Voluntary Data Preservation Mechanism in Base Station-less Sensor Networks

Yutian Chen[1], Jennifer Ly[2], and Bin Tang[2]

[1]Economics Department, California State University Long Beach, Yutian.Chen@csulb.edu
[2]Computer Science Department, California State University Dominguez Hills,
jenniferly95@gmail.com, btang@csudh.edu

June 27, 2022

Abstract

We consider the problem of preserving a large amount of data generated inside *base station-less sensor networks*, when sensor nodes are controlled by different authorities and behave selfishly. We modify the VCG mechanism to guarantee that each node, including the source nodes with overflow data packets, will voluntarily participate in data preservation. The mechanism ensures that each node truthfully reports its private type and network achieves efficiency for all the preserved data packets. Extensive simulations are conducted to further validate our results.

Keywords – VCG Mechanism, Sensor Networks, Data Preservation

1 Introduction

Background. Wireless sensor networks are ad hoc multi-hop wireless networks formed by a large number of low-cost sensor nodes. They have been used in a wide range of applications such as military surveillance, environmental monitoring, and target tracking Yick et al. (2008). Recent years observe emerging wireless sensor networks deployed in challenging environments such as 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 underwater or ocean sensor networks Jang and Adib (2019); Ghaffarivardavagh et al. (2020), wind and solar harvesting Li et al. (2012); Cammarano et al. (2012), volcano eruption monitoring and glacial melting monitoring et al. (2020); Werner-Allen et al. (2006); Martinez et al. (2004), and seismic sensor networks Cochran et al. (2009). In these scenarios, it is not practical to deploy data-collecting base stations with power outlets in or near such inaccessible sensor fields. These sensor networks are referred to as *base station-less sensor networks (BSNs)* because of the absence of the base stations.

A major task of BSNs is that sensory data generated have to be stored inside the network for some unpredictable period of time before being collected by periodic visits of robots or data mules Shah et al. (2003), or by low rate satellite link Colitti et al. (2008). We focus on one scenario as follows. Some sensor nodes are close to the events of interest and are constantly generating sensory data, depleting their own storage spaces. These sensor nodes with depleted storage spaces (referred to as *source nodes*) need to offload their overflow data to sensor nodes with available storage (referred to as *storage nodes*) to avoid data loss. The process of preserving overflow data within the sensor network is called as *data preservation in base station-less sensor networks.*

As sensor nodes are characterized by limited battery power, storage spaces, and processing capacity, and wireless communication consumes most of the battery power of sensor nodes, the key challenge of data preservation in BSNs is to conserve sensors' battery power in this process. Tang et al. (2013) proposed a centralized algorithm to minimize the total energy consumption of data preservation in BSNs. They showed that this problem is equivalent to the minimum cost flow problem, which can be solved optimally and efficiently Ahuja et al. (1993). This centralized algorithm is applicable if all the storage nodes cooperate, in the sense that they willingly contribute their battery power and storage spaces to help to relay and store overflow data from source nodes.

Our Contributions. However, in large-scale distributed sensor networks, sensor nodes could be controlled by different users or authorities, each of which pursues self-interest. Therefore sensor nodes can behave selfishly only to maximize their own benefit. As data preservation unavoidably consumes sensor nodes' limited resources, including battery power and storage capacity, these sensor nodes have minimum or zero motivation to help with data preservation. In addition, data preservation cost parameters of sensor nodes can be specific to the nodes. For example, the type of battery of a sensor node could affect the energy it consumes in data preservation. Due to the complexity of related features among different sensor nodes, parameters of data preservation cost specific to a sensor node should be taken as its private information instead of treated as known by the public. While this cost information is fundamental for a centralized algorithm to configure the efficient data preservation routes, selfish sensor nodes may not have an incentive to report their private cost parameters/types truthfully.

In the sight of issues rooted in selfishness, literature has adopted algorithmic mechanism design (AMD) Nisan (1999); Nisan and Ronen (2007, 1999), a subfield of game theory, to motivate selfish players to participate in the game (individual rationality) and also be truthful about their private type (efficiency). The central idea is to compensate (or charge) each player based on its task accomplished (or objects assigned) in the game. In this paper, we propose a voluntary data preservation mechanism by augmenting existing AMD techniques. In our model, each source node has a value of its overflow data and will choose to preserve the data only when its value exceeds the corresponding preservation cost. Therefore, our model avoids the inefficiency due to the *over preservation* of data, namely, when the data value falls short of its preservation cost but still gets preserved. In addition, our model requires source nodes to compensate storage nodes for help in preserving their data, thus shifting the compensation responsibility from the central authority to the source nodes. Moreover, it guarantees that every single node in the network will *voluntarily* participate in the data preservation game, including the source nodes and the storage nodes. Therefore, by taking into consideration the data value, our model extends individual rationality to the entire network, and improves the network efficiency of data preservation. The central authority in our model serves mainly two functions: it calculates the efficient data preservation route of each data and the compensation of each node; it also serves as a residual claimant, who covers the deficit in the system when the amount paid by the source nodes is below the calculated compensation, or holds the surplus when the opposite is true.

Paper Organization. The rest of the paper is organized as follows. Section 2 reviews all the related work. Section 3 formulates the data preservation problem. Section 4 presents the algorithmic mechanism design approach and presents the voluntary VCG model. Section 5 presents our detail simulation results and analysis. Section 6 concludes the paper with a discussion of future work.

2 Related Work

Data preservation in base station-less sensor networks has been studied extensively. It has been modeled as a minimum cost flow problem to achieve the energy optimization Tang et al. (2013); Crary et al. (2015) as well as fault-tolerance Tang et al. (2014). Xue et al. (2013) designed a maximum weighted flow algorithm to preserve data packets of different values. Some works studied a suite of new multiple traveling salesman placement problems for data aggregation Tang (2018); Tang et al. (2021). Recently, Hsu et al. (2020) designed a quadratic programming solution to maximize the survival time of preserved data packets for base station-less sensor networks. However, none of the above research considered the selfishness of sensor nodes.

Chen and Tang (2016) was one of the first to study data preservation in base station-less sensor networks while considering that sensor nodes are selfish. They took a game-theoretic approach and designed a computationally efficient and truthful Vickrey-Groves-Clark (VCG) mechanism Vickrey (1961); Groves (1973); Clarke (1971); Nisan (1999) wherein truth-telling is always a dominant strategy. Yu et al. (2022) further identified that when storage nodes have a limited amount of energy power, the VCG mechanism proposed in Chen and Tang (2016) is no longer truthful and nodes can manipulate the VCG mechanism in order to gain more utilities. They further designed a data preservation game that traces and punishes manipulative nodes in the BSN and delivers dominant strategies for truth-telling nodes while achieving provably optimal data preservation with cheat-proof guarantees.

However, there are several issues in the VCG model of Yu et al. (2022). First, they assume that all the overflow data will be stored in the network without considering the benefit and cost tradeoff of data preservation. As a result, no matter whether it is very valuable or of little value, each data will be preserved regardless of how much cost its preservation entails. Although total preservation cost is minimized, the outcome may not be truly efficient for the system, as it could take much effort to preserve data with little value. Second, the central authority is in charge of making all the compensation payments to the system, which can be a considerable amount. Meanwhile, the source nodes, as the direct beneficiaries of data preservation, are not responsible for compensating any other node. Third, they assume that the source nodes will participate in data preservation and tell the truth, therefore do not need any motivation and compensation from the central algorithm. However, source nodes are also selfish as participating in data preservation is not costless to them.

Our work is inspired by Eidenbenz et al. (2005), which is a sender-centric truthful ad hoc routing protocol that considers the rational and selfish source nodes. However, in this work, the only private information of a sender is its willingness to pay to establish the connection with the destination, and the sender's action is simply to establish this connection or not. In contrast, in our work, the relationship between a node's cost parameters and its incurred costs is more complicated. By lying about different cost parameters to different extents, a node might manipulate its cost and switch from one action to another.

3 Data Preservation Problem Formulation

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 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 a large amount of data, which depletes their storage spaces. Nodes with overflow data are referred to as *source nodes*. There are nodes in the system with empty storage space for data storage, which are referred to as *storage nodes*. WLOG there are $k < n$ source nodes $V_s = \{1, 2, ..., k\}$ and q storage nodes $V_{ss} = \{k+1, k+2, ..., k+q\}$, with $k + q \leq n$. Thus the rest nodes viz. $V - V_s - V_{ss} = \{k+q+1, k+q+2, ..., n\}$ have neither overflow data nor storage spaces thus can only relay data from source nodes in the data preservation process. We refer to them as *transition nodes*. Note that any node, including the source node, storage node, and transition node, can carry out the function of relaying data.

Let d_i denote the number of overflow data source node i generates, which must be offloaded to some storage nodes to avoid being lost. Let $d = \sum_{i=1}^{k} d_i$ be the total number of overflow data, and let $D = \{1, 2, ..., d\}$ denote the set of these d data, and j a single overflow data. Let $s(j) \in V_s$, $1 \leq j \leq d$, denote data j's source node. Let $m_i > 0$ be the available free storage space (in bits) at storage node $i \in V_{ss}$. Note that $m_i = 0$ for any source or transition node i. We assume that $\sum_{i=1}^{n} m_i \geq d \cdot a$; that is, the total size of the overflow data is not larger than the total available storage spaces in the network. Moreover, all the overflow data has certain values, indicating the importance of the data in a specific application. Let $g_i > 0$ be the value of each overflow data of source node $i \in V_s$.[1]

We assume that each node has enough energy to participate the data preservation process. Three different kinds of energy consumptions are involved in data preservation, as described below:

- *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 $E_i^t(j) = a \cdot \epsilon_i^a \cdot l_{i,j} + a \cdot \epsilon_i^e$. 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

[1] For simplicity, we assume that all the overflow data at the same source node has the same value. Our model is readily extended to the case when a source node has different values over its overflow data.

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. A *preservation function* as $p : D \rightarrow V_{ss}$ indicates that a data packet $j \in D$ is offloaded from its source node $s(j) \in V_s$ to a storage node $p(j) \in V_{ss}$ to be preserved. Let $P_j = \{s(j), ..., p(j)\}$ be the *preservation path* along which j is offloaded. Let $c_{i,j}$ denote node i's energy consumption in preserving j, which be represented as Equation (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 $(1 \leq j \leq d)$ to minimize the *total preservation cost.* The minimized total preservation cost is given by

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

under the storage constraint that the total size of data offloaded to storage node i can not exceed i's storage capacity: $|j|1 \leq j \leq d, p(j) = i| \cdot a \leq m_i$, $\forall i \in V_{ss}$. The corresponding optimal preservation function is p^* and the optimal preservation path of data j is P_j^*.

Algorithm. Tang et al. (2013) 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 Ahuja et al. (1993). We adopt and implement the scaling push-relabel algorithm proposed in Goldberg (1997). It has the time complexity of $O(|V|^2 \cdot |E| \cdot \log(|V| \cdot C))$, where C is the maximum capacity of an edge in the transformed graph. We denote the algorithm designed in Tang et al. (2013) 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.

Discussion. In this work, instead of cooperative sensor notes, we consider the situation when each node is selfish, intended to maximize its own interest instead of the system interest. We design a mechanism to incentivize selfish nodes to comply with efficient data preservation directed by the centralized algorithm. Note that incentives among different types of sensor notes are diversified. First, as a source node $i \in V_s$ wants to offload its overflow data to be preserved, we assume that information related to i is public, including its cost parameters and value g_i of its overflow data. For a source node, it will have its data offloaded so long as the compensation it needs to pay to other helper nodes does not exceed the value it holds for the data. Second, a non-source node (i.e., a storage and transition node) has private information about its cost parameters, denoted as *private type*: $t_i = \{\epsilon_i^a, \epsilon_i^e, \epsilon_i^s\}, \forall i \in V - V_s$. Private types of each node need to be truthfully reported. On the other hand, the cost incurred by each node should be compensated, so that every node will willingly participate in data preservation.

Moreover, as we take into consideration the data value of each source node, the concept of efficiency includes two aspects. The first one is the *preservation efficiency* same as in the previous literature. That is, given the set of data to be offloaded, the total preservation cost in the system shall be minimized. This part can be solved using the centralized algorithm in Tang et al. (2013) under the condition that all the nodes willingly cooperate. The second one is referred to as *data efficiency.* If the data value is below the corresponding minimum preservation cost found by the centralized algorithm, the data should not be preserved. This is a novel feature in our work because it compares the benefit and cost of preserving each data, and thus truly maximizes the net gain from data preservation.

4 Voluntary VCG Mechanism

We first introduce the concepts and notations of the algorithmic mechanism design (AMD) model. The goal of AMD model is to design a game in which selfish players who maximize their own utility have equilibrium strategies that result in a social optimal outcome. If the equilibrium is a *dominant strategy solution*, we say that this mechanism is *strategy proof.*[2]

[2] Dominant strategy of a player is a strategy that maximizes his utility regardless of the other players' strategies. If each player has a dominant strategy, in equilibrium each will play his dominant strategy and the strategy profile

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 $\pi_i(a_1, .., a_n) = v_i(t_i, o) + p_i$.

In the context of data preservation, for $i \in V - V_s$, the private type $t_i = \{\epsilon_i^e, \epsilon_i^a, \epsilon_i^s\}$. Node i's strategy set A_i includes any value of private type t_i it can report. For $i \in V - V_s$, $v_i(t_i, o) = -c_{i,j}$ given by Equation (1), and its utility is $\pi_i(t_i, t_{-i}) = p_i - \sum_{j=1}^{d} c_{i,j}$. I.e., a non-source node's utility is the difference between its received payment and its cost in data preservation. For a source node $i \in V_s$, our model will make it pay $g_i - c_{i,j}$ for each of its preserved data j, leaving node i zero utility from preserving its own data. On the other hand, a source node can relay other source nodes' data and will be compensated according to its cost for performing such tasks. Therefore, its utility is also zero by helping with other nodes' data preservation. Thus the payoff to a source node is zero.

As sensor nodes may have private information about the cost of data preservation, we can adopt the Vickrey-Groves-Clark (VCG) mechanism Vickrey (1961); Groves (1973); Clarke (1971), which is a strategy-proof mechanism Nisan (1999) frequently used when the objective function is the sum of all agents' valuations. Yu et al. (2022) used the VCG mechanism to motivate non-source nodes to participate in data preservation. Compared to Yu et al. (2022), our model will motivate all the nodes, including source nodes, to voluntarily participate in data preservation. As such, we name our mechanism as *voluntary VCG mechanism.* We start by formally defining the payment and utility of each node, denoted as the *payment model.*

4.1 Payment Model

We use c_i to denote node i's true total cost in data preservation, and p_i the total payment to node i. Thus for each node $i \in V$, $\pi_i = p_i - c_i$. Let $t_{-i} = \{t_1, ..., t_{i-1}, t_{i+1}, ..., t_n\}$ denote the vector of private types of all other nodes except node i, and $c_{-i} = \{c_1, ..., c_{i-1}, c_{i+1}, ..., c_n\}$ denote the data preservation costs of all other nodes except node i.

Definition 1. (**Payment and Utility.**) Based on Green and Laffont Green and Laffont (1979),

is called the dominant strategy solution. Note that a dominant strategy solution is also a Nash equilibrium since no player has an incentive to unilaterally deviate from its strategy.

under VCG mechanism, given any type $\tilde{t}_i$ reported by node i, the amount of payment given to node $i \in V$ 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:

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

where $c_{V-\{i\}}$ is the minimum total cost of the preservation path when i is excluded from the network; $\tilde{c}_V$ is the minimum total cost of the preservation path when i is included in the network and reports its cost $\tilde{t}_i$. Therefore i's utility is 0 when it is not chosen by the centralized algorithm; and when i is chosen, its utility is

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

Moreover, we define c_V as the minimum total cost of the preservation path that goes through i when i truthfully reports its type, i.e., when $\tilde{t}_i = t_i$.

In addition, to decide whether to preserve data j or not requires knowing the associated payment due to the preservation of data j. We define the payment to storage node i due to its help in preserving data j as

$$p_{i,j}(\tilde{t}_i, t_{-i}) = c_{V-\{i\},j} - (\tilde{c}_{V,j} - \tilde{c}_{i,j}). \tag{5}$$

That is, based on the reported private types $(\tilde{t}_i, t_{-i})$, the payment to node i for its help in preserving data j is the difference between the total data preservation cost of j when i is excluded and when i is involved, plus the reported preservation cost of j by node i. Here $c_{V,j}$ is the total preservation cost of j when the total preservation cost of the network is c_V; and $c_{V-\{i\},j}$ is the total preservation cost of j given that i is removed therefore the total preservation cost of the network is $c_{V-\{i\}}$. □

Let $h(i) = \{j \in D | i \in P_j, i \neq s(j)\}$ be the set of data for which node i is not its source node but belongs to its preservation path according to the centralized algorithm. That is, $h(i)$ is the set of data that node i helps (by either relaying or storing) in their preservation. Lemma 1 below says that node i's payment $p_i(\tilde{t}_i, t_{-i})$ comes from two parts: one part is its help to data $j \in h(i)$ (called as the direct help of i), and the second part is its help to data $j \notin h(i)$ (called as the indirect help of i). While the first part is straightforward, the second part is due to the holistic procedure of the centralized algorithm aiming at minimizing total preservation cost. Namely, removing node i may also affect the preservation path of those data which do not use node i for their preservation when node i is included in the network.

Lemma 1. *It holds that* $p_i(\tilde{t}_i, t_{-i}) = \sum_{j \in D} p_{i,j}(\tilde{t}_i, t_{-i}), \forall i \in V - V_s$.

Proof:

$$\begin{aligned} p_i(\tilde{t}_i, t_{-i}) = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i) &= \sum_{j \in h(i)} [c_{V-\{i\},j} - \tilde{c}_{V,j}] + \sum_{j \notin h(i)} [c_{V-\{i\},j} - \tilde{c}_{V,j}] + \tilde{c}_i \\ &= \sum_{j \in h(i)} [c_{V-\{i\},j} - \tilde{c}_{V,j}] + \sum_{j \notin h(i)} [c_{V-\{i\},j} - \tilde{c}_{V,j}] + \sum_{j \in h(i)} \tilde{c}_{i,j} \\ &= \sum_{j \in h(i)} [c_{V-\{i\},j} - \tilde{c}_{V,j} + \tilde{c}_{i,j}] + \sum_{j \notin h(i)} [c_{V-\{i\},j} - \tilde{c}_{V,j}] \\ &= \sum_{j \in D} p_{i,j}(\tilde{t}_i, t_{-i}). \end{aligned}$$

■

Time complexity of the payment model. The time taken to compute the payment is the time taken for the minimum cost flow calculation, which is $O(|V|^2 \cdot |E| \cdot \log(|V| \cdot C))$, where C is the maximum capacity of an edge in the transformed graph Goldberg (1997).

An Example of Incentive to Lie. For a source node $s(j) \in V_s$ of data packet j, the incentive compatibility is that $s(j)$ pays no more than the value j if j is preserved. That is, to decide whether data j should be preserved or not, it compares $g_{s(j)}$ to $c_{V,j} = \sum_{i \in P_j} c_{i,j}$, the total preservation cost for data j, and preserves j only if $g_{s(j)} \geq c_{V,j}$. However, this could distort storage nodes' incentives and lead them to lie, as illustrated by the below example.

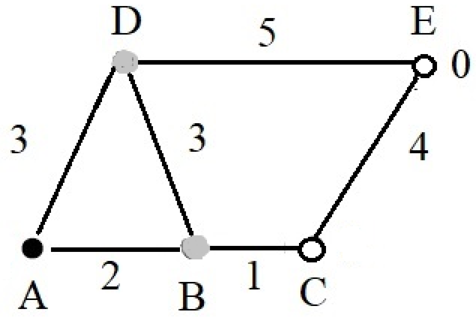

Figure 1: An example for incentive to lie.

In Fig. 1, node A is the source node with 1 unit overflow data (data size is $a = 1$). Nodes C and E are storage nodes, each with 1 unit of storage capacity. Nodes B and D are transition nodes. The weights (i.e., the distance between the two end nodes) on edges AB, BC, AD, DE, DB, and

EC are $2, 1, 3, 5, 3, 4$, respectively. The cost parameters are $\epsilon_i^a = 1, \epsilon_i^e = 0$ for $i \in \{A, B, C, D, E\}$, and $\epsilon_C^s = 2, \epsilon_E^s = 0$. That is, the transmission cost is the same as the distance; the storage cost of C is 2 and of E is 0. Suppose the data value of node A is $g_A = 4$. Therefore, when each non-source node truthfully reports the private type, as the preservation path is A, B, and C, with cost of $2+1+2 = 5 > 4$, the data will be dropped and $\pi_B = \pi_C = 0$.

Next, we show that each of B and C will have an incentive to lie to gain some positive utility. First, consider node B. Given that other nodes are telling the truth, let B lie by reporting $\epsilon_B^a = 0$. Then the cost along path A, B, C becomes $2+0+2 = 4 = g_A$, and data will be preserved. Utility of node B is $\tilde{\pi}_B = (3+5+0)-(2+0+2)+0-1 = 3$, strictly higher than the case when it tells the truth (i.e., $\pi_B = 0$). Here, $3+5+0$ is the cost of the preservation path A, D, E, when B is removed from the network. Thus node B has an incentive to lie. Similarly, we can see that node C can be better off by lying to $\epsilon_C^s = 1$ when other nodes are truth-telling. Under lying, the path cost along ABC is $2+1+1 = 4 = g_A$ and the data will be preserved. Now $\tilde{\pi}_C = (3+5+0) - (2+1+1) + 1 - 2 = 3$. Again $3+5+0$ is the path cost of preserving the data through D to E, when C is removed from the network. So node C has incentive to lie.

Global Replacement Path. To preserve the incentives for storage nodes to report their types truthfully, we consider a methodology similar to in Eidenbenz et al. (2005) with several modifications. That is, to decide whether data j should be preserved, we compare j's value to its *preservation cost along the global replacement path of* j. To find the global replacement path of j, we remove the original preservation path of j, i.e., P_j^* from the network, then let the central algorithm find the new path for j's preservation, which is j's global replacement path. The total preservation cost along this path is denoted as $c_{V-P_j^*,j}$. There are several diversifications from Eidenbenz et al. (2005). First, to remove the original path P_j^*, we only need to remove the non-source nodes on the path but not those source nodes (including j's source node $s(j)$), since source nodes have no private information. Second, the removal includes the destination (the storage node) of j.[3] We let a source node i preserve its data j as long as $g_{s(j)} \geq c_{V-P_j^*,j}$. If $g_{s(j)} \geq c_{V-P_j^*,j}$, we say that preservation of data j is feasible, and data j will be preserved. Otherwise, the preservation of data j is infeasible and data j will be dropped.

We define the set of data with feasible preservation as D^*. For D^*, the centralized algorithm will find the preservation path with minimum total cost, which may or may not be the same

[3] In Eidenbenz et al. (2005), the removal of the preservation path does not include the destination, which is a common storage to every data. Also, the source does not participate in other nodes' data preservation in their work.

as in P^*. As this will be the finalized preservation path used for the preservation of D^*, we denote it as P^f and the corresponding preservation path cost as c_V^f, with $c_V^f = \Sigma_{i \in P^f} \Sigma_{j \in D^*} c_{i,j}$. Correspondingly, $c_{V,j}^f = \Sigma_{i \in P_j^f} c_{i,j}$ is the total preservation cost of data j including its source node $s(j)$ and other nodes on the preservation path P_j^f. On the one side, with data j preserved, its value $g_{s(j)}$ is realized for $s(j)$, which will be paid by $s(j)$ to cover related cost. On the other side, according to the voluntary VCG mechanism, total payments for the preservation of j is $H_j \equiv \Sigma_{i \in V_s, i \in P_j} c_{i,j} + \Sigma_{i \in V - V_s} p_{i,j}$. Here $\Sigma_{i \in V_s} c_{i,j}$ is the total cost of the source nodes on the preservation path of data j (including j's source node), as their costs in helping j's preservation are directly observable; and $\Sigma_{i \in V - V_s} p_{i,j}$ is the total payment to all the other nodes due to their (direct or indirect) help in the preservation of data j. With some abuse of notation, we will call H_j as the *path payment of* j. Let the path payment along the final preservation path P^f be $H_j^f = \Sigma_{i \in V_s, i \in P_j^f} c_{i,j} + \Sigma_{i \in V - V_s} p_{i,j}^f$.

In general, for each data j, the payment made by the source node $g_{s(j)}$ and the finalized path payment H_j^f will not be equal, and the central authority will be the residual claimant to balance them. That is, if $g_{s(j)} > H_j^f$, there is a surplus in preserving data j, which will be held by the central authority; instead, if $g_{s(j)} < H_j^f$, there is a shortage in preserving data j, which will be covered by the central authority. Below we formally present the procedure of the voluntary VCG mechanism.

4.2 The Voluntary VCG Mechanism

Definition 2. (**The Voluntary VCG Mechanism.**) It includes four stages:

Stage 1. Each non-source node reports its private type t_i to the central authority.

Stage 2. For data set D, the centralized algorithm finds the optimal preservation path P^*, which minimizes the total preservation cost. For each data $j \in D$, it also finds the global replacement path and the corresponding replacement path cost $c_{V - P_j^*, j}$. The central authority then chooses to preserve data j if and only if $g_i \geq c_{V - P_j^*, j}$. The set of data chosen to be preserved is denoted as D^*.

Stage 3. For D^*, the centralized algorithm finds the *final optimal data preservation path* P^f, and calculate payment to each node based on Equation (3).

Stage 4. Each of the nodes in the finalized data preservation path P^f chooses to participate in data preservation or not. For nodes who choose to participate, each source node i pays $g_i - c_{i,j}^f$

for each of its preserved data $j \in d_i, j \in D^*$, and get reimbursed its cost in relaying other source nodes' data; each non-source node realizes the data preservation cost and also the payment given by Equation (3), and gets utility given by Equation (4). $\square$

Note that each node makes strategic moves only in stages 1 and 4, whereas stages 2 and 3 are non-strategic: in the absence of base stations, the centralized algorithm is provided by an outsider of the system. Although the outsider is denoted as a central authority, it cannot enforce the outcome in the system: Each node makes decisions based on its own interest. Nonetheless, our major result (presented below) indicates that the voluntary VCG mechanism provides each node the incentive to truthfully report private type and also participate in data preservation as instructed by the central algorithm. Thus the mechanism achieves data preservation efficiency in the sense that it minimizes preservation cost for the set of preserved data. On the other side, there can be data dropped due to its exorbitant preservation cost (relative to its value), therefore our mechanism also improves data efficiency for the network.

Assumptions. Several assumptions are needed for the mechanism to work. First, the payment model and the algorithm are common knowledge to each node. Second, each node has enough energy to perform data preservation tasks. Third, the network satisfies the "minimum-energy 2-connectivity", i.e., to any data, after a data preservation route is removed, there always exists an alternative route to preserve that data.[4]

For the voluntary VCG mechanism to be strategy-proof, it needs to satisfy two properties namely individual-rationality and incentive-compatibility, as explained below:

1. Individual-rationality (**IR**). It is the participation constraint that makes sure that each node, when truthfully reporting its type, will participate in data preservation once it is chosen by the centralized algorithm. That is, $\pi_i(t_i, t_{-i}) \geq 0 \ \forall t_{-i}$ and $\forall i \in V$.

2. Incentive-compatibility (**IC**). It requires that truthfully reporting private type is the dominant strategy of each node. Namely, each node gets the highest utility under truth-telling regardless of reported types of other nodes: $\pi_i(t_i, t_{-i}) \geq \pi_i(\tilde{t}_i, t_{-i}) \ \forall t_{-i}, \ \forall \tilde{t_i} \neq t_i$ and $\forall i \in V$.

Theorem 1. *The voluntary VCG mechanism satisfies the conditions IR and IC. In other words,*

[4] This implies that $\sum_{i=k+1}^{n} m_{-i} \geq d \cdot a$. After removing any single storage node, the system still can preserve all the overflow data. Although the "minimum-energy 2-connectivity" requirement looks restrictive, Yu et al. (2022) found that it is satisfied with high probability, in general over 90% in their simulations.

it is a dominant strategy solution that each non-source node shall truthfully report its private type, and each node follows the centralized algorithm in data preservation.

Proof: For a non-source node i that helps in preserving a data packet for which node i is not its source, node i either relays the data packet to a successor node, or stores the data packet. Therefore, it incurs one of the two costs below:

- *Relaying Cost $c_i^r(j)$.* When node i sends the data 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, $c_i^s = a \cdot \epsilon_i^e + a \cdot \epsilon_i^s$.

For given D^*, the proof of IR and IC among non-source nodes is similar as in Yu et al. (2022) and is omitted here. On the other hand, the decision on D^* considers the replacement path cost of each data, therefore lying or not by a node does not affect D^*, leaving each node no incentive to lie.

For source nodes, they will pay g_i for each preserved data and get zero utility. Therefore, participating data preservation is also a (weakly) dominant strategy of each source node. ∎

5 Simulations

In this section, we conduct extensive simulations to validate our theoretical results. Our simulator is written in Python. For the minimum cost flow implementation, we use NetworkX net (2022), a Python package for network analysis. To visualize our theoretical analysis, we focus on 4×4 grid networks with 16 sensor nodes, with IDs from 0 to 15. Unless otherwise mentioned, in any of the 4×4 grid networks generated, five nodes are randomly selected as source nodes, each having one data item. The rest nodes are either storage nodes, each having one or five units of storage spaces (i.e., $m = 1$ or 5); or transition nodes, each with zero storage spaces. Note that the number of storage nodes must be one more than the number of source nodes in order for the VCG mechanism to work. Each data point is the average of 20 simulation runs, and the error bars indicate 95% confidence intervals.

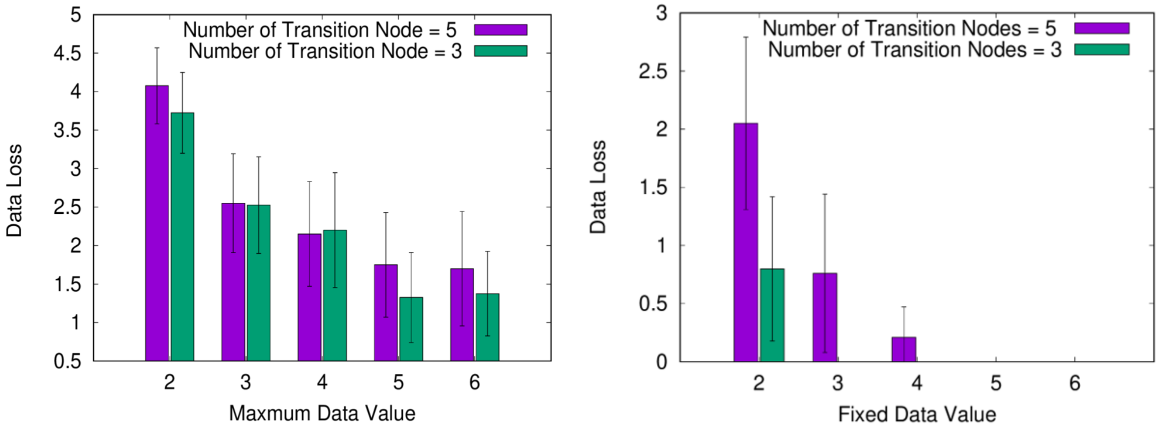

Figure 2: The data loss in (a) random data values and (b) fixed data values. $m = 1$.

Energy Model. We adopt a simplified energy model as follows for the purpose of easy illustration, although our work is based on a more general energy model. We assume the energy consumption of either sending to receiving a data item is 0.5 and storing a data item costs zero. Consequently, the energy consumption of offloading one data item from any source node to any storage node is the number of hops the data item traverses in the grid. That is, the weight (or cost) of each edge is one unit of energy (0.5 of transmitting and 0.5 of receiving).

Investigating Data Loss. We first investigate the data loss in the network. Fig. 2(a) shows the number of data losses by varying the values of the data items, which are random numbers between zero and a maximum data value. First, it shows that the number of data losses decreases with the increase of maximum data values. This is because our voluntary VCG mechanism preserves only the data items whose values exceed the costs of the corresponding replacement paths defined in Section 4.2. Therefore, the more valuable the data, the more likely it is preserved. Second, it shows that in most cases, the data loss for the number of transition nodes being 5 is larger than that when the number of transition nodes is 3. This is because more transition nodes generally result in a longer preservation path for a data item to reach its storage node. With a longer preservation path (and with a more significant cost), preserving data items becomes less desirable; thus more data loss occurs. In Fig. 2(b), we assign a fixed value to each data item and have the same observations as in Fig. 2(a), except that now the data loss for most of the data values are zeros for the number of transition nodes being 3. This is because being in the same range of [2, 6], a fixed data value means more valuable data items than the random case in Fig. 2(a). Consequently, both cases of transition numbers being 5 and 3 have less data loss. In particular, for the number of transition nodes being 3, there are zero data losses for values ranging from 3 to 6; for the number of transition nodes being 5, there are zero losses for values at 5 and 6.

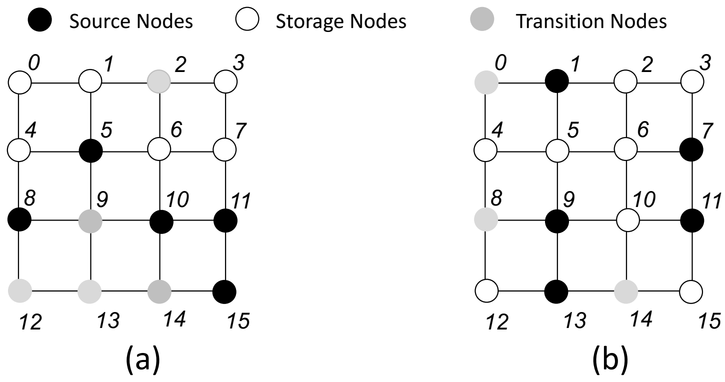

Figure 3: The grid network (a) has 5 transition nodes while (b) has 3 transition nodes. Both have 5 source nodes and the rest are storage nodes.

Comparing Truth-telling and Lying Utilities. Next, we validate the proposed VCG mechanism for the non-source nodes (i.e., storage and transition nodes) by comparing their truth-telling and lying utilities. We define the *scaling factor*, denoted as α, as the ratio between the reported and true values of a cost parameter of a non-source node. Following our energy model, when a non-source node reports (i.e., lies) about its energy cost of either transmitting or receiving one data item with a scaling factor of α, the reported costs become $0.5 \cdot \alpha$. When $\alpha < 1$, the node *under-reports* its cost by claiming it costs less energy than necessary; when $\alpha > 1$, it *over-reports* its cost by claiming it costs more energy than necessary; when $\alpha = 1$, it is truth-telling.

Table 1: Comparing truth-telling and lying utilities for non-source nodes with number of transition nodes = 5. Node IDs with * are transitions nodes, the rest are storage nodes with $m = 5$.

Non-source node	0	1	2*	3	4	6	7	9*	12*	13*	14*
Truth-telling (i.e., $\alpha = 1$)	0.0	0.0	0.0	0.0	2.0	1.0	2.0	0.0	0.0	0.0	0.0
Over-reporting, $\alpha = 2$	0.0	0.0	0.0	0.0	2.0	1.0	2.0	0.0	0.0	0.0	0.0
Under-reporting, $\alpha = 0.5$	0.0	0.0	0.0	0.0	2.0	1.0	2.0	0.0	0.0	0.0	0.0

<u>Number of Transition Nodes = 5.</u> Fig. 3(a) shows a randomly generated grid with nodes 5, 8, 10, 11, and 15 being source nodes, each having one data item. The non-source nodes include 5 transition nodes 2, 9, 12, 13, and 14 and six storage nodes 0, 1, 3, 4, 6, and 7. We first consider that $m = 5$, i.e., each storage node can store all the data items, as shown in Table 1. The three rows indicate

a non-source node's utility under truth-telling (i.e., $\alpha = 1$), over-reporting with $\alpha = 2$, and under-reporting with $\alpha = 0.5$, respectively. First, each non-source node's lying utility is no more than its truth-telling utility, indicating that truth-telling is the dominant strategy for the non-source nodes. Second, it shows the truth-telling utilities of most non-source nodes are zeros. This is because as each storage node has a storage capacity of 5, each can store all the data items in the network. As such, each of them is not critical as removing any of them does not increase the total preservation cost in the network. Third, nodes 4, 6, and 7 have the same utility under truth-telling and lying. This is because under either truth-telling or lying, the data preservation paths for the data items are mainly the same, due to regular toplogies of the grid network.

Table 2: Comparing truth-telling and lying utilities for non-source nodes with number of transition nodes = 5. Node IDs with * are transitions nodes, the rest are storage nodes with $m = 1$.

Non-source node	0	1	2*	3	4	6	7	9*	12*	13*	14*
Truth-telling (i.e., $\alpha = 1$)	0.0	1.0	0.0	3.0	3.0	3.0	6.0	0.0	0.0	0.0	0.0
Over-reporting, $\alpha = 2$	0.0	1.0	0.0	3.0	3.0	3.0	6.0	0.0	0.0	0.0	0.0
Under-reporting, $\alpha = 0.5$	0.0	1.0	0.0	3.0	3.0	3.0	6.0	0.0	0.0	0.0	0.0

We then consider that each storage node can only store one data item in the BSN (i.e., $m = 1$), as shown in Table 2. Again, we observe that truth-telling utility is the dominant strategy for the non-source nodes. Compared to Table 1, it shows non-source nodes have more positive truth-telling utilities. This is because as each storage node has a storage capacity of 1, removing any of them could prolong the data items' preservation paths, resulting in a higher total preservation cost for the entire network and a positive marginal cost for each node.

Number of Transition Nodes = 3. Finally, we consider that the number of transition nodes is 3. We found that only one node, storage node 12, has positive utilities. This is because 12 is critical to source node 13's data preservation; however, other storage nodes are not critical to any source nodes' data preservation. Due to the specific layout of the nodes, $m = 5$ or $m = 1$ do not make any difference in the utility computation.

6 Conclusion and Future Work

In this work, we study the data preservation problem in base station-less sensor networks wherein sensor nodes behave selfishly. Selfishness is reflected in two aspects. First, for non-source nodes,

Table 3: Comparing truth-telling and lying utilities for non-source nodes with number of transition nodes = 3. Node IDs with * are transitions nodes, the rest are storage nodes with $m = 5$ or $m = 1$.

Non-source node	0*	2	3	4	5	6	8*	10	12	14*	15
Truth-telling (i.e., $\alpha = 1$)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
Over-reporting, $\alpha = 2$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
Under-reporting, $\alpha = 0.5$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0

their cost parameters are private information. They may not want to participate in data preservation nor have the incentive to report their private information truthfully. Second, source nodes may not have an incentive to store data once the data value exceeds the payment needed for its preservation. We design a voluntary VCG mechanism under which the individual sensor nodes, motivated solely by self-interest, achieve a good system-wide data preservation solution. In particular, the mechanism guarantees truthfulness among non-source nodes and data preservation efficiency among the data set chosen to be preserved. In addition, the mechanism makes each source node pay only its data value to preserve the data, thus guaranteeing the voluntariness of the source nodes.

Currently, we adopt grid topologies for BSNs for ease of illustration and visualization. In the future, we will consider a more realistic BSN topology wherein energy consumption of data preservation depends on the distance among nodes. We will also investigate the budget imbalance of the voluntary VCG model when storage nodes are energy-constrained. After that, we will validate theoretical findings using simulation results. First, simulation results shall verify the truthfulness and efficiency of the mechanism by contrasting the utility of each non-source node under the truth-telling strategy to what it is under lying. Second, the simulation will illustrate the number of data dropped due to their considerable preservation cost, and examine how such number changes in the network topology. Third, the simulation shall look into the budget imbalance of the voluntary VCG. While it verifies the upper-bound budget imbalance without capacity constraint, the simulation shall study the budget imbalance in the scenario with capacity constraint. Other future work includes extending 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 nodes to cooperate and achieve optimal data preservation.

Acknowledgment

This work was supported in part by NSF Grants CNS-2131309, CNS-1911191, and CNS-1923956.

References

2022. NetworkX: Network Analysis in Python. https://networkx.org/.

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Inc.

A. Cammarano, C. Petrioli, and D. Spenza. 2012. 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).*

Y. Chen and B. Tang. 2016. Data Preservation in Base Station-less Sensor Networks: A Game Theoretic Approach. In *Proc. of the 6th EAI International Conference on Game Theory for Networks (GameNets 2016).*

E. H. Clarke. 1971. Multipart pricing of public goods. *Public Choice* (1971).

E. Cochran, J. Lawrence, C. Christensen, and A. Chung. 2009. A novel strong-motion seismic network for community participation in earthquake monitoring. *IEEE Inst and Meas* 12, 6 (2009), 8–15.

W. Colitti, K. Steenhaut, N. Descouvemont, and A. Dunkels. 2008. Satellite Based Wireless Sensor Networks: Global Scale Sensing with Nano- and Pico-satellites. In *Proc. of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08).* 445–446.

N. Crary, B. Tang, and S. Taase. 2015. Data Preservation in Data-Intensive Sensor Networks With Spatial Correlation. In *Proc. of the International Workshop on Mobile Big Data (MobiData 2015) in conjunction with Mobihoc 2015.*

S. Eidenbenz, G. Resta, and P. Santi. 2005. COMMIT: a sender-centric truthful and energy-efficient routing protocol for ad hoc networks with selfish nodes. In *19th IEEE International Parallel and Distributed Processing Symposium (IPDPS).*

L. Terray et al. 2020. From Sensor to Cloud: An IoT Network of Radon Outdoor Probes to Monitor Active Volcanoes. *Sensors* 20, 10 (2020).

R. Ghaffarivardavagh, S. S. Afzal, O. Rodriguez, and F. Adib. 2020. Ultra-Wideband Underwater Backscatter via Piezoelectric Metamaterials. In *Proc. of the ACM SIGCOMM.*

A. V. Goldberg. 1997. An Efficient Implementation of a Scaling Minimum-Cost Flow Algorithm. *J. Algorithms* 22 (1997), 1–29.

J. Green and J. Laffont. 1979. Incentives in public decision making. *Studies in Public Economics* 1 (1979), 65–78.

T. Groves. 1973. Incentives in teams. *Econometrica* (1973).

S. Hsu, Y. Yu, and B. Tang. 2020. DRE^2: Achieving Data Resilience in Wireless Sensor Networks: A Quadratic Programming Approach. In *Proc. of IEEE MASS.*

J. Jang and F. Adib. 2019. Underwater Backscatter Networking. In *Proc. of the ACM SIGCOMM.*

Y. Li, X. Li, and P. Wang. 2012. A Module Harvesting Wind and Solar Energy for Wireless Sensor Node. *Advances in Wireless Sensor Networks, the series Communications in Computer and Information Science* 334 (2012), 217–2224.

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

N. Nisan. 1999. Algorithms for selfish agents: Mechanism design for distributed computation. *STACS 1999. LNCS, Meinel, C., Tison, S. (eds.)* 1563 (1999).

N. Nisan and A. Ronen. 1999. Algorithmic mechanism design. In *Proc. of the thirty-first annual ACM symposium on Theory of computing (STOC 1999).* 129–140.

N. Nisan and A. Ronen. 2007. Algorithmic mechanism design. *Games and Economic Behavior* 35 (2007), 166–196.

R. C. Shah, S. Roy, S. Jain, and W. Brunette. 2003. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *Proc. of SNPA.*

B. Tang. 2018. DAO^2: Overcoming Overall Storage Overflow in Intermittently Connected Sensor Networks. In *Proc. of IEEE INFOCOM 2018.*

B. Tang, N. Jaggi, and M. Takahashi. 2014. Achieving Data K-Availability in Intermittently Connected Sensor Networks. In *Proc. of the International Conference on Computer Communications and Networks (ICCCN).*

B. Tang, N. Jaggi, H. Wu, and R. Kurkal. 2013. Energy-efficient Data Redistribution in Sensor Networks. *ACM Trans. Sen. Netw.* 9, 2 (April 2013), 11:1–11:28.

B. Tang, H. Ngo, Y. Ma, and B. Alhakami. 2021. *DAO^2: Overcoming Overall Storage Overflow in Intermittently Connected Sensor Networks.* Technical Report 2021-1, `http://csc.csudh.edu/btang/papers/infocom18_journal.pdf`. Computer Science Department, CSUDH.

W. Vickrey. 1961. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance* (1961).

G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. 2006. Fidelity and Yield in a Volcano Monitoring Sensor Network. In *Proc. of OSDI.*

X. Xue, X. Hou, B. Tang, and R. Bagai. 2013. Data Preservation in Intermittently Connected Sensor Networks With Data Priorities. In *Proc. of SECON.*

J. Yick, B. Mukherjee, and D. Ghosal. 2008. Wireless sensor network survey. *Computer Networks* 52 (2008), 2292–2330. Issue 12.

Y. Yu, S. Hsu, A. Chen, Y. Chen, and B. Tang. 2022. *Truthful and Efficient Data Preservation in Base Station-less Sensor Networks.* Technical Report 2022-1, `http://csc.csudh.edu/btang/papers/bsn_new.pdf`. Computer Science Department, CSUDH. Submitted to ACM Transactions on Sensor Networks.