MAXIMIZING DATA PRESERVATION TIME IN INTERMITTENTLY CONNECTED SENSOR NETWORKS

A Thesis by

Xiang Hou

Bachelor of Science, Beijing Union University, 2009

Submitted to the Department of Electrical Engineer and Computer Science
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

May 2012

© Copyright 2012 by Xiang Hou

All Rights Reserved

MAXIMIZING DATA PRESERVATION TIME IN INTERMITTENTLY CONNECTED SENSOR NETWORKS

The following faculty members have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Computer Science.

_____
Bin Tang, Committee Chair

_____
Rajiv Bagai, Committee Member

_____
Bayram Yildirim, Outside Member

ACKNOWLEDGEMENTS

I would like to appreciate my adviser Dr. Bin Tang for guiding me in this thesis work even in my life. Due to his edification and motivation to me, I could finish my graduate study. I appreciate Dr. Rajiv Bagai and Dr. Bayram Yildirim for being my committee members and their time for reviewing my thesis. In addition, I would also like to thank Mr. Keenan Jackson helping me on my coursework. Moreover, I thank all my friends that assistant me on my graduate study.

Finally, I would like to express my great appreciation to my family.

ABSTRACT

In intermittently connected sensor networks, wherein sensor nodes do have connected paths to the base station periodically, preserving generated data inside the network is a new and challenging problem. We propose to preserve 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 data items, in order to preserve them for maximum amount of time until next uploading opportunity arises. We refer to this problem as *storage-depletion induced data preservation problem* (SDP). First, we give feasibility condition of this issue 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. Moreover, we develop a sufficient condition to solve SDP optimally. Finally, we design a distributed algorithm with less time complexity then compare it with flow based algorithm then show via simulations that distributed algorithm performs close to optimal solution.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

With cost of microcomputer becomes cheaper and the size of it becomes smaller, a great number of large-scale Wireless Sensor Networks (WSNs) with a series of data-intensive sensing applications have been deployed recently. Not as WSNs deployed in clement and reachable environment such as farmland where the data generated from sensors can be gathered easily, some particular WSNs are deployed in extremely dangerous or inaccessible places. They include acoustic sensor networks [1], solar-powered sensor networks [2, 3] underwater or ocean sensor networks [4, 5], and geophysical monitoring [6, 7]. Many of these sensor network applications are deployed in remote areas and challenging environments, where the deployed sensor network must operate without a nearby base station for a long period of time. In such scenarios, large volumes 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 [9, 10]. 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*. One of the main functionalities of the intermittently connected sensor networks is to store the large amounts of data inside the network before next uploading opportunity arises.

When events of interest take place, the sensors close to them may collect data more frequently than nodes far away, therefore run out their storage space more

quickly than others and cannot store any newly generated data. The overflow data thus must be distributed or 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 events of interests while waiting for the data uploading opportunities, draining their battery energy constantly. 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 thesis, 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. We refer to this problem as *storage-depletion induced data preservation problem* (SDP). The SDP can be naturally divided into two sub-problems. They are feasibility of data preservation and data preservation maximization. Whether it is feasible or not for data preservation is the prerequisite of data preservation maximization. So we investigate data preservation maximization in a feasible instance of SDP in this thesis. Figure 1 gives an example of intermittently connect sensor network.

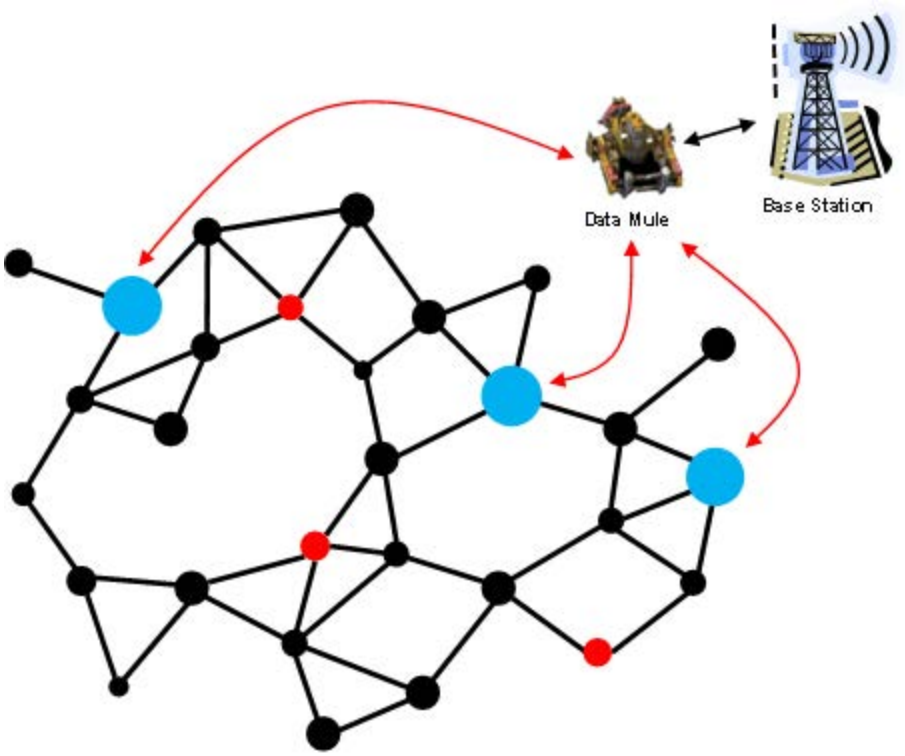

Figure 1: An intermittently connected sensor network

In Figure 1, each sensor has energy level, storage capacity and data items. The size of each node denotes their energy level; red nodes refer to data generators which already had overflow data items; blue nodes refer to desired destination nodes with enough available storage space. So it is preferable that overflow data on red nodes is offloaded to blue nodes via available connected paths. When next uploading opportunity meets, the data preserved on destination nodes can be transferred to base station by data mules.

To address the above challenges, we formulate the SDP as a graph-theoretic problem and solve the above sub-problems both theoretically as well as empirically. Specifically, we show that the feasibility problem is equivalent to the maximum flow

problem on an appropriately transformed flow network, and solve it by modifying the well-known Edmonds-Karp algorithm [11]. For the data preservation maximization, we show that it is NP-hard and provide a maximum flow-based algorithm to achieve the optimal solution when certain conditions are satisfied. However, the optimal nature makes it unsuitable for large-scale distributed sensor networks. Therefore, we propose a distributed algorithm that only depends on message passing among nodes. Then show distributed algorithm performs close to optimal solution.

1.1 Paper Organization

The rest of the thesis is organized as follows. Chapter 2 briefly introduces some existing works. In Chapter 3, we give network and energy cost models used, and formulate the data preservation problem. Chapter 4 describes two approaches based upon network flow algorithm to solve for feasibility problem and optimal solution when certain conditions are satisfied. Chapter 5 focuses on a distributed algorithm that only depends on message passing among nodes. In Chapter 6, we show the performance by comparing those two algorithms. Then we give both conclusion and future work in Chapter 7.

CHAPTER 2

RELATED WORK

Unlike most prevalent researches [12, 13, 14] which data collected from different sensors can always be transmitted to a nearby mobile or static base station via multi-hop communication under energy constraint, the data preservation problem in such an intermittently connected sensor network with energy and storage constraint becomes a new challenge.

Tang et al. [15] have studied energy-efficient data redistribution problem in data-intensive sensor networks. Valero et al. [16] 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 thesis, we consider that each node has finite and unreplenishable energy. To preserve data for the 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. [17] propose centralized and distributed 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 sensor node to

have arbitrary storage capacity. Furthermore, we present both centralized and distributed algorithms with performance guarantee. Besides, all above works do not identify and solve the feasibility problem in data preservation.

However, considering only energy constraint, Liu et al. [18] propose a redistribution method called DEDC taking data priority overall consideration to preserve data in isolated energy-harvesting wireless sensor networks which is similar to our intermittently connected sensor network, wherein the network consists of hundreds of static sensor nodes and only one sink node gathers all data to upload to base station by an external vehicle. Also, to reduce cost, they set up a specific zone named conservation area which is a predefined region adjacent to this single node to conserve forwarded data. Furthermore, they ignore all communication cost.

Our work was inspired by a sequence of system research in disconnection-tolerant storage networks (EnviroStore [19], EnviroMic [1], SolarStore [3], and AdaptSens [2]). 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 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 balancing, as graph-theoretic

problems, and focuses on their hardness and the optimality of their centralized and distributed solutions.

Maximizing data preservation time bears some resemblance to maximizing network lifetime in message routing [20, 21, 22, 23], where messages are routed from multiple source nodes to multiple sink nodes, and data gathering [24, 25, 26, 27], where data items are routed from multiple source nodes to a single sink node. Our work is most related to the former, which is reviewed as follows. Chang and Tassiulas [21] 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. Kar et al. [22] develop an admission control capacity competitive (the capacity is the number of messages routed over some time period) algorithm, CMAX (capacity maximization), with logarithmic competitive ratio. Park and Sahni [23] study how to route a sequence of messages each for a source and destination nodes pair. They propose a heuristic, called online maximum lifetime (OML), and show via simulations that OML is superior to CMAX in terms of network lifetime maximization, energy consumption and energy balancing.

CHAPTER 3

MODELS AND PROBLEM FORMULATION

3.1 Network Model

The sensor network is represented as a general undirected graph $G(V, E)$, where $V = \{1, 2, \dots, N\}$ is set of N nodes, and E is the set of edges. Two nodes are connected by an undirected edge if they are within the transmission range of each other and thus can communicate directly. There are p data generators, denoted as V_s. Without loss of generality, let $V_s = \{1, 2, \dots, p\}$. 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 be distributed. 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 sensor node $i \in V$. If $i \in V_s$, then $m_i = 0$, implying that a DG node has zero available storage space. If $i \in V - V_s$, then $m_i \geq 0$, implying that non-DG node i can store another m_i data items.

Finally, we assume that the total storage capacity of the entire network is no less than the total size of the data to be offloaded, i.e. $\sum_{\mathrm{i=p+1}}^{\mathrm{N}} \mathrm{m_i} \geq \mathrm{q}$. Otherwise, the problem becomes trivially infeasible.

3.2 Energy Model

Each sensor node i (including DGs) has unreplenishable and a finite initial energy E_i, which is an integer number. In our energy model, we assume that 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). Therefore, energy consumption of sending a data item from a DG to a sensor node equals the number of hops the data item traverses. We assume that there exists a contention-free MAC protocol (e.g. [28]) that provides channel access to the nodes.

3.3 Problem Formulation

Let $D = \{D_1, D_2, \dots, D_q\}$ 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 $D_i \in D$ is distributed from $S(i)$ to its destination node $r(i) \in V - V_s$. Let V_d denote the set of destination nodes, i.e. $V_d = \{r(i) | 1 \le i \le q\} \subseteq V - V_s$. Let $P_i : S(i), \dots, r(i)$, referred to as the distribution path of D_i, be the sequence of distinct sensor nodes along which D_i is distributed from $S(i)$ to $r(i)$ (Note that $S(i) \neq r(i)$, 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 $S(i)$ to $r(i)$, and let E_i' denote i's energy level after all the q data items are distributed. Then,

$$E_i' = E_i - \sum_{j=1}^{q} x_{ij}, \qquad \forall i \in V, \tag{1}$$

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. Here, i could be the DG or the destination node of D_j (with energy cost 0.5), or an intermediate relaying node of D_j (with energy cost one), or not involved with the

distribution at all (with energy cost zero). In this thesis we assume that a node can still relay data items even though it has a full storage. Table 1 lists all the notations.

TABLE 1

NOTATION SUMMARY

Notation	Explanation
$\boldsymbol{V_s}$	The set of data generators (DGs)
$\boldsymbol{V_d}$	The set of destination nodes
$\boldsymbol{x_{ij}}$	The energy cost of node i on distribution D_j
$\boldsymbol{s_i}$	The number of data items DG i has
$\boldsymbol{m_j}$	The storage capacity of node j
$\boldsymbol{S(i)}$	The DG node of data item D_i
$\boldsymbol{r(i)}$	The destination node of D_i
$\boldsymbol{P_i}$	The distribution path of D_i

The objective of the SDP is to find a data preservation strategy, including a distribution function r and a set of paths $P = \{P_1, P_2, \dots, P_q\}$, 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.

$$\max_{r,P} \min_{1 \le i \le q} E'_{r(i)} \tag{2}$$

under the energy constraint that each node cannot spend more energy than its initial energy level,

$$E'_i \ge 0 \qquad \forall i \in V, \tag{3}$$

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,

$$|\{j | r(j) = i,\ 1 \le j \le q\}| \le m_i\,, \qquad \forall i \in V. \tag{4}$$

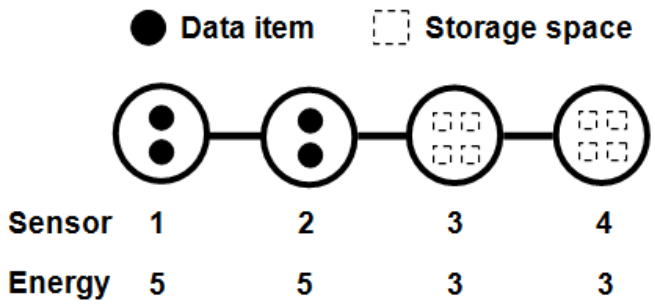

Figure 2: Illustration of the SDP problem.

Figure 2 shows an example of the SDP problem 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 be offloaded, respectively. Nodes 3 and 4 are non-DGs, with 4 and 4 available storage spaces, respectively. To maximize the minimum energy of the destination nodes, the optimal solution is that all data are offloaded to node 3, resulting in the minimum remaining energy of destination nodes as 1 (remaining energy of node 3 post distribution). However, 1 data item offloaded to node 4 while 3 data items offloaded to node 3 results in minimum remaining energy of destination nodes as 0.5 (remaining energy of node 3 post distribution). Even worse, if 3 data items offloaded to node 4, then it is not possible to offload the left one data item to either node 3 or node 4, since node 3's energy level already reaches 0.

Theorem 1: The SDP is NP-hard.

Proof: We prove it by reduction from the static data preservation problem [17], which is proven as NP-hard and is a special case of SDP. Specifically, in [17], it assumes each data DG i or each non-DG i has only single data item or storage capacity, respectively. Therefore, compared with their work [17], our work is more generalized

that each data DG i could have an arbitrary number of data items and each non-DG could have an arbitrary storage capacity. Thus, The SDP is NP-hard.

CHAPTER 4

FEASIBILITY PROBLEM AND CENTRALIZED DATA PRESERVATION ALGORITHM

4.1 Feasibility of Data Preservation

For a given instance of the SDP, it is possible that there does not exist a feasible solution. That is, given a network topology, the set of DGs with data items each needs to offload, the set of non-DG nodes with their available storage space, the initial energy level of all the nodes (including DGs), it is possible that not all the data items can be distributed due to energy constraint at nodes. For example, when a DG has 10 data items while its energy level is 2, it will exhaust its energy before finishing distributing all its data items.

To find the feasibility condition of the problem, we first transform undirected graph $G(V,\ E)$ into a new directed graph $G'(V',\ E')$ as follows.

1. $V' = V \cup \{s\} \cup \{t\}$, where s is the new source node and t is the new sink node.
2. 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.
3. Split each node $i \in V$ into two nodes: in-node i' and out-node i''. Add a directed edge $(i',\ i'')$ with capacity as E_i, the initial energy of node i. All the incoming directed edges of node i are incident on i' and all the outgoing directed edges of node i emanate from i''.

4. Connect s to each node i', the in-node of DG $i \in V_s$ with an edge of capacity s_i. Connect each node j'', the out-node of non-DG sensor node $j \in V - V_s$, to t with an edge of capacity m_j.

Therefore, $E' = \{(i, j): (i, j) \in E\} \cup \{(j, i): (i, j) \in E\} \cup \{(i', i''): i \in V\} \cup \{(s, i'): i \in V_s\} \cup \{(j'', t): j \in V - V_s\}$. We have $|V'| = 2|V| + 2$ and $|E'| = 2|V| + 2|E|$. Figure 3 shows the transformed network graph corresponding to linear sensor network in Figure 2.

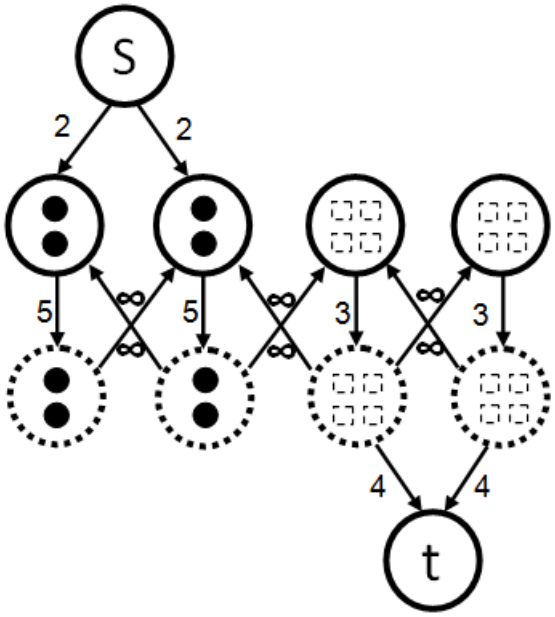

Figure 3: The transformed graph of the linear senor network in Figure 2 for feasibility problem.

Edmonds-Karp algorithm [11] is an efficient maximum flow algorithm, wherein it always finds a shortest path between the source and destination using breadth-first-search (BFS), and uses it as the next augmenting path in the residual network. 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 blow definitions.

Definition 1: In graph G', 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

Begin

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** (if there exists a path P from s to t in G_f 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 $c_f(P) \leftarrow \min\{\ c_f(u,\ v)\colon (u,\ v)\ is\ in\ P\}$;

7. Augmenting flow f along P;

8. **for** each edge $(u,\ v)$ in P

9. **if** $(u,\ v)$is the sending edge or the receiving edge

10. $c_f(u,\ v) = 0.5 \times c_f(u,\ v) - 0.5 \times\ c_f(P)$;

else

11. $c_f(u,\ v) = c_f(u,\ v) -\ c_f(P)$;

12. $c_f(v,\ u) = c_f(v,\ u) +\ c_f(P)$;

13. end while;

14. return f

End

The time complexity of MEA is the same as Edmonds-Karp, which is $O(|V'||E'|^2) = O(N^5)$ [11]. 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 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 first doubled in order to find the amount of flow (data item) that can be sent along P. Third, in lines 10, the residual capacities of the sending and receiving edge in P are first halved, to bring back 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, while other nodes in P each costs one unit of energy relaying one data item.

Below we present one critical observation of MEA.

Lemma 1: MEA guarantees that a DG distributes all its data before serving as relaying node, and that a non-DG receives data until its capacity is full before serving as relaying node.

Proof: By way of contradiction, assume that a DG serves as a relaying node before offloading its entire data item. That is, in one round of MEA, an augmenting path P : $A, \dots, B, \dots, C$ is chosen to offload data items from DG A to non-DG destination node C, and DG B is an intermediate relaying node in this path that still has data items of its own to be distributed. Therefore there exists a path P': $B, \dots, C$, wherein some data items of B can be offloaded to C. P' is obviously shorter that P, contradicting that P is the shortest path among all the available augmenting paths. By a similar argument, assume that one non-DG node with available storage serves as a relaying node of some data items. If this node is chosen to serve as the destination node of at least part of those data items, it results in a shorter path, which contradicts again with the MEA algorithm.

Theorem 2: Under our energy model, the value of flow found by MEA is no less than the value of the flow found by other max flow algorithms.

Proof: By way of contradiction, assume that another maximum flow algorithm, called O, which does not always choose shortest path between s and t as the augmenting path, yields a larger flow than MEA. This algorithm could be the Ford-

Fulkerson algorithm [11]. Note that for O, like MEA, it will not take a non-DG node with available storage as relaying node, because an augmenting path can always end at the first non-DG with available storage. Therefore we only consider that in some rounds of O, an augmenting path is selected between a DG and non-DG, while another DG is an intermediate relaying node in this path that still has its own data items to offload.

Assume O takes k rounds. Without loss of generality, assume that in round l $(1 \leq l \leq k)$, O offloads a data items from DG A to non-DG destination node C, and DG B is an intermediate relaying node in this path that still has b data items of its own to be distributed. In this case, we can change it such that B offloads b data items to C while A offloads $(a - b)$ data items to C, therefore saving the energies for all the nodes on the path: $A, \ldots, B$ while producing the same amount of flow. These saved amounts of energy can potentially be used to generate some flow, therefore giving more flow than that of O, a contradiction.

We use Figure 2 to illustrate Theorem 2, by changing node 2's initial energy from 5 to 1. Using MEA, node 2's two data items can be offloaded to node 3, resulting in maximum flow of 2. However, using other maximum flow algorithms, if node 1 is selected to offload one of its data item to node 3, it results in maximum flow of 1 since after relaying one data item, node 2's energy becomes zero.

Theorem 3: 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 there is a maximum s-t flow of value q by the MEA in $G'(V',\ E')$. MEA also gives the distribution path for each data items.

Proof: According to Integrality Theorem [11], if all the edge capacities are integers in a flow network, then there exists a maximum flow for which flow value on each edge is an integer. This integrality property is needed in our problem since the size of each data item is unsplittable. If there is a maximum flow of value $q = \sum_{i=1}^{p} s_i$ from s to t, there exists an integer flow: with s_1 amount of flow on edge$(s,\ 1')$, s_2 amount of flow on edge $(s,\ 1')$, … , and s_p amount on $(s,\ p')$. Therefore, there must be s_1 amount of net flow out of DG 1, s_2 amount of net flow out of DG 2, ..., and s_p amount of net flow out of DG p, meaning that it is feasible to distribute all the data items from all DGs. On the other hand, if there is a feasible distribution, that is, DG i can offload its s_i number of data items, each following a distribution path from DG i to a destination node in V_d. Then we can send s_i units of flow from s to i', and from i' following the paths in this feasible distribution to t, without violating the capacity conditions of edges. It is obvious that the flow obtain $q = \sum_{i=1}^{p} s_i$ is the maximum flow.

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.

In Figure 3, the transformed graph $G'(V',\ E')$ of the linear sensor network $G(V,\ E)$ in Figure 2 for feasibility problem. Whether there exists a maximum flow of 4 in G' indicates whether there exists a feasible data preservation strategy in the original linear sensor network.

4.2 Optimal Solution When V_d is Given

Below we first show that for any instance of feasible data preservation, when certain conditions are met, finding the optimal solution is equivalent to solving MEA on another appropriately transformed graph.

Theorem 4: When feasibility satisfies, 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 a correctly transformed graph using MEA.

Proof: First we show how to transform undirected graph $G(V,\ E)$ into another new directed graph $G''(V'',\ E'')$.

1. $V'' = V \cup \{s\} \cup \{t\}$, where s is the new source node and t is the new sink node.
2. 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.
3. Split each 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. All the incoming directed edges of node i are incident on i' and all the outgoing directed edges of node i emanate from i''.
4. Connect s to each node i', the in-node of DG $i \in V_s$ with an edge of capacity s_i. Connect each node j'', the out-node of non-DG sensor node $j \in V - V_s$, to t with an edge of capacity m_j.

Then, the rest of the proof is similar to that for Theorem 3.

In above $E'' = \{(i,\ j): (i,\ j) \in E\} \cup \{(j,\ i): (i,\ j) \in E\} \cup \{(i',\ i''): i \in V\} \cup \{(s,\ i'): i \in V_s\} \cup \{(j'',\ t): j \in V_d\}$, We have $|V''| = 2|V| + 2$ and $|E''| = |V| + 2|E| + |V_s| + |V_d|$.

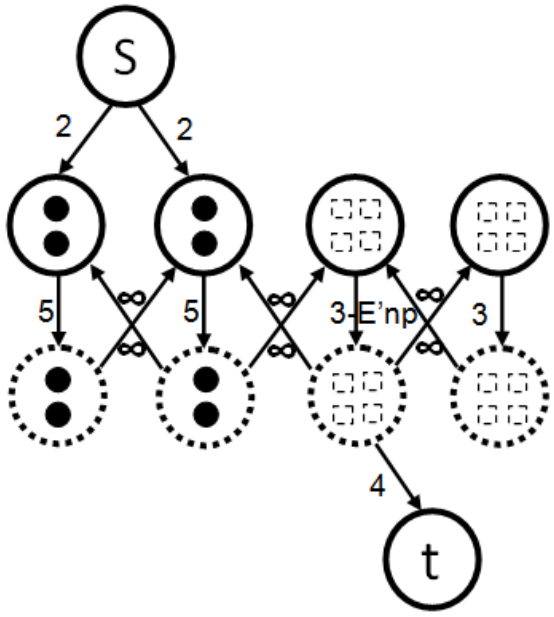

Figure 4: The transformed graph of the linear senor network in Figure 2.

In Figure 4, we assume that both the optimal destination nodes (nodes 3) and the minimum energy destination node n_p(node 3) with its energy level post distribution E'_{n_p} = 1 is all known. The flow with value of 4 in the transformed graph shows how data is distributed from DGs (node 1 and 2) to destination nodes (nodes 3).

Note the differences between Figure 3 and Figure 4: in Figure 3, new sink node t is connected to all the non-DG nodes while in Figure 4, t is connected to all the destination nodes; besides, some edge capacities are different.

Algorithm 2: Optimal Algorithm.

Input: $G''(V'',\ E''), V_d$

Output: Minimum remaining energy of destination nodes E_{min}

Begin

1. $E_{min} = 0$;

2. **for** each node $i \in V_d$

3. Let $n_p = i$ and $E_{n_p} = E_i$;

4. $x = E_{n_p}$ and $y = E_{n_p} - q$;

5. **while** $(x - y) > 0.5$

6. $E'_{n_p} = (x + y)/2$;

7. Transform $G(V, E)$ to $G''(V'',\ E'')$;

8. Run MEA on $G''(V'',\ E'')$;

9. **if** (maximum flow value == q) and $E'_{n_p} > E_{min}$

10. $\tau_{min} = E'_{n_p}; y = E'_{n_p}$;

11. **else** $x = E'_{n_p}$;

12. **end while;**

13. **end for;**

14. return E_{min}

End

4.3 Analysis of Optimal Solution

With the support of Theorem 4, next, we show that if the optimal destination set V_d is given, then the optimal data preservation time can be found in polynomial time using algorithm 2 (optimal algorithm). The key observation is that for each destination

node $i \in V_d$, it participates at most q times of relaying the data items, therefore its energy level after distribution E_i' satisfies the following: $E_i - q \leq E_i' \leq E_i$. The optimal algorithm works as follows. For each of the 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 try to find the maximum E'_{n_p} value that still yields a maximum flow of value q (meaning that all the data items are distributed from DGs to destination nodes). Such maximum E'_{n_p} is the maximum data preservation time of the network.

4.4 Time Complexity.

Since the time complexity of MFA is $O(N^5)$ and there are most $N - p$ destination nodes. Therefore, the time complexity of algorithm 2 is $(\mathrm{N} - p) \times \log q \times N^5$. Now, to find the optimal solution for any feasible instance of the SDP, we need to exhaust all the $2^{(N-p)}$ possible destination node sets whose total storage capacity are greater than or equal to q. For each such destination node set, we run algorithm 2 to find the maximum of minimum remaining energy. Therefore, the time complexity of the optimal algorithm is $2^{(N-p)} \times (N - p) \times \log q \times N^5$.

CHAPTER 5

DISTRIBUTED DATA PRESERVATION ALGORITHM

5.1 Distributed Data Preservation

We design a distributed data preservation algorithm and discuss its optimality under some assumptions. The distributed algorithm takes place in iterations. Each iteration consists of the following four stages:

Input: $G(V,\ E)$

Output: Minimum remaining energy of destination nodes**;** average energy level of destination nodes; total energy consumption of all nodes.

1. **Selecting local-high-energy node among non-DGs.** Each non-DG with free storage broadcasts its energy level to its one-hop neighbors. If its energy is the maximum among all its neighbors, it declares itself as a local-high node. Note that DG nodes do not participate in this stage.
2. **Advertisement of DGs.** Each DG i broadcasts an advertisement message in the network containing its ID, and number of data items to offload s_i. Every non-local high node (including DGs) forwards the advertisement message the first time it receives it. A local-high node does not forward it. An integer (initialized as 0) is also included in the advertisement message and incremented every time the message is forwarded, to capture the distance between DG node i and a local-high node.
3. **Commitment and notification of local-high nodes.** Among the set of DGs that local-high node j receives advertisement from, it selects the closest DG,

say, DG i, to commit. It commits $C_{ji} = min\,(m_j\,, s_i)$ storage space to DG i and sends a commitment message to DG i with its ID, its current energy level E_j , the number of storage space committed, and its distance to the DG. j also sends a notification message to each of the DGs that it does not commit to with the commitment message that j sends to DG i.

4. **Data offloading of DGs**. After receiving all commitment and notification messages, DG i performs the following computation.

   A. Sort the local-high nodes in descending order of their energy levels: $LH(0), LH(1), \dots, LH(m)$, each with energy level of $E(0)$, $E(1), \dots, E(m)$ respectively. Let $f(n)$ be the total number of data items that $LH(0), LH(1), \dots, LH(n-1)$, could store while each still having energy level greater than or equal to $E(n)$. That is,

$$f(n) = \sum_{k=0}^{n-1} \min((E(k) - E(n)) \times 2, \qquad m_{LH_{(k)}})$$

   When $\mathrm{f(n)} \geq \sum \mathrm{s_i} \geq \mathrm{f(n-1)}$, it needs the top n local-high nodes (that is, $\mathrm{LH(0), LH(1), \dots, LH(n-1)}$) to store $\sum \mathrm{s_i}$ data items in this iteration. $m_{LH_{(k)}}$ is the $LH(k)$'s available storage.

   B. For $LH(k)$, $(n-1 \geq k \geq 0)$, denote the number of data items it could store as $S\big(LH(k)\big)$, then

$$S\big(LH(k)\big) = \min(\big(E(k) - E(n-1)\big) \times 2 + \Delta_k, \quad m_{LH_{(k)}})$$

   Where $\big(E(k) - E(n-1)\big) \times 2$ is number of data items $LH(k)$ should store before its energy decreases to $E(n-1)$, Δ_k is number of data

items $LH(k)$ shares with other nodes in L_i that still have available storage for the rest $\sum s_i - f(n-1)$ data items

$$\Delta_k = \begin{cases} \frac{\sum s_i - f(n-1)}{n'} & if\ m_{LH_{(k)}} \geq 2\big(E(k) - E(n-1)\big), \\ m_{LH_{(k)}} & Otherwise, \end{cases}$$

where $n' = \{LH(k) | 0 \leq k \leq n-1\}$.

C. If DG i receives commitment message from local-high node j, and if j is going to store $S(j)$ amount of data items according to above calculation, then DG i offloads $min\ (\ C_{ji},\ S(j))$ to j. Otherwise, DG i will not offload any data to j. If DG i still has data items left, it starts next iteration.

5.2 Discussion of Optimality

In each iteration of the distributed algorithm, if every DG's advertisement message reaches every local high node and all local-high nodes are optimal destination nodes, it guarantees that the minimum energy of destination nodes is maximum post data distribution, compared with other offloading strategy. First, under this assumption, at the end of Stage 3, each DG receives complete and consistent information as which local-high nodes commits how many storages to which DG. Therefore at Stage 4, each DG's computation results are the same. Second, according to Steps A and B in Stage 4, after data offloading in this iteration, the remaining energy levels of local-high nodes $LH(0), LH(1), \dots, LH(n)$ are roughly the same, achieving the effect that the minimum remaining energy is maximized.

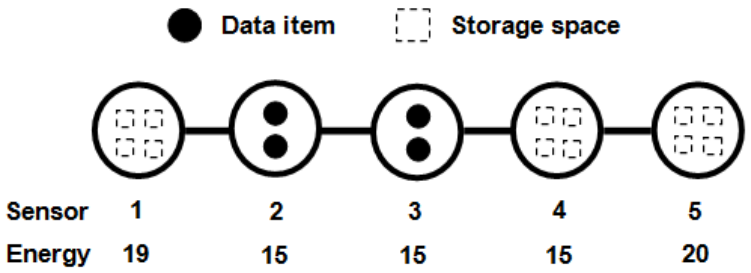

Figure 5: An example illustrating the optimality of the distributed algorithm.

Figure 5 shows a simple example upon which optimal solution is obtained in distributed algorithm. In this example, nodes 1 and 5 are local-high energy nodes. In Stage 3, node 1 commits 2 storages to node 2 and sends a notification to node 3, while node 5 commits 2 storages to node 3 and sends a notification to node 3. In Stage 4, the calculated results at both node 2 and 3 are that node 1 should receive 1 data item and node 5 should receive 3 data items. Since node 2 receives 2 commitments from node 1, it eventually sends only 1 data to node 1. Since node 3 receives 2 commitments from node 5, it eventually sends 2 data items to node 5. In the second iteration, node 2 advertises its one data item. Via the four stages, this data item will be offloaded to node 5.

5.3 Message Complexity

The total number of message transmissions in each iteration is $O(p^2N + qN)$.

CHAPTER 6

PERFORMANCE EVALUATION

We compare the performance of the network flow-based optimal algorithm (referred to as **Optimal**) and the distributed algorithm (referred to as **Distributed**). Given a data preservation instance, we either increase or decrease the energy levels of sensor nodes and then use MEA applying upon transformed graph G' to check whether it results in feasible or infeasible data preservation. We adopt grid topology for sensor networks 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. We vary the network size at 10x10, 15x15, or 20x20. We set the number of data items at each DG as 50 and the storage capacity of each non-DG node as 100. In all plots, the error bars indicate 95% confidence interval.

In feasible data preservation, since the Optimal assumes known destination nodes, we decide such known 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 $[50x/100] = [x/2]$ destination nodes. We choose $[x/2]$ nodes from the non-DG nodes and assign 1200 as their initial energy level, and assign other nodes

energy level in the range of [1000, 1100]. This way, we try to guarantee that those $[x/2]$ destination nodes are the optimal destination nodes.

6.1 Minimum Initial Energy Level for Feasibility.

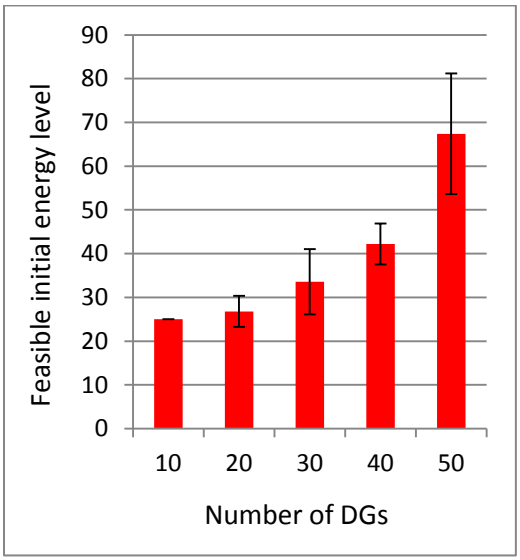

Figure 6: Minimum initial energy level for DGs in 10x10 network.

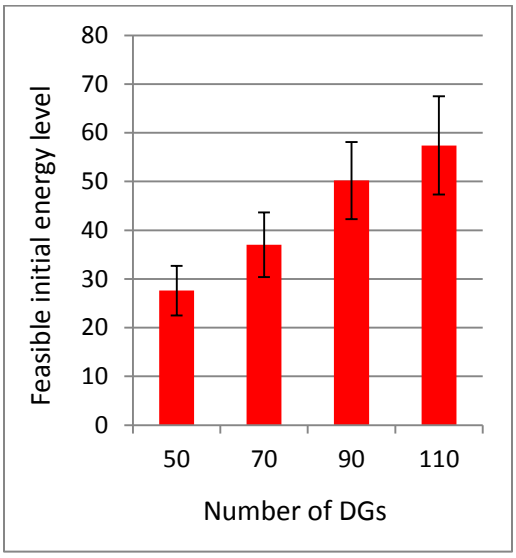

Figure 7: Minimum initial energy level for DGs in 15x15 network.

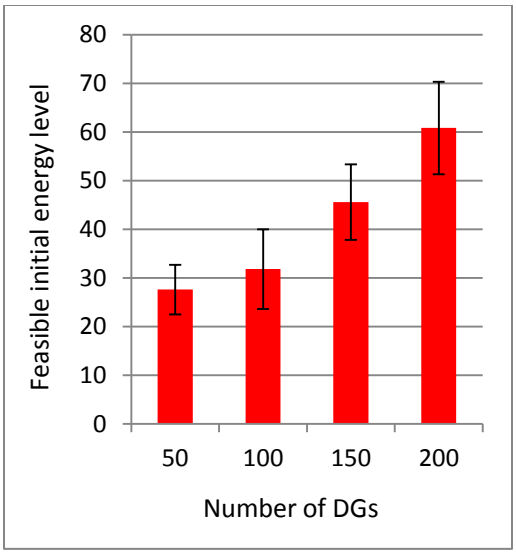

Figure 8: Minimum initial energy level for DGs in 20x20 network.

The data distribution becomes infeasible when nodes do not have enough initial energy. Therefore it is worthwhile to find the minimum initial energy level of each node that sustains feasible distribution of all the data items. Figure 6, 7 and 8 show the minimum initial feasible energy of all the three network topologies with varying number of DGs. Each DG has 50 data items and the storage capacity of each non-DG node is 100. It shows that in each topology, when the number of DGs is small, the minimum initial feasible energy level is almost 25, which is the energy needed for each DG to offload its 50 data items. When number of DGs increases, the minimum initial feasible energy increases. This is because that when the number of DGs is small, not many data items are to be offloaded. Therefore, each DG does not relay data items for other DGs, except offloading its own data. When number of data items increases, DGs need to relay data items for each other, increasing their minimum energy level for feasibility. For the rest of the simulations, to guarantee feasible data offloading, we set the initial energy of each node to be higher than its corresponding minimum energy level for feasibility.

6.2 Minimum Energy Level among Destination Nodes.

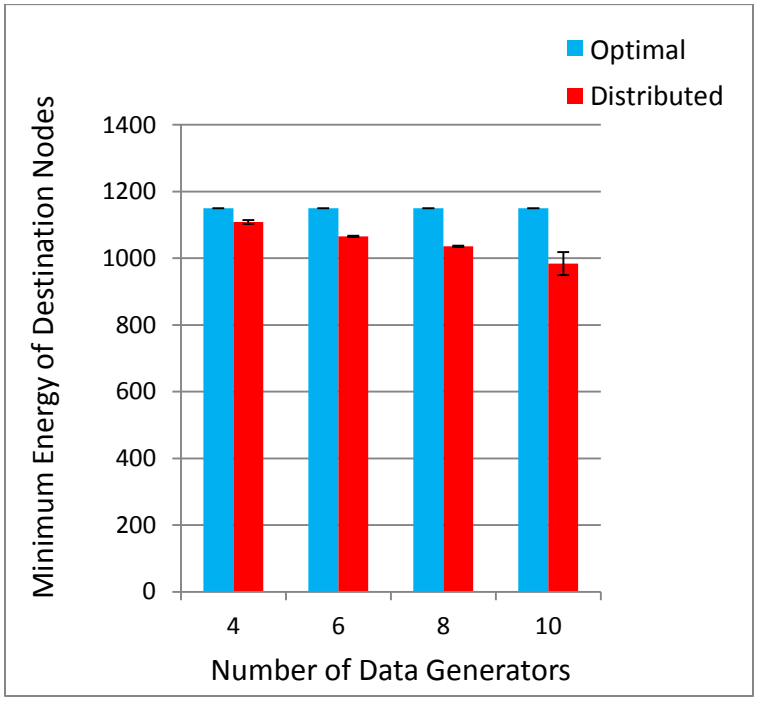

Figure 9: Minimum remaining energy of destination nodes in 10x10 network.

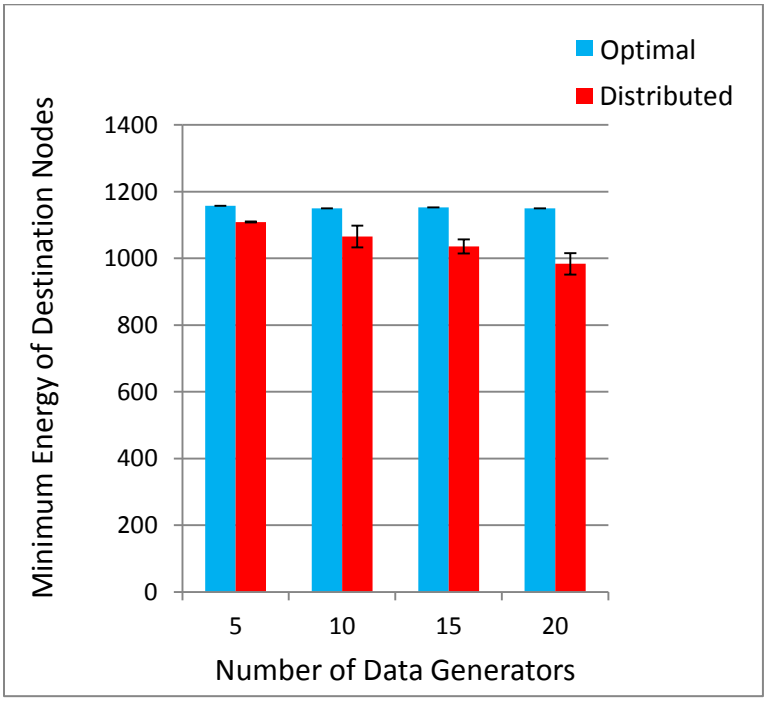

Figure 10: Minimum remaining energy of destination nodes in 15x15 network.

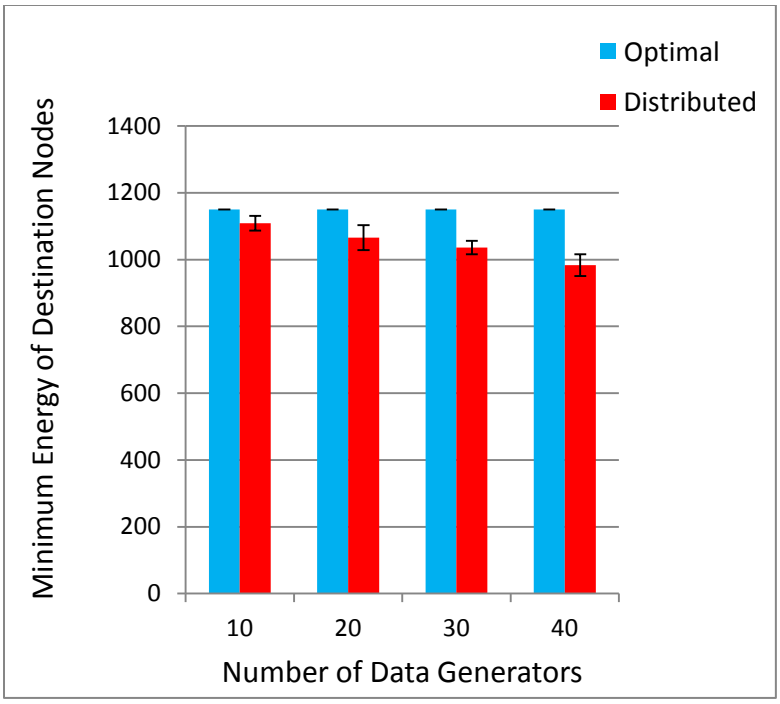

Figure 11: Minimum remaining energy of destination nodes in 20x20 network.

We have the following observations from Figure 9, 10 and 11. When the network size and the number of DGs are not large, the Distributed performs very close to Optimal. When either the number of DG gets larger, the Distributed does not perform as well as the Optimal because of the message overhead incurred in any distributed algorithm.

6.3 Average Remaining Energy Level of Destination Nodes.

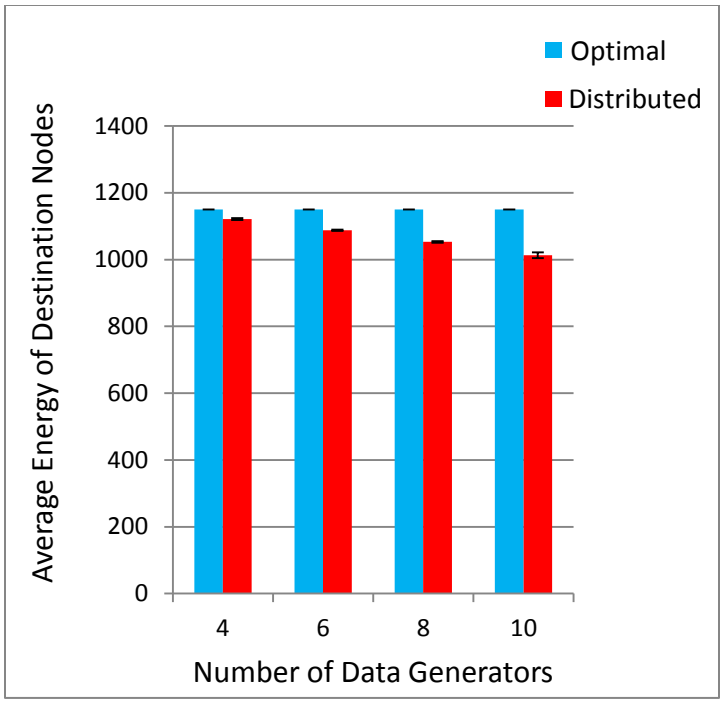

Figure 12: Average energy of destination nodes in 10x10 network.

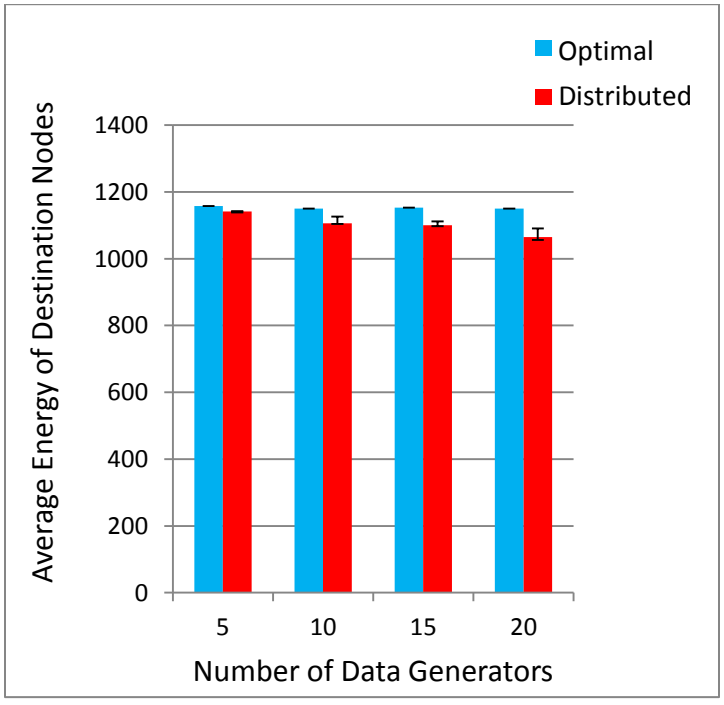

Figure 13: Average energy of destination nodes in 15x15 network.

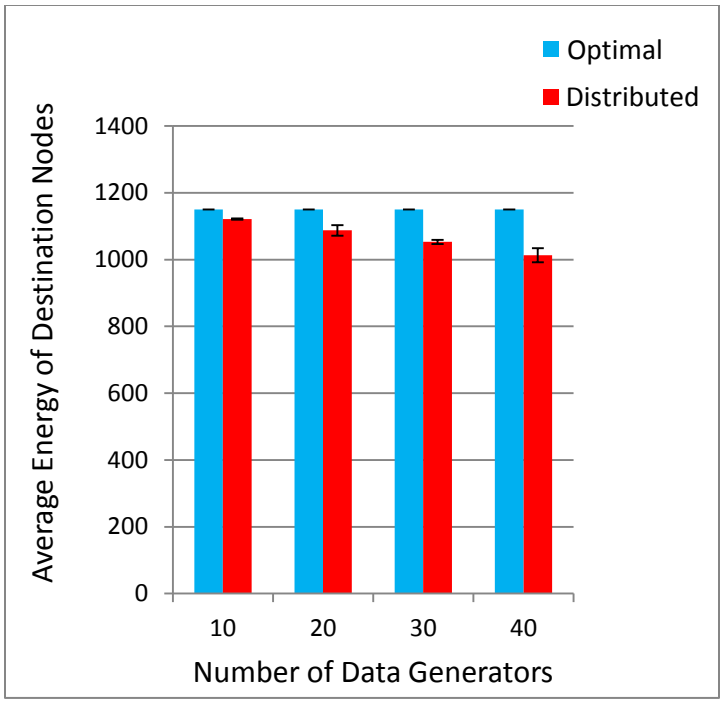

Figure 14: Average energy of destination nodes in 20x20 network.

The average remaining energy of all the destination nodes (or average data preservation time) equals to the sum of the remaining energy of all the destination nodes divided by the number of destination nodes. This is an important indicator for data preservation schemes, since it demonstrates each algorithm's ability to preserve all the data items in the network. Figure 12, 13 and 14 show again that all the algorithms have comparable performances in most cases. We conclude the same as result in previous section, which is that Distributed performs close to Optimal.

6.4 Total Energy Consumption of All Nodes.

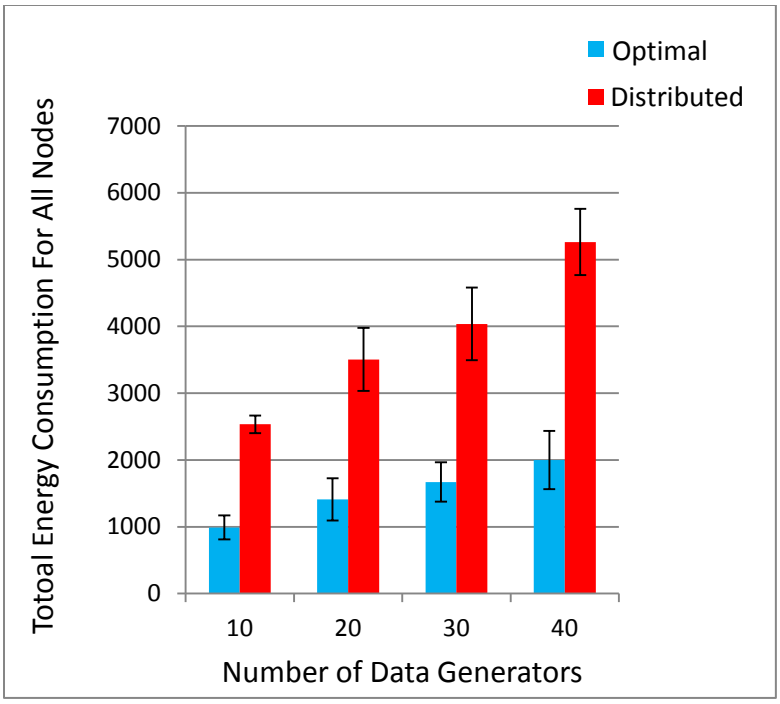

Figure 15: Total energy consumption for all nodes in 10x10 network.

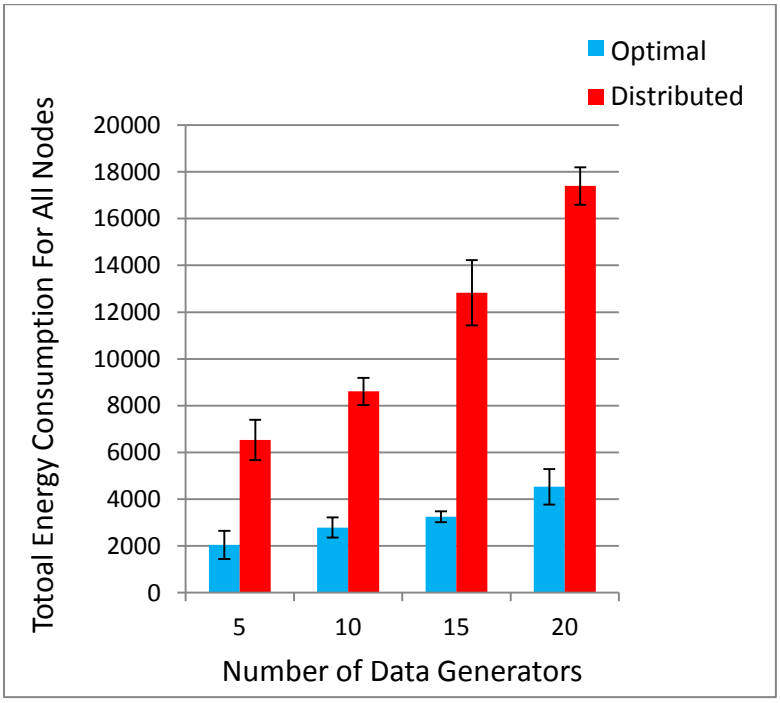

Figure 16: Total energy consumption for all nodes in 15x15 network.

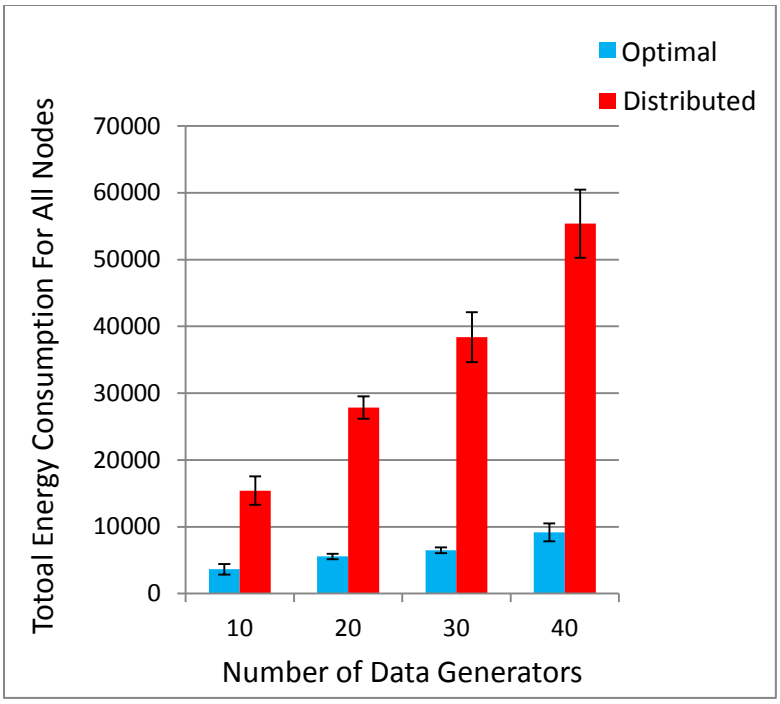

Figure 17: Total energy consumption for all nodes in 20x20 network.

Finally we investigate the total energy consumption in the entire network. From Figure 15, 16 and 17, we find that the total energy consumption of the Distributed is always two to four times larger than Optimal with increasing the size of network and the number of DGs. This reason is caused by the nature of any of distributed algorithms. Since we only focus on individual sensor node in distributed algorithm rather than having prerequisite knowledge about the entire network in centralized algorithm, communicating with other sensors to acquire part of information from the entire network is necessary procedure for each individual sensor nodes, and then it takes plenty of energy consumption for each sensor nodes.

CHAPTER 7

CONCLUSION AND FUTURE WORK

We formulate and solve the SDP as a graph-theoretic problem in intermittently connected sensor networks, which is a new problem that has not attracted much attention. The intermittently connected sensor network is inspired by an emerging new array of disconnection-tolerant sensor network applications, which are also a relatively new research field. Firstly, we show that the feasibility problem is equivalent to the maximum flow problem on an appropriately transformed flow network, and modify Edmonds-Karp [11] to solve it. In addition, we prove that The SDP is NP-hard and apply MEA on another transformed graph to achieve the optimal solution under feasible case when certain conditions are satisfied. Finally, we propose a distributed algorithm that performs close to optimal solution via several simulations, and then discuss its optimality.

In future, we would like to study both centralized algorithm and distributed algorithm under the infeasible case. 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. Finally, we would like to combine data sensing, data preservation and data retrieval into one framework, where energy

and storage dynamics and the interaction among the three pose more comprehensive and challenging problems.

REFERENCES

REFERENCES

[1] 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.

[2] 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).

[3] 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.

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

[5] 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.

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

[7] 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.

[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] 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.

[10] 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.

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

[12] V. Potdar, A. Sharif, E. Chang, "Wireless Sensor Networks: A Survey," waina, International Conference on Advanced Information Networking and Applications Workshops, pp.636-641, UK, 2009.

[13] S. A. Camtepe and B. Yener, “Key distribution mechanisms for wireless sensor networks: a survey,” Rensselaer Polytechnic Inst., Comput. Sci. Dept., Troy, NY, Tech. Rep. TR-05-07, 2005.

[14] U. Acharya and M. Younis, “Increasing Base-Station Anonymity in Wireless Sensor Networks,” Journal of Ad-hoc Network, Vol. 8, No. 8, pp. 791-809, November 2010.

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

[16] Marco Valero, Mingsen Xu, Nicholas A Mancuso, Wen-Zhan Song, and Raheem Beyah. "EDR2: A Sink Failure Resilient Approach for WSNs." To appear in the proceedings of IEEE International Communications Conference (ICC), June 2012.

[17] Masaaki Takahashi, Bin Tang, and Neeraj Jaggi. Energy-effcient 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.

[18] Liu Peng, Zhang Song, Qiu Jian, Shen Xingfa and Zhang Jianhui. A redistribution method to conserve data in isolated energy-harvesting sensor networks, Computer Science and Information Systems Volume 8, Issue 4, Pages: 1009-1025, 2011.

[19] 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.

[20] J. Aslam, Q. Li, and D. Rus. Three power-aware routing algorithms for sensor network. Wireless Comm. and Mobile Computing, 1:187–208, 2003.

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

[22] K. Kar, M. Kodialam, T. Lakshman, and L. Tassiulas. Routing for network capacity maximization in energy-constrained ad-hoc networks. In Proc. of the IEEE INFOCOM 2003.

[23] 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.

[24] 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.

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

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

[27] 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.

[28] 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.

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