Nash Equilibria of Data Preservation in Base Station-less Sensor Networks

Giovanni Rivera[1], Yutian Chen[2], Bin Tang[1]

[1] Department of Computer Science, California State University Dominguez Hills
[2] Economics Department, California State University, Long Beach

Abstract

We study the Nash Equilibrium *(NE) for data preservation in* base station-less sensor networks *(BSNs), wherein sensor nodes could behave selfishly. We design a suite of data preservation games that achieve NEs while minimizing energy consumption in the data preservation process. We analyze and quantify the efficiency loss of data preservation NEs by studying* price of anarchy *and* price of stability*, the ratios between the quality of the worst and best NEs to the quality of a minimum cost flow-based social optimal solution. Finally, we conduct extensive simulations to validate our theoretical results.*

Keywords – Nash Equilibria, Price of Anarchy/Stability, Data Preservation, Base Station-less Sensor Networks.

1. Introduction

Background and State-of-the-Arts. Recently, *Base station-less sensor networks* (BSNs) have drawn much attention from the research community [1,2]. BSNs refer to an array of emerging data-intensive sensing applications developed and deployed in recent years, including underwater or ocean exploration [3,4], volcano eruption monitoring [5], and seismic sensor networks [6]. The above applications are all deployed in some challenging environments and remote areas; thus, it is not feasible to deploy high-power and high-storage data-collecting base stations in or near the sensing area. One important function of BSNs is storing large volumes of sensory data inside the network between uploading opportunities including autonomous underwater vehicles (AUVs) and robots.

In the BSN, some sensor nodes are close to the events of interest and are constantly generating sensory data, thus have depleted their storage spaces; they are referred to as *source nodes*. To avoid data loss, such overflow data must be offloaded to other sensor nodes in the BSN with available storage (referred to as *storage nodes*). The process of offloading overflow data from source to storage nodes is called *data preservation in BSNs*.

Motivation. With the strides made in sensor network development and IoT applications over the past decade, the technologically advanced sensor nodes could become more intelligent [10]. Unlike the traditional sensors that can only sense, compute, and communicate signals to an outer 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 [11].

We apply game-theoretical techniques to analyze the selfish behavior of sensor nodes in the data preservation of BSNs. Game theory [12] is increasingly attracting more attention as a mechanism to solve various problems related to intelligent multi-agent systems. In particular, we attempt to achieve Nash Equilibrium (NE) of data preservation in the BSN. NE is a solution in a non-cooperative game wherein each player's strategy is optimal when considering the decisions of other nodes [13,14]. In particular, we intend to answer the following two questions. Do there exist data preservation games that achieve NEs? If so, what is the efficiency loss of data preservation due to the selfishness of sensor nodes? We design three data preservation games and prove that they all achieve NEs. In particular, the minimum cost flow-based game achieves NEs and minimum preservation cost.

2. Related Work

Game theory techniques have been applied to solve research problems in wireless sensor networks [12]. Voulkidis et al. [15] proposed a coalitional game-theoretic scheme that maximizes the lifetime of sensor networks. Niyato et al. [16] 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 solution of the game.

However, none of the above works addressed data preservation in BSNs. As it is always assumed in traditional sensor networks that the generated data packets would go to the well-known destination (i.e., the base station), the existing works mainly focused on incentivizing

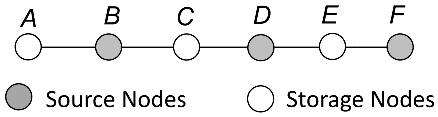

Figure 1. Illustrating the DPP.

the nodes on the shortest path or minimum spanning tree among the data source and the base station. In contrast, in our BSN model, we need to find the destination nodes for the data packets and then the routing paths for the data packets to go to the destination nodes.

The only works that applied game theory to tackle data preservation in the BSN are [17, 18]. They used algorithmic mechanism design techniques to incentivize selfish sensor nodes to participate in the data preservation. However, none studied NEs in data preservation. NE is the most fundamental game theory concept that determines the optimal strategies for selfish players in a non-cooperative game without being incentivized. We design a suite of data preservation games that not only reach NEs, but some of them give minimum total preservation cost in the BSN.

3. Data Preservation in the BSN

Network Model. We model a BSN as an undirected connected graph $G(V, E)$, where $V = V_s \cup V_r$ includes a set of source nodes V_s and a set of storage nodes V_r. Assume $|V| = n$, $|V_s| = k$, $|V_r| = q$, where $k + q = n$, and denote $V_s = \{S_1, S_2, ..., S_k\}$, and $V_r = \{R_1, R_2, ..., R_q\}$. Let $d_i > 0$ denote the number of overflow data packets generated at source node $S_i \in V_s$, and each packet is a bits. Let $d = \sum_{i=1}^{k} d_i$ be the total number of overflow data packets and let $\mathcal{D} = \{D_1, D_2, ..., D_d\}$ denote the set of d data packets. Let $s(j) \in V_s$, $1 \leq j \leq d$ denote the source node where D_j is generated. Let $m_j > 0$ be the available free storage space (in terms of the number of data packets) at storage node $R_j \in V_r$. We assume that $\sum_{j=1}^{q} m_j \geq d$; otherwise, there is not enough space to store all the overflow data packets.

Cost Model. Following the first-order radio model [19], when node u sends a data packet to its neighbor v over their distance $l_{u,v}$, the amount of *transmitting energy* spent by u is $E_u^t(v) = a \cdot \epsilon^a \cdot l_{u,v} + a \cdot \epsilon^e$. Here, ϵ^a and ϵ^e are the energy consumption of transmitting one bit on the transmit amplifier and circuit of node u, respectively. Their values are $100\, pJ/bit/m^2$ and $100\, nJ/bit$ respectively, following [19]. When node v receives a data packet, the amount of *receiving energy* it spends is $E_v^r = a \cdot \epsilon^e$. Here, ϵ^e is the energy consumption of receiving one bit on the circuit of node v. Given an edge $(u, v) \in E$, we define its weight $w(u, v)$ as the total energy consumption sending the packet from v to v; that is, $w(u, v) = E_u^t(v) + E_v^r$.

Let $P_j = \{S_{s(j)}, ..., R_{f(j)}\}$ be the *preservation path* along which D_j is offloaded from its source node $S_{s(j)} \in V_s$ to a storage node $R_{f(j)} \in V_r$, which is also the shortest path between $S_{s(j)}$ and $R_{f(j)}$. The total energy cost of preserving D_j, denoted as $c_j = \sum_{(u,v) \in P_j} w(u, v)$, is referred to as D_j's *preservation cost*. Let $c(i)$ denote the preservation cost of all the d_i data packets at source node S_i; $c(i) = \sum_{j=1}^{d_i} c_j$.

Problem Formulation of DPP. We define a *preservation function* as $f : \mathcal{D} \to V_r$, signifying that data packet $D_j \in \mathcal{D}$ is offloaded from its source node $S_{s(j)} \in V_s$ to a storage node $R_{f(j)} \in V_r$. The goal of the DPP is to find an f and a preservation path P_j to offload each D_j to $R_{f(j)}$, such that the *total preservation cost* $\sum_{j=1}^{d} c_j = \sum_{i=1}^{k} c(i)$ is minimized under the storage constraint of storage nodes: $|\{j | 1 \leq j \leq d, f(j) = i\}| \leq m_i, \forall R_i \in V_r$.

EXAMPLE 1: Fig. 1 shows a linear BSN with 6 sensor nodes viz. A to F and the energy cost on each edge is 1. Nodes B, D, and F are source nodes; each has one overflow data packet; nodes A, C, and E are storage nodes, each having a storage capacity of 1. The optimal data preservation solution is to offload B's packet to A, D's packet to C, and F's packet to E, resulting in a minimum preservation cost of 3. There are many other solutions; however, none of them is optimal. □

4. Data Preservation Games of DPP

In this section, we first define the Nash Equilibrium (NE) in the context of data preservation and introduce the concepts that quantify the efficiency loss in the NE. We then present a suite of data preservation games that achieve the NE and analyze their efficiency loss.

4.1. Nash Equilibrium (NE) in Data Preservation Games.

In data preservation games, the players are the k source nodes $\{S_1, S_2, \ldots, S_k\}$. Player S_i has a set of strategies $\mathcal{A}_i$, each indicating for its d_i data packets, how many are offloaded to which storage nodes following the shortest path between them. Given a strategy $s_i \in \mathcal{A}_i$ and its incurred preservation cost $c(i)$, player S_i receives a corresponding *utility* or *payoff* of $u_i = -c(i)$.

In the data preservation game, all the players act selfishly to maximize their payoff, thus minimizing the data preservation cost incurred in offloading its data packets. Given that s_i is a particular strategy chosen by player S_i, let s_{-i} denote the particular strategies chosen by all other players in the game. A set of strategies $s = \{s_i, s_{-i}\}$ is called a *strategy profile* of the game, and different choices of strategies from the players produce a different strategy profile. We are interested in a strategy profile $s = \{s_1, s_2, \ldots, s_k\}$ that results in a steady state condition of our data preservation game. We refer to it

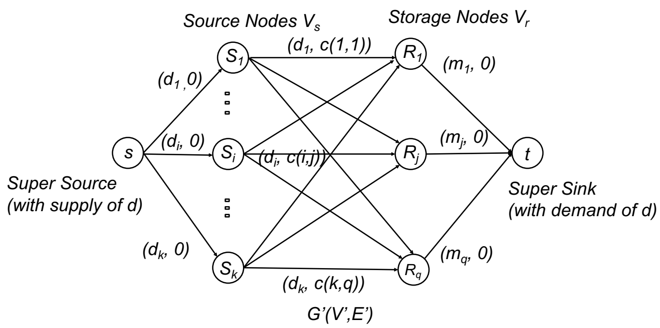

Figure 2. DPP in BSN graph $G(V, E)$ is equivalent to MCF problem in transformed flow network $G'(V', E')$. The first number in each parenthesis is the edge's capacity, and the second is its cost.

as a *data preservation NE*. In data preservation NE, each player knows the equilibrium strategies of the other players, and no player has any incentive to deviate from its chosen strategy (otherwise, its utility will be decreased); i.e., $u_i(s_i, s_{-i}) \geq u_i(s_i^*, s_{-i})$ for all $s_i^* \in S_i$.

Price of Anarchy (PoA) and Price of Stability (PoS). Both PoA and PoS are concepts in economics and game theory that measure how the efficiency of a system degrades due to the selfish behavior of its players. Given any BSN instance, its PoA is the ratio between the largest possible total preservation cost in an NE (i.e., the worst equilibrium) and the minimum total preservation cost in an optimal DPP solution, while PoS measures the ratio between the best equilibrium and the optimal centralized solution.

4.2. Data Preservation Games.

In this subsection, we present three data preservation games, viz. minimum cost flow (MCF) Game, Greedy Game, and Random Game. We analyze their NEs together with their price of anarchy and the price of stability.

MCF Game. The MCF Game has two steps.

Step 1: Transforming a BSN to a Flow Network. We first convert the BSN graph $G(V, E)$ to a flow network $G'(V', E')$ in Fig. 2 following below four steps.

First, we construct the nodes in G' as $V' = \{s\} \cup \{t\} \cup V_s \cup V_r$, where s is the super source node and t of the super sink node in the flow network. V_s is the set of source nodes, and V_r is the set of storage nodes.

Second, we construct the edges in G' as $E' = \{(s, S_i)\} \cup \{(S_i, R_j)\}\} \cup \{(R_j, t)\}\}$, where $S_i \in V_s$ and $R_j \in V_r$. There is a complete bipartite graph between V_s and V_r.

Third, for each edge (s, S_i), set its capacity as d_i, the number of data packets S_i has, and cost as 0. For each edge (R_j, t), set its capacity as m_j, the storage capacity of storage node R_j, and its cost as 0. For each edge (S_i, R_j), set its capacity as d_i and cost as $c(i, j)$. Here, $c(i, j)$ is the energy cost of offloading one data packet from node S_i to R_j along their shortest path.

Finally, we set the supply at s and demand at t as d, the total number of data packets in the BSN.

Step 2: Applying MCF-Algorithm on the Flow Network. We then apply MCF algorithms to the above flow network. MCF can be solved efficiently by many combinatorial algorithms [20]. We adopt the implementation by Goldberg [21], which is a scaling push-relabel algorithm with the highest performance among all the algorithms. It has the time complexity of $O(l^2 \cdot m \cdot \log(l \cdot n))$, where l, m, and n are the number of nodes, number of edges, and maximum edge capacity of $G'(V', E')$.

Theorem 1: The MCF game gives optimal total preservation cost, thus its $PoS = 1$; it also reaches NE.

Proof: Its optimality has been proved in our previous work [22], which shows that DPP in BSN graph $G(V, E)$ is equivalent to MCF in flow network $G'(V', E')$. Given an MCF-based data preservation solution, if a source node has the incentive to switch its data preservation path for any of its data packets, it must be the preservation cost of the new path will be smaller than that of the previous path, resulting in a smaller total preservation cost. This contradicts the optimality of the MCF. ∎

Greedy Game. The Greedy Game viz. Algo. 1 below is a more time-efficient greedy algorithm that applies directly to the BSN graph. Each source node i offloads its d_i data packets to its closest storage nodes with available spaces until all the data packets in the BSN are offloaded. Finding the shortest storage node for any source node takes $O(|V|^2 \cdot \log(|V|))$. Therefore, the time complexity of Algo. 1 is $O(k \cdot V|^2 \cdot \log(|V|))$.

Algorithm 1: The Greedy Data Preservation Game.
Input: A BSN graph $G(V, E)$;
Output: Data preservation paths $f : \mathcal{D} \rightarrow V_r$;
for $(1 \leq i \leq k)$ // current data packets at S_i
  $l_i = d_i$;
for $(1 \leq j \leq q)$ // current storage space at R_j
  $h_j = m_j$;
for $(1 \leq i \leq k)$ // each source node S_i
  while $(l_i > 0)$
    Find the storage node in V_r closest to S_i that still has available spaces, say R_j;
    Offload $\min(l_i, h_j)$ packets to R_j along the the shortest path between S_i and R_j;
    $l_i = l_i - \min(l_i, h_j)$, $h_j = h_j - \min(l_i, h_j)$;
  end while;
end for;
RETURN $f : \mathcal{D} \rightarrow V_r$.

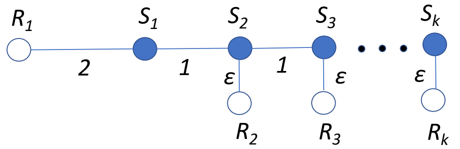

Figure 3. PoA of Greedy Game is $O(k)$.

Theorem 2: Greedy Game reaches NE. When $d_i = m_j = 1$, its PoA is $O(k)$.

Proof: Given a greedy game solution, if source node S_i has the incentive to switch its data preservation path for any of its data packets while other source nodes do not switch their strategies, the preservation cost of the new path must be smaller than that of the previous path. This contradicts the execution of the Greedy Game, wherein when its S_i's turn to offload its data packets, it will offload to the storage nodes with the smallest cost.

We use the example in Fig. 3 to illustrate that the PoA of the Greedy Game could be as high as $O(k)$. There are k source nodes S_1, S_2, ..., S_k and k storage nodes R_1, R_2, ..., R_k with $d_i = m_j = 1$. The preservation cost of one packet is shown on each edge where $\epsilon \ll 1$. Greedy Game will offload S_1's one packet to R_2, S_2's to R_3, ...,. and S_k's to R_1, resulting in total preservation cost of $(k-1)\cdot(1+\epsilon)+2$. The optimal MCF algorithm will offload S_1's one packet to R_1, S_2's to R_2, ...,. and S_k's to R_k, resulting in total cost of $2+(k-1)\cdot\epsilon$. ∎

Random Game. We also design a Random Game for comparison purposes. In the Random Game, each source node offloads its data packets to randomly chosen storage nodes with available spaces. The time complexity of the Random Game is $O(k \cdot |V|)$. Unlike MCF and Greedy, Random does not always achieve NE. Below is the sufficient condition for Random Game to achieve NE.

Theorem 3: When the total number of data packets equals the total storage spaces available in the BSN, the Random Game always reaches NE.

Proof: The BSN will have no free spaces once the source nodes offload all of their data packets. Given such a data preservation solution, for any source node, there is no other storage node available to switch to. Thus it does not have an incentive to switch, resulting in an NE. ∎

5. Simulations

We write our own simulator in Java on a Windows 10 machine with an Intel Processor (Intel Core i7-10750H) and 32GB of memory. We randomly place 150 sensor nodes in a $2000m \times 2000m$ sensor field. The transmission range of the sensor nodes is set as 200m, which means there exists an edge between any two sensor nodes if their distance is within 200m. Unless otherwise mentioned, the number of data packets at each source node is 100, and the storage capacity of each storage node is 100. We denote the PoA of Greedy Game and Random Game as PoA$_G$ and PoA$_R$, respectively. Given any instance of BSN, its PoA$_R$ is the ratio of the total preservation costs between Greedy Game and MCF Game, and PoA$_G$ is the ratio of the total preservation costs between Random Game and MCF Game. Each data point averages 20 simulation runs, and the error bars indicate 95% confidence intervals.

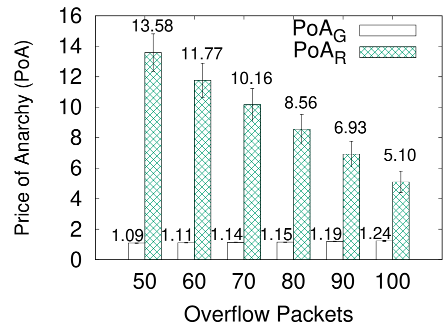

Figure 4. Varying Number of Data Packets.

Effects of Varying Data Packets d_i. Fig. 4 shows the values of PoA$_G$ and PoA$_R$ by varying the number of data packets d_i from 50, 60, ..., to 100 while fixing the storage capacity m_j as 100. First, we observe that PoA$_R$ is much larger than PoA$_G$ in all the cases. This show that Random Game has a much larger efficiency loss than Greedy does, as it does not attempt to save the preservation cost when offloading data packets. On the other hand, the largest PoA$_G$ is 1.24, showing the efficiency loss of the Greedy Game is at most 24% of the optimal preservation cost. Second, it shows that when increasing d_i, the PoA$_G$ increases gradually. Although the total preservation costs of both Greedy Game and MCF Game increase, Greedy increases more than MCF does, resulting in increased PoA$_G$. Third, we observe that PoA$_R$ decreases dramatically when increasing d_i, showing that Random Game performs much better in more "crowded" scenarios. This is because when there are more data packets to be offloaded, the randomness' negative effect of getting large preservation costs will be gradually eliminated.

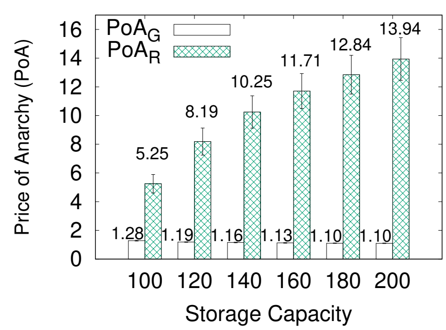

Figure 5. Varying Storage Capacity of Storage Nodes.

Effects of Varying Storage Capacity m_j. We then vary the storage capacity m_j of the storage node from 100, 120, ..., to 200 while fixing d_i as 100. Fig. 5 again shows the big difference between PoA_G and PoA_R. However, unlike Fig. 4, Fig. 5 shows that with the increase of the m_j, PoA_R increases evidently. This is because, with the increase of the m_j, the preservation cost of the MCF Game decreases while the preservation cost of the Random Game increases, resulting in a larger PoA_R. It also shows that with the increase of the m_j, PoA_G decreases slightly. As the preservation costs of both MCF and Greedy decrease with the increase of the m_j, the only explanation is that Greedy has more space to reduce its preservation cost than MCF.

6. Conclusion and Future Work

We studied the Nash Equilibrium (NE) for data preservation in emerging base station-less sensor networks (BSNs), wherein the resource-constrained sensor nodes could behave selfishly only to conserve their own resources and maximize their own benefit. We design a suite of data preservation games to achieve NE while minimizing energy consumption in the data preservation process. We used the price of anarchy and stability to analyze the efficiency loss in the NEs. As future work, we will design a data preservation game considering different data packets have different values remains a challenge.

Acknowledgment

This work was supported by NSF Grant CNS-2131309. We thank Chris Gonzalez for helping with the plots.

References

[1] Tang B. dao^2: Overcoming overall storage overflow in intermittently connected sensor networks. In Proc. of IEEE INFOCOM. 2018; .

[2] Hsu S, Yu Y, Tang B. dre^2: Achieving data resilience in wireless sensor networks: A quadratic programming approach. In Proc. of IEEE MASS. 2020; .

[3] Jang J, Adib F. Underwater backscatter networking. In Proc. of the ACM SIGCOMM. 2019; .

[4] Ghaffarivardavagh R, Afzal SS, Rodriguez O, Adib F. Ultra-wideband underwater backscatter via piezoelectric metamaterials. In Proc. of the ACM SIGCOMM. 2020; .

[5] et al. LT. From sensor to cloud: An iot network of radon outdoor probes to monitor active volcanoes. Sensors 2020; 20(10).

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

[7] Basagni S, Boloni L, Gjanci P, Petrioli C, Phillips CA, Turgut D. Maximizing the value of sensed information in underwater wireless sensor networks via an autonomous underwater vehicle. In Proc. of INFOCOM. 2014; .

[8] Wichmann A, Korkmaz T, Tosun AS. Robot control strategies for task allocation with connectivity constraints in wireless sensor and robot networks. IEEE Transactions on Mobile Computing 2018;17(6):1429–1441.

[9] Huang H, Savkin AV, Ding M, Huang C. Mobile robots in wireless sensor networks: A survey on tasks. Computer Networks 2019;148:1–19.

[10] Elhoseny M, Shankar K, Abdel-Basset M. Artificial Intelligence Techniques in IoT Sensor Networks. Chapman and Hall/CRC, 2020.

[11] Chen Z, Qiu Y, Liu J, Xu L. Incentive mechanism for selfish nodes in wireless sensor networks based on evolutionary game. Computers Mathematics with Applications 2011; 62(9):3378–3388.

[12] AlSkaif T, Zapata MG, Bellalta B. Game theory for energy efficiency in wireless sensor networks: Latest trends. Journal of Network and Computer Applications 2015;54:33 – 61.

[13] Nash J. Equilibrium points in n-person games. Proceedings of the National Academy of Sciences 1950;36(1):48–49.

[14] Nash J. Non-cooperative games. The Annals of Mathematics 1951;54(2):286–295.

[15] Voulkidis AC, Anastasopoulos MP, Cottis PG. Energy efficiency in wireless sensor networks: A game-theoretic approach based on coalition formation. ACM Trans Sen Netw July 2013;9(4).

[16] Niyato D, Hossain E, Rashid MM, Bhargava VK. Wireless sensor networks with energy harvesting technologies: a game-theoretic approach to optimal energy management. IEEE Wireless Communications 2007;14(4):90–96.

[17] Chen Y, Tang B. Data preservation in base station-less sensor networks: A game theoretic approach. In Proc. of the 6th EAI International Conference on Game Theory for Networks (GameNets 2016). 2016; .

[18] Chen Y, Ly J, Tang B. Voluntary data preservation mechanism in base station-less sensor networks. In 12th EAI International Conference on Game Theory for Networks (GameNets 2022); .

[19] Heinzelman W, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks. In Proc. of HICSS 2000; .

[20] Ahuja RK, Magnanti TL, Orlin JB. Network Flows: Theory, Algorithms, and Applications. Prentice-Hall, Inc., 1993.

[21] Goldberg AV. An efficient implementation of a scaling minimum-cost flow algorithm. Journal of Algorithms 1997; 22(1):1–29.

[22] Tang B, Jaggi N, Wu H, Kurkal R. Energy-efficient data redistribution in sensor networks. ACM Trans Sen Netw April 2013;9(2):11:1–11:28.

[23] Tauman Y. The aumann-shapley prices: A survey. A Roth Ed The Shapley Value Essays in Honor of Lloyd S Shapley Cambridge Cambridge University Press 1988;279–304.