similar to that for Theorem 2. ■

Figure 2 (b) shows the transformed graph assuming that destination nodes (node 3) are given and node 3 is node n_p. Note the differences between Figure 2 (a) and Figure 2 (b): in Figure 2 (a), new sink node t is connected to all the non-DG nodes while in Figure 2 (b), t is connected to all the destination nodes; besides, some edge capacities are different.

Optimal Algorithm When V_d is Known. With the support of Theorem 3, next, we show that if the optimal destination set V_d is given, then the optimal data distribution can be found in polynomial time, using Algorithm 2. The key observation is that for each destination node $i \in V_d$, it participates in at most q times of data relaying, therefore its energy level after distribution E_i' satisfies: $E_i - q \leq E_i' \leq E_i$. Algorithm 2 works as follows. For each node $i \in V_d$, we assume that it is the node n_p (i.e., the destination node with minimum energy post distribution) and then run a binary search of its remaining energy E'_{n_p} in the range

$[E_i - q, E_i]$. We find the maximum E'_{n_p} value that still yields a maximum flow of value q. Such maximum E'_{n_p} is the maximum of the minimum remaining energy of all destination nodes.

Algorithm 2: Optimal Algorithm.
Input: $G''(V'', E'')$, V_d
Output: Minimum remaining energy of destination nodes E_{min}

E_{min} = 0;
for each node $i \in V_d$
  Let $n_p = i$ and $E_{n_p} = E_i$;
  $x = E_{n_p}$; $y = E_{n_p} - q$;
  while $(x - y) > 0.5$;
    $E'_{n_p} = (x + y)/2$;
    Transform $G(E, V)$ to $G''(E'', V'')$;
    Run MEA on $G''(E'', V'')$;
    if (maximum flow value == q) and $E'_{n_p} > E_{min}$
      $E_{min} = E'_{n_p}$; $y = E'_{n_p}$;
    else $x = E'_{n_p}$;
  end while;
end for;
RETURN E_{min}.

Time Complexity. The time complexity of the optimal algorithm is $2^{(N-p)} \times (N - p) \times \log q \times N^5$, without detailed explanation due to space constraints.

B. BFS-Based Data Preservation Algorithm

The high time complexity of the optimal algorithm and the fact that it does not work for infeasible data preservation lead us to design new algorithm. Below we design a Breadth-First-Search (BFS)-based heuristic algorithm with much lower time complexity than that of the optimal. The heuristic includes the following three mechanisms, each starting from a non-DG node with available storage space X. It eventually finds a DG from which a data item is offloaded to X.

- **modified_BFS(X):** Using BFS that enqueues *only non-destination nodes that are reachable from* X, it finds the first DG that still has data items to offload. By not enqueuing destination nodes, it avoids adopting them as relaying nodes, in order to conserve their energy. It returns NULL if no such DGs can be reached from X this way.
- **modified_BFS(X, M):** Using BFS that enqueues *any nodes excluding minimum-energy destination nodes* M *that are reachable from* X, it finds the first DG that has data items to offload. By not enqueuing minimum-energy destination nodes, it avoids lowering the minimum-energy of destination nodes. It returns NULL if no such DGs can be reached.
- **BFS(X):** This is the regular BFS which enqueues *any nodes in the network that are reachable from* X. It finds the first DG that still has data items to offload. It returns NULL if no DG can be reached in this way; in this case, X is isolated and can not store any data items from DGs even if it has available storage.

Definition 2: (Data Preserving Node.) In sensor network graph $G(V, E)$, $X \in V$ is a data preserving node if a) it is a non-DG sensor node, b) it has available storage space, and c) there is a path between X and a DG that still has data items. That is, for a data preserving node X, $BFS(X) \neq NULL$. □

Algorithm 3 is the BFS-based data preservation algorithm. In each iteration, when there is still data item not yet distributed, it finds a data preserving node with maximum remaining energy to receive a data item. When it can no longer find any data preserving node, the algorithm stops. To maximize the minimum remaining energy of destination nodes, the algorithm tries to avoid existing destination nodes, especially destination nodes with minimum remaining energy, as intermediate relaying nodes. Algorithm 3 is designed to work for both feasible and infeasible data preservations. When infeasible data preservation takes place, it can still offload data as long as there are data preserving nodes in the same network partition.

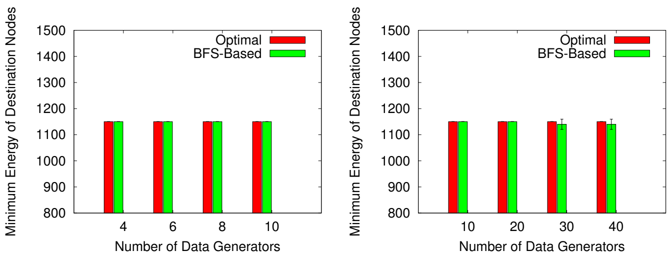
Algorithm 3: BFS-Based Data Preservation Algorithm.
Input: $G(V, E)$
Output: Minimum remaining energy of destination node E_{min}; Set of destination nodes with minimum remaining energy M;

$M = \phi$ (empty set);
$E_{min} = \infty$;
while (not all data items are offloaded)
  Find data preserving node X with maximum energy;
  if (X == NULL) Stops;
  if (X is not marked as a destination node)
    Mark X as a destination node;
  if ($X.energy < E_{min}$)
    $E_{min} = X.energy$; $M = \{X\}$
  if ($X.energy == E_{min}$)
    $M = M \cup \{X\}$;
  S = modified_BFS(X);
  if (S == NULL)
    S = modified_BFS(X, M);
    if (S == NULL)
      S = BFS(X);
      if (S == NULL)
        Mark X as a non-data preserving node;
        continue;
  end if;
  $s_S = s_S - 1$;
  $m_X = m_X - 1$;
  Let P be the distribution path from S to X;
  Update the energy levels of all the nodes in P;
  if (there exists a destination node K in P that $K.energy < E_{min}$)
    $M = \{K\}$; $E_{min} = K.energy$;
  if (there exists a destination node K in P that $K.energy == E_{min}$)
  $M = M \cup \{K\}$;
  For any node in P whose energy level reaches zero, delete all the edges incident on this node;
end while;
RETURN M and E_{min}.

Time Complexity. In each iteration of Algorithm 3, there are at most three BFS computations, and it takes $O(N)$ to find a data preserving node. Since the time complexity of a standard BFS algorithm is $O(|V| + |E|) = O(N^2)$, the time complexity of Algorithm 3 is $O(qN^3)$.

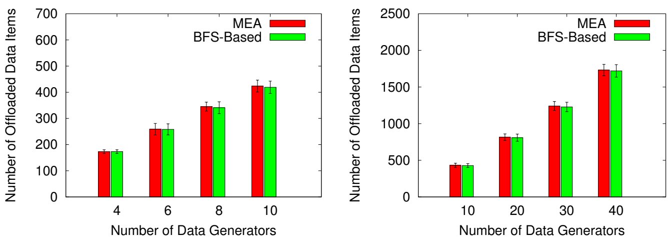
V. Performance Evaluation

We adopt grid topology since it facilitates the algorithm implementation without compromising the algorithms and their comparison. All the comparisons can be applied to a general network topology as well. To compare, all the algorithms take the same input files, which specify network topology, initial energy of each node, set of DGs, number of data items of each DG, and storage capacity of each non-DG. Each data point is an average over five runs, each run we randomly select a set of DGs. The network size is set as10×10 and 20×20. We set the number of data items at each DG as 50 and the storage capacity of each



(a) 10×10 network. (b) 20×20 network.

Fig. 3. Minimum remaining energy of destination nodes.



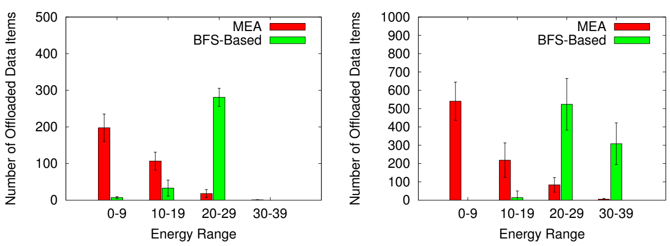
(a) 10×10 network. (b) 20×20 network.

Fig. 4. Number of successfully offloaded data items.

non-DG node as 100. In all plots, the error bars indicate 95% confidence interval.

Feasible Data Preservation. In feasible data preservation, since the Optimal assumes known destination nodes, we decide such destination nodes as follows. We assume that each destination node in Optimal has a full storage post data distribution: therefore, for x DGs, there are $\lceil 50x/100 \rceil = \lceil x/2 \rceil$ destination nodes. We choose $\lceil x/2 \rceil$ nodes from the non-DG nodes and assign 1200 as their initial energy, and set other nodes energy level in the range of [1000, 1100]. This way, we try to guarantee that those $\lceil x/2 \rceil$ destination nodes are the optimal destination nodes. Figure 3 shows that the BFS-Based performs very close to the Optimal in different network sizes.

Infeasible Data Preservation. When total data preservation becomes infeasible, we are interested in finding for each algorithm, how many data items are successfully offloaded and more specifically, the energy range the data items fall into. Since Optimal is not applicable to infeasible case and according to Corollary 1, the MEA gives maximum number of data items when not all the data items are offloaded, we compare MEA with BFS-Based. In all the plots, the initial energy level of each node is in the range of [10, 40]. Figure 4 shows that in terms of number of offloaded data items, BFS-Based performs almost the same as Optimal. Figure 5 shows for both algorithms, the



(a) 10×10 network (8 DGs). (b) 20×20 network (20 DGs).

Fig. 5. Number of successfully offloaded data items in each energy range.

number of data items that "fall into" each energy range. There are 8 and 20 DGs in 10×10 and 20×20 network respectively. In all the cases, BFS-Based performs the best, offloading data items to destination nodes with higher energy range of [20, 39]), while MEA offloads to lower half of the energy range ([0, 19]). For MEA, since its goal is to offload maximum amount of data items, it does not pay special attention to destinations with higher energy when it offloads data.

VI. Conclusion and Future Work

We have formulated and solved data preservation problem in intermittently connected sensor networks, which is a new problem that has not attracted much attention. We design several data preservation techniques that not only try to achieve optimal performance but also work for challenging network scenarios such as sensor energy depletion and network partitions. Currently the SDP is a static problem, in which the data to be offloaded is generated at the beginning and only once. We would like to address a real-time problem where data is generated and transmitted dynamically and periodically. As a second step, we will consider heterogeneous sensor networks wherein the data generated by different data generators are of different priorities and values, which is a more common sensor network scenario, but no doubt is a more challenging problem compared to the one studied in this paper.

VII. Acknowledgement

This work was supported in part by NSF Grants CNS-1116849 and EPS-0903806.

References

[1] Thomas Corman, Charles Leiserson, Ronald Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2009.

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

[3] 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.

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

[5] 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.

[6] 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*.

[7] K. Martinez, R. Ong, and J.K. Hart. Glacsweb: a sensor network for hostile environments. In *Proc. of SECON 2004*.

[8] 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.

[9] Masaaki Takahashi, Bin Tang, and Neeraj Jaggi. Energy-efficient data preservation in intermittently connected sensor networks. In *Proc. of the International Workshop on Wireless Sensor, Actuator and Robot Networks (WiSARN), in conjunction with IEEE INFOCOM 2011*.

[10] Bin Tang, Neeraj Jaggi, Haijie Wu, and Rohini Kurkal. Energy efficient data redistribution in sensor networks. In *Proc. of IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2010)*.

[11] Marco Valero, Mingsen Xu, Nick Mancuso, Wen-Zhan Song, and Raheem Beyah. Edr2: A sink failure resilient approach for wsns. In *Proc. of the IEEE ICC 2012*.

[12] Lili Wang, Yong Yang, Dong Kun Noh, Hieu Le, Tarek Abdelzaher, Michael Ward, and Jie Liu. Adaptsens: An adaptive data collection and storage service for solar-powered sensor networks. In *Proc. of the 30th IEEE Real-Time Systems Symposium (RTSS 2009)*.

[13] Geoff Werner-Allen, Konrad Lorincz, Jeff Johnson, Jonathan Lees, and Matt Welsh. Fidelity and yield in a volcano monitoring sensor network. In *Proc. of OSDI 2006*.

[14] Yong Yang, Lili Wang, Dong Kun Noh, Hieu Khac Le, and Tarek F. Abdelzaher. Solarstore: enhancing data reliability in solar-powered storage-centric sensor networks. In *Proc. of MobiSys 2009*, pages 333–346, 2009.