Data-VCG: A Data Preservation Game for Base Station-less Sensor Networks with Performance Guarantee

Jennifer Ly[†], Yutian Chen[*], Bin Tang[†]
[*]Economics Department, California State University, Long Beach, Yutian.Chen@csulb.edu
[†]Department of Computer Science, California State University Dominguez Hills, jenniferly95@gmail.com, btang@csudh.edu

***Abstract*—We design a data preservation game called Data-VCG for *base station-less sensor networks (BSNs)*. In the BSN, sensor nodes do not have connected paths to a base station; thus, sensory data must be preserved inside the network before uploading opportunities arise. Data-VCG incorporates data values into the classic Vickrey-Clark-Groves (VCG) mechanism, a generic truthful mechanism for achieving a socially-optimal solution. It motivates all the sensor nodes (i.e., source, storage, and transition) to voluntarily participate in the data preservation while achieving minimum data preservation cost. We give a detailed analysis of the performance guarantee of the Data-VCG and show that under certain conditions, its worst-case budget imbalance is at most $n-3$ times the efficiency gain, where n is the number of sensor nodes in the network. We conduct extensive simulations to validate our results in both grids and randomly generated BSNs under different network dynamics.**

Keywords – VCG Mechanism, Base Station-less Sensor Networks, Data Preservation, Budget Imbalance, Worst Efficiency Loss

I. Introduction

Background and Motivation. In this paper, we consider *base station-less sensor networks* (BSNs) in challenging environments (e.g., underwater or ocean exploration [13] and volcano eruption monitoring [10]). In a BSN, as deploying high-power and high-storage data-collecting base stations in or near the sensing field is not feasible, its essential function is to store large volumes of sensory data inside the network between two uploading opportunities (e.g., the periodic visits of autonomous underwater vehicles (AUVs) [5] and robots [20]).

We consider a BSN model as follows. Some sensor nodes are close to the events of interest and constantly generate sensory data, thus depleting their storage spaces; they are referred to as *source nodes*. To avoid data loss, source nodes need to offload their *overflow data* to nearby sensor nodes with available storage (referred to as *storage nodes*) if uploading opportunities are not available; this process is called *data preservation in BSNs*. Besides, there exist some sensor nodes called *transition nodes* that have neither overflow data nor available storage, and their sole role in the BSN is to relay overflow data packets from source nodes to storage nodes. Storage and transition nodes are called *non-source nodes*.

As sensor nodes usually have limited battery power, storage spaces, and processing capacity, existing research on BSNs has mainly focused on various optimal resource allocations in data preservation [17], [12], [18]. However, there are two areas for improvement in all the above works. First, they fail to consider that in many BSN applications, different sensory data could have different importance (i.e., data values), thus contributing differently to scientists analyzing the physical environment. For example, in a volcano monitoring application [10], while all the sensed data (i.e., seismic, infrasonic, temperature) are essential for scientists to analyze the volcano activities, seismic and infrasonic data are usually more crucial than the temperature data to interpret the key features of a volcano such as its scale and magnitude.

Second, all the above work assumes that the sensor nodes are cooperative and are willing to contribute their resources to the data preservation process. Unlike the traditional sensors that only sense, compute, and communicate, the smart sensor can also perceive, reason, and learn in the process. Besides, many emerging IoT sensing applications are on a global scale and distributed in nature, with the sensor nodes being controlled by different entities, each aiming to pursue its self-interest and maximize its benefit. As such, the resource-constrained and distributed sensor nodes in the BSN can behave selfishly only to conserve their resources and have little incentive to participate in data preservation. If not dealt with satisfactorily, selfishness could impede the data preservation process and compromise the functions and missions of the above BSN applications.

Our Contributions. In this paper, we address the above challenge by proposing a mechanism design framework called Data-VCG. Data-VCG integrates data values with the Vickrey-Clark-Groves (VCG) mechanism [19], a well-known mechanism design methodology to achieve truthfulness and efficiency. We make the following contributions.

First, preserving all the overflow data packets could consume lots of energy, resulting in system performance degradation. To eliminate such inefficiency, we propose that each source node preserves its overflow data only when its values exceed the incurred energy consumption.

Second, unlike existing work [22] that assumes that a central authority is solely responsible for the compensation, our Data-VCG game requires source nodes to compensate storage nodes for preserving their data, which is more sustainable and self-sufficient. The Data-VCG guarantees that every node, including the source and non-source nodes, will *voluntarily* participate in the data preservation game. In contrast, existing work [7], [22] only motivates the storage nodes to participate in the data preservation process.

Third, we examine the budget imbalance of our Data-VCG game and investigate the ratio of budget imbalance to the system efficiency gain in the worst scenario (referred to as the *worst efficiency loss*). In the case of a budget deficit, we find that it is bounded $n - 3$ times the efficiency gain, with n being the number of sensor nodes in the network. In the case of a budget surplus, the budget surplus never exceeds the efficiency gain.

Paper Organization. Section II reviews all the related work. Section III and Section IV present the data preservation problem and the Data-VCG game, respectively. Section V presents our detailed simulation results and analysis. Section VI concludes the paper with a discussion of future work.

II. Related Work

Algorithmic mechanism design (AMD) [15] takes an objective-first approach to designing economic mechanisms or incentives to motivate rational and selfish players toward desired objectives. There exists extensive research applying AMD techniques to motivate wireless ad hoc and sensor nodes to participate in the packet forwarding [21], topology and power control [16], duty cycling and MAC protocols [8], data aggregation [4] and task allocation [3]. However, none of them focused on data preservation in BSNs.

Chen et al. [7] studied data preservation in BSNs, considering sensor nodes are selfish. They designed a computationally efficient and truthful VCG-based data preservation game wherein truth-telling is always a dominant strategy. Yu et al. [22] further identified that when storage nodes have a limited energy power, the VCG mechanism proposed in [7] is no longer truthful. Nodes can manipulate the VCG mechanism to gain more utilities. However, both works assumed that the source nodes must offload their overflow data packets to other storage nodes. We consider the selfishness of both the source and storage nodes and propose a holistic approach to achieve their truth-telling. In both existing works, data packets with little values can be preserved with much energy costs, yielding less efficient outcomes for the entire BSN system.

Our work is inspired by Eidenbenz et al. [9], which proposed a sender-centric truthful ad hoc routing protocol. However, it assumed the only private information of a sender (i.e., a source node) is its willingness to establish a connection with the destination. In contrast, in our work, the relationship between a node's cost parameters and its incurred costs is more complicated (see cost model in Section III). By lying about different cost parameters to different extents, a node might manipulate its cost and switch from one action to another. Chen et al. [6] introduced a VCG-based mechanism wherein both source nodes and storage nodes voluntarily participate in data preservation; however, it does not provide a performance guarantee of the mechanism. In this paper, we fill such a gap by designing a performance metric *efficiency loss* to quantify the game's efficiency. We give a detailed analysis of its performance guarantee and show that under certain conditions, the worst-case budget imbalance of the mechanism is at most $n-3$ times the efficiency gain, with n number of sensor nodes in the network.

III. Data Preservation Problem in the BSN

System Model. We model a BSN 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. WLOG let's assume there are $k < n$ source nodes $V_s = \{1, 2, ..., k\}$, $q < n - k$ storage nodes $V_r = \{k + 1, k + 2, ..., k + q\}$, and $n - k - q$ transition nodes $V_t = \{k+q+1, k+q+2, ..., n\}$. Storage and transition nodes are *non-source nodes*.

The sensory data are modeled as a sequence of data packets, each of which is a bits. Let d_i denote the number of overflow data packets source node $i \in V_s$ 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 packets. Let $s(j) \in V_s$, $1 \le j \le d$ denote data packet j's source node and D_i be the set of data packets at source node i; that is, $D_i = \{j \in D | s(j) = i\}$. Let m_i be the available free storage space (in bits) at sensor node $i \in V$. Note that $m_i = 0$ for $i \in V_s \cup V_t$ while $m_i > 0$ for $i \in V_r$. We assume that $\sum_{i=k+1}^{k+q} m_i > d \cdot a$.

Cost Model. Each overflow data packet has some value, indicating its importance in the BSN application. Let $g_i > 0$ be the value of each overflow data packet at source node $i \in V_s$. Here, we assume that the overflow data packets from the same source node have the same value, although our model can be easily extended to the d data packets all have different values.

When node i sends a data packet to its neighbor i' over their distance $l_{i,i'}$, the amount of *transmitting energy* spent by i is $E_i^t(i') = a \cdot \epsilon_i^a \cdot l_{i,i'}^2 + a \cdot \epsilon_i^e$. Here, $\epsilon_i^a = 100pJ/bit/m^2$ and $\epsilon_i^e = 100nJ/bit$ are the energy consumption of transmitting one bit on the transmit amplifier and circuit of node i, respectively. When node i receives a data packet, the amount of *receiving energy* it spends is $E_i^r = a \cdot \epsilon_i^e$. Note that ϵ_i^e is involved in transmission and receiving energy, while ϵ_i^a is involved only in transmission energy. By lying about different cost parameters to different extents, a node might manipulate its cost and switch from one action to another.

Problem Formulation. A *preservation function* $f : D \to V_r$ indicates that a data packet $j \in D$ is offloaded from its source node $s(j) \in V_s$ to a storage node $f(j) \in V_r$. Let $P_j = \{s(j), ..., f(j)\}$ be the *preservation path* along which j is offloaded. Let $c_{i,j}$ denote node i's energy consumption in preserving j. The objective is to find a preservation function f and P_j ($1 \le j \le d$) to minimize the *total preservation cost* $c = \min_f \sum_{j=1}^{d} \sum_{i=1}^{n} c_{i,j} = \min_f \sum_{i=1}^{n} \sum_{j=1}^{d} c_{i,j}$ under the storage constraint of storage nodes: $|\{j | 1 \le j \le d, p(j) = i\}| \cdot a \le m_i$, $\forall i \in V_r$.

Cooperative Minimum Cost Flow (MCF) Algorithm. When all the nodes are cooperating, the data preservation problem

TABLE I
NOTATION SUMMARY

Notation	Description
$G(V,E)$	BSN graph, $V = V_s \cup V_r \cup V_t$, $\|V\| = n$
V_s	Set of k source nodes
V_r	Set of q storage nodes
V_t	Set of $n-k-q$ transition nodes
d_i	Number of overflow data packets at source node $i \in V_s$
g_i	Value of overflow data packets at source node $i \in V_s$
d	Total number of overflow data packets
D	The set of d overflow data packets
D_i	The set of overflow data packets at source node i
i	Index for sensor nodes, $1 \le i \le n$
j	Index for overflow data packets, $1 \le j \le d$
$s(j)$	The source node of data j, $1 \le j \le d$
m_i	Storage capacity of storage node $i \in V_r$
$E_i^t(i')$	Transmission energy spent by i to transmit one packet to i'
E_i^r	Receiving energy spent by i to receive one data packet
f	Data offloading function
P_j	The *offloading path* of data packet j
$\sigma(i,j)$	Node i's successor node in P_j
$c(i,j)$	Node i's preservation cost for j
c	The total preservation cost of the entire network
D_f	The set of preserved data packets in Data-VCG game

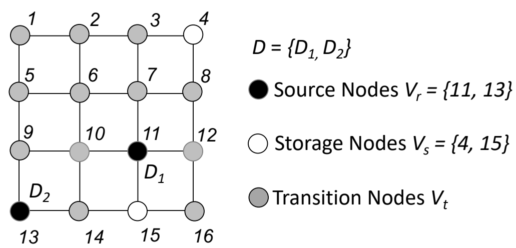


Fig. 1. Illustrating data preservation in a 4×4 grid BSN.

in the BSN is equivalent to the minimum cost flow (MCF) problem [18]. MCF can be solved optimally and efficiently, and there are many cooperative MCF algorithms [2]. This paper adopts the scaling push-relabel algorithm proposed in [11] as it has the highest performance C codes available. We refer to it as the *MCF algorithm*.

EXAMPLE 1: Fig. 1 shows a 4×4 grid BSN with 16 sensor nodes. Nodes 11 and 13 are source nodes, each having one overflow data packet to offload, $D = \{D_1, D_2\}$; nodes 4 and 15 are storage nodes, each having two units of storage capacity; the others are transition nodes. Assume each edge costs one unit of energy. The MCF algorithm will offload D_1 and D_2 to node 15, giving a total preservation cost of 3. □

IV. The Data-VCG Game in the BSN

We present the Data-VCG game and prove it is a voluntary data preservation game with a performance bound.

A. The Data-VCG Game

Payment Model. In Data-VCG, there is a central authority whose role is to collect values of all the preserved data packets from source nodes and uses them as payment to compensate all the nodes that help preserve data. We introduce the payment model for non-source and source nodes, respectively.

Source Nodes. As the source nodes own their overflow data and thus want to offload and preserve them, the source node compensates other nodes involved in preserving its data packet in the Data-VCG game (by paying the central authority, which compensates all other nodes). As such, a source node will have its data offloaded only if the compensation it pays others does not exceed the data value. And source node i pays $g_i - c_{i,j}$ to others for preserving its data packet j (recall $c_{i,j}$ is node i's cost in offloading data packet j). On the other hand, if source node i relays other source nodes' data packets, it will be compensated with the preservation cost it spends. Therefore, a source node has zero utility from preserving its data, and its utility is always zero. We assume all the source nodes make their information public, including cost parameters and data values, and we leave the incentive for the source node to lie about its data value as future work.

Non-Source Nodes. Any non-source node (i.e., a storage node or a transition node) $i \in V - V_s$ considers its cost parameters as its private information, referred to as its *private type* $t_i = \{\epsilon_i^a, \epsilon_i^e\}$. And its strategy set A_i includes any value of private type t_i it can report. Let $c_i = \sum_{j=1}^{d} c_{i,j}$ denote node i's true total cost and p_i the total payment to i in the Data-VCG. Let $t_{-i} = \{t_1, ..., t_{i-1}, t_{i+1}, ..., t_n\}$ denote the vector of private types of all other nodes except i. Given any type $\tilde{t}_i$ reported by non-source node i, its received payment is 0 if it does not participate; otherwise, it is

$$p_i(\tilde{t}_i, t_{-i}) = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i), \quad (1)$$

where $c_{V-\{i\}}$ is the minimum total preservation cost when i is excluded from the network; $\tilde{c}_V$ and $\tilde{c}_i$ are the minimum total preservation cost and i's reported preservation cost when i reports its private type $\tilde{t}_i$. Therefore i's utility is 0 when it does not participate; otherwise, its utility is

$$\pi_i(\tilde{t}_i, t_{-i}) = p_i(\tilde{t}_i, t_{-i}) - c_i = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i) - c_i. \quad (2)$$

The above implies that after removing any non-source node from the BSN, the rest of the BSN is still connected, and there is still enough storage space in the BSN to store all the overflow data packets. This is indeed the case in Example 1.

The Data-VCG Game. Next, we present the Data-VCG Game. We first present the below definition.

Definition 1: (Payment to Non-Source Node i for Preserving Data j.) The *payment to i for preserving j* is

$$p_{i,j}(\tilde{t}_i, t_{-i}) = c_{V-\{i\},j} - (\tilde{c}_{V,j} - \tilde{c}_{i,j}). \quad (3)$$

$c_{V,j}$ and $c_{V-\{i\},j}$ are the total preservation cost for j if i is included in or removed from the BSN, respectively. □

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. Definition 1 shows that i's decision to preserve a data packet j or not requires knowing how much payment it can receive. It will also be used in analyzing the worst budget imbalance defined later.

Definition 2: (**Replacement Path and Replacement Cost of Data Packet** j**.**) Let P_j^* be the original preservation path of j computed by the MCF algorithm. Then we remove all the nodes in P_j^* except j's source node $s(j)$ from the BSN and rerun the MCF algorithm to find a new preservation path for j. This new path is j's *replacement path*, and its cost is the *replacement cost* of j, denoted as $c_{(V-P_j^*),j}$. □

EXAMPLE 2: For the BSN in Fig. 1, the preservation path of the data packet D_1 is 11, 15, and the preservation for the data packet D_2 is 13, 14, 15. For D_1, after removing 15, there are three replacement paths with a replacement cost of 3; one is 11, 7, 3, 4. For D_2, after removing nodes 14 and 15, there are many replacement paths with a replacement cost of 6; one is 13, 9, 5, 1, 2, 3, 4. □

The Data-VCG is shown in Algo. 1. First, a non-source node i reports its private type $\tilde{t}_i$ (line 1), and the MCF algorithm finds the optimal data preservation paths (line 2). Then, it computes the replacement path and cost for data packet j. If its value is greater than or equal to its replacement cost, it will be preserved (line 3); such packet is called *preserved data packet*. Next, it reruns the MCF algorithm for preserved data packets and finds packet j's final preservation path P_j^f (line 4). Finally, it calculates the payment for source and non-source nodes, based on which each non-source node voluntarily participates in data preservation and gets payment (line 5).

Algorithm 1: The Data-VCG Game in the BSN.
Input: A data preservation instance $G(V, E)$, $(\tilde{t}_i, t_{-i})$;
Output: A payment for $i \in V$.
Notations: $\tilde{t}_i$: the private type reported by node i;
P^*: the optimal preservation paths for D with $\tilde{t}_i$;
D^f: the final set of data packets to be preserved;
P_j^f: the final data preservation path for packet $j \in D^f$;
$c_{i,j}^f$: node i's energy cost preserving data packet $j \in D^f$;
0. $D^f = \phi$ (empty set);
1. Non-source node i reports its private type $\tilde{t}_i$;
2. Runs the MCF algorithm for D to find the optimal set of preservation paths $P^* = \{P_j^*\}$, $j \in D$;
3. **for** each data packet $j \in D$
    Compute j's replacement cost $c_{(V-P_j^*),j}$ by removing j' preservation path (except $s(j)$) and rerun MCF;
    if $(g_{s(j)} \geq c_{(V-P_j^*),j})$
        $D^f = D^f \cup \{j\}$; // Data j is preserved
    end if;
  end for;
4. Runs the MCF algo. for D^f and finds the P_j^f, $i \in D^f$;
5. **for** each preserved data packet $j \in D^f$
    Source node $s(j)$ pays $g_{s(j)} - c_{s(j),j}^f$ to central authority;
    Non-source node $i \in P_j^f - \{s(j)\}$ receives payment given by Eq. (1) from central authority, and realizes utility given by Eq. (2);
  end for;
6. **RETURN** A payment for $i \in V$.

Algo. 1 achieves voluntary participation for both source and non-source nodes. The proof is omitted due to space constraints.

EXAMPLE 3: For the BSN in Fig. 1, assume the values of both D_1 and D_2 are 6. The replacement costs of these two packets are 3 and 6, respectively (Example 2). As each does not exceed the corresponding data value, both packets will be preserved and $D^f = \{D_1, D_2\}$. □

Incentive for Non-Source Nodes to Lie. For a source node to willingly offload a data packet, the value of the packet should be more than its preservation cost. A simple rule seems to have the data preserved when its data value is larger than the reported preservation. However, this could incentivize non-source nodes to lie about their cost parameters, failing the VCG mechanism. Consider the example in Fig. 1 again and let the value of D_2 be 1.5. Assume the cost parameters are $\epsilon_i^a = 1$ and $\epsilon_i^e = 0$ for all the node i. When each non-source node truthfully reports the private type, node 13 will drop its data as the minimum preservation cost is $1+1 = 2 > 1.5$ by storing the data to node 15 through node 14. Hence under truth-telling, $\pi_{14} = 0$. Now suppose 14 lies by reporting 0 costs. Then the path cost along 13, 14, and 15 becomes $1 + 0 = 1 < 1.5$, and the overflow data is preserved. Following Eq. (2), 14's utility $\tilde{\pi}_{14} = (1 + 1 + 1 + 1) - (1 - 0) - 1 = 2$, strictly higher than its truth-telling utility.

B. The Worst Budget Imbalance of Data-VCG

In Data-VCG, the central authority collects values from all the preserved data packets and uses them as payment to compensate all the nodes that help preserve data. Generally, the collected data values and the payment to the nodes are unequal; we refer to their difference as the *budget imbalance* in the Data-VCG. In particular, the budget imbalance could be either surplus (i.e., data values larger than payment) or deficit (i.e., data values smaller than payment) in the system. Below, we quantify the total payment spent for preserved data packet $j \in D^f$ computed in Data-VCG viz. Algo. 1.

Definition 3: (**Total Payment for Preserving Data** $j \in D^f$**.**) Denote the total payment for preserved data packet $j \in D^f$ as H_j^f, $H_j^f = \Sigma_{i \in V_s, i \in P_j^f} c_{i,j} + \Sigma_{i \in V - V_s} p_{i,j}^f$. Here, the first term on the r.h.s. is the total cost of the source nodes on P_j^f (including j's source node), as their costs in helping j's preservation are directly observable; and the second term is the total payment to all the non-source nodes due to their help in the preservation of data $j \in D^f$. □

For any preserved data packet $j \in D^f$, the payment made by its source node to the central authority is $g_{s(j)}$, which covers the cost along the preservation path. Yet, the total payment (including the preservation cost of its source node) needed for its preservation is H_j^f. If $g_{s(j)} < H_j^f$, the central authority finances the system to have data j preserved; if $g_{s(j)} > H_j^f$, the central authority collects the surplus. As a result, the net balance the central authority injects into the system may not be too large. Below we investigate the worst scenario of the Data-VCG game from the perspective of the budget imbalance.

Worst Efficiency Loss. Inspired by Moulin et al. [14], we define *worst efficiency loss* as below to measure the degree of budget imbalance in a Data-VCG. Denote the cost of the final optimal preservation path $P^f = \{P_j^f\}$, $j \in D^f$, as c_V^f. We have $c_V^f = \Sigma_{i \in P^f} \Sigma_{j \in D^f} \; c_{i,j}$. Let $c_{V,j}^f = \Sigma_{i \in P_j^f} \; c_{i,j}$ be the total preservation cost of data j including its source node $s(j)$ and other nodes on the preservation path P_j^f.

Definition 4: (**Budget Imbalance, Efficiency Gain, Worst Efficiency Loss in the BSN.**) We define the *budget imbalance (BI)* as the difference between the total payment for preserving D^f and the total data values of the D^f; that is, $BI = \Sigma_{j \in D^f}(H_j^f - g_{s(j)})$. BI is thus the amount needed from the central authority to finance the system to run Data-VCG.

We define *efficiency gain (EG)* of preserving D^f as the difference between the total data values of D^f and the total cost keeping D^f; that is, $EG = \Sigma_{j \in D^f} g_{s(j)} - c_V^f$.

We define the *worst efficiency loss (L)* as the maximum ratio between budget imbalance and efficiency gain in the BSN. That is, $L = \max \frac{BI}{EG} = \max \frac{\Sigma_{j \in D^f}(H_j^f - g_{s(j)})}{\Sigma_{j \in D^f} g_{s(j)} - c_V^f}$. L characterizes the worst scenario of the overall performance of the Data-VCG over all possible profiles of valuations. □

EXAMPLE 4: Following Example 3, the payment for preserving node 13's data is (5-3)+(9-3)+2=10, and it is (9-3)+1=7 for the preservation of node 11's data. Therefore, the efficiency loss is calculated as (10+7)/(6+6-3)=17/9. □

Below we show that the worst efficiency loss l of Data-VCG is upper- and lower-bounded when the storage capacity of the storage nodes is large. Proofs are omitted due to space constraints. Lemma 1 below says that the payment to node i is now based only on the direct help of node i on relaying/storing a data packet. If node i is not on the preservation path of data, then node i receives no payment for the preservation of that data. Lemma 2 says that removing any node of the system always (weakly) increases the preservation cost of any data.

Lemma 1: Let $h(i) = \{j \in D | i \in P_j, i \neq s(j)\}$, the set of data for which node i is not its source node but belongs to its preservation path according to the MCF algorithm. Suppose $m_i \geq d, \forall i \in V_r$. It holds that $p_i(\tilde{t}_i, t_{-i}) = \sum_{j \in h(i)} p_{i,j}(\tilde{t}_i, t_{-i}), \forall i \in V - V_s$.

Lemma 2: Suppose $m_i \geq d, \forall i \in V_r$. It holds that $c_{V-i,j} \geq c_{V,j}, \forall j \in D$.

Proof: Suppose $c_{V-i,j} < c_{V,j}$ for some $j \in D$. Then consider keeping the preservation path for j as in c_{V-i}, and adopting the preservation path of all other data as in c_V (which is feasible without storage capacity constraint of storage nodes). The total preservation cost is

$$\hat{c}_{V-i} \equiv c_{V-i,j} + \sum_{\xi \in D, \xi \neq j} c_{V,\xi} < c_{V,j} + \sum_{\xi \in D, \xi \neq j} c_{V,\xi} = c_V$$
$$\Rightarrow \hat{c}_{V-i} < c_V,$$

which contradicts that c_V is the optimal preservation cost. ■

Theorem 1: It holds that $L \in [-1, n-3]$ for a BSN with $n \geq 3$ and $m_i \geq d, \forall i \in V_r$.

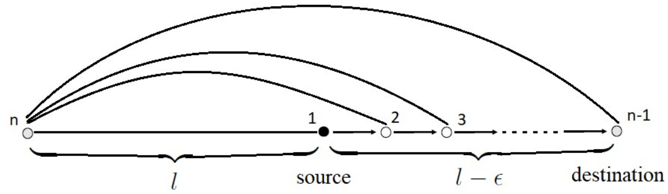


Fig. 2. The Worst Efficiency Loss.

Proof: From the definition of L, we have

$$\begin{aligned} L &= \max \frac{\Sigma_{j \in D^f}(H_j^f - g_{s(j)})}{\Sigma_{j \in D^f} \; g_{s(j)} - \Sigma_{j \in D^f} \; c_{V,j}} \\ &= \max \frac{\Sigma_{j \in D^f}(H_j^f - g_{s(j)})}{\Sigma_{j \in D^f} \; (g_{s(j)} - c_{V,j})} \\ &\leq \max_{j \in D^f} \frac{H_j^f - g_{s(j)}}{g_{s(j)} - c_{V,j}}. \end{aligned}$$

Note $H_j^f = \Sigma_{i \in V_s, i \in P_j^f} c_{i,j} + \Sigma_{i \in V - V_s} p_{i,j}^f$. By Lemma 1, payment to node i is based on i's direct help to the set of data which have i on their preservation path. Thus $\Sigma_{i \in V - V_s} p_{i,j}^f = \Sigma_{i \in P_j^* / V_s} p_{i,j}^f$. The above inequality becomes $L \leq \max_{j \in D^f} \frac{\Sigma_{i \in P_j^* / V_s}(c_{V-i,j} - c_{V,j}) + (\Sigma_{i \in P_j^* / V_s} c_{i,j} + c_{s(j),j}^*) - g_{s(j)}}{g_{s(j)} - c_{V,j}}$

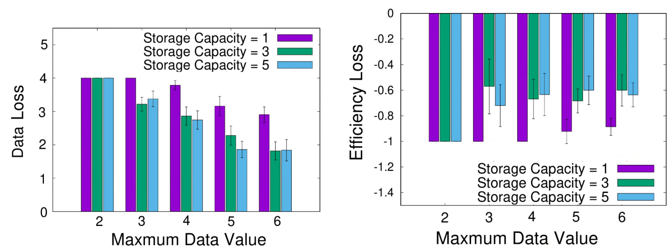
$$\begin{aligned} &= \max_{j \in D^f} \frac{\Sigma_{i \in P_j^* / V_s}(c_{V-i,j} - c_{V,j}) + c_{V,j} - g_{s(j)}}{g_{s(j)} - c_{V,j}} \\ &\leq \max_{j \in D^f} \frac{\Sigma_{i \in P_j^* / V_s}(c_{V-i,j} - c_{V,j}) + c_{V,j} - c_{V-P_j^*,j}}{c_{V-P_j^*,j} - c_{V,j}} \\ &\overset{\text{Lem. 2}}{\leq} \max_{j \in D^f} \frac{\Sigma_{i \in P_j^* / V_s}(c_{V-P_j^*,j} - c_{V,j}) + c_{V,j} - c_{V-P_j^*,j}}{c_{V-P_j^*,j} - c_{V,j}} \\ &\leq \frac{(n-2)(c_{V-P_j^*,j} - c_{V,j}) + c_{V,j} - c_{V-P_j^*,j}}{c_{V-P_j^*,j} - c_{V,j}} = n - 3. \end{aligned}$$

The last equality holds because the number of nodes on the preservation path for data $j \in D^f$ (excluding the source node) can be at most $(n-2)$, since there exists at least one alternative route for j's preservation. ■

Second, we show that $L \geq -1$. By Lemma 1, $\Sigma_{j \in D^f} H_j^f = \Sigma_{j \in D^f} [\Sigma_{i \in V_s, i \in P_j^f} c_{i,j} + \Sigma_{i \in V - V_s, i \in P_j^f} p_{i,j}]$. Since $\Sigma_{i \in V - V_s, j \in P_j^f} p_{i,j} = \Sigma_{i \in V - V_s, j \in P_j^f} [c_{V-i,j} - c_{V,j} + c_{i,j}]$, we have $\Sigma_{j \in D^f} H_j^f = \Sigma_{j \in D^f} \Sigma_{i \in V, i \in P_j^f} c_{i,j} + \Sigma_{j \in D^f} \Sigma_{i \in V - V_s, i \in P_j^f} [c_{V-i,j} - c_{V,j}] = c_V + [c_{V-i} - c_V] \geq c_V$ by the cost minimization of MCF. Thus $L \geq \frac{c_V - \Sigma_{j \in D^f}(g_{s(j)})}{\Sigma_{j \in D^f} \; g_{s(j)} - c_V} = -1$. ■

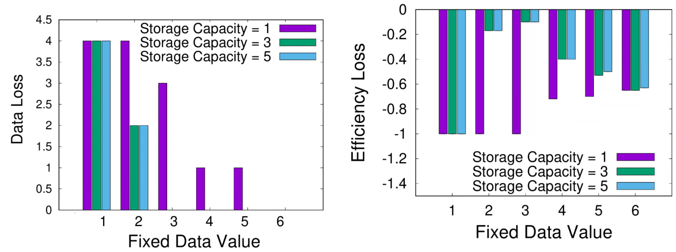
Lemma 3: When $m_i \geq d, \forall i \in V_r$, the upper bound of L in Theorem 1 is binding.

Proof: Consider a linear BSN shown in Fig. 2. Node 1 is the source node with 1 unit overflow data; nodes $n-1$ and n are two storage nodes, each with 1 unit storage. All the other nodes $2, 3, ..., n-2$ are transition nodes evenly located between node 1 and $n-1$. The distance from node 1 to node n is l and from node 1 to node $n-1$ is $l - \epsilon$, with $0 < \epsilon < l$. Suppose



(a) Data Loss. (b) Efficiency Loss.

Fig. 3. Random Data Values in Grid BSNs.



(a) Data Loss. (b) Efficiency Loss.

Fig. 4. Fixed Data Values in Grid BSNs.

data preservation only incurs transmitting energy cost; that is, $\epsilon_i^e = 0, \epsilon_i^a = 1, \forall i \in V$. The MCF algorithm offloads the data from node 1 to destination $n-1$ through the other nodes $2, 3, ..., n-2$, achieving minimized total cost of $a(l-\epsilon)$. Suppose $g_1 = l$. Thus, the data value is no less than the global replacement path cost (which is from the node 1 to node n and costs l), and the data will be preserved.

The VCG payment to each node $i \in 2, 3, ..., n-2$ who relays the data is $c_{V-i} - c_V + c_i = l - (l-\epsilon) + \frac{l-\epsilon}{n-2} = \epsilon + \frac{l-\epsilon}{n-2}$. The VCG payment to the storage node $n-1$ is $c_{V-i} - c_V + c_i = l - (l-\epsilon) = \epsilon$. And the cost to the source node is $\frac{l-\epsilon}{n-2}$. Therefore, the path payment is $H^f = \frac{l-\epsilon}{n-2} + [\epsilon + \frac{l-\epsilon}{n-2}](n-3) + \epsilon = l + (n-3)\epsilon$. The efficiency loss is $L = \frac{H^f - g_1}{g_1 - c_V} = \frac{l+(n-3)\epsilon - l}{l-(l-\epsilon)} = n-3$. ∎

V. Simulation Results

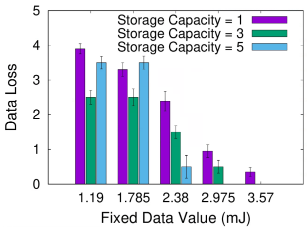
Simulation Setup. We write our simulator in Python to validate our theoretical results. For the MCF algorithm, we use NetworkX [1], a Python package for network analysis. We conduct extensive simulations on both grid BSNs and random BSNs. In grid BSNs, we first focus on 4×4 grid BSNs with 16 sensor nodes to visualize our theoretical analysis; we then use the 10×10 grid of 100 nodes to show that Data-VCG is still efficient in large-scale BSNs. In both cases, five nodes are randomly selected as source nodes, each with one data item; six are storage nodes, each having 1, 3, or 5 storage space units. The rest are transition nodes, each with zero storage spaces.

For the random BSNs, 30 sensor nodes are randomly placed in an area of $100m \times 100m$. We set the transmission range of sensor nodes as 30m. Among all the 30 sensor nodes, there are 5 data nodes (each with one overflow data) and six storage nodes (each with one storage capacity); the rest are transition nodes. Each data point averages 20 simulation runs, and the error bars indicate 95% confidence intervals.

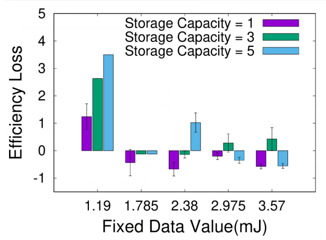
Grid BSNs. We first study grid BSNs wherein each data packet's value is a random number between 0 and a maximum data value. Fig. 3(a) shows that with the increase of the maximum data values, the data loss in the network decreases. This is because the more valuable the data, the more possible it gets preserved for data efficiency in the BSN. Fig. 3(b) shows that the efficiency loss ratio of Data-VCG is negative, meaning that there is a surplus of the system (i.e., total preserved data value exceeds total payment). This is because, in the setup of our grid BSNs, each node's marginal contribution to data preservation is relatively small, leading to a relatively small payment to each helping node. Moreover, the efficiency loss ratio may increase while increasing the maximum data values. When more data packets are preserved, the system efficiency gain increases; therefore, the negative ratio between budget imbalance and efficiency gain tends to increase.

Next, we study the scenario of fixed data values, wherein all the data packets have the same values. Fig. 4(a) shows that when data value increases, the dropped data decreases independently of each storage node's storage capacity. Moreover, the number of dropped data also lowers at the higher storage capacity of each storage node for a given data value. Both results are pretty intuitive. Fig. 4(b) shows that how the value of each data affects the efficiency loss is more complicated. On the one side, increasing data value increases the efficiency gain. On the other side, an increase in data value means more preserved data, which tightens the preservation capacity of the network. As a result, the total payment could get more prominent. Since the total preserved data value also increases, the directional change in budget imbalance is generally ambiguous. Fig. 4(b) shows that when each storage node has only 1 unit storage capacity, the absolute value of the efficiency loss ratio drops due to the higher efficiency gain. However, when each storage node has more storage capacity (3 and 5 units), it drops and then increases in data value. The implication is that when more and more data are preserved due to the spacious storage capacity, the increase in preserved data value increases the absolute value of the budget imbalance, resulting in a more significant system surplus and total value efficiency loss ratio.

Random BSNs. Fig. 5 shows our results of fixed data values in randomly generated BSNs. To enable data values to be comparable to the energy cost, we have to assume that some energy consumption equals one unit of data value. In particular, we treat the total energy consumption of sending and receiving one data packet of 512B 30m away, which is 1.19mJ, as one unit of data value. We vary the data values as 1, 1.5, 2, 2.5, and 3 units, as shown in Fig. 5. In contrast to Fig. 4, which shows that the efficiency loss of the Data-VCG



(a) Data Loss.



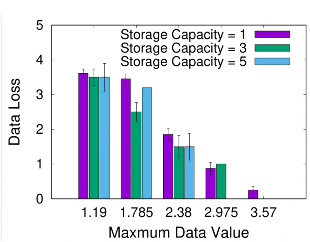
(b) Efficiency Loss.

Fig. 5. Fixed Data Values in Random BSNs of $200m \times 200m$.

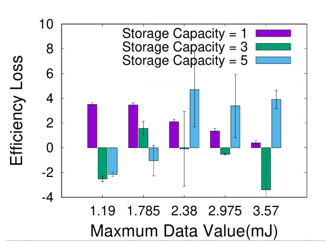
is negative, Fig. 5(b) shows that for a relatively small data value, there is a positive efficiency loss ratio, meaning that there is system budget deficit (total payment exceeds total preserved data value). The reason is that in random BSNs, there can be non-storage nodes of huge marginal contribution to the data preservation, leading to significant payments to those important non-source nodes, and hence a positive budget imbalance and a positive efficiency loss ratio. When increasing data values, as the total preserved data values also increase, it happens again that there is a system surplus and the efficiency loss ratio becomes negative (although the negative values are more significant than -1 compared to Fig. 4). Fig. 6 are the results for random data values, which shows the same trend as in Fig. 5.

VI. **Conclusion and Future Work**

We propose a data preservation game called Data-VCG for BSNs, which can model many emerging sensing applications. Preserving sensing data directly in the distributed sensor network is critical to make BSN scalable and energy efficient. Data-VCG integrates the data values with the classic VCG mechanism; it motivates the participation of all the sensor nodes and achieves a performance guarantee for worst-case budget imbalance. Our extensive simulations under different network parameters show that Data-VCG achieves both preservation and data efficiency. As MCF algorithms can not differentiate flows of different overflow packets, one technical challenge faced by Data-VCG is that it can not tell different data packets apart from the same source node. Integrating the MCF algorithms and Data-VCG at the flow level to compute the payment for each data packet becomes a new and challenging problem.



(a) Data Loss.



(b) Efficiency Loss.

Fig. 6. Random Data Values in Random BSNs of $200m \times 200m$.

ACKNOWLEDGMENT

This work was supported by NSF Grant CNS-2131309.

REFERENCES

[1] Networkx: Network analysis in python, 2022. https://networkx.org/.
[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., 1993.
[3] T. AlSkaif, M. G. Zapata, and B. Bellalta. Game theory for energy efficiency in wireless sensor networks: Latest trends. *Journal of Network and Computer Applications*, 54:33 – 61, 2015.
[4] A. Attiah, M. Amjad, M. Chatterjee, and C. Zou. An evolutionary routing game for energy balance in wireless sensor networks. *Comput. Netw.*, 138:31–43, 2018.
[5] S. Basagni, L. Boloni, P. Gjanci, C. Petrioli, C. A. Phillips, and D. Turgut. Maximizing the value of sensed information in underwater wireless sensor networks via an autonomous underwater vehicle. In *Proc. of INFOCOM*, 2014.
[6] Y. Chen, J. Ly, and B. Tang. Voluntary data preservation mechanism in base station-less sensor networks. In *12th EAI International Conference on Game Theory for Networks (GameNets 2022)*.
[7] Y. Chen and B. Tang. 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)*, 2016.
[8] M. Doudou, J. M. Barcelo-Ordinas, D. Djenouri, J. Garcia-Vidal, A. Bouabdallah, and N. Badache. Game theory framework for mac parameter optimization in energy-delay constrained sensor networks. *ACM Trans. Sen. Netw.*, 12(2), May 2016.
[9] S. Eidenbenz, G. Resta, and P. Santi. 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)*, 2005.
[10] L. Terray et al. From sensor to cloud: An iot network of radon outdoor probes to monitor active volcanoes. *Sensors*, 20(10), 2020.
[11] A. V. Goldberg. An efficient implementation of a scaling minimum-cost flow algorithm. *J. Algorithms*, 22:1–29, 1997.
[12] S. Hsu, Y. Yu, and B. Tang. dre^2: Achieving data resilience in wireless sensor networks: A quadratic programming approach. In *Proc. of IEEE MASS*, 2020.
[13] J. Jang and F. Adib. Underwater backscatter networking. In *Proc. of the ACM SIGCOMM*, 2019.
[14] Hervé Moulin. Almost budget-balanced vcg mechanisms to assign multiple objects. *Journal of Economic Theory*, 144(1):96–119, 2009.
[15] N. Nisan. Algorithms for selfish agents: Mechanism design for distributed computation. *STACS 1999. LNCS, Meinel, C., Tison, S. (eds.)*, 1563, 1999.
[16] S. Sengupta, M. Chatterjee, and K. Kwiat. A game theoretic framework for power control in wireless sensor networks. *IEEE Transactions on Computers*, 59(2):231–242, 2010.
[17] B. Tang. dao^2: Overcoming overall storage overflow in intermittently connected sensor networks. In *Proc. of IEEE INFOCOM*, 2018.
[18] B. Tang, N. Jaggi, H. Wu, and R. Kurkal. Energy-efficient data redistribution in sensor networks. *ACM Trans. Sen. Netw.*, 9(2):11:1–11:28, April 2013.
[19] Zhe Wang, Tansu Alpcan, Jamie S. Evans, and Subhrakanti Dey. Truthful mechanism design for wireless powered network with channel gain reporting. *IEEE Transactions on Communications*, 67(11):7966–7979, 2019.
[20] A. Wichmann, T. Korkmaz, and A. S. Tosun. Robot control strategies for task allocation with connectivity constraints in wireless sensor and robot networks. *IEEE Transactions on Mobile Computing*, 17(6):1429–1441, 2018.
[21] F. Wu, K. Gong, T. Zhang, G. Chen, and C. Qiao. Como: A game-theoretic approach for joint multirate opportunistic routing and forwarding in non-cooperative wireless networks. *IEEE Transactions on Wireless Communications*, 14(2):948–959, 2015.
[22] Y. Yu, S. Hsu, A. Chen, Y. Chen, and B. Tang. Truthful and efficient data preservation in base station-less sensor networks. *Accepted for publication in ACM Transactions on Sensor Networks*, 2023.