Truthful and Optimal Data Preservation in Base Station-less Sensor Networks: An Integrated Game Theory and Network Flow Approach

YUNING YU, SHANGLIN HSU, and ANDRE CHEN, California State University Dominguez Hills
YUTIAN CHEN, California State University Long Beach
BIN TANG, California State University Dominguez Hills

We aim to preserve a large amount of data generated inside *base station-less sensor networks* (BSNs) while considering that sensor nodes are selfish. BSNs refer to emerging sensing applications deployed in challenging and inhospitable environments (e.g., underwater exploration); as such, there do not exist data-collecting base stations in the BSN to collect the data. Consequently, the generated data has to be stored inside the BSN before uploading opportunities become available. Our goal is to preserve the data inside the BSN with minimum energy cost by incentivizing the storage- and energy-constrained sensor nodes to participate in the data preservation process. We refer to the problem as DPP: data preservation problem in the BSN. Previous research assumes that all the sensor nodes are cooperative and that sensors have infinite battery power and design a minimum-cost flow-based data preservation solution. However, in a distributed setting and under different control, the resource-constrained sensor nodes could behave selfishly only to conserve their resources and maximize their benefit.

In this article, we first solve DPP by designing an integer linear programming (ILP)-based optimal solution without considering selfishness. We then establish a game-theoretical framework that achieves provably truthful and optimal data preservation in BSNs. For a special case of DPP wherein nodes are not energy-constrained, referred to as DPP-W, we design a data preservation game DPG-1 that integrates algorithmic mechanism design (AMD) and a more efficient minimum cost flow-based data preservation solution. We show that DPG-1 yields dominant strategies for sensor nodes and delivers truthful and optimal data preservation. For the general case of DPP (wherein nodes are energy-constrained), however, DPG-1 fails to achieve truthful and optimal data preservation. Utilizing packet-level flow observation of sensor node behaviors computed by minimum cost flow and ILP, we uncover the cause of the failure of the DPG-1. It is due to the packet dropping by the selfish nodes that manipulate the AMD technique. We then design a data preservation game DPG-2 for DPP that traces and punishes manipulative nodes in the BSN. We show that DPG-2 delivers dominant strategies for truth-telling nodes and achieves provably optimal data preservation with cheat-proof guarantees. Via extensive simulations under different network parameters and dynamics, we show that our games achieve system-wide data preservation solutions with optimal energy cost while enforcing truth-telling of sensor nodes about their private cost types. One salient feature of our work is its integrated game theory and network flows approach. With the observation of flow level sensor node behaviors provided by the network

This work was supported in part by NSF Grants CNS-2131309 and CNS-1911191.
Authors' addresses: Y. Yu, S. Hsu, A. Chen, and B. Tang, California State University Dominguez Hills, 1000 E. Victoria Street, Carson, CA, 90747; e-mails: {yyu19, shsu10, achen12}@toromail.csudh.edu, btang@csudh.edu; Y. Chen, California State University Long Beach, 1250 Bellflower Blvd, Long Beach, CA; e-mail: Yutian.Chen@csulb.edu.

1550-4859/2023/10-ART5 $15.00
https://doi.org/10.1145/3606263

flows, our proposed games can synthesize "microscopic" (i.e., selfish and local) behaviors of sensor nodes and yield targeted "macroscopic" (i.e., optimal and global) network performance of data preservation in the BSN.

CCS Concepts: • **Networks** → **Network algorithms**; **Network economics;**

Additional Key Words and Phrases: Base station-less sensor networks, data preservation, energy-efficiency, network flow (minimum cost flow and maximum flow), integer linear program, game theory, algorithmic mechanism design

ACM Reference format:
Yuning Yu, Shanglin Hsu, Andre Chen, Yutian Chen, and Bin Tang. 2023. Truthful and Optimal Data Preservation in Base Station-less Sensor Networks: An Integrated Game Theory and Network Flow Approach. *ACM Trans. Sensor Netw.* 20, 1, Article 5 (October 2023), 40 pages.
https://doi.org/10.1145/3606263

1 INTRODUCTION

Base Station-less Sensor Networks. Wireless sensor networks are formed by a large number of small-size, battery-powered and resource-constrained, and spatially dispersed nodes with sensing, computing, and communication capabilities [99]. Since their inception more than two decades ago, they have continuously evolved into various new forms on a global scale, such as the Internet of Things [91] and mobile crowdsensing (MCS) [49, 97] and permeate many parts of our lives including healthcare monitoring [65], smart home and cities [71], and smart farming [32].

In this article, we focus on some emerging sensing applications deployed in challenging environments, such as inaccessible or inhospitable regions or under extreme weather. Such applications include underwater or ocean sensor exploration [10, 30, 45, 56, 70, 103], wind and solar harvesting [53, 60], seismic sensor networks [68, 78], and volcano eruption and glacial melting monitoring [23, 79]. These sensing systems are designed to tackle some of the most fundamental problems facing human beings, including scientific exploration, disaster warnings, and climate change. As they are all deployed in challenging environments, it is not feasible to deploy high-power and high-storage data-collecting base stations in or near the sensing field. We refer to emerging sensor network applications without base stations as *base station-less sensor networks (BSNs)*. BSNs are in sharp contrast to the traditional sensor networks and the current IoT applications wherein base stations are always available in the field to collect sensory data.

Due to the base station-less nature of a BSN, one of its essential functions is to preserve large volumes of generated data inside the BSN. Later they can be collected by the periodic visits of uploading opportunities, including autonomous underwater vehicles (AUVs) [10, 31] and robots [44, 92, 98]. Figure 1 illustrates our BSN network model. In a BSN, some sensor nodes are close to the events of interest and are constantly generating sensory data, thus depleting their storage spaces. We refer to the sensor nodes with depleted storage spaces while still generating data as *data nodes* and the sensor nodes with available storage as *storage nodes*.[1] The newly generated data that can no longer be stored at storage-depleted data nodes is called *overflow data*. To avoid data loss, overflow data must be offloaded to storage nodes to be preserved and wait for the arrival of the uploading opportunities. The goal is to select storage nodes and offload the overflow data to them while minimizing the total energy consumption in this process. Those storage nodes selected to store overflow data are called *destination nodes*. We refer to the process of finding the destination nodes and offloading overflow sensory data from data nodes to these destination nodes as *data preservation in the BSN*.

[1]Sensor nodes that generate data but whose storage spaces are not full are considered storage nodes, as their storage can be used to store data from other sensor nods. Besides, a data node can relay overflow data packets from another data node.

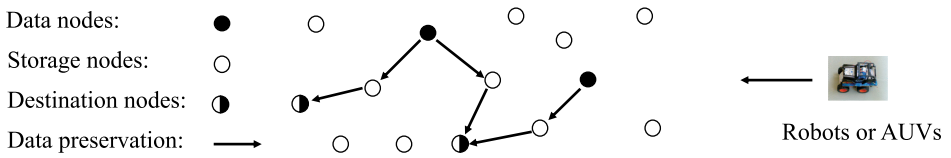


Fig. 1. The BSN model.

Network Flows-based Solutions. Existing research has proposed a suite of network flows-based algorithms to achieve different objectives in data preservation in the BSN [5, 19, 41, 42, 80, 82, 96]. Network flow problems [4, 34] (e.g., maximum flow, minimum cost flow, and multi-commodity flow) are a class of computational problems that study how to move objects (e.g., goods, vehicles, and network packets) within a flow network efficiently and cost-effectively. For example, when all the data packets can be preserved, Tang et al. [82] designed a *minimum cost flow*-based optimal algorithm to minimize the total energy consumption in data preservation. When not all the packets can be preserved due to the insufficiency of energy power of sensor nodes, Xue et al. [96] designed a *maximum weighted flow*-based optimal algorithm to maximize the total values of preserved data packets. When not all the packets can be preserved due to the insufficiency of storage spaces of sensor nodes, Tang et al. [80] proposed aggregating the data packets to reduce their sizes before offloading them to destination nodes. They modeled the energy-efficient data aggregation in BSNs as a new graph-theoretical problem called *multiple traveling salesmen placement problem* and designed a constant-factor approximation algorithm. Recently, Hsu et al. [42] proposed a quadratic programming-based algorithm to maximize data resilience in BSNs by preserving the overflow data for the maximum time.

Motivation. All the above network flows-based solutions for data preservation in BSNs assume that the sensor nodes are cooperative and willing to contribute their resources, including battery power and storage spaces, to the data preservation process. With the strides made in sensor network development and IoT applications over the past decade, these assumptions are no longer invalid [16]. First, in a global-scale and distributed decision-making environment, sensor nodes could be under the control of different users, each aiming to pursue its own self-interest and maximize its own benefit. For example, in the oil and gas industry, the IoT sensors that detect possible oil and gas leaks could belong to different companies with different business incentives [95]. Second, the technologically advanced sensor nodes could become more intelligent [25]. Unlike the traditional sensors that can only sense, compute, and communicate signals to an external system, the intelligent sensor can also perceive, reason, and learn. As such, the resource-constrained sensor nodes in the BSN can behave selfishly, only to conserve their own resources and have little incentive to participate in the data preservation. Finally, being selfish, sensor nodes can not only hold their critical information (e.g., storage capacities and energy costs of preserving the data packets) as *private information* and not report them, but also, in the worst case, misreport them to gain more utilities and maximize their own benefit in the data preservation process. The tension between node-centric selfishness and data-centric data preservation in the BSN, if not dealt with well, could impede the data preservation process and compromise the functions and missions of the aforesaid emerging sensing applications.

Our Contributions. In this article, we mitigate the above tension by carefully creating a truthful and optimal game theory framework for data preservation in the BSN. Focusing on strategic interaction among rational and selfish decision-makers, game theory is an ideal tool for analyzing

decentralized and self-organizing networks like BSNs. We make the following contributions to the article:

First, we formulate and solve a general data preservation problem in the BSN referred to as **DPP: data preservation problem**. The goal of DPP is to minimize the total energy consumption in data preservation while considering that sensor nodes have limited storage capacity and energy power. We design an integer linear program (ILP) to solve DPP optimally. In addition, we design another ILP to solve a relevant *DPP feasibility problem,* referred to as DPP-F. Given any instance of DPP with the energy-constrained sensor nodes, DPP-F checks if all the overflow data packets can be successfully preserved inside the BSN or not; that is, if the data preservation is feasible for this DPP instance. In contrast, existing work only solved a special case of DPP wherein nodes have infinite energy power [82, 83]. We refer to this special case as DPP-W in the article.

Second, we consider the selfishness of sensor nodes and design a non-cooperative game for DPP-W. We refer to it as **DPG-1: data preservation game-1**. DPG-1 is based on *algorithmic mechanism design (AMD)* [61–63], a subfield of game theory and network optimization. AMD designs computationally efficient games, including strategies and payoffs, such that individual players, motivated solely by self-interest, achieve a good system-wide solution. Therefore, AMD techniques are suitable for achieving truthful and optimal data preservation. In particular, we consider Vickrey-Clark-Groves (VCG) mechanism [17, 39, 84]. We identify the challenges of directly applying VCG techniques in our BSN model. We prove that with these challenges, DPG-1 can still (a) guarantee that truth-telling its private cost information (i.e., energy costs of receiving, saving, and transmitting packets) is a node's dominant strategy and (b) sufficiently motivate each node to participate in optimal data preservation. Thus, DPG-1 achieves a truthful and optimal system-wide data preservation solution with self-interested sensor nodes.

Third, we show that in DPP, wherein nodes have a finite amount of energy, DPG-1 can no longer provide truth-telling and optimal data preservation. We find that, unlike DPP-W, a sensor node can manipulate the VCG model and lie about its costs to receive more utilities in DPP. In doing so, a lying node can receive more data packets than it can possibly process, resulting in data loss and sub-optimal data preservation in the BSN. We thus design another data preservation game, viz., DPG-2 to fix the flawed VCG model used in DPG-1. We show via proof and simulations that DPG-2 delivers truth-telling as a node's dominant strategy and achieves optimal data preservation in the BSN while accommodating the selfish behavior of energy-constrained sensor nodes.

Finally, in contrast to many existing works that applied game theory and related techniques to solve sensor networking problems, our work takes a network flow approach to facilitate the game design and game-theoretical analysis. Utilizing data preservation flows computed by push-relabel-based and ILP-based minimum cost flow algorithms, our designed games can synthesize "microscopic" (i.e., selfish and local) behaviors of sensor nodes and transform them into targeted "macroscopic" (i.e., optimal, global, and fault-tolerant) network performance of data preservation in the BSN. This integrated game theory and network flow approach helps us to understand the selfish behaviors of sensor nodes quantitatively to achieve truthful and optimal data preservation of the BSN. Consequently, all our game-theoretical findings can be validated by our progressive experiments. Table 1 summarizes our data preservation games, corresponding data preservation problems, and network flow techniques.

Article Organization. The rest of the article is organized as follows: Section 2 reviews all the related work and sets the stage for the contributions of our work. Section 3 formulates the DPP and introduces its energy model and the cost parameters of sensor nodes. Section 4 proposes the centralized network flow-based algorithmic solutions: push-relabel-based MCF for DPP-W and ILP-based MCF for DPP, respectively. Section 5 considers selfish sensor nodes in DPP-W and

Table 1. Summary of Data Preservation Games (DPGs) and Their Corresponding Data Preservation Problems with Underlying Network Flow Techniques

Data preservation games	Data preservation problems	Network flow techniques
DPG-1 (Section 5)	DPP-W (Section 3)	Push-relabel-based MCF, viz., PR-MCF (Section 4.1)
DPG-2 (Section 6)	DPP and DPP-F (Section 3)	ILP-based MF and MCF, viz., ILP (A) and (B) (Section 4.2)

ILP: integer linear programming; MF: maximum flow; MCF: minimum cost flow; PR-MCF: push-relabel-based MCF.

designs data preservation game DPG-1 that provably provides optimal and truthful data preservation solutions. Section 6 shows that DPG-1 does not work for the general case of DPP, finds the cause of such, and designs a game named DPG-2 that provably provides an optimal and truthful solution to DPP. Section 7 validates our game designs via progressive simulation results, starting from the macroscopic network characteristics of data preservation, progressing deeper into microscopic views of data packet flows, and finally pinpointing the game-theoretical behaviors of the selfish sensor nodes. Section 8 concludes the article with a discussion of future work.

2 RELATED WORK

Game theory techniques have been extensively applied to solve research problems in computer networks in general, and wireless ad hoc and sensor networks in particular [6, 11, 66, 75]. There are four main classes of game theoretical techniques that have been employed in the existing research. *Non-cooperative game theory* studies strategies between interactions among individual competing players, with Nash Equilibrium (NE) being its solution concept that describes a steady-state condition for the players. *Cooperative game theory* models situations where players form groups (i.e., coalitions) rather than acting individually. One of its central notations is *the core*, which is the payoff allocation that no group of players has the incentive to leave its coalition to form another coalition. The third one is called *cooperation enforcement games*, wherein selfish players are incentivized to cooperate to maximize the social optimal of the system. The fourth is *mechanism design*, also called reverse game theory, which takes an objectives-first approach to design economic mechanisms or incentives to motivate rational and selfish players toward desired objectives. In this article, we mainly take a mechanism design approach, as it suits our goal of motivating selfish nodes to achieve optimal total data preservation cost in the BSN. Below, we review the mechanism design work in both general networks and wireless ad hoc networks, the game-theoretical techniques in sensor networks, including mobile crowdsensing, and the existing data preservation research in BSNs.

Mechanism Design in General Networks. The seminal work by Nisan and Rosen [61–63] introduced the framework of applying mechanism design to solve algorithmic problems where participants are selfish. In particular, they showed that the classic VGC mechanism provides a truthful solution for *utilitarian problems* where the objective function is the sum of all agents' valuations (i.e., costs). They studied shortest path routing between one pair of source and destination nodes where the edges are strategic players. Feigenbaum et al. [26] treated nodes as strategic players instead and considered multiple source-destination pairs. They designed distributed algorithms to compute the lowest-cost routes and payments for transit nodes on all routes. In their follow-up work [27], they studied cost-sharing multicast, wherein a source communicates with multiple receivers along a multicast tree. They assumed the link cost is publicly known and analyzed the network complexity of two mechanisms *marginal cost* and *Shapley value*. Such incentives for sharing as well as profit-maximization were also investigated in peer-to-peer networks [35, 76], content distribution networks [8, 20], and mobile cloud computing [14].

Mechanism Design in Ad Hoc Networks. Zhong et al. [105] was one of the first to apply mechanism design to the mobile ad hoc network and designed a practical cryptographic model for payment delivery in such networks. Unlike the previous works where each player must have a private cost type, it assumed the information held by each player is not totally private (as each player maintains a public-private key pair). It also assumed all the transit nodes get the same payment amount, indifferent from its cost in its payment model. Anderegg and Eidenbenz [7] focused on the cost-of-energy parameters and the route discovery and recovery in a mobile ad hoc setting and designed a variation of the VCG mechanism that achieved truthful and cost-efficient ad hoc routing. They assumed that the source nodes were always truthful. Wang et al. [86, 89] studied multicast and showed that when all the costs are private information, there is no efficient VCG-based mechanism that guarantees truth-telling due to the NP-hardness of the problem. Their designed multicast protocols are minimum spanning tree-based approximation algorithms wherein each agent maximizes its profit when it truthfully reports its cost. However, their mechanism only worked on tree-based structures. Chen et al. [12] considered the multicast under unreliable wireless links and proposed a distributed game-based algorithm that achieves Nash Equilibrium. Recently, some research [69, 93, 94] deviated from traditional ad hoc routing and considered opportunistic routing wherein nodes overhear the packets to participate in packet forwarding. They designed games that achieved social efficiency and Pareto-efficient Nash equilibrium with faithfulness as a given property. Felegyhazi et al. [28] studied the Nash equilibria of packet forwarding strategies in wireless ad hoc networks. They showed the equilibrium conditions for cooperative and noncooperative strategies using the theory of iterative games.

Different from the above payment (or price)-based incentive mechanism, other works in ad hoc networks proposed to use reputation or acknowledgment systems to detect routing misbehavior and to motivate cooperation among selfish players [46, 52, 55, 58]. In particular, Li and Shen [52] studied the incentive strategies in both reputation and price-based systems and observed that combining a reputation and price-based system is a promising method to provide strong incentives.

Game-theoretical Techniques in Sensor Networks. In contrast to ad hoc network research that mainly focuses on routing between source and destination communication pairs (i.e., *one-to-one* model), sensor networks sense the environment and send the sensory data at multiple nodes back to the base station (i.e., *many-to-one* model) for storage, viewing, and analysis. Existing research works applying game theory to solve various sensor network problems include topology (and power) control, connectivity, and coverage [43, 73, 74], duty cycling and media access control (MAC) protocols [2, 13, 21, 100], data routing (i.e., packet forwarding) and aggregation [9, 48, 59, 64, 85], security and threats prevention [3, 54, 101], and task allocation [6, 75]. As data preservation in BSNs is essentially a packet forwarding mechanism, below, we review all the existing works that apply game theory to study data routing in sensor networks.

Attiah et al. [9] proposed an evolutionary routing game for energy balance in wireless sensor networks. The goal was to reduce the load and avoid collisions on the most used routes in a distributed manner. They derived a mixed strategy Nash equilibrium and showed that it also achieves fairness. Voulkidis et al. [85] proposed a coalitional game-theoretic scheme that maximizes the network lifetime. It employed the spatial correlation among sensory data to reduce the transmitted data packets. Kannan et al. [48] focused on length and energy-constrained information routing in sensor networks and proposed several payoff models and utility functions. They showed that for each utility function, there exists a Nash equilibrium. Niyato et al. [64] studied the solar-powered sensor network that uses a sleep and wakeup strategy for energy conservation. They modeled nodes' sleep and wake-up strategies as a bargaining game and derived the Nash equilibrium as the game's solution.

A careful study reveals that no existing work addressed data preservation in BSNs. Besides, they mainly focused on incentivizing the nodes on the shortest path between the source and well-known destination (e.g., base stations) or minimum spanning tree covering them. As shown in all the above literature, such paths are one-to-one, many-to-one, or one-to-many (i.e., multicast) communication models. Different from all existing work, our BSN model is indeed an *any-to-any* model where data from *any* data nodes can be offloaded to *any* storage nodes in the BSN. As such, it needs to decide the destination for each data packet and the routing from its data node to the destination, which prompts a new challenge that has yet to be tackled by any previous game theory research. To tackle this challenge, we convert the data preservation problem into a minimum-cost network flow problem [4]. Minimum cost flow generalizes the shortest path and minimum spanning tree and can find destinations and the routing to the destinations for the data packets in the BSN. Compared to the existing approach, network flow techniques are better suited for BSNs to achieve more robust data preservation. However, how game theory plays a role in nodes' behaviors in network flow-based problems remains largely unexplored.

Incentive Mechanisms in Crowdsensing. In recent years, incentive mechanisms have been applied in mobile crowdsensing to exploit the "wisdom" of many mobile users [102]. They mainly focused on designing economic mechanisms to stimulate user participation and to incentivize users to provide more accurate sensing data [22, 36, 37, 47, 50, 67, 97, 104]. Our data preservation incentive model is similar to the crowdsensing model in that both motivate nodes or users to contribute their resources (i.e., storage, computations, and energy); however, with different network models and application goals. In mobile crowdsensing, the main task is enlisting users to sense the surrounding environment for monitoring traffic, health, or social behavior. It mainly used the Stackelberg game [29], where the leader moves first, and then the followers move. However, a base station (i.e., a centralized server) still collects all the sensed data and executes the tasks. In data preservation in BSNs, the main task is to enlist the nodes to preserve the sensed data, as no base station is available to collect and store the data. Besides, our model has no clear distinction between leaders and followers, as a data node can be either a leader (to offload its overflow data) or a follower (to relay data for other data nodes). As a result, Stackelberg games are unsuitable for solving our data preservation problem.

Data Preservation in the BSN. With its theoretical rigor and powerful applicability, network flows and their related algorithms have been widely used in computer science, operations research, and engineering [4, 34, 77]. Our previous data preservation research in BSNs has indeed used the minimum cost flow to model the energy optimization [19, 82] and to achieve fault-tolerance [81], designed a maximum weighted flow algorithm to preserve data packets of different values [96], uncovered a suite of new multiple traveling salesman placement problems for data aggregation [80, 83], and designed a quadratic programming solution to maximize survival time of preserved data packets [42]. Our BSN model, with its theoretical roots in network flows and its simplicity, may inspire new architectures for future network infrastructures and applications. It is a very general information producer and consumer model that has not been adequately explored in any other context.

Chen et al. [15] designed a computationally efficient and truthful VCG-based data preservation game for BSNs assuming that storage nodes are selfish and all sensor nodes have infinite energy. We show that when storage nodes have a limited energy power, the VCG mechanism proposed in Reference [15] is no longer truthful. They further considered data packets to have different values, and that both data and storage nodes are selfish [15]. They designed a voluntary data preservation game for all the nodes in the BSN. Recently, Ly et al. [57] gave an analytical analysis of the performance guarantee of this game, showing that under certain conditions, its worst-case

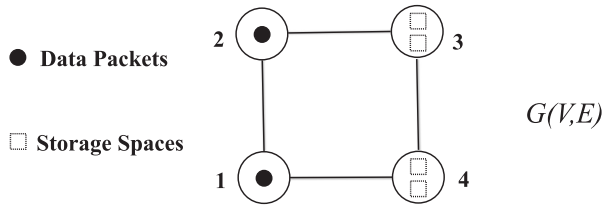


Fig. 2. A BSN $G(V, E)$ with two data nodes 1 and 2, each having one data packet to offload, and two storage nodes 3 and 4, each having two storage spaces.

budget imbalance is at most $n3$ times the efficiency gain, where n is the number of sensor nodes in the network. Rivera [72] designed a suite of data preservation games that achieve NEs, and they analyze their efficiency loss in terms of the price of anarchy and the price of stability [51].

3 PROBLEM FORMULATION OF DPP

Network Model. We represent the BSN as an undirected connected graph $G(V, E)$, where $V = \{1, 2, \ldots, n\}$ is the set of n sensor nodes and E is the set of m edges. In the BSN, some sensor nodes are close to the event of interest and generate many data packets, thus having already depleted their storage spaces; they are referred to as *data nodes*. Their newly generated data packets that cannot be stored locally are *overflow data packets*. W.L.O.G., there are k data nodes $V_s = \{1, 2, \ldots, k\}$, where data node i currently has d_i overflow data packets, each is a bits. Let $d = \sum_{i=1}^{k} d_i$ and $D = \{D_1, D_2, \ldots, D_d\}$ be this set of d overflow data packets. Let $s(j) \in V_s$, $1 \leq j \leq d$ denote D_j's data node. The rest sensor nodes in $V - V_s = \{k + 1, k + 2, \ldots, n\}$ are referred to as *storage nodes*, as they have local storage spaces available. Let m_i be the available free storage space (in bits) at storage node $i \in V - V_s$. Note that sensor nodes that have generated some sensory data locally but have not depleted their storage spaces are considered storage nodes. We assume there is a central authority existing in the system to compute the data preservation solution and design the corresponding mechanism, as well as to pay the rewards to the sensor nodes.

Because of the storage depletion of the data nodes, their generated overflow data packets must be offloaded to some storage nodes to be preserved and to wait for the arrival of uploading opportunities. The process of offloading packets from data nodes to storage nodes is referred to as *data preservation* in a BSN. Figure 2 shows a BSN $G(V, E)$ with two data nodes 1 and 2, each having one overflow data packet to offload, and two storage nodes 3 and 4, each having two storage spaces available. When the sensor nodes are selfish, offloading data packages from data nodes to storage nodes may cause data privacy problems, which can solve by privacy-preserving incentive mechanisms such as PrivAim [87].

Energy Model. Each sensor node $i \in V$ has an initial energy level E_i. We augment the first-order radio model [40] and consider three different kinds of energy consumption incurred in data preservation.

- *Receiving Energy* E_i^r. When node i receives an a-bit data packet from one of its one-hop neighbors, the amount of *receiving energy* it spends is $E_i^r = a \cdot \epsilon_i^e$. Here, $\epsilon_i^e = 100nJ/bit$ is the energy consumption per bit on the receiver or transmit circuit of node i. We refer to it as *receiving parameter*. Note that E_i^r only depends on the size of the data i receives, not the distance between it and the sender.
- *Transmission 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 *transmission energy* spent by i is $E_i^t(j) = a \cdot \epsilon_i^a \cdot l_{i,j}^2 + a \cdot \epsilon_i^e$. Here, $\epsilon_i^a = 100pJ/bit/m^2$ is called *transmission parameter*, which is the energy

consumption of sending one bit on transmit amplifier of node i. Note that $E_i^t(j)$ includes the energy consumption on both i's transmit amplifier and transmit circuit and depends on the distance between nodes i and j and the data size.

- *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 referred to as the *storing parameter* and is the energy consumption of storing one bit at node i; its default value is $100nJ/bit$. Note: E_s only depends on the size of the data it stores.

When a node relays (i.e., receives then transmits) a data packet, it still needs to buffer it in its local memory before transmitting it to the next node. However, we assume the energy cost of storing on the memory is much smaller than the energy cost of storing on the storage, thus is negligible.

Cost Parameters of Sensor Nodes. We refer to $\{\epsilon_i^a, \epsilon_i^e, \epsilon_i^s\}$ in the energy model as node i's *cost parameters*. Our energy model differs from the well-known first-order wireless radio model [40] and most existing research in two aspects. First, the first-order model does not consider storing energy parameterized by ϵ_i^s, as storing energy of a sensor node is usually considered negligible compared to transmission and receiving energy. However, Mathur et al. [1] examined the energy consumption of different currently available flash memory, a viable storage technology for low-power, energy-constrained wireless sensor networks. They found that read, write, and erase energy consumption per byte for Hitachi MultiMedia Cards (MMC) and NAND flash memory are 1.108 μJ and 0.062 μJ, respectively (which are equivalent to 139 nJ/bit and 8 nJ/bit, respectively). Therefore, the energy consumption of storing data packets on sensor nodes can not be neglected, especially considering that large amounts of data are generated and stored at sensor nodes in a BSN. Second, existing research (e.g., Reference [48]) assumes the energy cost of transmission only depends on distances between sensors, which implies that ϵ_i^a and ϵ_i^e have the same values for different sensor nodes i. However, Wang and Yang [88] pointed out that energy consumption is a function of the features of devices. Specifically, they found that the energy consumption on circuits varies significantly from state-to-state of a sensor device and among different types of sensor devices. Following this spirit, we assume that different sensor nodes could have different values of the same parameter. All three cost parameters ϵ_i^a, ϵ_i^e, and ϵ_i^s are node-dependent. Table 2 shows all the notations.

Problem Formulation of DPP. Define a *preservation function* as $p : D \rightarrow V - V_s$, indicating that a data packet $D_j \in D$ is offloaded from its data node $s(j) \in V_s$ to a storage node $p(j) \in V - V_s$ to be preserved. Let $P_j = \{s(j), \ldots, p(j)\}$ be the *preservation path* along which D_j is offloaded. Let $c_{i,j}$ denote node i's energy consumption in preserving D_j and $c_i = \sum_{j=1}^{d} c_{i,j}$ be the total energy consumption of node i in preserving all the data packets. $c_{i,j}$ can 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 of DPP is to find a preservation function p and P_j $(1 \le j \le d)$ to minimize the *total preservation cost* c in the BSN, where

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

under the storage constraint of storage nodes: $|\{j | 1 \le j \le d, p(j) = i\}| \cdot a \le m_i, \forall i \in V - V_s$, and the energy constraint of all sensor nodes: $\sum_{j=1}^{d} c_{i,j} \le E_i, i \in V$.

Table 2. Notation Summary

Notation	Description
V	$V = \{1, 2, \ldots, n\}$ is the set of n sensor nodes
V_s	$V_s = \{1, \ldots, k\}$ is the set of k data nodes, and
$V - V_s$	$V - V_s = \{k+1, l+2, \ldots, n\}$ is the set of $n-k$ storage nodes
d_i	Number of overflow data packets from data node $i \in V_s$
m_i	Storage capacity of storage node $i \in V - V_s$
D	$D = \{D_1, D_2, \ldots, D_d\}$ is the set of d overflow data packets
$s(j)$	The data node of $D_j \in D$
E_i	Initial energy level of sensor node i
$E_i^{'}$	Remaining energy level of sensor node i after data offloading
$E_i^t(j)$	Transmission energy spent by i to transmit one packet to j
E_i^r	Receiving energy spent by i to receive one data packet
E_i^s	Storing energy spent by i to store one data packet
ϵ_i^a	Transmission parameter of node i
ϵ_i^e	Receiving parameter of node i
ϵ_i^s	Storing parameter of node i
p	Data offloading function $p : D \rightarrow V - V_s$
P_j	The *offloading path* of data packet $D_j \in D$
$\sigma(i, j)$	Node i's successor node in P_j
$c_{i,j}$	Node i's energy cost of offloading data packet D_j
t_i	$t_i = \{\epsilon_i^e, \epsilon_i^a, \epsilon_i^s\}$ is the true private type of node i
$\tilde{t}_i$	Reported private type by node i
$x_{i,j}$	The amount of flows on edge (i, j) in flow networks for ILP
c_i	The true cost of node i
$\tilde{c}_V$	The minimum total preservation cost of the network when i reports its cost $\tilde{c}_i$
$c_{V-\{i\}}$	The minimum total preservation cost of the network when i is removed
$p_i(\tilde{t}_i, t_{-i})$	The payment received by node i when it reports $\tilde{t}_i$ in DPG-1
$\pi_i(\tilde{t}_i, t_{-i})$	Node i's utility when it reports $\tilde{t}_i$ in DPG-1
$p_i^x(\tilde{t}_i, t_{-i})$	The payment received by node i when it reports $\tilde{t}_i$ in DPG-2
$\pi_i^x(\tilde{t}_i, t_{-i})$	Node i's utility when it reports $\tilde{t}_i$ in DPG-2
$G'(V', E')$	The flow network used to solve DPP-W, where $E_i = \infty$
$G''(V'', E'')$	The flow network used to check if a DPP instance with E_i being finite is feasible (i.e., DPP-F)
$G'''(V''', E''')$	The flow network used to compute the minimum total data preservation cost when feasible

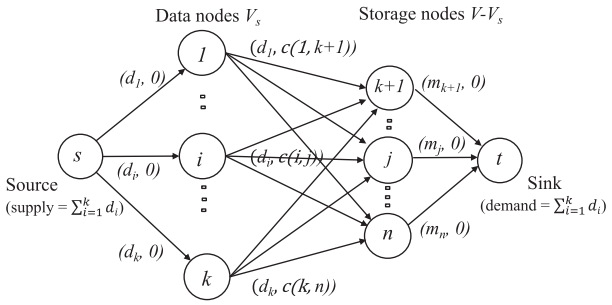
4 ALGORITHMIC SOLUTIONS OF DPP

We first consider a special case that sensor nodes are not energy-constrained (i.e., $E_i = +\infty$, $1 \leq i \leq n$) and refer to it as DPP-W. Tang et al. [82] have solved DPP-W as a minimum cost flow (MCF) problem, which is presented below for completeness. We then consider general DPP and solve it with an MCF-based ILP optimal solution.

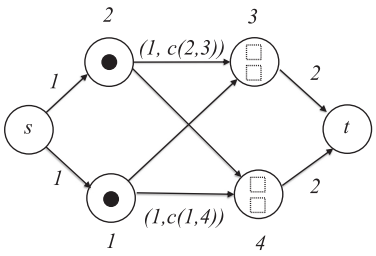
4.1 Solving DPP-W

Minimum Cost Flow (MCF) Solution. We formally introduce MCF as follows: Given a directed graph $G' = (V', E')$ with a source node s and a sink node t, each edge $(u, v) \in E'$ has a capacity $a(u, v)$ as well as a cost $d(u, v)$. Let $f(u, v)$ be the flow on edge $(u, v) \in E'$. The goal of MCF is to find a flow function f to minimize the total cost of transmitting y amount of flow from s to t, i.e., $\Sigma_{(u,v)\in E'}(d(u, v) \cdot f(u, v))$, subject to (a) capacity constraint: $f(u, v) \leq a(u, v), \forall (u, v) \in E'$, (b) flow conservation constraint: $\sum_{u \in V'} f(u, v) = \sum_{u \in V'} f(v, u)$, for each $v \in V' - \{s, t\}$, and (c) the net flow out of s and the net flow into t are both y.

Tang et al. [82] has proved that the DPP-W in BSN graph $G(V, E)$ is equivalent to the MCF problem in a flow network $G'(V', E')$, shown in Figure 3(a), that is properly transformed from the



(a) G'(V',E')



(b) G'(V',E') for the BSN in Fig. 2

Fig. 3. (a) shows the flow network $G'(V', E')$ transformed from BSN graph $G(V, E)$. (b) shows the flow network $G'(V', E')$ transformed from the BSN graph $G(V, E)$ in Figure 2.

$G(V, E)$. We refer to this transformation that constructs Figure 3(a) as Transformation I, which includes the following five steps:

Transformation I

(1) $V' = \{s\} \cup \{t\} \cup V$, where s is the source node and t is the sink node in the flow network.
(2) $E' = \{(s, i)\} \cup \{(i, j)\} \cup \{(j, t)\}$, where $i \in V_s$ and $j \in V - V_s$. Note that it is a complete bipartite graph between V_s and $V - V_s$.
(3) For each edge (s, i), set its capacity as d_i, the number of data packets at $i \in V_s$, and cost as 0. For each edge (j, t), set its capacity as m_j, the storage capacity of j, and the cost as 0.
(4) For each edge (i, j), set its capacity as d_i and cost as $c(i, j)$. Here, $c(i, j)$ is the minimum energy consumption sending one data packet from data node i to storage node j.
(5) Set the supply at s and the demand at t as $d = \sum_{i=1}^{k} d_i$, the total number of overflow data packets.

We have $|V'| = |V'| + 2$, $|E'| = |V| + |V_s| \cdot |V - V_s|$. The time complexity of Transformation I is $O(|V|^3)$, as we can use Floyd-Warshall all-pair shortest path algorithm [18] to compute $c(i, j)$ between any data node i and any storage node j. As an example, with this transformation, the BSN graph shown in Figure 2 is now converted to the flow network shown in Figure 3(b).

Next, we apply MCF algorithms on $G'(V', E')$ to find the minimum total data preservation cost. There are many optimal and efficient combinatorial MCF algorithms, including cycle canceling, successive shortest path and capacity scaling, and network simplex [4]. We indeed adopt the scaling push-relabel algorithm proposed in Reference [33]. We refer to this scaling push-relabel-based MCF algorithm as PR-MCF. PR-MCF has the highest performance codes available for network optimization and has worked well over many problem classes. 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 $G'(V', E')$ [33]. As $|V'| = |V'| + 2$, $|E'| = |V| + |V_s| \cdot |V - V_s|$ following Transformation I, PR-MCF takes $O(|V|^4 \cdot \log(|V| \cdot \max\{d_i, m_j\}))$ to solve DPP-W.

4.2 Solving DPP

In the general case of DPP, where sensor nodes have limited battery power, some sensors may exhaust their energy power during data preservation. This can cause network partition between data nodes and storage nodes and obstruct the data preservation process. Consequently, some data packets may not be offloaded successfully from their data nodes to some storage nodes to be preserved. When all the d overflow data packets can be offloaded, we say the data preservation is

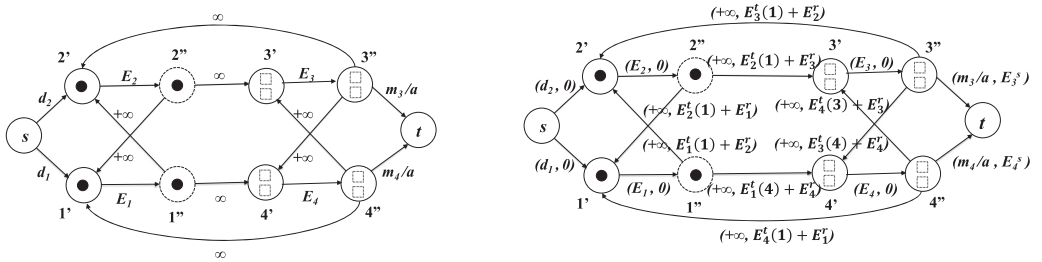


Fig. 4. (a): the flow network $G''(V'', E'')$ that solves DPP-F for the BSN in Figure 2. (b): the flow network $G'''(V''', E''')$ that finds the minimum preservation cost for the BSN in Figure 2, given that it is feasible.

feasible; otherwise, it is not feasible. Therefore, before solving DPP, we first study *data preservation feasibility problem*: Given any DPP instance, can all its d overflow data packets be offloaded from their data nodes into the BSN for data preservation due to energy constraints of sensor nodes? We refer to this problem as DPP-F and propose a maximum flow-based ILP solution below.

Note that infeasible data preservation could be attributed to either insufficient storage spaces or insufficient energy power of sensor nodes. In this article, we assume there are enough storage spaces in the BSN (i.e., $\sum_{i=k+1}^{n} m_i \geq d \cdot a$) thus, infeasible data preservation is attributed only to insufficient energy.[2]

4.2.1 Solving DPP-F. To solve DPP-F, we transform the BSN graph $G(V, E)$ to another flow network $G''(V'', E'')$. This transformation strives to represent the initial battery power E_i of sensor node i as an edge capacity so the energy constraint of sensor nodes can play a role in the data preservation process. When a sensor's energy is depleted, it can no longer participate in data preservation. This transformation, which constructs $G''(V'', E'')$ from $G(V, E)$ and is referred to as Transformation II, takes the following four steps:

Transformation II

(1) In $G(V, E)$, replace each undirected edge $(i, j) \in E$ with two directed edges (i, j) and (j, i). Set the capacities of all the directed edges as $+\infty$.
(2) Split node $i \in V$ into two nodes: *in-node* i' and *out-node* i''. Add a directed edge (i', i'') with a capacity of E_i, the initial energy level of node i. All the incoming directed edges of node i are incident on i', and all the outgoing directed edges of node i emanate from i''. Therefore, the two directed edges (i, j) and (j, i) in Step (1) are now changed to (i'', j') and (j'', i').
(3) Add a source node s, and connect s to the in-node i' of the data node $i \in V_s$ with an edge. Set the capacity of this edge as d_i, the number of data packets at data node i.
(4) Add a sink node t, and connect out-node j'' of the storage node $j \in V - V_s$ to t. Set its edge capacity as $\frac{m_j}{a}$, the storage capacity of storage node j in terms of the number of data packets. Recall that m_j is node j's storage capacity, and a is the size of a data packet in bits.

Therefore, $V'' = \{s\} \cup \{t\} \cup \{i' : i \in V\} \cup \{i'' : i \in V\}$ and $E'' = \{(i'', j') : (i, j) \in E\} \cup \{(j'', i') : (i, j) \in E\} \cup \{(i', i'') : i \in V\} \cup \{(s, i') : i \in V_s\} \cup \{(j'', t) : j \in V - V_s\}$. With Transformation II, the BSN graph $G(V, E)$ in Figure 2 is converted to flow network $G''(V'', E'')$ in Figure 4(a). Next, we formulate and solve ILP program (A) below on the obtained flow network $G''(V'', E'')$ to find the

[2]We refer to infeasible data preservation due to insufficient storage spaces (i.e., $\sum_{i=k+1}^{n} m_i < d \cdot a$) as overall storage overflow. Overall storage overflow can be solved by aggregating the overflow data packets following their spatial correlation pattern [80]. We leave the design of a data aggregation game under overall storage overflow as future work.

maximum amount of data packets that can be offloaded under energy constraint. In ILP (A), x_{ij} is the number of flows on edge $(i, j) \in E''$.

$$(A) \quad \text{maximize} \quad \sum_{i \in V_s} x_{si'} \tag{3}$$

$$\text{s.t.} \quad x_{si'} \leq d_i, \qquad i \in V_s \tag{4}$$

$$x_{i''t} \leq \frac{m_i}{a}, \qquad i \in V - V_s \tag{5}$$

$$x_{si'} + \sum_{j:(i,j)\in E} x_{j''i'} = \sum_{j:(i,j)\in E} x_{i''j'}, \qquad i \in V_s \tag{6}$$

$$\sum_{j:(i,j)\in E} x_{j''i'} = \sum_{j:(i,j)\in E} x_{i''j'} + x_{i''t}, \qquad i \in V - V_s \tag{7}$$

$$E_i^r \times \sum_{j:(i,j)\in E} x_{j''i'} + \sum_{j:(i,j)\in E} (E_i^t(j) \times x_{i''j'}) \leq E_i, \qquad i \in V_s \tag{8}$$

$$E_i^r \times \sum_{j:(i,j)\in E} x_{j''i'} + \sum_{j:(i,j)\in E} E_i^t(j) \times x_{i''j'} + E_i^s \times x_{i''T} \leq E_i, \qquad i \in V - V_s \tag{9}$$

In ILP (A), Objective (3) is to find the maximum amount of packets that can be offloaded in the entire network. Inequality (4) indicates the maximum number of packets data node i can offload is d_i, the initial number of data packets data node i has. Inequality (5) indicates the maximum number of packets storage node i can store is $\frac{m_i}{a}$, where m_i is the storage capacity of storage node i, and a is the size of each packet. Equation (6) shows the flow conservation for data nodes, where the number of its own data packets offloaded plus the number of data packets it relays for other data nodes equals the number of data packets it transmits. Equation (7) is the flow conservation for storage nodes, which says that data packets a storage node i receives are either relayed to other nodes or stored by i. Inequalities (8) and (9) represent the energy constraints for data and storage nodes, respectively.

Theorem 1. *Given any DPP instance $G(V, E)$, if $\sum_{i \in V_s} x_{si'} = d$ when applying ILP (A) on $G''(V'', E'')$, then this DPP instance is feasible.*

Proof. First, when $\sum_{i \in V_s} x_{si'} = d$, there are d amount of flows going from s to t in $G''(V'', E'')$. As the edge capacity between s and i' is d_i, it must be that d_i amount of flows going from s into i'. Due to flow conservation in all nodes $V'' - \{s, t\}$, it must be that d_i amount of flows going out of i''.

Second, with the node-splitting technique in Step (2) in Transformation II, the initial energy level of node i (i.e., E_i) is now the capacity of edge $(i', i'') \in E''$. As the energy consumption of any node i, which is represented by the l.h.s of Inequality (8) and (9) in ILP (A), is less than E_i, it guarantees that node i does not exceed its energy capacity during data preservation in $G(V, E)$.

Therefore, all the d packets in BSN $G(V, E)$ can be offloaded from their data nodes to storage nodes while satisfying the energy constraints of sensor nodes, yielding feasible data preservation. □

For a feasible DPP instance, we next compute its minimum total energy consumption (i.e., total preservation cost) in Section 4.2.2.[3]

[3]For an infeasible instance due to insufficient energy of sensor nodes, our previous work [96] proposed a maximum weighted flow-based data preservation technique to maximize the total values preserved, assuming that data packets have different values. We leave designing a data preservation game under insufficient energy power of sensor nodes as future work.

4.2.2 Solving DPP. To compute the minimum total data preservation cost for a feasible DPP instance, we again transform the BSN graph $G(V, E)$ to another different flow network $G'''(V''', E''')$ following below four steps. We refer to it as Transformation III.

Transformation III

(1) In $G(V, E)$, replace each undirected edge $(i, j) \in E$ with two directed edges (i, j) and (j, i). Split node $i \in V$ into two nodes: *in-node* i' and *out-node* i'' and add a directed edge (i', i'') with a capacity of E_i, the initial energy level of node i, and cost of zero. All the incoming directed edges of node i are incident on i', and all the outgoing directed edges of node i emanate from i''.
(2) Add a source node s and connect it to the in-node i' of the data node $i \in V_s$, set its capacity as d_i, the number of data packets at data node i, and set its cost as zero.
(3) For directed edge (i'', j'), set its capacity as infinity and cost as $E_i^t(j) + E_j^r$, the sum of node i's transmission energy and node j's receiving energy. For directed edge (j'', i'), set its capacity as $+\infty$ and cost as $E_j^t(i) + E_i^r$, the sum of node j's transmission energy and node i's receiving energy.
(4) Add a sink node t and connecting the out-node i'' of the storage node $j \in V - V_s$ to t, set the capacity of the edge as $\frac{m_i}{a}$, the storage capacity of i, and the cost of the edge as E_i^s, the energy consumption of storing one data packet at node i.

Similar as $G''(V'', E'')$ in Figure 4(a), now $V''' = \{s\} \cup \{t\} \cup \{i' : i \in V\} \cup \{i'' : i \in V\}$ and $E''' = \{(i'', j') : (i, j) \in E\} \cup \{(j'', i') : (i, j) \in E\} \cup \{(i', i'') : i \in V\} \cup \{(s, i') : i \in V_s\} \cup \{(j'', t) : j \in V - V_s\}$. However, $G'''(V''', E''')$ is different from $G''(V'', E'')$, as each edge in E'' only has a capacity, whereas each edge in E''' has a capacity as well as a cost. With Transformation III, the BSN graph $G(V, E)$ shown in Figure 2 is now converted to the flow network $G'''(V''', E''')$ shown in Figure 4(b).

Next, we formulate below ILP (B) and apply it upon $G'''(V''', E''')$ to solve DPP. Here, x_{ij} and c_{ij} are the flow and cost on edge $(i, j) \in E'''$, respectively.

$$(B) \qquad \text{minimize} \quad \sum_{(i,j)\in E'''} x_{ij} \times c_{ij} \tag{10}$$

$$\text{s.t.} \quad x_{si'} = d_i, \qquad i \in V_s \tag{11}$$

$$x_{i''t} \le \frac{m_i}{a}, \qquad i \in V - V_s \tag{12}$$

$$x_{si'} + \sum_{j:(i,j)\in E} x_{j''i'} = \sum_{j:(i,j)\in E} x_{i''j'}, \qquad i \in V_s \tag{13}$$

$$\sum_{j:(i,j)\in E} x_{j''i'} = \sum_{j:(i,j)\in E} x_{i''j'} + x_{i''t}, \qquad i \in V - V_s \tag{14}$$

$$E_i^r \times \sum_{j:(i,j)\in E} x_{j''i'} + \sum_{j:(i,j)\in E} (E_i^t(j) \times x_{i''j'}) \le E_i, \qquad i \in V_s \tag{15}$$

$$E_i^r \times \sum_{j:(i,j)\in E} x_{j''i'} + \sum_{j:(i,j)\in E} E_i^t(j) \times x_{i''j'} + E_i^s \times x_{i''t} \le E_i, \qquad i \in V - V_s \tag{16}$$

Objective (10) is to minimize $\sum_{(i,j)\in E''} x_{ij} \times c_{ij}$, the total energy cost of all the flows in the flow network (i.e., total preservation cost). Note that Constraints (11)–(16) are similar to Constraints (4)–(9) in ILP (A), except that Constraint (11) is now $x_{si'} = d_i$, as all the d_i data packets at data node i can be offloaded in a feasible case. Theorem 2 below shows ILP (B) finds the minimum total preservation cost for any feasible DPP graph $G(V, E)$.

Theorem 2. *Given any feasible DPP instance in $G(V, E)$ and its transformed flow network $G'''(V''', E''')$, applying ILP (B) on G''' gives the minimum total data preservation cost in $G(V, E)$.*

Proof. We need to show that the energy consumptions of sensor nodes in data preservation are accurately represented in Transformation III, and ILP (B) results in minimum total data preservation cost while satisfying the storage and energy constraints of all sensor nodes.

First, recall that in Transformation III, an edge $(i, j) \in E$ changes to two directed edges (i'', j') and $(j'', i') \in E'''$, with their costs being $E_i^t(j) + E_j^r$ and $E_j^t(i) + E_i^r$, respectively. Consider any data packet in G offloaded from data node $i \in V_s$, goes through a sequence of intermediate nodes (if any), and gets stored at a storage node $j \in V - V_s$. It has a corresponding flow in G''' that starts at source node s, goes to i' and i'', the in-node and out-node of data node i, and then goes through a sequence of in-node and out-node of intermediate nodes (if there are any), then goes to j' and j'', the in-node and out-node of storage node j, and finally ends at sink node t. Along the way, due to the cost setup of edges (i'', j') and (j'', i'), the transmission energy of i, and receiving and transmission energy of all other intermediate nodes (if there are any), and the receiving and storing energy of storage node j is all accurately captured in c_{ij}, the cost of edge $(i, j) \in E'''$. Second, because of its Objective (10), ILP (B) computes the minimum total data preservation cost. Finally, Inequalities (12), (15), and (16) guarantee the storage constraint of storage nodes, the energy constraint of data nodes, and the energy constraint of storage nodes are satisfied, respectively. □

The above graph transformation is similar to the one used in Reference [42] that solved a data resilience maximization problem in the BSN. However, the objectives and the techniques in both works are different. In Reference [42], the data resilience is defined as the sum of the remaining energy of the destination nodes of all the preserved data packets. It is formulated as a quadratic programming problem. For the DPP studied in this article, the goal is to minimize the total energy consumption in data preservation, formulated as an ILP problem. Note that, so far, we have introduced two optimal centralized data preservation solutions: ILP (B) for the DPP and PR-MCF for DPP-W. Although both solutions minimize the total preservation cost in the BSN, they have different time complexity. PR-MCF is efficient with polynomial time complexity, as discussed in Section 4.1, while ILP (B) is time-consuming, as the general class of ILP is NP-hard. However, both solutions assume that all the sensor nodes are fully cooperative and will follow the algorithmic computation to participate in the data preservation. This assumption is invalid for selfish nodes, which have zero incentive to participate in data preservation. Or even if they are incentivized to participate, they tend to lie about their costs to gain more utilities. We thus design suitable data preservation games to not only incentivize sensor nodes to participate in data preservation but also to guarantee that reporting their true cost parameters is their dominant strategy. Our games seamlessly integrate algorithmic mechanism design and its variations with network flow computations of PR-MCF and ILP (B). In the below two sections, we design data preservation games for DPP-W and DPP, respectively.

5 DPG-1: DATA PRESERVATION GAME FOR DPP-W

In this section, we design a data preservation game for DPP-W and refer to it as DPG-1: data preservation game-1. We first present the DPG-1 in Section 5.1 and then prove in Section 5.2 that truthfulness is achieved in DGP-1. In DPG-1, it is a dominant strategy for every storage node to truthfully report its private cost type.

5.1 Data Preservation Game DPG-1

We make two assumptions for the DPG-1 and the DPP-W instance it is applied upon. First, the BSN graph must remain connected, and its represented DPP-W instance must be feasible for data

preservation even if any one of the sensor nodes is removed from the BSN. This is required by the payment model used in DPG-1. For example, for the BSN instance shown in Figure 2, after removing any node, the BSN is still connected, and its data preservation is still feasible. Second, the data nodes must offload their overflow data packets to other storage nodes and thus need not be motivated. That is, we assume that data nodes are always truthful and only the storage nodes are players in DPG-1. How to motivate data nodes has been addressed in Reference [24]. As DPG-1 is based on the classic Vickrey-Clark-Groves (VCG) mechanism [17, 39, 84], we first introduce the VCG mechanism.

VCG Mechanism. There are n players or agents in the network—player i has some private information t_i, called its *private cost type*. Each player i has a set of strategies A_i. When i plays strategy $a_i \in A_i$, the mechanism computes an *output* $o = o(a_1, \ldots, a_n)$ and a *payment vector* $p = (p_1, \ldots, p_n)$, where $p_i = p_i(a_1, \ldots, a_n)$ is the payment to player i. Player i's cost is given by *cost function* $v_i(t_i, o)$, which depends on t_i and o. Player i wants to maximize its *utility function* $\pi_i(a_1, .., a_n) = v_i(t_i, o) + p_i$. Classic VGC mechanism provides a truthful solution for utilitarian problems where the objective function is the sum of all agents' valuations (i.e., costs) [61]. This suits our data preservation problem well, wherein the total preservation cost is the sum of each node's cost in data preservation. Next, we introduce the corresponding private cost type, cost function, and strategy set used in DPG-1.

Private Cost Type, Cost Function, and Strategy Set in DPG-1. In DPG-1, storage node i's *private cost type* $t_i = \{\epsilon_i^e, \epsilon_i^a, \epsilon_i^s\}$ consists of three cost parameters, viz., transmission parameters ϵ_i^a, receiving parameter ϵ_i^e, and storing parameter ϵ_i^s.[4] Node i's *cost function* is $v_i(t_i, o) = -\sum_{j=1}^{d} c_{i,j}$, wherein $c_{i,j}$ is node i's energy cost in preserving data packet D_j and is given by Equation (1). Node i has three possible actions for each data packet: Either it does not participate in its data preservation, or when it participates, it may either relay (i.e., receive and then transmit) or store (i.e., receive and then store) the data packet. The *output* of the DPG-1 is the data preservation process computed by the minimum cost flow algorithm PR-MCF [33], which indicates how each data packet should be relayed or stored by each storage node to achieve the minimum total preservation cost. Therefore, node i's *strategy set* A_i includes its different ways to report t_i as well as its actions for each data packet following the PR-MCF computation.

As both the data node and storage node can relay data packets while only the storage node can store data packets, following the energy model defined in Section 3, we define the *relaying cost* of data node and storage node i and *storing cost* of storage node i:

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

We assume that the amount of data a data node generates during sensing is public knowledge. As a data node wish to offload its overflow data packets into the BSN for preservation, it has no incentive to lie about how many data packets it has.

[4]As the goal of data nodes is to offload their overflow data packets, we assume they do not lie about their cost parameters and they are public knowledge in the BSN.

Challenges of Applying VCG. We observe that there are two features of $t_i = \{\epsilon_i^e, \epsilon_i^a, \epsilon_i^s\}$ that make directly applying classic VCG to solve DPG-1 challenging.

(1) In contrast to the scalar private cost type defined in classic VCG and used by existing works (e.g., the channel gain in a wireless channel [90] and the emission signal strength used in Reference [7]), t_i in DPG-1 is a *composite type* consisting of three cost parameters. Therefore, there are $2^3 = 8$ different combinations of the cost parameters that a node can lie about t_i. Whether choosing different combinations of the components of a composite private cost type affects the truthfulness of classic VCG has yet to be explored.
(2) Consequently, many existing works assumed that an agent has a scalar private cost type, which only induces its *binary* action of either doing it or not doing it.[5] In contrast, in DPG-1, the relationship between node i's cost parameters in t_i and its incurred costs is more complicated: ϵ_i^e is involved in both receiving cost $c_i^r(j)$ and storing cost c_i^s, while ϵ_i^a is involved only in $c_i^r(j)$ and ϵ_i^s is involved only in c_i^s. By lying about different cost parameters to different extents, a node might manipulate its cost and switch from one action to another.

For (1), we prove rigorously that the truthfulness of classic VCG still holds for composite private cost types and then validate it by our simulation results. For (2), as there is a need to study which cost parameters are more impactful in the truthfulness of DPG-1, we conduct extensive simulations to investigate and give some insights into how they play a role in the truthfulness of DPG-1.

Payment and Utility Model in DPG-1. Next, we derive the payment and utility functions used in the DPG-1. Let $c_i = \sum_{j=1}^{d} c_{i,j}$ be the total data preservation cost of node i computed by the PR-MCF proposed in Section 4.1, and $t_{-i} = \{t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n\}$ be the vector of cost types of all other nodes except node i. We give the below definitions:

Definition 1 (Payment and Utility in DPG-1). Based on Green and Laffont [38], given any cost $\tilde{t}_i$ reported by node i, the amount of payment given to node i depends on whether node i is chosen to participate in data preservation or not according to the PR-MCF computation. Its payment is 0 if it is not chosen; when it is chosen, its payment is:

$$p_i(\tilde{t}_i, t_{-i}) = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i). \tag{17}$$

Here, $\tilde{c}_V$ is the minimum total preservation cost of the network when i reports its cost $\tilde{c}_i$, and $c_{V-\{i\}}$ is the minimum total preservation cost of the network when i is removed. Both can be computed using the PR-MCF algorithm. i's utility is 0 when it is not chosen; and when i is chosen, its utility is

$$\pi_i(\tilde{t}_i, t_{-i}) = p_i(\tilde{t}_i, t_{-i}) - c_i = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i) - c_i, \tag{18}$$

where c_i is node i's true cost (i.e., its energy cost based on the true values of its cost parameters). Moreover, we define c_V as the minimum total preservation cost in the BSN when i truthfully reports its cost (i.e., $\tilde{t}_i = t_i$).

The above payment and utility models are common knowledge to each node. That is, each node knows that based on their reported cost types, their payment and utility are computed by Equations (17) and (18), respectively. For example, for the BSN instance in Figure 2, suppose the total preservation cost when node 3 is removed (i.e., $c_{V-\{3\}}$) is 10, and suppose that when node 3 participates by truth-telling its private cost type, its cost is 2 and the total preservation cost is 8. Then

[5]For example, Ad hoc-VCG [7], a VCG-based routing protocol for ad hoc networks, assumed that each ad hoc node has only one cost parameter called *cost-of-energy* and has binary action of forwarding the packet or not. In COMMIT [24], a sender-centric truthful ad hoc routing protocol, 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 to establish this connection or not.

according to Equations (17) and (18), 3's payment is $10 - 8 + 2 = 4$ and its utility is $4 - 2 = 2$ when truth-telling.

Discussions. The amount of payment given to a specific node i equals the total preservation cost when i is removed from the network minus all other participating nodes' costs when i participates. That is, i's payment is the energy cost that node i helps to reduce for the entire network when it participates in data preservation. i's utility is its payment minus its true cost, which is i's *marginal contribution* to the network in preserving data packets. As node i is compensated with its share of contribution in the payment amount, it is thus motivated to participate if its received utility is greater than 0. The time taken to compute the payment is the time taken for the PR-MCF algorithm.

Data Preservation Game DPG-1. With the above preparations, we present the DPG-1. It has three stages.

(1) Each storage node reports its private cost type t_i.
(2) Based on the reported cost types, the PR-MCF algorithm computes the data preservation process and its incurred minimum total preservation cost.
(3) Each storage node follows the computed data preservation to either participate or not. If it participates, then it receives the payment given by Equation (17) and gets the utility given by Equation (18).

Note that each storage node takes strategic moves in Stages 1 and 3, while Stage 2 is non-strategic, with only the PR-MCF algorithm being executed. In Stage 1, a storage node reports its (either truth-telling or lied) private cost type. In Stage 3, it decides whether to participate in data preservation based on its received utility. It will participate if its utility is greater than zero; otherwise, it will not. Since there is a time sequence between the two decisions in Stage 1 and Stage 3, the solution concept of the game is subgame perfect Nash equilibrium (SPNE). SPNE is a Nash equilibrium (NE) in which players achieve NE in every subgame of the whole game tree. Next, we prove the truthfulness of the DPG-1.

5.2 Truthfulness of the DPG-1

The truthfulness is achieved if the resulting equilibrium of DPG-1 satisfies the following two properties:

(1) **Individual-rationality (IR)**. It is the participation constraint that makes sure that each node, when truthfully reporting its private cost type and is chosen by the PR-MCF, will receive positive utility and participate in the data preservation. That is,

$$\pi_i(t_i, t_{-i}) \geq 0 \ \forall t_{-i} \text{ and } \forall i \in V - V_s.$$

(2) **Incentive-compatibility (IC)**. It requires that truthfully reporting private cost 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:

$$\pi_i(t_i, t_{-i}) \geq \pi_i(\tilde{t}_i, t_{-i}) \ \forall t_{-i}, \ \forall \tilde{t}_i \neq t_i \text{ and } \forall i \in V - V_s.$$

Note that when IR and IC are satisfied, it is a dominant strategy solution to the DPG-1 that each node truthfully reports its private type and participates in the game whenever chosen by the centralized algorithm PR-MCF.[6]

[6]Note that we do not argue that truthfulness is the unique equilibrium, since IR and IC only impose weakly dominance of truthfulness to each node. There could be cases when a node is indifferent between lying and truth-telling. Therefore, other Nash equilibria involving one or multiple nodes lying could exist. However, the dominant strategy solution is a strong

Next, we present Lemma 1, which shows that for any node, even if it reports its private cost type differently, as long as it is assigned the same data preservation tasks, the total preservation cost of the network does not change.

Lemma 1. *For given t_{-i} , consider two different types of node i denoted as $\hat{t}_i \neq \tilde{t}_i$. If node i performs the same tasks following the centralized algorithm PR-MCF under $\hat{t}_i$ and $\tilde{t}_i$, then it must be that the real total minimized data preservation costs remain the same under $\hat{t}_i$ and $\tilde{t}_i$.*

Proof. We use contradiction to prove that when node i are performing the same tasks under $\hat{t}_i$ and $\tilde{t}_i$, the total data preservation costs for all nodes other than i are the same under $\hat{t}_i$ and $\tilde{t}_i$. Recall that the minimized total data preservation cost c_V is the sum of c_i and c_{-i}. Denote $\hat{c}_V = \hat{c}_i + \hat{c}_{-i}$ the total minimized cost under $(\hat{t}_i, t_{-i})$ and $\tilde{c}_V = \tilde{c}_i + \tilde{c}_{-i}$ the total minimized cost under $(\tilde{t}_i, t_{-i})$. By way of contradiction, suppose $\hat{c}_{-i} \neq \tilde{c}_{-i}$. W.L.O.G., let $\hat{c}_{-i} < \tilde{c}_{-i}$. Since node i takes the same tasks under $\hat{t}_i$ and $\tilde{t}_i$, it holds that all nodes other than i can use the data preservation solution under $(\hat{t}_i, t_{-i})$ to offload data packets under $(\tilde{t}_i, t_{-i})$. Thus, under $(\tilde{t}_i, t_{-i})$, switching to the same data preservation solution used under $(\hat{t}_i, t_{-i})$ gives a total cost $\hat{c}_i + \tilde{c}_{-i} < \tilde{c}_i + \tilde{c}_{-i} = \tilde{c}_V$, a contradiction to cost minimization of the centralized algorithm under $(\tilde{t}_i, t_{-i})$. □

Theorem 3. *In the DPG-1, it is a dominant strategy of every storage node i to truthfully report its private cost type in stage 1; then to participate in data preservation in stage 3 following PR-MCF. IR and IC are satisfied under the payment given by Equation (17).*

Proof. Node i can either report truthfully or tell a lie about its cost type $t_i = \{\epsilon_i^a, \epsilon_i^s, \epsilon_i^e\}$. There are four cases for the outcome of i under truthfulness and lying. Below, we show that IR and IC are satisfied in all four cases.

Case I: Node i is not in the preservation path of a data packet when reporting either t_i or $\tilde{t}_i$. In this case, $\pi_i(t_i, t_{-i}) = \pi_i(\tilde{t}_i, t_{-i}) = 0$.

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

Case III: Node i is not in the preservation path of a packet when reporting t_i; however, it is in the preservation path of the packet when reporting $\tilde{t}_i$. Thus, $\pi_i(t_i, t_{-i}) = 0$. Denote $\tilde{c}_V^{t_i}$ as the true data preservation cost (based on (t_i, t_{-i})) when the data preservation route is determined under $(\tilde{t}_i, t_{-i})$. We have $c_{V-\{i\}} \leq \tilde{c}_V^{t_i}$ due to cost minimization under t_i. Node i's payoff under $\tilde{t}_i$ is $\pi_i(\tilde{t}_i, t_{-i}) = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i) - c_i = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i + c_i) = c_{V-\{i\}} - \tilde{c}_V^{t_i} \leq 0$. Thus, $\pi_i(t_i, t_{-i}) \geq \pi_i(\tilde{t}_i, t_{-i})$.

Case IV: Node i is in the preservation path when reporting either t_i or $\tilde{t}_i$. There are different subcases according to the tasks assigned to node i. We discuss each subcase below:

Subcase IVa. Node i is assigned exactly the same tasks when reporting either t_i or $\tilde{t}_i$. By Lemma 1, node i is getting the same payoff, since $\pi_i(\tilde{t}_i, t_{-i}) = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i) - c_i = c_{V-\{i\}} - c_V = \pi_i(t_i, t_{-i}) \geq 0$.

Subcase IVb. The tasks assigned to node i are different under t_i and $\tilde{t}_i$. Note that all data packets are preserved in either case, since storage nodes are not energy-constrained. The payment of i when it reports truthfully is $\pi_i(t_i, t_{-i}) = c_{V-\{i\}} - (c_V - c_i) - c_i = c_{V-\{i\}} - c_V \geq 0$. Instead, when it lies by reporting $\tilde{t}_i$, its payoff is $\pi_i(\tilde{t}_i, t_{-i}) = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i) - c_i = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i + c_i)$. Here, $\tilde{c}_V - \tilde{c}_i + c_i \equiv \tilde{c}_V^{t_i}$, the total preservation cost calculated according to (t_i, t_{-i}) using the route found

solution concept, since, to each node, truthfulness is never worse than lying, no matter what other nodes choose to do. In some scenarios, truthfulness is strictly better than lying.

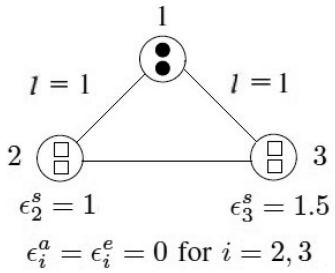


Fig. 5. An example shows the invalidity of standard VCG. Initial energy $E_2 = E_3 = 2$. As data nodes are always truthful, data node 1's cost parameters are not specified, and we assume E_1 is large enough to offload node 1's two data packets.

by the centralized algorithm under $(\tilde{t}_i, t_{-i})$. Since there is no data loss, by cost minimization of the centralized algorithm under (t_i, t_{-i}), $\tilde{c}_V^{t_i} \geq c_V$. Thus, $\pi_i(\tilde{t}_i, t_{-i}) \leq c_{V-\{i\}} - c_V = \pi_i(t_i, t_{-i})$.

Thus, IC is satisfied in all cases. Note that IR, the participation constraint under truth-telling, is also satisfied in all cases, since $\pi_i(t_i, t_{-i}) \geq 0$ holds everywhere, as indicated in the above proof. Therefore, each node has truth-telling as the dominant strategy and will willingly participate in data preservation. □

Discussions. Let us understand the intuition of Theorem 3 and its proof from the economic point of view. The idea of the VCG mechanism is to give each storage node a net payoff (i.e., utility) according to its marginal contribution to data preservation. Let us consider the scenario wherein lying by storage node i leads to data preservation routes different from those found under truth-telling. We can replace the preservation routes under node i's truth-telling with those found under its lying (this is always possible, because nodes are not energy-constrained in DPG-1, thus no data loss occurs). Now, since such replacement will only increase the total data preservation cost in the network, it reduces node i's marginal contribution to the network, implying a lower utility to node i. As such, the VCG mechanism provides incentives for node i to tell the truth and to participate in data preservation. However, the above argument will no longer hold when nodes are energy-constrained, which is studied in the following section.

6 DPG-2: DATA PRESERVATION GAME FOR DPP

In this section, we design a data preservation game called DPG-2 for the general case of DPP, wherein sensor nodes have limited battery power. DPG-2 is based on our key finding Theorem 4 below. Theorem 4 states that if a storage node has limited energy power, then lying about its private cost type can possibly increase the node's utility compared to truth-telling. This observation invalidates the truthfulness of the DPG-1.

Theorem 4. *When nodes are energy-constrained, the classic VCG mechanism proposed in DPG-1 cannot guarantee that truthfulness is the dominant strategy for each storage node.*

Proof. We design an example to demonstrate this. Consider a BSN instance in Figure 5, where node 1 is the data node with 2 data packets, and node 2 and node 3 are the storage nodes, each having a storage capacity of 2 and each having a distance of 1 to node 1. Nodes 2 and 3 each have 2 units of initial energy. For $t_i = \{\epsilon_i^a, \epsilon_i^s, \epsilon_i^e\}$, suppose $\epsilon_i^a = \epsilon_i^e = 0$ for $i = 2, 3$, and $\epsilon_2^s = 1, \epsilon_3^s = 1.5$. Nodes 2 and 3 have zero relaying cost, but their storing costs for one data packet are 1 and 1.5, respectively. Therefore, with their energy capacity, node 2 can store two data packets from node 1, and node 3 can store one data packet from node 1.

First, consider truth-telling by nodes 2 and 3. Following the optimal ILP (B) proposed in Section 4.2.2, it will offload node 1's two data packets to node 2, and node 3 does not store any data packet. The total data preservation cost is 2 and node 3's utility is $\pi_2(t_2, t_3) = 0$.

Next, consider that node 3 lies by downsizing its private cost type $\tilde{t}_3 = \epsilon_3^s$ as 0.8 (as $\epsilon_3^a = \epsilon_3^e = 0$) while node 2 is telling the truth. In this case, the ILP (B) will choose node 3 to store node 1's two data packets, and the ILP (B) computes the total data preservation cost as $2 * 0.8 = 1.6$ according to node 3's reported private cost type. With only 2 units of energy, however, node 3 can only store one data packet and has to drop the other; its true cost is thus $c_3 = 1.5$. Under the VCG mechanism in DPG-1 and following Equation (18), node 3's utility is $\pi_3(t_2, \tilde{t}_3) = c_{V-\{3\}} - \tilde{c}_V + \tilde{c}_3 - c_3 = 2 - 1.6 + 1.6 - 1.5 = 0.5 > 0$. Unlike its truth-telling case, in which node 3's utility is $\pi_2(t_2, t_3) = 0$, node 3 is incentivized to lie about its private cost type to gain a positive utility of 0.5. We conclude that in DPP, the classic VCG mechanism can fail to satisfy IC or/and IR and no longer guarantee the truthfulness of storage nodes. □

Discussions. Note that a node has no incentive to upsize its private cost type (i.e., by exaggerating that it takes more energy than needed to participate in the data preservation). With such a claim and to minimize the total preservation cost, the MCF-based ILP (B) will not assign a heavier data preservation load to this upsizing node than when it is truth-telling. An extreme case is that if a node claims infinite cost in any data preservation task, it will then be excluded from data preservation. Consequently, with less load than truth-telling, the upsizing node will never run out of its energy; therefore, no data loss will occur. In this upsizing case, the argument in Theorem 3 continues to hold that a node cannot strictly improve its utility through exaggeration; i.e., truth-telling is its dominant strategy. Figure 5 shows two possible consequences when a storage node i downsizes its private cost type. First, the value of $\tilde{c}_V$ in Equation (18) decreases. Second, node i can get assigned more data packets than it can possibly store or relay with its energy capacity. As such, the node must drop some of the received data packets. As the value of $c_{V-\{i\}}$ does not change whether i lies or not, i's utility $\pi_i(\tilde{t}_i, t_{-i})$ increases following Equation (18).[7] Such data loss uniquely arises in DPP and is not addressed by DPP-W and the classic VCG mechanism in DPG-1. Therefore, unlike the DPP-W wherein a storage node does not have the incentive to lie about its private cost type (Theorem 3), in DPP, by following the classic VCG mechanism and downsizing its private cost type, an energy-constrained storage node has the incentive to lie to gain more utility at the cost of data loss of the system (Theorem 4).

DPG-2. Next, we present DPG-2, which improves DGP-1 and restores the truthfulness of the VCG. It consists of a data loss-finding mechanism and a data loss-prohibiting mechanism.

Finding Data Loss. Figure 6 illustrates how the data loss finding mechanism works, which first finds if a lying storage node i drops packets, and if so, how many packets are dropped. Finding the number of dropped data packets (i.e., data loss) is made possible by the computation of ILP (B) proposed in Section 4.2.2. Such network flow-level computation can find out the number of data packets i receives, relays, and stores at a flow level and thus can find if it drops data packets. Figure 6 illustrates how this works. We refer to the total number of packets that i is assigned to receive, relay, and save as *assigned load*, *assigned relay*, and *assigned save*, respectively. Due to flow conservation in ILP (B), assigned load = assigned relay + assigned save.

Next, using the above information, we calculate how many packets in node i's assigned load are *actually* saved and relayed with its actual energy power available. We denote these two numbers as *actual save and actual relay*, respectively, and compute as follows: First, among all the data packets

[7]We investigate the effect of downsizing of i's private cost upon $\tilde{c}_i$ and c_i via simulations in Section 7.

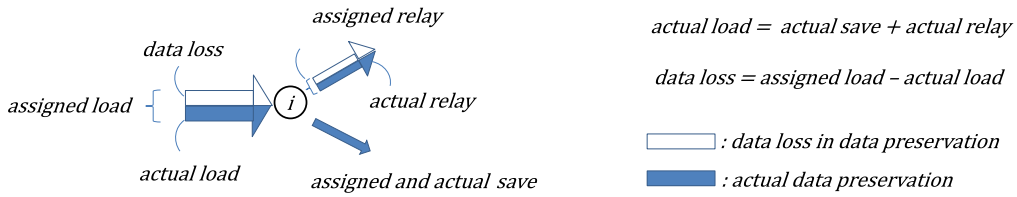


Fig. 6. Finding data packets dropped by lying storage node i.

it receives (i.e., assigned load), i saves as many packets as its storage capacity allows. If packets are still left from the assigned load after the storage is full, i then relays those that require the smallest transmission energy (i.e., packets that are relayed to closest neighbors) until its energy power is depleted. The number of data packets relayed this way is *actual relay*. In this case, actual save equals assigned save, as shown in Figure 6. Otherwise, actual save is less than the assigned save, and the actual relay is zero. In either case, the sum of actual save and actual relay must equal the *actual load*, the number of packets the storage node handles. Finally, we compare the actual load with the assigned load. If they are equal, then there is no data packet dropped. Otherwise, it must be that the actual load is less than the assigned load, and their difference is the number of data packets dropped by i. We refer to this as *data loss* in Figure 6. Algorithm 1 shows the detailed execution of how to find the data loss.

ALGORITHM 1: Finding data loss at storage node i.

Data: A BSN graph $G(V, E)$.
Result: Detect if data loss occurs at storage node i.
Transform $G(V, E)$ to a flow network $G'''(V''', E''')$ following Transformation III in Section 4.2.2;
Apply ILP (B) on $G'''(V''', E''')$ to compute the assigned load, assigned relay, and assigned save;
if *assigned load* > $\lfloor \frac{m_i}{a} \rfloor$, *where* $\lfloor \frac{m_i}{a} \rfloor$ *is node i's storage capacity* **then**
  actual save = assigned save = $\lfloor \frac{m_i}{a} \rfloor$;
  assigned relay = assigned load − $\lfloor \frac{m_i}{a} \rfloor$;
  actual relay = 0;
  Sort the packets in the assigned relay in the non-descending order of their transmission energy;
  Let E^i_{curr} be the remaining energy of node i after receiving the assigned load amount of data packets following ILP (B);
  // In the assigned relay, relays those incurring minimum transmission energy until i's energy power is depleted;
  while $E^i_{curr} > 0$ **do**
    $E^i_{curr} = E^i_{curr} - \lfloor \frac{m_i}{a} \rfloor \cdot a \cdot \epsilon^s_i$; //compute the remaining energy after saving data packets;
    Relay the packet in the assigned reply with minimum transmission energy;
    Update E^i_{curr};
    actual relay = actual relay + 1;
  end
else
  actual save = assigned load;
  actual relay = 0;
end
actual load = actual save + actual relay;
data loss = assigned load - actual load;
Return data loss;

Data Loss Inhibiting Mechanism. As the failure of the VCG mechanism in DPG-1 is due to dropping data packets by energy-constrained storage nodes, we now modify the standard VCG model to inhibit such data loss. Our main idea is to introduce a data loss inhibiting mechanism into the standard VCG model to punish any node that drops data packets. We first use Algorithm 1 to find out if a storage node i drops the packets and, if so, how many are dropped. Note that as Algorithm 1 accommodates the reported private cost types of storage nodes, it is still valid in a game-theoretic context. In a practical operation, as sensor nodes use wireless communication that is broadcast in nature, dropping packets of a node can be observed by its neighbors easily. As the number of received packets minus the number of stored packets is the number of transmitted packets while each node's storage capacity is public knowledge, whether a node drops received data packets or not thus can be easily observed. For any node, we assume at least one of its neighbors can observe how many data packets it receives and transmits. As the storage capacity of a storage node is public knowledge, we thus can infer if the node discards any received data packets. Since data loss can be easily detected by the network, storage nodes cannot secretly drop data packets and collect payment for pretended data preservation. Therefore, it is not a concern that a storage node may drop data packets to gain any benefits from the data preservation game. For ease of exposition, we simply assume that a node will not drop data packets if it has enough energy to save or transmit them.

We propose the modified payment model as follows: Given reported type $(\tilde{t}_i, t_{-i})$, whenever node i is chosen to participate in data preservation, its payment is given by

$$p_i^x(\tilde{t}_i, t_{-i}) = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i) - I_i \cdot [c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i)]. \tag{19}$$

Here, $I_i = 1$ if node i ever drops any data packet; $I_i = 0$ if not. Different from Equation (17), when node i drops any data packet, it is now punished by receiving zero payment. The corresponding utility of node i is thus

$$\pi_i^x(\tilde{t}_i, t_{-i}) = p_i^x(\tilde{t}_i, t_{-i}) - c_i. \tag{20}$$

Therefore, if node i drops any data packet, then its utility calculated in Equation (20) becomes $-c_i$. In contrast, a node's utility is at most zero when it does not participate. Thus, this prevents storage nodes from dropping packets.

DPG-2. With above preparations, we present the DPG-2 below.

(1) Each storage node reports its private cost type t_i.
(2) Based on the reported cost types, the ILP (B) computes the data preservation process and its incurred minimum total preservation cost.
(3) It calls Algorithm 1 to find the data loss if there is any.
(4) Each storage node follows the computed data preservation to either participate or not. If it participates, then it receives the payment given by Equation (19) and gets the utility given by Equation (20).

Theorem 5. *In DPG-2, with the payment given by Equation (19) and $t_i \in \{\epsilon_i^a, \epsilon_i^s, \epsilon_i^e\}$, both IR and IC are satisfied. Therefore, DPG-2 guarantees that truthfulness is the dominant strategy for each storage node.*

Proof. When there is no data loss of node i through reporting $\tilde{t}_i$, the payment is the same as in Equation (17) and the proof follows the same as in Theorem 3. We consider the case when node i drops at least one data packet under $\tilde{t}_i$, implying that node i must be in the preservation path under $\tilde{t}_i$. There can be two cases.

Case I: Node i is not in the preservation path when reporting t_i and is in the preservation path when reporting $\tilde{t}_i$. Thus, $\pi_i^x(\tilde{t}_i, t_{-i}) \leq 0 = \pi_i(t_i, t_{-i})$.

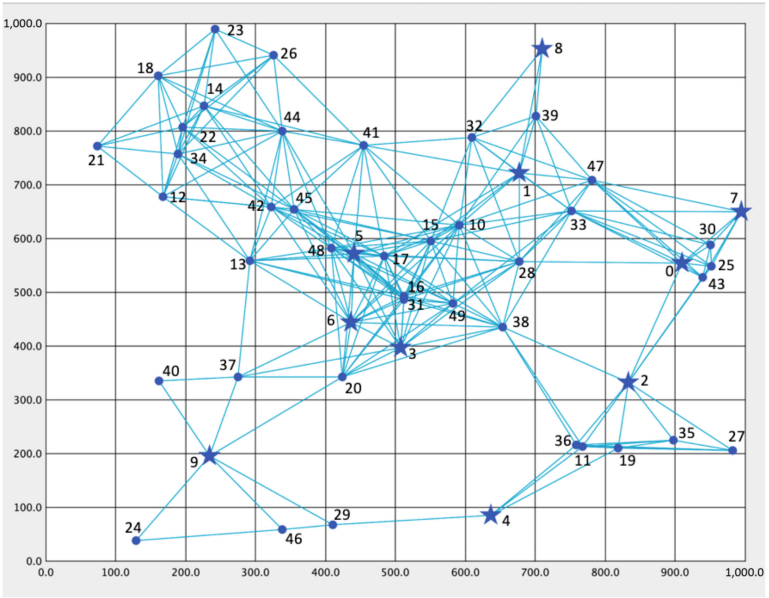


Fig. 7. A BSN with 50 nodes. Nodes 0–9 are data nodes and 10–49 are storage nodes.

Case II: Node i is in the preservation path when reporting either t_i or $\tilde{t}_i$. The utility of i when it reports truthfully is $\pi_i(t_i, t_{-i}) = c_{V-\{i\}} - (c_V - c_i) - c_i = c_{V-\{i\}} - c_V \geq 0$. Instead, when it lies by reporting $\tilde{t}_i$, its utility is $\pi_i^x(\tilde{t}_i, t_{-i}) = -c_i \leq 0$. Thus, $\pi_i(t_i, t_{-i}) \geq \pi_i^x(\tilde{t}_i, t_{-i})$.

The results immediately follow. □

7 SIMULATION RESULTS AND ANALYSES

Network Topology. Figure 7 shows the BSN topology used for our simulation experiments. Fifty sensor nodes are randomly placed in a field of 1,000 meters by 1,000 meters. The transmission range of each sensor node is 250 meters; that is, two nodes can directly communicate with each other by sending or receiving data packets when they are in this range. Among the 50 sensors, nodes 0–9 are data nodes and 10–49 are storage nodes. Each data node has 100 data packets, each of which has a size of 512 B. The storage capacity m_i of storage node i varies from 26 to 50 data packets (although 25 is the minimum storage capacity to successfully offload all the 1,000 data packets, as we need to take out one storage node when computing utilities while making the data preservation still feasible, we set the minimum storage capacity as 26). After all the data packets are offloaded, the network is almost full with $m_i = 26$ and is exactly half-full with $m_i = 50$. Compared to a half-full, an almost-full network represents a more stressful network where a storage node participates in more data preservation tasks (by either relaying or storing).

For the cost parameters ϵ_i^e, ϵ_i^a, and ϵ_i^s, we consider both the *homogeneous case* wherein different storage nodes have the same values for the same cost parameter (thus simplified as ϵ^e, ϵ^a, and ϵ^s) and the *heterogenous case* wherein different nodes have different values for the same cost parameter. For homogeneous case, the true values of ϵ^e, ϵ^a, and ϵ^s are 100 nJ/bit, 100 $pJ/bit/m^2$, and 100 nJ/bit, respectively; for heterogeneous case, they are random numbers in the range of [100,200] nJ/bit, [100,200] $pJ/bit/m^2$, and [100, 200] nJ/bit, respectively.

We define the *scaling factor*, denoted as α, as the ratio between the reported and true values of a cost parameter for a storage node. That is, when a storage node reports (i.e., lies) about a cost

Table 3. Scaling Factors of Cost Parameters for Storage Nodes

Node	11	13	15	17	19	21	23	25	27	29	32	33	35	37	39	41	43	45	47	49
ϵ_i^a	8.2	5.7	8.6	6.9	9.5	8.4	6.7	8.8	5.9	0.4	2.4	0.1	4.5	3.9	8.6	9.4	7.1	0.4	4.4	6.1
ϵ_i^e	4.9	4.2	6.7	4.1	6	7.6	7	6.4	1.9	4	3.8	4.2	7.2	2	3.6	9	1.5	0.4	0.9	8.9
ϵ_i^s	2	1.4	3.3	2.1	8	3.6	0.6	8	2.8	0.5	4.4	0.1	6.7	1.9	8.3	3.7	3.1	4.4	8.4	5.8

parameter of either ϵ_i^e, ϵ_i^a, or ϵ_i^s with a scaling factor of α, the reported values become $\alpha \cdot \epsilon_i^e$, $\alpha \cdot \epsilon_i^a$, or $\alpha \cdot \epsilon_i^s$, respectively. When $\alpha < 1$, we say the node *scales down* its cost by claiming it costs less energy than necessary; when $\alpha > 1$, it *scales up* its cost by claiming it costs more energy than necessary (i.e., it exaggerates its cost); when $\alpha = 1$, it is truth-telling. When a storage node i lies about its private cost type $t_i = \{\epsilon_i^a, \epsilon_i^s, \epsilon_i^e\}$, it can choose to lie any combination of the three cost parameters.

7.1 Evaluating DPP-W and DPG-1

In DPP-W and its data preservation game DPG-1, each node has an infinite amount of energy, and the total preservation cost is computed using the minimum cost flow algorithm, viz., PR-MCF in Section 4.1. We start with the general case where each storage node lies all three parameters simultaneously. Then, we look into the case where the storage node lies one cost parameter at a time and examine how different cost parameters affect the utility of the storage node.

Lying Three Parameters Simultaneously. We consider the homogeneous cost parameters, and each storage node lies all three cost parameters by randomly choosing its scaling factor α in the range of [0.1, 10]. The scaling factors of the storage nodes relevant to our findings are shown in Table 3.

Figure 8 compares the truth-telling and lying utilities of storage nodes using the utility model introduced in DPG-1. It shows that for each of the 40 storage nodes, its truth-telling utility is always greater than or equal to its lying utility. This demonstrates the effectiveness of DPG-1 in achieving minimum total preservation cost while accommodating the selfishness of storage nodes. Second, however, there are a few nodes whose lying utilities are significantly lower than their truth-telling utilities, including both scaling-up nodes (e.g., node 32) and nodes that mainly scale down (e.g., node 33). This indicates that nodes get different data preservation assignments under truth-telling and lying. By utility Equation (18), $\pi_i(\tilde{t}_i, t_{-i}) = p_i(\tilde{t}_i, t_{-i}) - c_i = c_{V-\{i\}} - (\tilde{c}_V - \tilde{c}_i) - c_i = c_{V-\{i\}} - \tilde{c}_V + \tilde{c}_i - c_i$. When scaling up, a node i gets fewer data preservation tasks, thus other nodes take more tasks, which increases $(\tilde{c}_V - \tilde{c}_i)$ and thus decreases utility $\pi_i(\tilde{t}_i, t_{-i})$. When scaling down, a node i gets more data preservation tasks, making its true preservation cost c_i larger and, consequently, its utility $\pi_i(\tilde{t}_i, t_{-i})$ becomes less. In an extreme case, a scaling-down node could get assigned so many preservation tasks that its real cost c_i increases to a level that its utility becomes negative, which is shown for the case of node 45. Finally, we note that node 23's truth-telling and lying utilities are both zeros. Figure 7 shows that as node 23 sits near the top edge of the sensor field and is farthest from all the data nodes, it does not participate in the data preservation process, incurring zero payment, cost, and utility.

Lying One Parameter at a Time. Figures 9(a), (b), and (c) show nodes lying their ϵ^s, ϵ^a, and ϵ^e in a half-full network, respectively (we only show nodes with non-zero utilities). We set α as 0.1, 1, and 10. We observe that while lying ϵ_e and ϵ_s results in equal truth-telling and lying utilities for most of the storage nodes, lying about ϵ_a results in some utilities that are significantly different from the corresponding truth-telling ones (e.g., nodes 25, 39, and 47). Our energy model in Section 3 can explain this: ϵ_e and ϵ_s are multiplied by data packet size, while ϵ_a is multiplied by the size of the packet as well as the square of the distances. As lying ϵ_a has a more evident effect of changing a node's energy cost and thus utility, we focus on ϵ_a for the rest of the simulations.

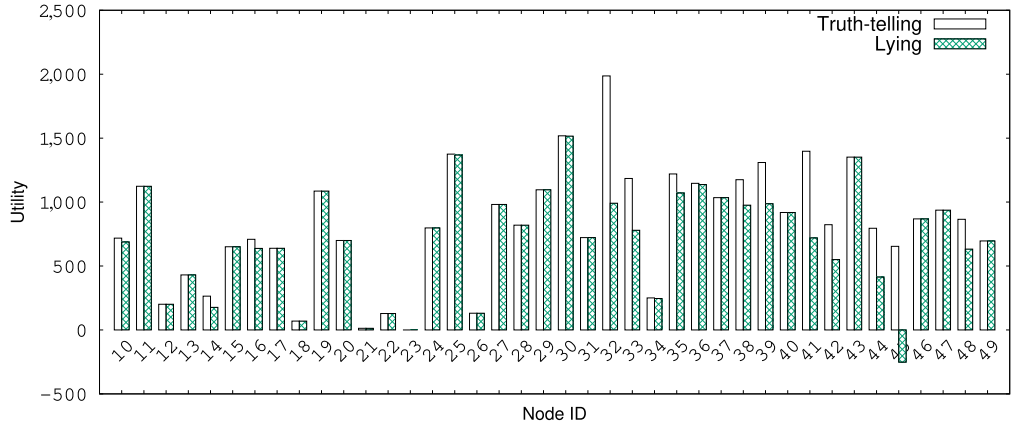


Fig. 8. Lying all three parameters simultaneously in a homogenous and half-full network.

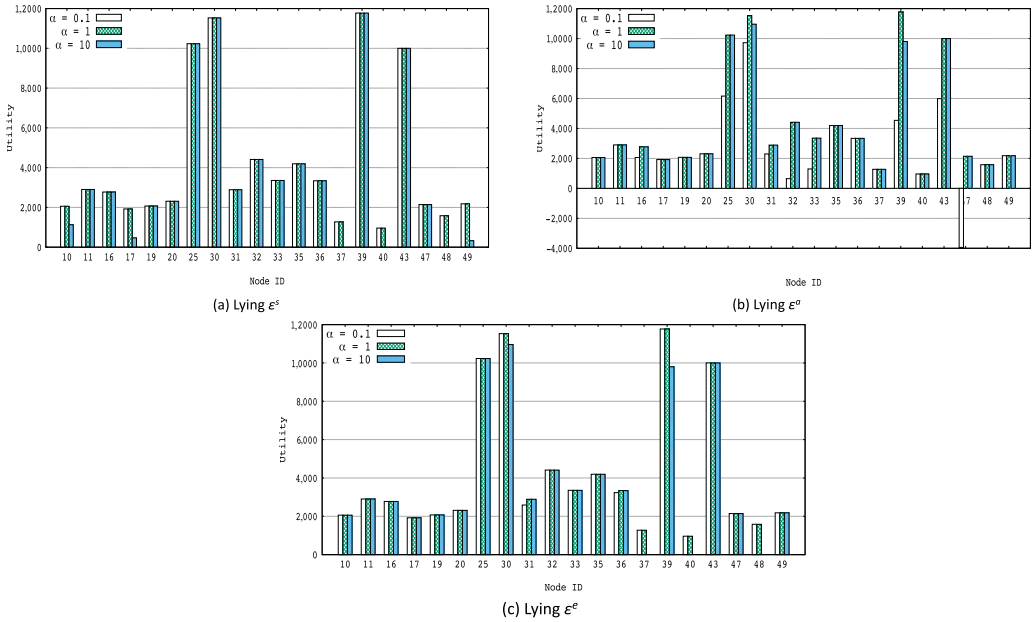


Fig. 9. Lying one parameter at a time in a heterogenous and half-full network.

Figure 10 considers the completely full case wherein all the storage nodes are more stressed by data preservation. Compared to Figure 9(b) wherein only one storage node receives a negative utility, Figure 10(b) shows that in this stressful network scenario, 12 scaling-down nodes are receiving negative utilities. This is because, with completely full storage, it is more likely that the lying storage node i is assigned a heavy load, thus increasing c_i dramatically, easily making its utility negative. As a completely full network better demonstrates the dynamism of the VCG behaviors for storage nodes, we adopt a completely full network for the rest of the simulations.

7.2 Evaluating DPP and DPG-2

In DPP and DPG-2, due to their energy constraints, the sensor nodes could deplete their energy power, and the total preservation cost is computed using the ILP-based minimum cost flow, viz., ILP

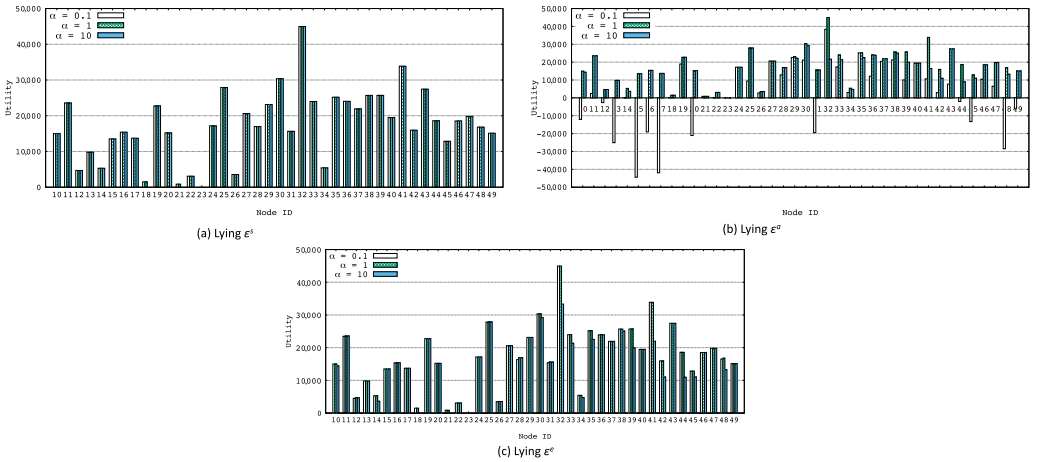


Fig. 10. Lying one parameter at a time in a heterogenous and completely full network.

(B) proposed in Section 4.2. We refer to any sensor node that exhausts its battery power as a *defunct node*. A data node is defunct if it does not have enough energy to relay (i.e., receive and transmit) at least one data packet to its closest neighbor. A storage node is defunct if its remaining energy is less than the smaller one between the energy cost of saving one data packet to its local storage and the energy cost of relaying one data packet to its closest neighbor. Finding defunct nodes is possible by calculating each node's energy consumption on its involved data preservation paths computed from ILP (B).

As defunct nodes occur and data preservation gets more challenging in DPP and DPG-2, we investigate progressively. In Section 7.2.1, we focus on truth-telling data preservation (i.e., DPP) and get a general view of its network characteristics, including fault tolerance and the energy consumption and workload of individual nodes. We then investigate the DPG-2 (i.e., VCG under energy constraints) in Section 7.2.2 and observe that in contrast to DPG-1, an anomaly arises wherein some nodes can yield a lying utility that is higher than the truth-telling utility in DPG-2, therefore invalidating VCG mechanism. In Section 7.2.3, we analyze the data preservation flows computed by network flow ILP (B) and obtain a "microscopic" view of each node's workload. As such, in Section 7.2.4, we can attribute the anomaly's cause to data loss due to packet dropping by some storage node. In Section 7.2.5, we implement our DPG-2 with a data loss inhibiting mechanism and show that it restores the truthfulness of classic VCG.

7.2.1 Network Characteristics in Truth-telling. We set the storage capacity of storage nodes m_i as 26, the minimum storage capacity that allows the DPG-2 to work. We assume that all the sensor nodes have the same initial energy levels E_i, as the sensor nodes usually have full and the same battery power when initially deployed (nonetheless, our game works for the case of different E_i as well). To investigate VCG behaviors, we are interested in finding *minimum feasible energy level* E_m, sensor nodes' minimum energy level at which all the 1,000 data packets in the network can still be offloaded. When $E_i = (E_m - 1)$ mJ, at least one data packet cannot be offloaded. As $m_i = 26$ and E_m provide the minimum storage and energy support for feasible data preservation, such "minimum" feasible condition can produce stressful data preservation scenarios that do not conform to the standard VCG, as we will show in Section 7.2.2 and later.

Fault-tolerance. To find E_m, we decrease E_i from 1,600 mJ, at which all the data can be offloaded, and record the total number of data packets offloaded as shown in Figure 11. At each E_i, we first

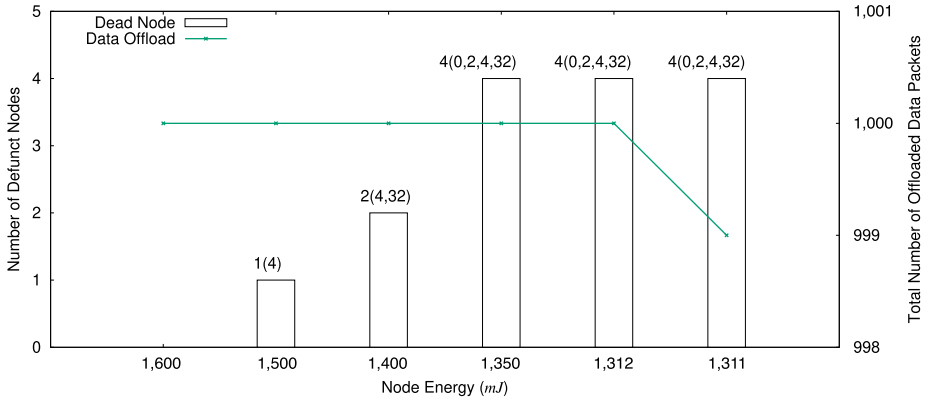


Fig. 11. Number of defunct nodes and total data packets offloaded in the network at different node energy E_i. The numbers in the parentheses are the IDs of defunct nodes.

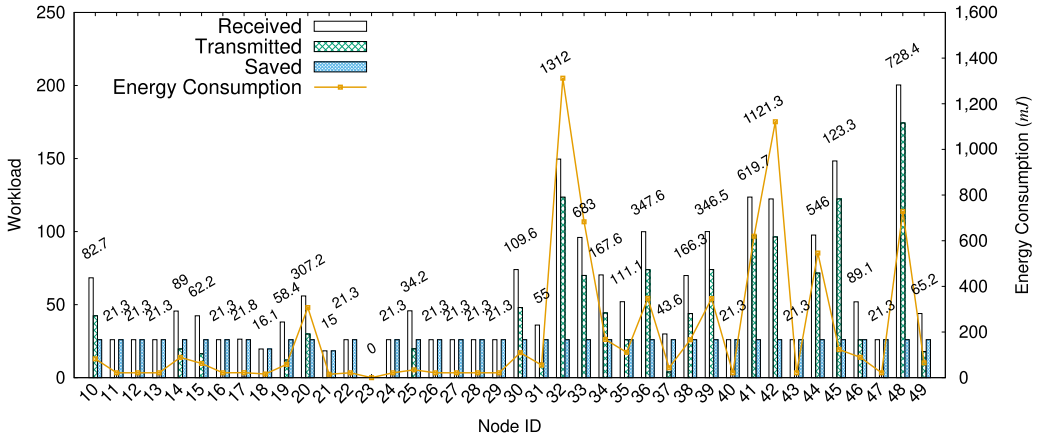


Fig. 12. Workload (i.e., number of received, transmitted, and saved packets) and energy consumption of truth-telling storage nodes.

use the maximum flow ILP, viz., ILP (A) proposed in Section 4.2.1, to find out the number of packets that can be offloaded. Then, we use the ILP (B) proposed in Section 4.2.2 to find out the minimum total energy consumption (i.e., total preservation cost) in offloading those data packets. Figure 11 shows the number of defunct nodes (and their IDs) at different E_i. It has zero defunct nodes at 1,600 mJ and increases to 4 at 1,312 mJ while the data preservation is still kept feasible, demonstrating the fault-tolerant capability of ILP (B). Among the four defunct nodes, three are data nodes (i.e., nodes 0, 2, and 4), and one is a storage node (i.e., node 32). When we further decrease E_i to 1,311 mJ, only 999 data packets can be offloaded. We thus set E_m as 1,312 mJ for the rest of the simulation experiments unless otherwise mentioned.

Workload and Energy Consumption of Storage Nodes. After getting a global view of the BSN's fault tolerance, we take a microscopic look at each storage node's workload and energy consumption. We define a storage node's *workload* as the total number of data packets it receives, which also equals the sum of the number of data packets it saves and relays (i.e., transmits). A node's workload is the node's "contribution" to the data preservation process. Figure 12 shows that node

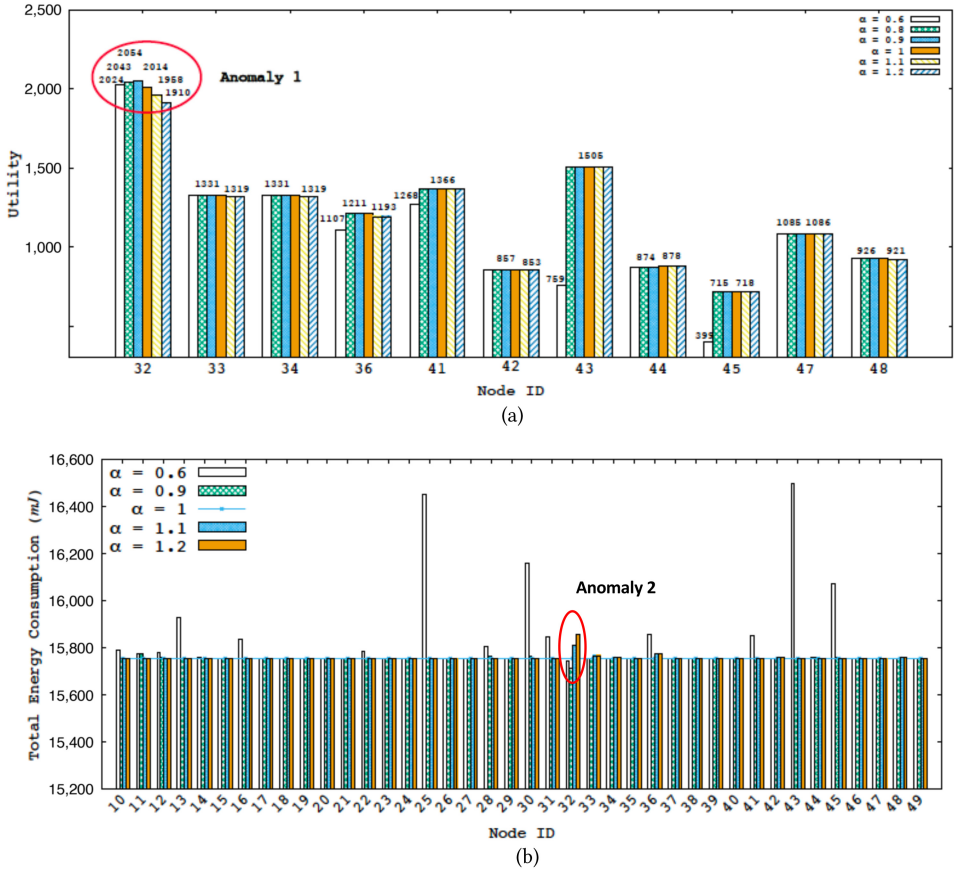


Fig. 13. Two anomaly cases in DPG-2. (a) Node 32's lying utility is larger that its truth-telling. (b) Total energy consumption (i.e., total preservation cost) resulted from node 32's lying is lower than the optimal total preservation cost when it is truth-telling.

48 has the largest workload, receiving around 200 data packets. Referring back to the BSN topology in Figure 7, node 48 is close to data nodes 3, 5, and 6, thus serving as the "traffic hub" to offload their data packets to the storage nodes located at the top-left region of the sensor field. However, although it is the most loaded, node 48's energy consumption of 728.4 mJ is not the highest. Instead, node 32 has the highest energy consumption among storage nodes, already depleting 1,312 mJ of its battery power. Like node 48, node 32 is close to a few data nodes (i.e., nodes 1 and 8); unlike node 48, node 32 is relatively distant from neighboring storage nodes, thus costing more transmission energy to relay data packets. Note that node 23 does not participate in the data process, as it sits on the very edge of the sensor field, incurring zero workloads and energy consumption.

7.2.2 Anomaly Cases in DPG-2. Now, we vary the scaling factor α of the node's transmission parameter ϵ^a from 0.6, 0.8. 0.9, 1, 1.1, to 1.2 and investigate each storage node's utility and the resulting total preservation cost of the network. We have observed two anomaly cases in Figure 13. First, Figure 13(a) shows that while most of the storage nodes comply with the VCG theory that their truth-telling utilities at $\alpha = 1$ are no less than their lying utilities, node 32 has a truth-telling utility that is less than its lying utilities at $\alpha = 0.6, 0.8$, and 0.9. Second, Figure 13(b) shows that

when node 32 lies at $\alpha = 0.6$ and 0.9, the resultant total preservation cost of the entire network is smaller than that under its truth-telling at $\alpha = 1$, contradicting the optimality of ILP (B). Note that for a few nodes (e.g., nodes 25 and 30 at $\alpha = 0.6$), the total preservation cost is larger than the optimal. This is because when some storage nodes claim they are more energy-efficient in data preservation than they actually are, the ILP (B) assigns them more data packets than it does in the optimal solution, resulting in much larger non-optimal energy consumption.

7.2.3 Investigating Data Loss. We conjecture that the lower than optimal energy consumption in Figure 13(b) is because some data packets are dropped in the BSN when node 32 lies with $\alpha = 0.6$ and 0.9. We thus implement Algorithm 1 to find if any storage node has dropped data packets during the data preservation process. Figure 14(a) shows when each storage node lies with $\alpha = 0.6$, its assigned (and actual) receive, transmit, and save, as illustrated in Figure 6, as well as its own individual energy consumption. By comparing the assigned load and actual load of each storage node following Algorithm 1, we find that nodes 25, 30, 32, 43, and 45 have dropped packets. This is further sustained by the fact that each has depleted its respective energy power of 1,312 mJ. Referring to the topology in Figure 7, these nodes are close to one or more data nodes and involve heavily relaying their data packets, thus consuming lots of energy. Therefore, we conclude that a storage node drops data packets due to energy depletion, causing data loss.

In particular, Figure 14(b) shows that node 32 has dropped data packets at $\alpha = 0.6, 0.8$, and 0.9, which explains why in Figure 13(b) the resultant total preservation cost when node 32 lies is less than optimal. Referring back to Figure 12, as node 32 already depletes its energy when truth-telling, when it scales down, it will be assigned more data packets than it is when truth-telling. Thus, it must drop any such extra packets assigned due to its energy depletion. As its energy consumption is still 1,312 mJ while other nodes get fewer data preservation tasks, the total preservation cost of the entire BSN is thus smaller than that of the optimal.

However, although nodes 25, 30, 43, and 45 at $\alpha = 0.6$ drop some data packets and cause data loss, the resultant total preservation costs are still larger than optimal, as shown in Figure 13(b). This indicates that data loss does not necessarily yield reduced total preservation cost. As each node is assigned more workloads when it lies, the resulting data preservation solutions could deviate from the optimal a lot and result in a much higher total preservation cost, although the node has dropped some packets due to insufficient energy. This shows that data loss is necessary for the less-than-optimal total preservation cost.

Another observation from Figure 14(b) is that when increasing α towards 1, the occurrences of data loss decrease. This is because as fewer data preservation tasks are assigned to a lying node, the less chance it will exhaust its battery power. Data loss does not occur for $\alpha = 1$, as data preservation is always feasible in truth-telling. When $\alpha > 1$, there is no data loss either; as each lying node claims to cost more than it actually does, it gets assigned fewer tasks than it normally does and thus has sufficient energy to finish the assigned tasks.

7.2.4 Why Does Only Node 32 Have the Incentive to Lie? Although a few nodes, including node 32, have dropped data packets when lying, why does only node 32 have a lying utility larger than its truth-telling one? This implies only node 32 has the incentive to lie. We define a storage node's *incentive to lie* as its lying utility minus its truth-telling utility; that is, the more utility it gets, the larger its incentive to lie.

This is attributed to two unique features of node 32 (among all the 40 storage nodes). First, node 32's energy has already been depleted in truth-telling data preservation. Therefore, when node 32's lying results in a heavier data preservation load, almost all the additional data load is discarded. Such additional data load does not increase node 32's true energy cost. Second, in the BSN, node 32 is located at a *strategically crucial* position where not only do multiple data nodes

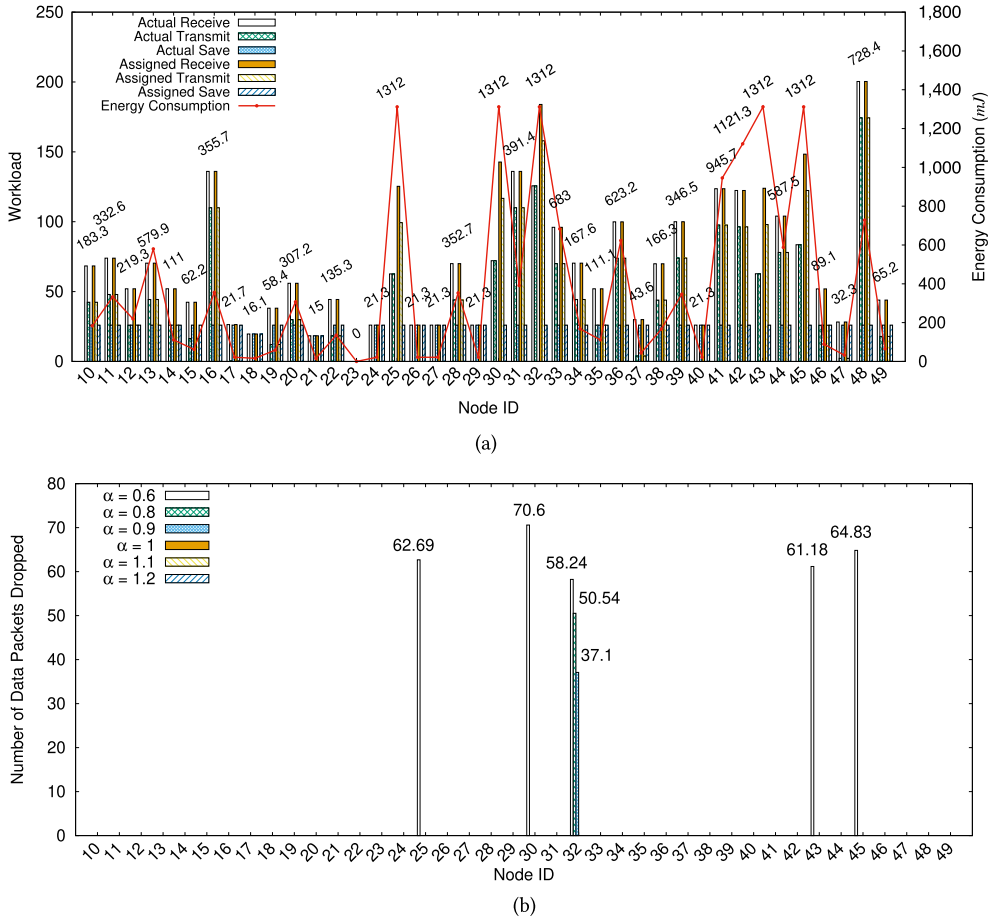
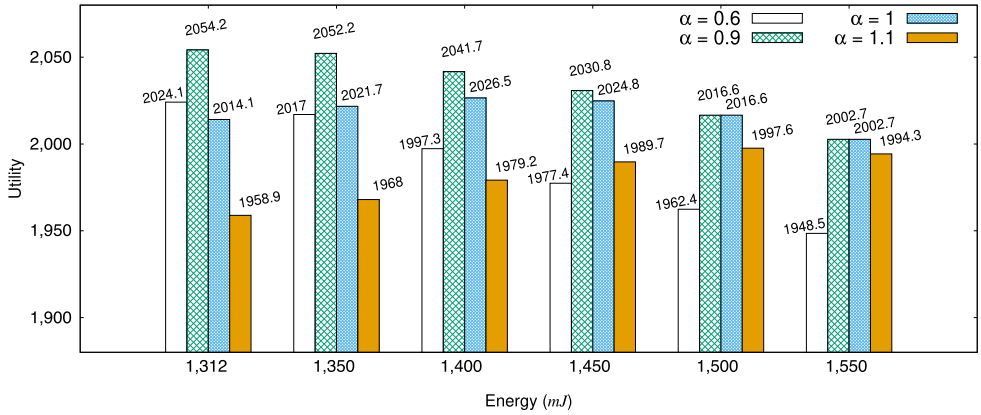


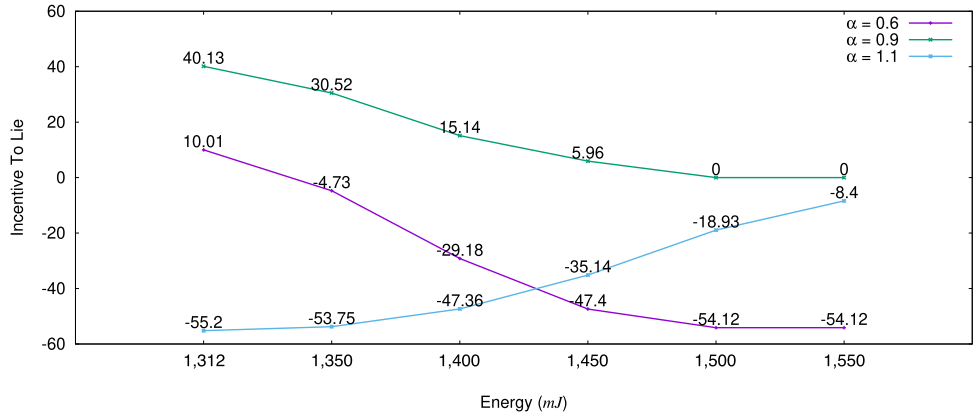
Fig. 14. (a) Each storage node's assigned (and actual) receive, transmit, and save, as well as energy consumption when $\alpha = 0.6$. (b) Number of dropped data packets by lying storage nodes.

(nodes 1 and 8) need it for data preservation, but also it has relatively long distances to any of its neighbors. As such, node 32 must carry out these costly data preservation tasks despite their needed large transmission energy. Meanwhile, when node 32 scales down its cost parameter with its additional data load assigned increased, the data preservation cost of all nodes other than node 32, which is $(\tilde{c}_V - \tilde{c}_i)$ in utility equation Equation (18), decreases. On the other side, constrained by its initial energy, node 32's true cost for data preservation c_i is roughly the same lying or not. Therefore, following Equation (18), node 32 garners a higher utility under lying than truth-telling.

In summary, for a storage node to find lying profitable, it must be on the verge of its energy depletion when truth-telling; it also must be in a strategically important position such that when it scales down, it will be assigned a much larger workload. *Then a scaling-down node can reduce the total preservation cost of the entire BSN by being assigned more workload while dropping data packets and not increasing its own energy cost, as it has already depleted its energy.* This, therefore, results in a larger payment for this strategically crucial lying node. Below, we take a closer look at node 32's selfish behavior and quantitatively examine how its incentive to lie changes w.r.t. the exogenous factors, including initial energy level E_i and storage capacity m_i.



(a) Utility of node 32.



(b) Incentive to lie of node 32.

Fig. 15. Utility and incentive to lie of node 32 when varying initial energy level E_i and scaling factor α.

Incentive to Lie when Varying E_i. Figure 15(a) shows the utility of node 32 lying with different values of α while increasing E_i from 1,312 mJ to 1,550 mJ and fixing m_i as 26. When $E_i \leq 1,450$ mJ, node 32 has a truth-telling utility smaller than at least one of its lying utilities. When $E_i \geq 1,500$ mJ, node 32 follows standard VCG theory in which the truth-telling utility is the dominant strategy. Such behavior can be explained by Figure 15(b), which shows that when $\alpha < 1$ (i.e., 0.6 and 0.9), node 32's incentive to lie decreases with the increase of E_i. As data loss by node 32 eventually drops with the increase of E_i, it shrinks the potential benefit accrued to node 32 through its data loss, reducing its incentive to lie.

Meanwhile, we have two observations of node 32's incentive to lie at $\alpha > 1$. First, it is always negative, giving node 32 no incentive to lie. This is because as no single node is critical to the data preservation feasibility (i.e., the data preservation is still feasible when any node is removed), no data loss could occur when $\alpha > 1$. Second, with the increase of E_i, node 32's incentive to lie increases and approaches zero, showing that its utility loss under lying drops when the energy constraint is relaxed. Node 32's lie becomes less harmful to the system performance when the energy constraint is lifted. The data preservation load assigned to node 32 gets less and less differentiated

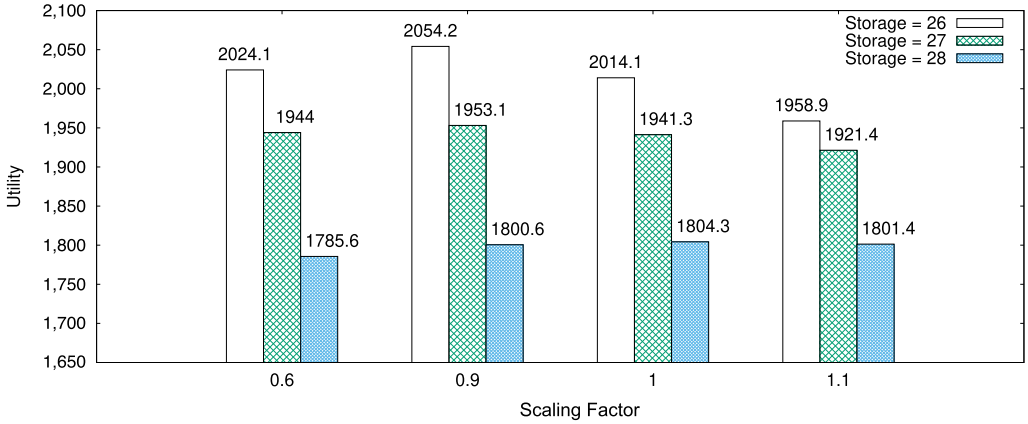


Fig. 16. Utility of node 32 when varying its storage capacity m_i and scaling factor α.

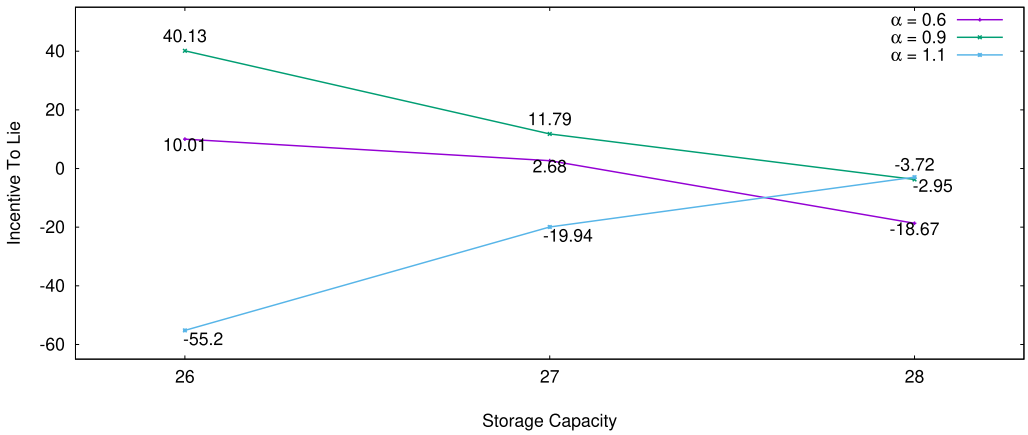


Fig. 17. Incentive to lie of node 32 when varying its storage capacity m_i and scaling factor α.

from its load under truth-telling. Indeed, if under a certain energy level, node 32 is assigned the same load under both $\alpha = 1.1$ and truth-telling, then its incentive to lie becomes zero.

Incentive to Lie when Varying m_i. Next, we fix the energy as 1,312 mJ and compare node 32's utilities by varying m_i. Figure 16 shows that when $\alpha = 0.6$ and 0.9 while $m_i = 26$ and 27, truth-telling utilities are less than the lying utilities, which is against the AMD theory. However, when the storage capacity reaches 28, the truth-telling utility will be larger than the lying utility. This shows that when m_i is small, nodes have more incentive to lie to gain more utilities, which is further validated by its incentive to lie, shown in Figure 17.

Figure 17 investigates the incentive to lie for node 32 under different storage capacities. When the scaling factor $\alpha < 1$ (i.e., 0.6 and 0.9), its intention to lie decreases when increasing the storage capacity. With a larger storage capacity, as each node does not need much energy to offload data, lying has less effect in distorting the data preservation route, thus resulting in less data loss and less intention to lie. We also observe while $\alpha = 1.1$ and we increase the storage capacity, the difference between lying utility and truth-telling utility will decrease.

7.2.5 Data Loss Inhibiting Mechanism in DPG-2. We have shown that under the traditional VCG mechanism, nodes can lie about their cost parameters, viz., ϵ^a, ϵ^e, ϵ^s to gain more utility. Below,

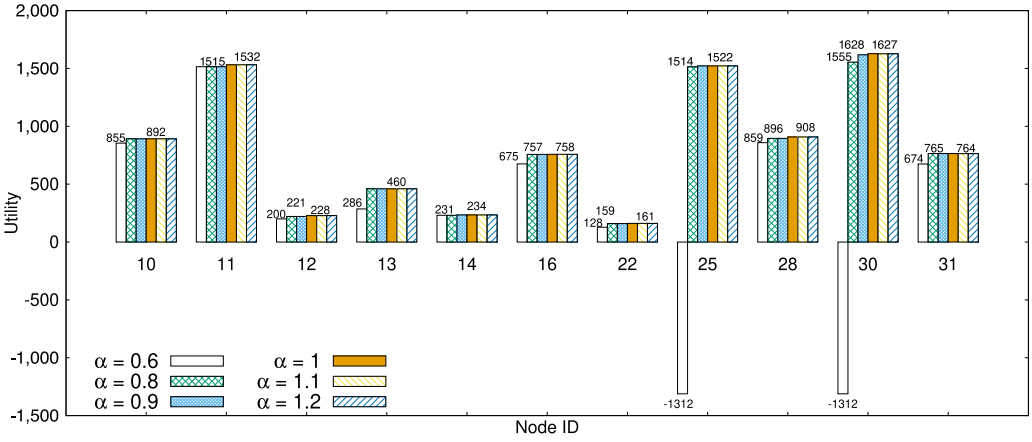


Fig. 18. Utilities of nodes 10–31 in DPG-2 with data loss inhibiting mechanism. $E_i = 1,312$ mJ and $m_i = 26$.

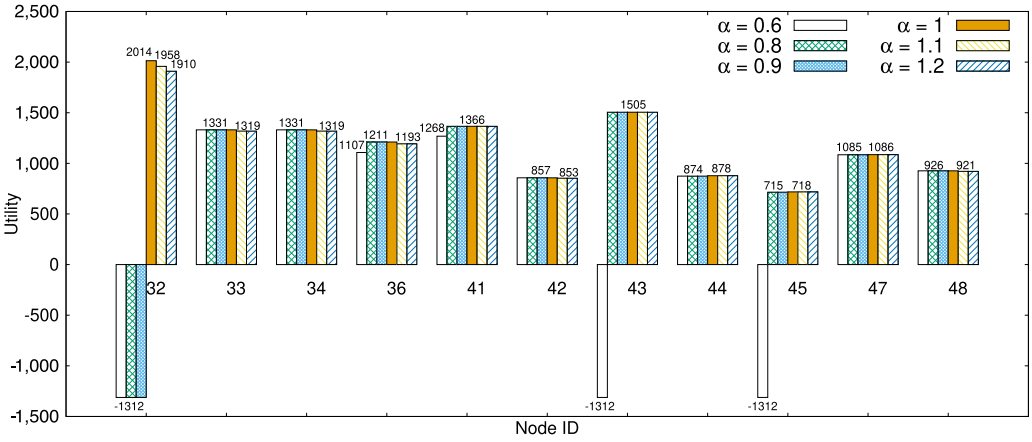


Fig. 19. Utilities of nodes 32–49 in DPG-2 with data loss inhibiting mechanism. $E_i = 1,312$ mJ and $m_i = 26$.

we implement DPG-2 to fix the problem. On top of DPG-1, DPG-2 introduced in Section 6 mainly contains a data loss inhibiting mechanism that, for any node that drops its received packages, the system gives it zero payment. Figures 18 and 19 calculate the utility of each node in DPG-2 when it is truth-telling or lies with different α. It can be seen that the truth-telling utility of each node is always greater than or equal to its lying utilities. This demonstrates that the data loss inhibiting mechanism in DPG-2 restores the effectiveness of VCG, with truth-telling again being the dominating strategy. Consequently, all the nodes will truthfully report their cost parameters, guaranteeing optimal system performance with minimum total data preservation cost.

Finally, we check if the truthfulness is still preserved in DPG-2 when each storage node lies all of its three parameters simultaneously. We set the initial energy as 1,312 mJ and m_i as 26. We first investigate a homogeneous case wherein each node lies about all its three cost parameters with the same $\alpha = 0.6$, as shown in Figure 20. We then investigate a heterogenous case wherein each node lies about all its three parameters with α a random number in $[0.1, 1]$, as shown in Figure 21. We compare their truth-telling and lying utility in each case. Both cases show that each storage node's lying utility is no more than its truth-telling utility, demonstrating that truth-telling is a

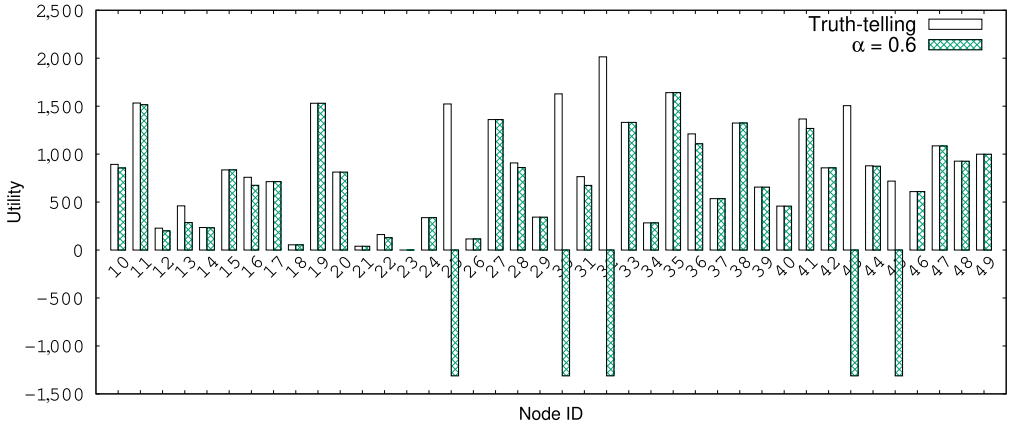


Fig. 20. Utilities of nodes in DPG-2 with homogeneous α. $\alpha = 0.6$, $E_i = 1,312$ mJ, and $m_i = 26$.

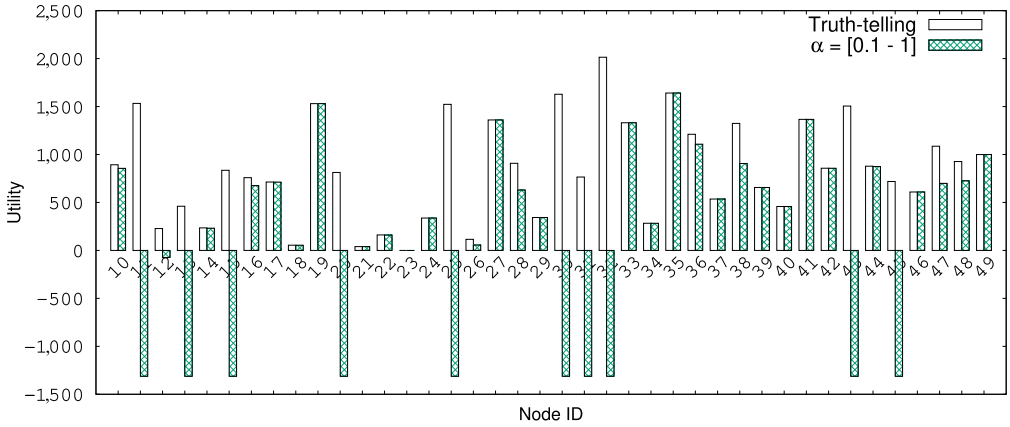


Fig. 21. Utilities of nodes in DPG-2 with heterogenous α. α is a random number in [0.1, 1], $E_i = 1,312$ mJ, and $m_i = 26$.

dominant strategy for the storage nodes in DPG-2. When α is a random number in [0.1, 1], some lying nodes will report their cost parameters by scales smaller than 0.6. Consequently, these nodes could be compensated less by the central algorithm although assigned a heavier workload due to its low reported cost for data preservation. This explains the more frequent occurrence of negative utilities in Figure 21 with Figure 20.

8 CONCLUSION AND FUTURE WORK

In this work, we study the data preservation problem in base station-less sensor networks (BSNs) wherein energy- and storage-constrained sensor nodes behave selfishly. BSNs find many emerging science applications in challenging environments, such as underwater exploration. Our goal is to minimize the total data preservation cost (i.e., total battery power consumption of sensor nodes) in the BSN while accommodating the selfish behavior of sensor nodes. We take a game-theoretic approach and design a suite of data preservation games wherein the individual sensor nodes, motivated solely by self-interest, achieve a good system-wide data preservation solution. Our BSN model, with its theoretical roots in network flows and simplicity, may inspire

new architectures for future network infrastructures and applications. It is indeed a very general information producer and consumer model that has not been adequately explored in any context.

In particular, we formulate and solve a general data preservation problem called DPP and design an integer linear program (ILP) to solve DPP optimally. To the extent of our knowledge, the DPP has not been studied in the existing literature. In a special case of DPP wherein energy constraint is not considered (i.e., DPP-W), we identify the challenges of applying VCG mechanism into our BSN model to motivate selfish sensor nodes. We show that with these challenges, our game (i.e., DPG-1) still achieves a truthful and optimal system-wide data preservation solution with self-interested sensor nodes. However, in the general case of DPP, wherein nodes have a finite amount of energy, we observe that DPG-1 can no longer provide truth-telling and optimal data preservation. We thus design another data preservation game, viz., DPG-2, to fix the flawed VCG model used in DPG-1. We show via proof and simulations that DPG-2 not only delivers truth-telling as a node's dominant strategy but also achieves optimal data preservation in the BSN while accommodating the selfish behavior of energy-constrained sensor nodes. In contrast to many existing works that applied game theory and related techniques to solve sensor networking problems, our work takes a network flow approach to facilitate the game design and game-theoretical analysis. Due to the theoretical roots of the game theory and network flows, the designed techniques in our DPG-1 and DPG-2 are applicable to any network environment where game theory and network flows play a role.

Currently, we do not consider the budget imbalance, which is the amount the central authority needs to finance the data preservation process. We will consider if there exists an upper bound of such budget imbalance. Second, we will consider that a storage node can also lie about its storage capacity. Although lying about either energy cost or storage capacity can be treated similarly in analysis, lying about both simultaneously in the data preservation game is a more challenging problem. Third, we have assumed that although sensor nodes have limited storage and energy capacity, data preservation is still feasible; that is, all the data can be offloaded from data nodes to some storage nodes. In future work, we will consider infeasible data preservation, wherein not all the data can be offloaded and preserved due to either energy insufficiency or storage insufficiency, or both, and study the game-theoretic behavior of sensor nodes in these more challenging scenarios. Finally, we will extend our analysis to a dynamic scenario wherein overflow data are generated from time to time at different nodes. It is well understood in game theory that an infinitely repeated game gives a much larger set of equilibria, and in certain scenarios, full cooperation can be achieved. In our setting of data preservation among selfish nodes, it is interesting to see to what extent we need to provide motivation for selfish storage nodes to cooperate to engage in optimal data preservation.

REFERENCES

[1] G. Aathur, P. Desnoyers, D. Ganesan, and P. Shenoy. 2009. Ultra-low power data storage for sensor networks. *ACM Trans. Sensor Netw.* 5, 4 (2009), 33:1–33:34.

[2] A. Abrardo, L. Balucanti, and A. Mecocci. 2013. A game theory distributed approach for energy optimization in WSNs. *ACM Trans. Sensor Netw.* 9, 4, Article 44 (July 2013).

[3] A. Agah, S. K. Das, and K. Basu. 2004. A game theory based approach for security in wireless sensor networks. In *IEEE IPCCC'04*. 259–263.

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

[5] B. Alhakami, B. Tang, J. Han, and M. Beheshti. 2019. DAO-R: Integrating data aggregation and offloading in sensor networks via data replication. *Int. J. Sensor Netw.* 29, 2 (2019), 134 – 146.

[6] T. AlSkaif, M. G. Zapata, and B. Bellalta. 2015. Game theory for energy efficiency in wireless sensor networks: Latest trends. *J. Netw. Comput.* 54 (2015), 33–61.

[7] L. Anderegg and S. Eidenbenz. 2003. Ad hoc-VCG: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In *MobiCom'03*. 245–259.

[8] A. Asheralieva and D. Niyato. 2019. Game theory and Lyapunov optimization for cloud-based content delivery networks with device-to-device and UAV-enabled caching. *IEEE Trans. Vehic. Technol.* 68, 10 (2019), 10094–10110.
[9] A. Attiah, M. Amjad, M. Chatterjee, and C. Zou. 2018. An evolutionary routing game for energy balance in wireless sensor networks. *Comput. Netw.* 138 (2018), 31–43.
[10] S. Basagni, L. Boloni, P. Gjanci, C. Petrioli, C. A. Phillips, and D. Turgut. 2014. Maximizing the value of sensed information in underwater wireless sensor networks via an autonomous underwater vehicle. In *INFOCOM'14*.
[11] D. E. Charilas and A. D. Panagopoulos. 2010. A survey on game theory applications in wireless networks. *Comput. Netw.* 54, 18 (Dec. 2010), 3421–3430.
[12] F. Chen and J. Kao. 2013. Game-based broadcast over reliable and unreliable wireless links in wireless multihop networks. *IEEE Trans. Mob. Comput.* 12, 8 (2013), 1613–1624.
[13] J. Chen, Q. Yu, P. Cheng, Y. Sun, Y. Fan, and X. Shen. 2011. Game theoretical approach for channel allocation in wireless sensor and actuator networks. *IEEE Trans. Automat. Contr.* 56, 10 (2011), 2332–2344.
[14] X. Chen. 2015. Decentralized computation offloading game for mobile cloud computing. *IEEE Trans. Parallel Distrib. Syst.* 26, 4 (2015), 974–983.
[15] Y. Chen and B. Tang. 2016. Data preservation in base station-less sensor networks: A game theoretic approach. In *EAI GameNets'16*.
[16] Z. Chen, Y. Qiu, J. Liu, and L. Xu. 2011. Incentive mechanism for selfish nodes in wireless sensor networks based on evolutionary game. *Comput. Math. Applic.* 62, 9 (2011), 3378–3388.
[17] E. H. Clarke. 1971. Multipart pricing of public goods. *Pub. Choice* 11, 1 (1971), 17–33.
[18] T. Corman, C. Leiserson, R. Rivest, and C. Stei. 2009. *Introduction to Algorithms*. MIT Press.
[19] N. Crary, B. Tang, and S. Taase. 2015. Data preservation in data-intensive sensor networks with spatial correlation. In *MobiData'15*.
[20] D. DiPalantino and R. Johari. 2009. Traffic engineering vs. content distribution: A game theoretic perspective. In *IEEE INFOCOM'09*. 540–548.
[21] M. Doudou, J. M. Barcelo-Ordinas, D. Djenouri, J. Garcia-Vidal, A. Bouabdallah, and N. Badache. 2016. Game theory framework for MAC parameter optimization in energy-delay constrained sensor networks. *ACM Trans. Sen. Netw.* 12, 2 (May 2016).
[22] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand. 2012. Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing. In *IEEE INFOCOM'12*.
[23] S. Edwards, T. Murray, T. O'Farrell, I. C. Rutt, P. Loskot, I. Martin, N. Selmes, R. Aspey, T. James, S. L. Bevan, and T. Baugé. 2013. A high-resolution sensor network for monitoring glacier dynamics. *IEEE Sensors J.* 14, 11 (2013), 3926–3931.
[24] 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 *IEEE IPDPS'05*.
[25] M. Elhoseny, K. Shankar, and M. Abdel-Basset. 2020. *Artificial Intelligence Techniques in IoT Sensor Networks*. Chapman and Hall/CRC.
[26] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker. 2005. A BGP-based mechanism for lowest-cost routing. *Distrib. Comput.* 18, 1 (2005), 61–72.
[27] J. Feigenbaum, C. Papadimitriou, and S. Shenker. 2001. Sharing the cost of multicast transmissions. *J. Comput. Syst. Sci. (Special Issue Internet Algor.)* 63 (2001), 21–41.
[28] M. Felegyhazi, J. P. Hubaux, and L. Buttyan. 2006. Nash equilibria of packet forwarding strategies in wireless ad hoc networks. *IEEE Trans. Mob. Comput.* 5, 5 (2006), 463–476.
[29] D. Fudenberg and J. Tirole. 1991. *Game Theory*. MIT Press.
[30] R. Ghaffarivardavagh, S. S. Afzal, O. Rodriguez, and F. Adib. 2020. Ultra-wideband underwater backscatter via piezoelectric metamaterials. In *ACM SIGCOMM'20*.
[31] S. M. Ghoreyshi, A. Shahrabi, and T. Boutaleb. 2018. An efficient AUV-aided data collection in underwater sensor networks. In *IEEE AINA'18*.
[32] D. Glaroudis, A. Iossifides, and P. Chatzimisios. 2020. Survey, comparison and research challenges of IoT application protocols for smart farming. *Comput. Netw.* 168 (2020).
[33] A. V. Goldberg. 1997. An efficient implementation of a scaling minimum-cost flow algorithm. *J. Algor.* 22 (1997), 1–29.
[34] A. V. Goldberg, E. Tardos, and R. E. Tarjan. 1990. Network flow algorithmss. *Paths, Flows VLSI-Des.* 9 (1990), 101–164.
[35] P. Golle, K. Leyton-Brown, and I. Mironov. 2001. Incentives in peer-to-peer file sharing. In *ACM EC'01*.
[36] X. Gong and N. Shroff. 2018. Incentivizing truthful data quality for quality-aware mobile data crowdsourcing. In *MobiHoc'18*. 161–170.
[37] X. Gong and N. Shroff. 2019. Truthful mobile crowdsensing for strategic users with private data quality. *IEEE/ACM Trans. Netw.* 27, 5 (2019), 1959–1972.
[38] J. Green and J. Laffont. 1979. Incentives in public decision making. *Stud. Pub. Econ.* 1 (1979), 65–78.

[39] T. Groves. 1973. Incentives in teams. *Econometrica* 41, 4 (1973), 617–631.
[40] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. 2000. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS*.
[41] X. Hou, Z. Sumpter, L. Burson, X. Xue, and B. Tang. 2012. Maximizing data preservation in intermittently connected sensor networks. In *MASS*.
[42] S. Hsu, Y. Yu, and B. Tang. 2020. DRE^2: Achieving data resilience in wireless sensor networks: A quadratic programming approach. In *IEEE MASS*.
[43] S. Hu, G. Li, and G. Huang. 2021. Dynamic spatial-correlation-aware topology control of wireless sensor networks using game theory. *IEEE Sensors J.* 21, 5 (2021), 7093–7102.
[44] H. Huang, A. V. Savkin, M. Ding, and C. Huang. 2019. Mobile robots in wireless sensor networks: A survey on tasks. *Comput. Netw.* 148 (2019), 1–19.
[45] J. Jang and F. Adib. 2019. Underwater backscatter networking. In *ACM SIGCOMM*.
[46] J. Jaramillo and R. Srikant. 2010. A game theory based reputation mechanism to incentivize cooperation in wireless Ad Hoc networks. *Ad Hoc Netw.* 8, 4 (June 2010), 416–429.
[47] H. Jin, L. Su, and K. Nahrstedt. 2017. Theseus: Incentivizing truth discovery in mobile crowd sensing systems. In *ACM MobiHoc*.
[48] R. Kannan and S. Iyengar. 2004. Game-theoretic models for reliable path-length and energy-constrained routing with data aggregation in wireless sensor networks. *IEEE J. Select. Areas Commun.* 22 (2004), 1141–1150.
[49] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad. 2013. Mobile phone sensing systems: A survey. *IEEE Commun. Surv. Tutor.* 15, 1 (2013), 402–427.
[50] I. Koutsopoulos. 2013. Optimal incentive-driven design of participatory sensing sys-tems. In *IEEE INFOCOM'13*.
[51] E. Koutsoupias and C. Papadimitriou. 2016. Worst-case equilibria. *Comput. Sci. Rev.* 3, 2 (2016), 65–69.
[52] Z. Li and H. Shen. 2012. Game-theoretic analysis of cooperation incentive strategies in mobile ad hoc networks. *IEEE Trans. Mob. Comput.* 11, 8 (2012), 1287–1303.
[53] W. Liang, X. Ren, X. Jia, and X. Xu. 2013. Monitoring quality maximization through fair rate allocation in harvesting sensor networks. *IEEE Trans. Parallel Distrib. Syst.* 24, 9 (2013), 1827–1840.
[54] H. Lim, G. Ghinita, E. Bertino, and M. Kantarcioglu. 2012. A game-theoretic approach for high-assurance of data trustworthiness in sensor networks. In *IEEE ICDE'12*.
[55] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan. 2007. An acknowledgment-based approach for the detection of routing misbehavior in MANETs. *IEEE Trans. Mob. Comput.* 6, 5 (May 2007), 536–550.
[56] L. Liu, R. Wang, D. Guo, and X. Fan. 2016. Message dissemination for throughput optimization in storage-limited opportunistic underwater sensor networks. In *SECON*.
[57] J. Ly, Y. Chen, and B. Tang. 2023. Data-VCG: A data preservation game for base station-less sensor networks with performance guarantee. In *TENSOR'23*.
[58] S. Marti, T. J. Giuli, K. Lai, and M. Baker. 2000. Mitigating routing misbehavior in mobile ad hoc networks. In *ACM MobiCom'00*.
[59] M. Maskery and V. Krishnamurthy. 2007. Decentralized adaptation in sensor networks: Analysis and application of regret-based algorithms. In *IEEE CDC'07*.
[60] D. Mosse and G. Gadola. 2012. Controlling wind harvesting with wireless sensor networks. In *IGCC*.
[61] N. Nisan. 1999. Algorithms for selfish agents: Mechanism design for distributed computation. In *STACS'99*..
[62] N. Nisan and A. Ronen. 1999. Algorithmic mechanism design. In *STOC'99)*. 129–140.
[63] N. Nisan and A. Ronen. 2007. Algorithmic mechanism design. *Games Econ. Behav.* 35 (2007), 166–196.
[64] D. Niyato, E. Hossain, M. M. Rashid, and V. K. Bhargava. 2007. Wireless sensor networks with energy harvesting technologies: A game-theoretic approach to optimal energy management. *IEEE Wirel. Commun.* 14, 4 (2007), 90–96.
[65] N. Pathak, A. Mukherjee, and S. Misra. 2020. Reconfigure and reuse: Interoperable wearables for healthcare IoT. In *IEEE INFOCOM'20*.
[66] F. Pavlidou and G. Koltsidas. 2008. Game theory for routing modeling in communication networks—A survey. *J. Commun. Netw.* 10, 3 (Sep. 2008), 268–286.
[67] D. Peng, F. Wu, and G. Chen. 2018. Data quality guided incentive mechanism design for crowdsensing. *IEEE Trans. Mob. Comput.* 17, 2 (2018), 307–319.
[68] D. E. Phillips, M. Moazzami, G. Xing, and J. M. Lees. 2017. A sensor network for real-time volcano tomography: System design and deployment. In *IEEE ICCCN'17*.
[69] X. Qin, X. Wang, L. Wang, Y. Lin, and Wang X. 2019. An efficient probabilistic routing scheme based on game theory in opportunistic networks. *Comput. Netw.* 149 (2019), 144–153.
[70] M. Rahmati and D. Pompili. 2019. UWSVC: Scalable video coding transmission for in-network underwater imagery analysis. In *IEEE MASS'19*.

[71] B. Rashid and M. H. Rehmani. 2016. Applications of wireless sensor networks for urban areas: A survey. *J. Netw. Comput. Applic.* 60 (2016), 192–219.
[72] J. Rivera, Y. Chen, and B. Tang. 2023. On the performance of Nash equilibria of data preservation in base station-less sensor networks. In *IEEE MASS 2023*.
[73] S. Sengupta, M. Chatterjee, and K. Kwiat. 2010. A game theoretic framework for power control in wireless sensor networks. *IEEE Trans. Comput.* 59, 2 (2010), 231–242.
[74] I. F. Senturk, K. Akkaya, and S. Yilmaz. 2014. Relay placement for restoring connectivity in partitioned wireless sensor networks under limited information. *Ad Hoc Netw.* 13 (2014), 487–503.
[75] H.-Y. Shi, W.-L. Wang, N.-M. Kwok, and S.-Y. Chen. 2012. Game theory for wireless sensor networks: A survey. *Sensors (Basel, Switz.)* 12 (12 2012), 9055–97.
[76] J. Shneidman and D. C. Parkes. 2003. Rationality and self-interest in peer to peer networks. In *IPTPS'03*.
[77] Y. Shoham. 2008. Computer science and game theory. *Commun. ACM* 51, 8 (Aug. 2008), 74–79.
[78] A. A. Syed, W. Ye, and J. Heidemann. 2008. T-Lohi: A new class of MAC protocols for underwater acoustic sensor networks. In *IEEE INFOCOM'08*. Retrieved from: http://www.isi.edu/ilense/snuse/index.html.
[79] R. Tan, G. Xin, J. Chen, W.-Z. Song, and R. Huang. 2013. Fusion-based volcanic earthquake detection and timing in wireless sensor networks. *ACM Trans. Sensor Netw.* 9 (2013).
[80] B. Tang. 2018. DAO^2: Overcoming overall storage overflow in intermittently connected sensor networks. In *IEEE INFOCOM'18*, 1–9.
[81] B. Tang, N. Jaggi, and M. Takahashi. 2014. Achieving data K-availability in intermittently connected sensor networks. In *ICCCN'14*.
[82] B. Tang, N. Jaggi, H. Wu, and R. Kurkal. 2013. Energy-efficient data redistribution in sensor networks. *ACM Trans. Sensor Netw.* 9, 2 (Apr. 2013), 11:1–11:28.
[83] B. Tang, H. Ngo, Y. Ma, and B. Alhakami. 2023. DAO^2: Overcoming overall storage overflow in intermittently connected sensor networks. *Accepted at IEEE/ACM Transactions on Networking* (2023), 1–16. DOI: 10.1109/TNET.2023.3273553
[84] W. Vickrey. 1961. Counterspeculation, auctions and competitive sealed tenders. *J. Finance* 16, 1 (1961), 8–37.
[85] A. C. Voulkidis, M. P. Anastasopoulos, and P. G. Cottis. 2013. Energy efficiency in wireless sensor networks: A game-theoretic approach based on coalition formation. *ACM Trans. Sensor Netw.* 9, 4 (July 2013).
[86] W. Wang and X.-Yang Li. 2006. Low-cost routing in selfish and rational wireless ad hoc networks. *IEEE Trans. Mob. Comput.* 5, 5 (2006), 596–607.
[87] D. Wang, J. Ren, Z. Wang, Y. Wang, and Y. Zhang. 2023. PrivAim: A dual-privacy preserving and quality-aware incentive mechanism for federated learning. *IEEE Trans. Comput.* 72, 7 (2023), 1913–1927.
[88] Q. Wang and W. Yang. 2007. Energy consumption model for power management in wireless sensor networks. In *SECON'07*.
[89] W. Wang, X. Y. Li, and Y. Wang. 2004. Truthful multicast routing in selfish wireless networks. In *MobiCom*. 245–259.
[90] Z. Wang, T. Alpcan, J. S. Evans, and S. Dey. 2019. Truthful mechanism design for wireless powered network with channel gain reporting. *IEEE Trans. Commun.* 67, 11 (2019), 7966–7979.
[91] A. Whitmore, A. Agarwal, and L. Da Xu. 2015. The internet of things—A survey of topics and trends. *Inf. Syst. Front.* 17 (2015), 261–274.
[92] A. Wichmann, T. Korkmaz, and A. S. Tosun. 2018. Robot control strategies for task allocation with connectivity constraints in wireless sensor and robot networks. *IEEE Trans. Mob. Comput.* 17, 6 (2018), 1429–1441.
[93] F. Wu, T. Chen, S. Zhong, L. Li, and Y. Yang. 2008. Incentive-compatible opportunistic routing for wireless networks. In *ACM MobiCom'08*. 303–314.
[94] F. Wu, K. Gong, T. Zhang, G. Chen, and C. Qiao. 2015. COMO: A game-theoretic approach for joint multirate opportunistic routing and forwarding in non-cooperative wireless networks. *IEEE Trans. Wirel. Commun.* 14, 2 (2015), 948–959.
[95] L. D. Xu, W. He, and S. Li. 2014. Internet of things in industries: A survey. *IEEE Trans. Industr. Inform.* 10, 4 (2014), 2233–2243.
[96] X. Xue, X. Hou, B. Tang, and R. Bagai. 2013. Data preservation in intermittently connected sensor networks with data priorities. In *SECON'13*.
[97] D. Yang, G. Xue, X. Fang, and J. Tang. 2016. Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones. *IEEE/ACM Trans. Netw.* 24, 3 (2016), 1732–1744.
[98] H. Yedidsion, A. Banik, P. Carmi, M. J. Katz, and M. Segal. 2017. Efficient data retrieval in faulty sensor networks using a mobile mule. In *WiOpt'17*.
[99] H. Yetgin, K. Cheung, M. El-Hajjar, and L. A. Hanzo. 2017. Survey of network lifetime maximization techniques in wireless sensor networks. *IEEE Commun. Surv. Tutor.* 19 (2017), 828–854.

[100] Q. Yu, J. Chen, Y. Fan, X. Shen, and Y. Sun. 2010. Multi-channel assignment in wireless sensor networks: A game theoretic approach. In *IEEE INFOCOM'10*. 1–9.
[101] C. Zhang, X. Zhu, Y. Song, and Y. Fang. 2010. A formal study of trust-based routing in wireless ad hoc networks. In *IEEE INFOCOM'10*.
[102] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao. 2016. Incentives for mobile crowd sensing: A survey. *IEEE Commun. Surv. Tutor.* 18, 1 (2016), 54–67.
[103] H. Zheng and J. Wu. 2015. Data collection and event detection in the deep sea with delay minimization. In *SECON'15*.
[104] Z. Zheng, Y. Peng, F. Wu, S. Tang, and G. Chen. 2017. Trading data in the crowd: Profit-driven data acquisition for mobile crowdsensing. *IEEE J. Select. Areas Commun.* 35, 2 (2017), 486–501.
[105] S. Zhong, J. Chen, and Y. R. Yang. 2003. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *IEEE INFOCOM'03*.

Received 17 June 2022; revised 8 May 2023; accepted 13 June 2023