Data Preservation in Base Station-Less Sensor Networks: A Game Theoretic Approach

Yutian Chen[1(✉)] and Bin Tang[2]

[1] Economics Department, California State University Long Beach, Long Beach, USA
Yutian.Chen@csulb.edu
[2] Computer Science Department, California State University Dominguez Hills, Carson, USA
btang@csudh.edu

Abstract. We aim to preserve the large amount of data generated inside *base station-less sensor networks* with minimum energy cost, while considering that sensor nodes are selfish. Previous research assumed that all the sensor nodes are cooperative and designed a centralized minimum-cost flow solution. However, in a distributed setting wherein energy- and storage-constrained sensor nodes are under different control, they could behave selfishly, only to maximize their own benefit. In this paper, we take a game theoretic approach and design a computationally efficient data preservation game. We show that in our game, individual sensor nodes, motivated solely by self-interest, achieve good system-wide data preservation solution.

Keywords: Sensor networks · Data preservation · Energy-efficiency · Game theory

1 Introduction

Sensor networks are ad hoc multi-hop wireless networks formed by a large number of low-cost sensor nodes with limited battery power, storage spaces, and processing capacity. Wireless sensor networks have been used in a wide range of applications such as military surveillance, environmental monitoring, and target tracking [22]. Recently, some of the emerging sensor networks are deployed in challenging environments such as in remote or inhospitable regions, or under extreme weather, to continuously collect large volumes of data for a long period of time. Such emerging sensor networks include seismic sensor networks [5], underwater or ocean sensor networks [10,17,21], wind and solar harvesting [3,11], and volcano eruption monitoring and glacial melting monitoring [12,20].

In the above scenarios, it is not practical to deploy data-collecting base stations with power outlets in or near such inaccessible sensor fields. Due to the absence of the base stations, these sensor networks are referred to as *base station-less sensor networks*. Sensory data generated therefore have to be stored inside the network for some unpredictable period of time and then being collected by

J. Cheng et al. (Eds.): GameNets 2016, LNICST 174, pp. 13–23, 2017.
DOI: 10.1007/978-3-319-47509-7_2

periodic visits of robots or data mules [16], or by low rate satellite link [6]. In particular, some sensor nodes are close to the events of interest and are constantly generating sensory data, depleting their own storage spaces. We refer to the sensor nodes with depleted storage spaces while still generating data as *source nodes*. The newly generated data that can no longer be stored at source nodes is called *overflow data*. To avoid data loss, overflow data is offloaded to sensor nodes with available storages (referred to as *storage nodes*). We call this process *data preservation in base station-less sensor networks*.

Since wireless communication consumes most of the battery power of sensor node, the key challenge is how to conserve sensors' battery power by minimizing the total energy consumption in data preservation. Tang et al. showed that this problem is equivalent to minimum cost flow problem [18], which can be solved optimally and efficiently [2]. Two fundamental assumptions are needed for the optimal data preservation algorithm in Tang et al. to work. First, the work assumes that all the storage nodes are selfless in the sense that they are willing to contribute their battery power and storage spaces to help offloading and storing the overflow data from the source nodes. Second, the optimal algorithm depends on full observability of the data preservation costs of each storage node, including the cost of relaying and storing the overflow data.

In this work, we tackle the data preservation problem when sensor nodes are selfish and are in lack of incentive to contribute to data preservation. Two reasons make it important to view sensor nodes as selfish players. First, sensor nodes are generally resource-constrained, with very limited amount of hardware resources including battery power, storage capacity, and processing power. Such resource constraints give sensor nodes minimum or zero motivation to be an altruistic player in data preservation. Second, in a large scale distributed sensor networks, sensor nodes could be under the control of different users or controllers, each of which pursues their own self-interest in the network. Under above scenarios sensor nodes can behave selfishly only to maximize their own benefit.

When sensor nodes are selfish, those assumptions in Tang et al. are no longer valid. First, in order to conserve their own battery power and storage spaces, the storage nodes will choose not to spend their energy and storage resources to help the source nodes to preserve the overflow data, obstructing the entire data preservation process. Second, the associated costs of data preservation of each storage node are normally private information which are not directly observed by outsiders. Due to selfishness, the storage nodes are in lack of incentive to truthfully report their costs. The reason is that each storage node needs to be paid in order to be motivated to participate in data preservation. Nonetheless through lying about its associated cost of data preservation, the storage node may successfully induce a data preservation path which generates itself a higher payoff compared to the payoff when it tells the truth. Such lying behavior of the storage nodes out of their selfishness clearly makes the data preservation path in Tang et al. inefficient. Therefore, with selfish sensor nodes, the challenge is to achieve good system performance, i.e., efficient data preservation with minimum energy cost, while still accommodating selfishness of the sensor nodes.

In this paper, we address the above challenge by utilizing the technique of **algorithmic mechanism design (AMD)** [13–15], a subfield of microeconomics and game theory. The goal of AMD is desirable – it designs computationally efficient game (including strategies and payoffs) such that individual players, motivated solely by self-interest, achieve good system-wide solution. We design computationally efficient data preservation game, in which a payment model is presented to compensate selfish nodes for participating in the data preservation. The payment to each node is designed in a way such that the following two purposes are achieved: first, each node, understanding how the payments are calculated, finds it optimal truthfully reporting its private cost information. Second, based on the reported cost of each node, the payment can sufficiently motivate each node who is involved in the optimal data preservation path calculated in Tang et al. to actually participate in data preservation. With these two goals achieved, the payment model in our game leads to good system-wide data preservation solution with each sensor node motivated solely by self-interest.

2 Data Preservation Problem

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 sensor nodes and E is the set of m edges. The sensory data are modeled as a sequence of data packets, each of which is a bits. Some sensor nodes are close to the event of interest and generate large amount of data packets and deplete their storage spaces; they are referred to as *source nodes*. WLOG there are k source nodes $V_s = \{1, 2, ..., k\}$. The rest nodes in $V - V_s$ are referred to as *storage nodes*. Let d_i denote the number of overflow data packets source node i generates. Because of the storage depletion of the source nodes, the overflow data packets must be offloaded from their source nodes to some storage nodes to be preserved. Let $d = \sum_{i=1}^{k} d_i$ be the total number of overflow data packets, and let $D = \{D_1, D_2, ..., D_d\}$ denote the set of these d data packets. Let $s(j) \in V_s$, $1 \le j \le d$, denote D_j's source node. Let m_i be the available free storage space (in bits) at sensor node $i \in V$. If $i \in V_s$, then $m_i = 0$, implying that a source node is storage-depleted and thus has zero available storage space. If $i \in V - V_s$, then $m_i \ge 0$, implying that a storage node i can store another m_i bits of data packets. We assume that $\sum_{i=k+1}^{n} m_i \ge d \cdot a$, that is, the total size of the overflow data packets can be accommodated by the total available storage spaces.

<u>Energy Model.</u> We consider three different kinds of energy consumptions incurred in data preservation.

- *Transmitting Energy* $E_i^t(j)$. When node i sends a data packet of a bits to its one-hop neighbor j over their distance $l_{i,j}$, the amount of *transmitting energy* spent by i is $E_i^t(j) = a \cdot \epsilon_i^a \cdot l_{i,j}^2 + a \cdot \epsilon_i^e$. Here, ϵ_i^a is energy consumption of sending one bit on transmit amplifier of node i, and ϵ_i^e is energy consumption of transmitting one bit on the circuit of node i.

- *Receiving Energy* E_i^r. When node i receives an a-bit data packet from one of its one-hop neighbor, the amount of *receiving energy* it spends is $E_i^r = a \cdot \epsilon_i^e$. Here, ϵ_i^e is energy consumption of receiving one bit on the circuit of node i. Note that E_i^r does not depend on the distance between nodes.
- *Storing Energy* E_i^s. When node i stores a-bit data into its local storage, the amount of *storing energy* it consumes is $E_i^s = a \cdot \epsilon_i^s$. Here ϵ_i^s is the energy consumption of storing one bit at node i.

Problem Formulation. Define a *preservation function* as $p : D \to V - V_s$, indicating that a data packet $D_j \in D$ is offloaded from its source node $s(j) \in V_s$ to a storage node $p(j) \in V - V_s$ to be preserved. Let $P_j = \{s(j), ..., p(j)\}$ be the *preservation path* along which D_j is offloaded. Let $c_{i,j}$ denote node i's energy consumption in preserving D_j. $c_{i,j}$ can be represented as Eq. 1 below, with $\sigma(i, j)$ being the successor node of i on P_j.

$$c_{i,j} = \begin{cases} E_i^t(\sigma(i,j)) & i = s(j) \\ E_i^r + E_i^s & i = p(j) \\ E_i^r + E_i^t(\sigma(i,j)) & i \in P_j - \{s(j), p(j)\} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The objective is to find a preservation function p and P_j $(1 \le j \le d)$ to minimize the *total preservation cost*, denoted as c, i.e.,

$$c = \min_p \sum_{j=1}^{d} \sum_{i=1}^{n} c_{i,j} = \min_p \sum_{i=1}^{n} \sum_{j=1}^{d} c_{i,j}, \tag{2}$$

under the storage constraint that the total size of data offloaded to storage node i can not exceed i's storage capacity: $|j|1 \le j \le d, p(j) = i| \cdot a \le m_i$, $\forall i \in V - V_s$.

Algorithm. Tang et al. [18] has shown that this problem is equivalent to the minimum cost flow problem in a properly transformed graph of the sensor network graph. The minimum cost flow problem can be solved optimally and efficiently [2]. We adopt and implement the scaling push-relabel algorithm proposed in [1,7]. It has the time complexity of $O(|V|^2|E|\log(|V|C))$, where C is the maximum capacity of an edge in the transformed graph. We denote the algorithm designed in Tang et al. [18] as *the centralized algorithm* to highlight that it minimizes data preservation energy based on the assumption that each node in the network is selfless and therefore fully cooperative.

In this work, we instead consider selfishness of nodes in the sense that each node is maximizing its own interest instead of the system interest. The central problem is to design a mechanism to incentivize selfish nodes to accomplish data preservation as in the centralized algorithm. Note that each source node is obligated to offload its data therefore selfishness does not apply to source nodes. On the other hand, storage nodes are selfish and need to be motivated. However, selfishness of storage nodes can lead to two problems. First, each storage node has no incentive to either relay or store data as either task consumes energy. Therefore, our mechanism needs to pay those storage nodes involved in data preservation path solved from the centralized algorithm, in order to give them incentive to participate in data preservation. The second problem is more subtle but fundamental. The centralized algorithm can figure out the minimum cost data preservation path only based on the assumption that data preservation costs of each storage node are observed. However, some of those cost parameters of each node (given by ϵ_i^e, ϵ_i^a and ϵ_i^s) are private information of each node and may not be directly observed by outsiders. Thus our mechanism needs to induce each node to truthfully report their unobserved cost parameters, so that the centralized algorithm can calculate the minimum cost path based on the reported cost parameters.

3 Algorithmic Mechanism Design (AMD) Approach

The goal of AMD is to design a game in which selfish players maximizing their own utility will choose strategies resulting in the social optimum specified by an optimal algorithm. Here the resulted state is referred to as the *dominant strategy equilibrium/solution*. Dominant strategy of a player is a strategy always maximizing his utility regardless of the other players' strategies. In a dominant strategy solution, each player is playing his dominant strategy. Note that a dominant strategy solution is also a Nash equilibrium since no player has an incentive to deviate from its strategy unilaterally. The challenge in the data preservation problem is to design utility function so that truthfully reporting its cost parameter is a *dominant strategy* to each storage node. Below we first introduce the concepts and notations of the AMD model. We then present the payment model, and prove that under this payment model, acting truthfully (that is, telling its true energy cost involved in data preservation) is each node's dominant strategy.

The AMD Model. There are n nodes in the network - node i has some private information t_i, called its type. There is an *output specification* that maps each type vector $t = \{t_1, ..., t_n\}$ to some output o. Node i's cost is given by *valuation function* $v_i(t_i, o)$, which depends on t_i as well as o. A *mechanism* defines for each node i is a set of strategies A_i. When i plays strategy $a_i \in A_i$, the mechanism computes an *output* $o = o(a_1, ..., a_n)$ and a *payment vector* $p = (p_1, ..., p_n)$, where $p_i = p_i(a_1, ..., a_n)$. Node i wants to maximize its utility function $\pi_i(a_1, .., a_n) = v_i(t_i, o) + p_i$.

There are three observations of the total preservation cost (Eq. 2). First, the total cost is the sum of all participating nodes' energy costs. We can therefore adopt the Vickrey-Groves-Clark (VCG) mechanism [4,9,19]. VCG mechanism applies to mechanism design optimization problems where the objective function is simply the sum of all agents' valuations, and it guarantees that each agent plays truthfully by reporting its true valuation [13]. Second, to minimize the total preservation cost for all the data packets, it only needs to minimize the preservation cost for each data packet D_j, given its source node and destination node. Therefore, our payment model focuses on only one packet, say D_j. Third, in the context of data preservation, $c_{i,j}$ is essentially the private information held by storage node i. To each storage node i, its strategy set includes to truthfully report its cost parameter (therefore $c_{i,j}$) or to lie about its cost parameter. That is, $(c_{i,1}, c_{i,2}, ..., c_{i,d}) \in A_i$ and $v_i(t_i, o) = -c_{i,j}$ for any i. Therefore i's utility is $\pi_i = p_i - c_{i,j}$.

Payment and Utility Model. Below we present the payment and utility model. Since we focus on any data packet D_j (and its preservation path P_j), we use c_i instead of $c_{i,j}$ to denote node i's true cost. p_i is the payment made to node i in order to motivate it to participate the data preservation, $\pi_i = p_i - c_i$. Let c_{-i} denote the strategies of all other nodes except node i.

Definition 1 Payment and Utility. *Based on Green and Laffont [8], under VCG mechanism, given any cost $\tilde{c}_i$ reported by node i, the amount of payment given to node i depends on whether node i is chosen to participate in data preservation according to the centralized algorithm. Its payment is* 0 *if it is not chosen; and its payment when it is chosen is:*

$$p_i(\tilde{c}_i, c_{-i}) = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i), \tag{3}$$

where $c_{V-\{i\}}$ is the minimum total cost of the preservation path that does not go through i; $\tilde{c}_V$ is the minimum total cost of the preservation path that goes through i, when i reports its cost $\tilde{c}_i$. Therefore i's utility is 0 *when it is not chosen by the centralized algorithm; and when i is chosen, its utility is*

$$\pi_i(\tilde{c}_i, c_{-i}) = p_i(\tilde{c}_i, c_{-i}) - c_i = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i) - c_i, \tag{4}$$

where c_i is node i's true cost. Moreover, we define c_V as the minimum total cost of the preservation path that goes through i when i truthfully reports its cost, i.e., when $\tilde{c}_i = c_i$. □

Time complexity of the payment model. The time taken to compute the payment model is the time taken for the minimum cost flow calculation, which is $O(|V|^2|E|\log(|V|C))$, where C is the maximum capacity of an edge in the transformed graph [1,7]. Under this model, the amount of payment given to a specific node i equals the total minimum cost of all the participating nodes when i does not participate minus all other participating nodes' cost when i participates. The rationale is that a node can be motivated to participate if it is paid its share of contribution, which in our case, is the amount of preservation energy this node helps to reduce when it participates.

An implication here is that the payment and utility model is common knowledge to each node. That is, each node understands that based on their reported cost types and the corresponding data preservation path calculated by the centralized algorithm, their payment and utility are given by (3) and (4), respectively. The timing of the game among the source nodes is given below.

Definition 2 Timing of the Game. *The game unfolds as follows. In stage 1, each storage node reports its private type c_i. In stage 2, the centralized algorithm is applied based on reported cost types to calculate the minimum cost data preservation path. In stage 3, each of the storage nodes chosen in the path chooses to* *participate in data preservation or not. If they participate, they realize the data preservation cost and also the payment given by Eq. (3), and each gets utility given by Eq. (4).* □

Note that each storage node moves only in stages 1 and 3, when each chooses how much to report for private type and whether to participate in data preservation based on the corresponding payment. Stage 2 is non-strategic: in the absence of base stations, the centralized algorithm is provided by an outsider of the system, and it cannot be enforced in the system by the outsider. Since there is a time sequence between the two decisions of each node in stage 1 and stage 3, the solution concept of the game is subgame perfect Nash equilibrium (SPNE). SPNE is a Nash equilibrium (NE) in which players are doing NE in every subgame of the whole game tree.

Assumptions. We assume that the source nodes are obliged to offload their overflow data packets to other storage nodes, thus need not to be motivated. Therefore, their types are known public knowledge, and they will be reimbursed according to true costs they entail. For storage node i that participates in the preservation of a specific data packet, it incurs one of the two costs below:

- *Relaying Cost* $c_i^r(j)$. When node i receives a data packet and then sends it to one of its one-hop neighbor j over their distance $l_{i,j}$, its *relaying cost*, denoted as $c_i^r(j)$, is the sum of its receiving energy and transmitting energy. That is $c_i^r(j) = E_i^r + E_i^t(j) = 2 \cdot a \cdot \epsilon_i^e + a \cdot \epsilon_i^a \cdot l_{i,j}^2$.
- *Storing Cost* c_i^s. When node i receives a data packet and then stores it into its storage, its *storing cost*, denoted as c_i^s, is the sum of its receiving energy and its storing energy. That is, $c_i^s = a \cdot \epsilon_i^e + a \cdot \epsilon_i^s$.

Note that node i has three energy parameters: ϵ_i^e, ϵ_i^a, and ϵ_i^s. Among them, ϵ_i^e affects both $c_i^r(j)$ and c_i^s, while ϵ_i^a only affects $c_i^r(j)$ and ϵ_i^s only affects c_i^s. Next, we will study the AMD model wherein for each node i, either ϵ_i^a or ϵ_i^s or ϵ_i^e is the private type of node i not directly observed by the public. Since ϵ_i^a and ϵ_i^s each only affects one cost or the other, we study $t_i = \epsilon_i^a$ or $t_i = \epsilon_i^s$ first.

3.1 AMD When $t_i = \epsilon_i^a$ or $t_i = \epsilon_i^s$

We focus on $t_i = \epsilon_i^a$ since $t_i = \epsilon_i^s$ can be studied similarly. Below we give a detailed proof that under above VCG payment model, for each node i, truth-telling (reporting its true type t_i) is a dominant strategy. We define $c_{-i}^r = \{c_1^r, ..., c_{i-1}^r, c_{i+1}^r, ..., c_n^r\}$ as the cost vector of other nodes except i. Since the optimal minimum cost flow algorithm determines i's successor node j, and for ease of notation, we use c_i instead of $c_i^r(j)$ to represent node i's relaying cost, and use $\tilde{c}_i$ instead of $\tilde{c_i^r}(j)$ in the following theorem and proof.

Theorem 1: *For any node i, suppose $t_i = \epsilon_i^a$ (that is, ϵ_i^a is i's private type). Reporting its true type ϵ_i^a is node i's dominant strategy. That is, $\pi_i(c_i, c_{-i}) \geq \pi_i(\tilde{c}_i, c_{-i})$, $\forall\ \tilde{c}_i \neq c_i$ and $\forall\ c_{-i}$.*

Proof: We consider that node i either reports truthfully or not. Under either case, node i could be chosen to participate in the data preservation or not according to the centralized algorithm. Therefore there are all together four cases. Below we show that $\pi_i(c_i, c_{-i})$ is always greater or equal to $\pi_i(\tilde{c_i}, c_{-i})$ in all the four cases.

Case I: Node i is in the preservation path when reporting either c_i or $\tilde{c_i}$. Thus the payment of i when it reports truthfully is $\pi_i(c_i, c_{-i}) = c_{V-\{i\}} - (c_V - c_i) - c_i = c_{V-\{i\}} - c_V$. On the other side, when it lies by reporting $\tilde{c_i}$, its payoff is $\pi_i(\tilde{c_i}, c_{-i}) = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c_i}) - c_i = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c_i} + c_i) = c_{V-\{i\}} - c_V$. Therefore in this case $\pi_i(c_i, c_{-i}) = \pi_i(\tilde{c_i}, c_{-i})$. Note that $\pi_i(c_i, c_{-i}) \geq 0$ because $c_{V-\{i\}} - c_V \geq 0$.

Case II: Node i is in the preservation path when reporting c_i, which implies that $c_{V-\{i\}} \geq c_V$; and it is not in the preservation path when reporting $\tilde{c_i}$, which gives payoff $\pi_i(\tilde{c_i}, c_{-i}) = 0$. Thus its payoff under truth-telling is $\pi_i(c_i, c_{-i}) = c_{V-\{i\}} - c_V \geq 0$. In this case $\pi_i(c_i, c_{-i}) \geq \pi_i(\tilde{c_i}, c_{-i})$.

Case III: Node i is not in the preservation path when reporting c_i, which gives $\pi_i(c_i, c_{-i}) = 0$ and also implies that $c_{V-\{i\}} \leq c_V$. However, it is in the preservation path when reporting $\tilde{c_i}$. Its payoff when it lies is $\pi_i(\tilde{c_i}, c_{-i}) = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c_i}) - c_i = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c_i} + c_i) = c_{V-\{i\}} - c_V \leq 0$. Therefore in this case $\pi_i(c_i, c_{-i}) \geq \pi_i(\tilde{c_i}, c_{-i})$.

Case IV: Node i is not in the preservation path when reporting either c_i or $\tilde{c_i}$. In this case $\pi_i(c_i, c_{-i}) = \pi_i(\tilde{c_i}, c_{-i}) = 0$.

Since $\pi_i(c_i, c_{-i}) \geq \pi_i(\tilde{c_i}, c_{-i})$ holds regardless of other nodes' strategy c_{-i} under all the cases, we conclude that reporting its true cost c_i is node i's dominating strategy.

Theorem 2. *With the payment given by (3), when ϵ_i^a or ϵ_i^s is unobserved, there exists SPNE of the game, in which every storage node i truthfully reports its cost type in stage 1. Moreover, in stage 3 each node i chosen by the centralized algorithm for data preservation will participate.*

Proof: In stage 3, each storage node chosen by the centralized algorithm will participate as long as its utility (i.e., payoff) is no less than zero. When we move back to stage 1, by Theorem 1, each storage node has a dominant strategy, which is to truthfully report the type. Therefore, the Nash equilibrium in stage 1 is that each node truthfully reports its type. Since each node in the data preservation path gets a non-negative utility (see the proof of Theorem 1), each will choose to participate in stage 3. We conclude that the strategy given in this theorem constitutes a SPNE. ∎

3.2 AMD When $t_i = \epsilon_e^i$

When ϵ_e^i is the unknown type of source node i, the reported value of the type affects the two costs simultaneously: the relaying cost $c_i^r(j)$ and the storing cost c_i^s. The complication here is that by lying about its type, node i might switch its role in data preservation from one task to a different task. For example, node

i might be assigned to relay the data packet according to ϵ_e^i, its true cost type; but by reporting $\tilde{\epsilon}_e^i \neq \epsilon_e^i$, node i might instead be assigned to store the data. It is not clear whether VCG can continue to apply in this case or not. To examine the situation when ϵ_e^i is the unknown type of node i, we first denote c_V^{is} as the minimum total cost of data preservation given that node i stores the data packet; and c_V^{ir} as the minimum total cost of data preservation given that node i relays the data packet. Note that $c_{V-\{i\}}$ is the minimum total cost of data preservation given that node i does not participate in data preservation. The following theorem shows that the basic idea of VCG continues to hold. Since the optimal minimum cost flow algorithm determines i's successor node j, and for ease of notation, we use c_i to represent $c_i^r(j)$ or c_i^s, and use $\tilde{c}_i$ to represent $\tilde{c_i^r}(j)$ or $\tilde{c_i^s}$ in the following theorem and proof.

Theorem 3. *For any node i, suppose $t_i = \epsilon_i^e$ (that is, ϵ_i^e is i's private type). Reporting its true type ϵ_i^e is node i's dominant strategy. That is, $\pi_i(c_i, c_{-i}) \geq \pi_i(\tilde{c}_i, c_{-i})$, $\forall\ \tilde{c}_i \neq c_i$ and $\forall\ c_{-i}$.*

Proof: Based on the reported ϵ_i^e of node i, node i could be chosen to participate in the data preservation or not according to the centralized algorithm. If it is chosen, it may be designated to either transmit or store the data packet. We need to show that regardless of other nodes' reported cost types, telling truth is always the optimal strategy of node i. There are in together six cases and we show that $\pi_i(c_i, c_{-i})$ is always greater or equal to $\pi_i(\tilde{c}_i, c_{-i})$ in all the six cases.

Case I. Node i is in the preservation path to relay the data packet when reporting either c_i or $\tilde{c}_i$.

Case II. Node i is in the preservation path to store the data packet when reporting either c_i or $\tilde{c}_i$.

Case III. Node i is in the preservation path to relay the data packet when reporting c_i and is doing nothing when reporting $\tilde{c}_i$.

Case IV. Node i is in the preservation path to store the data packet when reporting either c_i or $\tilde{c}_i$.

Proof for $\pi_i(c_i, c_{-i}) \geq \pi_i(\tilde{c}_i, c_{-i})$ for the four cases are similar as in the proof of Theorem 1 and are omitted. We focus on the following two cases.

Case V. Node i is in the preservation path to relay the data packet when reporting c_i and is in the preservation path to store the data packet when reporting $\tilde{c}_i$. This implies that $c_V^{is} \geq c_V^{ir}$. When node i reports c_i, its payoff is $c_{V-\{i\}} - c_V^{ir} + c_i - c_i = c_{V-\{i\}} - c_V^{ir}$. When node i reports $\tilde{c}_i$, its payoff is $c_{V-\{i\}} - \tilde{c}_V^{is} + \tilde{c}_i - c_i = c_{V-\{i\}} - c_V^{is}$. It follows that $c_{V-\{i\}} - c_V^{ir} \geq c_{V-\{i\}} - c_V^{is}$.

Case VI. Node i is in the preservation path to store the data packet when reporting c_i and is in the preservation path to relay the data packet when reporting $\tilde{c}_i$. This implies that $c_V^{is} \leq c_V^{ir}$. When node i reports c_i, its payoff is $c_{V-\{i\}} - c_V^{is} + c_i - c_i = c_{V-\{i\}} - c_V^{is}$. When node i reports $\tilde{c}_i$, its payoff is $c_{V-\{i\}} - \tilde{c}_V^{ir} + \tilde{c}_i - c_i = c_{V-\{i\}} - c_V^{ir}$. It follows that $c_{V-\{i\}} - c_V^{is} \geq c_{V-\{i\}} - c_V^{ir}$. ∎

Theorem 4: *With the payment given by (3), when ϵ_e^i is unobserved, there exists SPNE of the game, in which every source node i truthfully reports its cost type* *in stage 1. Moreover, in stage 3 each node i chosen by the centralized algorithm for data preservation will participate.*

Proof: It follows the same argument as the proof of Theorem 2 and is omitted here. ∎

4 Conclusion and Future Work

In this work, we study data preservation problem in base station-less sensor networks wherein energy- and storage-constrained sensor nodes behave selfishly. We take a game theoretic approach and design a payment model under which the individual sensor nodes, motivated solely by self-interest, achieve good system-wide data preservation solution. In particular, we break down the data preservation cost of each storage node into two parts: relaying cost and storing cost, where cost parameters are node-dependent. The payment model is designed in a way such that no matter which cost parameter (related only to the relaying cost or only to the storing cost or to both) is private to the node, truthfully reporting the cost parameter is a dominant strategy to each node. We show that as a result, in the game it is an equilibrium that each storage node first truthfully reports its cost parameter, then participates in data preservation if it is chosen by the centralized data preservation algorithm.

In the next step of the work, we will validate theoretical findings using simulation results. By contrasting the payment of each storage node in the sensor network under truth-telling strategy to what it is under lying, we will show that truth-telling is never worse off and in certain cases is strictly better off to each storage node regardless of the choice of the other nodes. The simulation results thus can verify that truth-telling is a dominant strategy of each source node. Other future work includes relaxing some assumptions in the current work. In particular, we have assumed that data preservation is feasible in the sensor network, i.e., all the nodes have enough energy to offload and preserve all the overflow data packets. If instead the network is infeasible so that some data packets will inevitably be lost, it is interesting to see how the payment model can work to induce the efficient data preservation. Finally, we will extend our analysis to a dynamic scenario wherein overflow data are generated from time to time at different nodes. It is well understood in game theory that an infinitely repeated game gives a much larger set of equilibrium and in certain scenarios full cooperation can be achieved. In our setting of data preservation among selfish nodes, it is interesting to see to what extent we need to provide motivation for selfish storage node to engage in the optimal data preservation.

Acknowledgment. This work was supported in part by NSF Grant CNS-1116849.

References

1. Andrew goldberg's network optimization library. http://www.avglab.com/andrew/soft.html
2. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice-Hall Inc., Englewood Cliffs (1993)
3. Cammarano, A., Petrioli, C., Spenza, D.: Pro-energy: a novel energy prediction model for solar and wind energy-harvesting wireless sensor networks. In: IEEE 9th International Conference on Mobile Adhoc and Sensor Systems (MASS 2012)
4. Clarke, E.H.: Multipart pricing of public goods. Public Choice (1971)
5. Cochran, E., Lawrence, J., Christensen, C., Chung, A.: A novel strong-motion seismic network for community participation in earthquake monitoring. IEEE Inst. Meas. **12**(6), 8–15 (2009)
6. Colitti, W., Steenhaut, K., Descouvemont, N., Dunkels, A.: Satellite based wireless sensor networks: Global scale sensing with nano- and pico-satellites. In: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, SenSys 2008, pp. 445–446 (2008)
7. Goldberg, A.V.: An efficient implementation of a scaling minimum-cost flow algorithm. J. Algorithms **22**, 1–29 (1997)
8. Green, J., Laffont, J.: Incentives in public decision making. Stud. Public Econ. **1**, 65–78 (1979)
9. Groves, T.: Incentives in teams. Econometrica (1973)
10. Li, S., Liu, Y., Li, X.: Capacity of large scale wireless networks under gaussian channel model. In: Proceedings of MOBICOM 2008 (2008)
11. Li, Y., Li, X., Wang, P.: A module harvesting wind and solar energy for wireless sensor node. In: Wang, R., Xiao, F. (eds.) CWSN 2012. CCIS, vol. 334, pp. 217–224. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36252-1_20
12. Martinez, K., Ong, R., Hart, J.K.: Glacsweb: a sensor network for hostile environments. In: Proceedings of SECON 2004 (2014)
13. Nisan, N.: Algorithms for selfish agents: mechanism design for distributed computation. In: Meinel, C., Tison, S. (eds.) STACS 1999. LNCS, vol. 1563, pp. 1–15. Springer, Heidelberg (1999). doi:10.1007/3-540-49116-3_1
14. Nisan, N., Ronen, A.: Algorithmic mechanism design. In: Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing (STOC 1999), pp. 129–140 (1999)
15. Nisan, N., Ronen, A.: Algorithmic mechanism design. Games Econ. Behav. **35**, 166–196 (2007)
16. Shah, R.C., Roy, S., Jain, S., Brunette, W.: Data mules: modeling a three-tier architecture for sparse sensor networks. In: Proceedings of SNPA 2003 (2003)
17. Syed, A.A., Ye, W., Heidemann, J.: T-lohi: a new class of mac protocols for underwater acoustic sensor networks. In: Proceedings of INFOCOM 2008 (2008). http://www.isi.edu/ilense/snuse/index.html
18. Tang, B., Jaggi, N., Wu, H., Kurkal, R.: Energy-efficient data redistribution in sensor networks. ACM Trans. Sen. Netw. **9**(2), 11: 1–11: 28 (2013)
19. Vickrey, W.: Counterspeculation, auctions and competitive sealed tenders. J. Finan. **16**, 8–37 (1961)
20. Werner-Allen, G., Lorincz, K., Johnson, J., Lees, J., Welsh, M.: Fidelity and yield in a volcano monitoring sensor network. In: Proceedings of OSDI 2006 (2006)
21. Xiao, Y.: Underwater Acoustic Sensor Networks. Auerbach Publications (2009)
22. Yick, J., Mukherjee, B., Ghosal, D.: Wireless sensor network survey. Comput. Netw. **52**, 2292–2330 (2008)

http://www.springer.com/978-3-319-47508-0

Game Theory for Networks
6th International Conference, GameNets 2016,
Kelowna, BC, Canada, May 11-12, 2016, Revised
Selected Papers
Cheng, J.; Hossain, E.; Zhang, H.; Saad, W.; Chatterjee,
M. (Eds.)
2017, X, 147 p. 40 illus., Softcover
ISBN: 978-3-319-47508-0