Harmony or Involution: Game Inspiring Age-of-Information Optimization for Edge Data Gathering in Internet of Things

XIAOYAN YIN, School of Information Science and Technology, Northwest University, State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications), and Shaanxi International Joint Research Centre for the Battery-Free Internet of Things, China
XIAOQIAN MI, China Unicom Software Research Institute, China
SIJIA YU, Computer Engineering, State University of New York at Stony Brook, NY, USA
YANJIAO CHEN, College of Electrical Engineering, Zhejiang University, China
BAOCHUN LI, Department of Electrical and Computer Engineering, University of Toronto, Canada

Age-of-Information (AoI) has been recently reckoned as a suitable parameter to evaluate the freshness of collected information, which is essential for data retrieval in Internet of Things, especially the monitoring tasks, e.g., the operating situation of equipments. To motivate a large number of sensor nodes and solicit more up-to-date information from these nodes, the control center usually allocates rewards to nodes according to their proportional contributions. This induces intense competitions among nodes who try to gain high payoffs by carefully balancing the rewards and the costs. In this article, we propose a novel stochastic game model to formulate the competition among sensor nodes, which considers AoI as a metric used by the control center to quantify the contributions of nodes. We also take into account the uncertainty of channel quality, which affects the transmission success ratio of packets generated by nodes. Finally, we design an ϵ-Nash learning algorithm, which adopts the θ-greedy exploration strategy, to derive the ϵ-approximate Nash equilibrium such that nodes can maximize their long-term payoffs. Our substantive simulation results and analysis verify that the proposed algorithm outperforms baseline algorithms in bringing higher payoffs to nodes and more fresh information to the control center.

CCS Concepts: • **Networks** → *Packet scheduling*; • **Theory of computation** → *Algorithmic game theory;*

Additional Key Words and Phrases: Age-of-Information, game theory, learning, Internet of Things

This research is sponsored by the National Key Research and Development Program of China under Grant No. 2018YFB1802401, the National Natural Science Foundation of China under Grant No. 61872295, the Shaanxi Natural Science Foundation under Grant No. 2020JM-416, and Open Foundation of State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) under Grant No. SKLNST-2020-2-03.
Authors' addresses: X. Yin, School of Information Science and Technology, Northwest University, State key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications), and Shaanxi International Joint Research Centre for the Battery-Free Internet of Things, China; email: yinxy@nwu.edu.cn; X. Mi, China Unicom Software Research Institute, Xi'an, China; email: mixq5@chinaunicom.cn; S. Yu, Computer Engineering, State University of New York at Stony Brook, New York, USA; email: 220019168@link.cuhk.edu.cn; Y. Chen (corresponding author), College of Electrical Engineering, Zhejiang University, Hangzhou, China; email: chenyanjiao@zju.edu.cn; B. Li, Department of Electrical and Computer Engineering, University of Toronto, Toronto, Canada; email: bli@ece.toronto.edu.

1550-4859/2023/02-ART46 $15.00
https://doi.org/10.1145/3565022

ACM Reference format:
Xiaoyan Yin, Xiaoqian Mi, Sijia Yu, Yanjiao Chen, and Baochun Li. 2023. Harmony or Involution: Game Inspiring Age-of-Information Optimization for Edge Data Gathering in Internet of Things. *ACM Trans. Sen. Netw.* 19, 2, Article 46 (February 2023), 23 pages.
https://doi.org/10.1145/3565022

1 INTRODUCTION

Fuelled by wide penetration and ubiquitous connectivity of portable devices and **Internet of Things (IoTs)**, the demand is ever increasing for real-time information, such as equipment supervision and traffic condition monitoring. In most cases, out-of-date information will be of little or no use. More specifically, for equipment supervision, equipment faults will be accurately predicted if the control center can timely obtain real-time equipment operation status, which can be accessed by sensors deployed on the monitored equipment to effectively avoid industrial safety accidents. To characterize the freshness of information, the metric **Age-of-Information (AoI)** has been proposed to measure the time elapsed after the packet is generated [9]. A lower AoI indicates that the information is more fresh.

There have been extensive efforts on studying the technical issues of AoI management. For instance, scheduling policies have been proposed to minimize AoI in a single-hop wireless network [8], and a zero-wait policy for AoI management has been designed in Reference [20]. However, generating and transmitting information consumes valuable resources. Therefore, control centers need to deploy certain incentive mechanisms to compensate sensor nodes and encourage them to contribute more up-to-date information. Such economic issues regarding AoI were first introduced in Reference [5], which formulated the competition between two platforms as a non-cooperative game. Different from Reference [5], we are interested in the competition among nodes who are trying to gain desirable profits from the center. Generally, the center allocates a fixed pool of rewards to nodes based on their proportional contributions to the specific tasks. This indicates that the reward obtained by a node not only depends on her own contribution but also relies on the contributions of other nodes, which may induce intense competitions among nodes. To make the problem more complicated, not all data generated by a node can be delivered successfully to the center due to uncertainty in wireless channel quality. With a poor channel quality, even if a node generates a large number of packets, few of them can reach the center, thus the payoff of the node is affected considering the cost of packet generation and transmission.

In this article, we investigate the dynamic and strategic competition between sensor nodes for data collection in IoTs by taking AoI management of data into account. Our target problem is to derive the optimal strategies for nodes to achieve expected payoffs, which imply the optimal allocation of system resources by trading off between the freshness of information and the cost based on the stochastic game from the perspectives of nodes, not only a monetary reward. Since each node's payoff depends on the joint actions of nodes in each time slot, we make the first attempt to utilize the general-sum game to formulate the competition among nodes for data gathering in IoTs. Furthermore, due to the uncertainty in channel quality, the system state of the game keeps changing over time, inspiring by nature of game theory; thus, we model the long-term dynamic packet generation in a stochastic setting. At each time stage, a node determines the packet generation rate according to the current observed channel quality and the average AoI of its packets. The generated packets will be transmitted to the control center but the fraction of successfully delivered packets are affected by the channel quality, i.e., more packets can reach the platform if the channel quality is high. With the newly generated packets, the platform will update the AoI of nodes. More specifically, if more packets are received in the current time slot, then the AoI value of

the node will be reduced by a larger amount, meaning that the information provided by the node is more fresh. The platform then distributes a fixed amount of reward to nodes proportional to their contributions calculated according to the number of received packets and AoI of their contents. To achieve the best strategy for involved nodes in the stochastic game, we design a learning algorithm leading to ϵ-Nash Equilibrium, which adopts the θ-greedy strategy, to achieve the optimal strategies for nodes that can maximize their long-term payoffs. Experimental results show that our proposed approach achieves superior performance over baseline algorithms.

In this article, we make the following key contributions.

- We present the first attempt to adopt a stochastic game model to analyze the AoI for the control center in IoTs, which captures the uncertainty of the environment and the dynamic interactions between nodes.
- We propose a novel incentive mechanism that considers both the quantity and quality (i.e., AoI) of information provided by nodes to the control center. We design an efficient ϵ-Nash learning algorithm, which adopts the θ-greedy exploration strategy, to achieve the optimal strategies for nodes, and then the nodes can learn and adjust their actions based on optional strategies and revenue expectations.
- We evaluate the proposed algorithm to verify its superiority over random, fixed, and myopically learned strategies in terms of node payoff and information freshness. We also conduct extensive validation about the impact of ϵ on convergence speed and node payoff.

The remainder of the article is organized as follows. Section 2 reviews related work. In Section 3, we describe our system model for data collection in IoTs. In Section 4, we formulate the competition among nodes based on the stochastic game. We present a learning algorithm, which utilizes the θ-greedy learning strategy and can lead to the ϵ-Nash Equilibrium, to achieve the best strategies for nodes in Section 5. The evaluation results and performance analysis of our scheme are demonstrated in Section 6. The limitation and guiding principles are discussed in Section 7. Finally, Section 8 concludes the article.

2 RELATED WORK

Age-of-information. AoI has been introduced as a new metric to evaluate the freshness of information, which measures the duration between the moment that the content is generated and the time the content is received. There has been a lot of research focusing on the technological issues of minimizing AoI. The existence of an optimal packet generation rate has been proved, which allows a source to keep its status as timely as possible [9]. To better control information updates, a general age penalty function has been introduced to characterize the staleness of data and an optimal update policy has been proposed for minimizing the average age penalty [20]. A method of minimizing AoI has been proposed that jointly considers multiple properties, including sampling behavior, sample size, and the transmission capacity [10]. To optimize AoI without throughput loss, a preemptive **Last-generated First-served (LGFS)** strategy has been proposed with multiple servers under an arbitrary task arrival rate [1]. The economics of AoI management has been investigated [5, 18, 21]. A non-pricing mechanism has been proposed to enforce cooperation among selfish platforms to balance their AoIs and sampling costs [5].

To encourage nodes to sample information at different rates over time, a dynamic pricing strategy has been designed for the provider to offer age-dependent monetary returns [18]. Two pricing schemes, i.e., optimal time-dependent and optimal quantity-based pricing, have been designed by modeling the interactions between information sources and nodes as a Stackelberg game [21]. However, there is a lack of works that study AoI from the perspectives of nodes. In particular, it is unexplored how nodes manage AoI to obtain optimal rewards from the control center. In this

article, we address this problem by considering the competition among nodes who carefully adjust their packet generation rate to attain the desirable AoI and reward.

Stochastic Game. On the basis of Markov Decision Process, the stochastic game takes interaction and competition among different players into consideration [13, 15]. Recently, a stochastic game-based model can characterize the interactions between multiple agents in a time-varying environment with uncertainties. A stochastic game framework has been designed for anti-jamming defence to improve the spectrum utilization in cognitive radio networks [16]. To protect user privacy, the interaction between the user and its competitor is modeled as a zero-sum stochastic game to find the best strategy in Reference [17]. The video rate adaptation problem can also be modeled based on the stochastic game to improve viewers' **Quality-of-Experience (QoE)** in Reference [2]. To formulate the interactive competition between Bitcoin mining pools, a general-sum stochastic game-based model has been developed in Reference [19].

Moreover, numerous works have leveraged the learning algorithms to find the optimal strategies for agents in the stochastic game. The minimax-Q learning algorithm has been proposed in Reference [12] for zero-sum Markov game. Based on the work of Reference [12], a multi-agent Q-learning method has been proposed for general-sum stochastic games in Reference [6]. Furthermore, Q-learning has been extended to a noncooperative multi-agent setting in general-sum stochastic games, where the convergence under different conditions has been studied [7]. Recently, two-timescale algorithms have been proposed, which can find stationary Nash equilibria in a general-sum stochastic game [14]. Different learning approaches have been presented for five variants of multi-agent inverse reinforcement learning in general-sum stochastic games [11]. In this article, we formulate the competition among nodes as a stochastic game, and adopt learning algorithms to find the optimal strategy for each node.

3 SYSTEM MODEL

We consider two sensor nodes u_1 and u_2 competing with each other for data gathering in an IoT, as shown in Figure 1. Each node samples information from the environment and attempts to transmit the newly sampled packets to the control center. We consider a time-slot-based dynamic time horizon. Each node generates multiple packets periodically in each time slot, and each packet contains certain information, including the time of generation, the order of generation, and the total number of generated packets in the current time slot. We use $f_i(n)$ and $q_i(n)$ to denote the average AoI and channel quality of node u_i at time slot n, respectively. The average AoI reflects the timeliness of the information provided by the node to the center in each time slot, and the channel quality determines the packet transmission success ratio, which will affect AoI in the next time slot.

Upon receiving the packets from the two nodes, the control center will divide up a fixed amount of reward among the two nodes, proportional to their contributions to the specific tasks, i.e., the number of delivered packets and the AoI of the contents. More specifically, nodes will receive the same amount of rewards provided that they performed equally well/equally bad or proportional payment (i.e., nodes with poor performance will receive less rewards). Therefore, the payoff of a node at a certain time slot relies upon various factors, including the number of successfully transmitted packets, the AoI of the contents, and the cost of packet generation and transmission. A node has incentives to adjust the packet generation rate to manage the AoI to maximize its long-term payoff. Therefore, the modeling considerations related to AoI manage when finding the best strategies for nodes are as follows.

- *Packet generation rate.* A higher packet generation rate incurs a higher cost but produces more packets that are more likely to be successfully transmitted to refresh AoI, which may

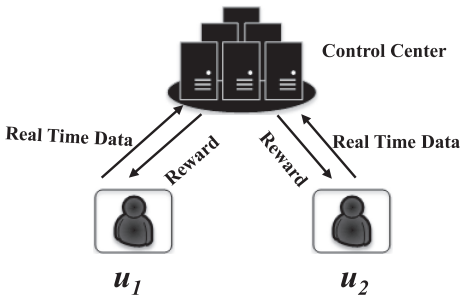


Fig. 1. System model. The nodes generate and send real-time data to the control center, then the control center assigns rewards to nodes based on their contributions.

yield a higher reward from the control center. Meanwhile, a node faces competition from the rival node who will share a fixed amount of reward from the control center. To gain the maximum payoff, the node has to find the optimal packet generation rate at each time slot by jointly considering its own cost and the channel quality, as well as possible actions of the rival node.

- *Channel quality.* The channel quality is determined by the environment where a node is in, and it will have an influence on the transmission success ratio. Given the packet generation rate chosen by the node, a larger fraction of the generated packets can be successfully delivered to the control center when the channel quality is better. In other words, if the channel quality is poor, even if the packet generation rate is high, then the payoff of the node may be low, since few packets can be received by the control center. Therefore, the node should take into account the current channel quality when deciding the packet generation rate. Note that the channel quality of two nodes are independent.

4 STOCHASTIC GAME FORMULATION

As an extension to the **Markov Decision Process (MDP)**, stochastic games [7] can capture the dynamic interactions among multiple agents. In this section, we analyze the interaction between two nodes and formulate the packet generation decision game as a two-agent stochastic game.

4.1 Game Formulation

We formulate the AoI management of two competing nodes in an IoT as a two-agent stochastic game, denoted as a six-tuple $\Gamma =< \mathbb{S}, \mathbb{A}_1, \mathbb{A}_2, \mathbf{r_1}, \mathbf{r_2}, \mathbf{P} >$, where (1) $\mathbb{S}$ is denoted as the state space, and the system state is characterized by the average AoI and the channel quality, $\mathbb{S} = \{s(1), \ldots, s(n)\}$, $s(n) = \{f_1(n), f_2(n), q_1(n), q_2(n)\}$, where $s(n)$ is the state at time slot n, $f_i(n)$ is the average AoI of node u_i, $i \in \{1, 2\}$, at time slot n, and $q_i(n)$ is the channel quality of node u_i, $i \in \{1, 2\}$, at time slot n, (2) $\mathbb{A}_\mathbf{i}, \mathbf{i} \in \{\mathbf{1}, \mathbf{2}\}$ is denoted as the action space that contains all possible actions that node u_i can take, (3) $\mathbf{r_i} : \mathbb{S} \times \mathbb{A}_\mathbf{1} \times \mathbb{A}_\mathbf{2} \mapsto \mathbf{r_i}$ is the payoff for u_i, where $\mathbf{r_i} \in \mathbb{R}, \mathbb{R}$ is denoted as the set of real numbers, (4) $\mathbf{P} : \mathbb{S} \times \mathbb{A}_\mathbf{1} \times \mathbb{A}_\mathbf{2} \mapsto \Delta(\mathbb{S})$ is described as the transition probability function of $\mathbb{S} = \{f_1(1), f_2(1), q_1(1), q_2(1), \ldots, f_1(n), f_2(n), q_1(n), q_2(n)\}$, and actions, i.e., $\mathbb{A}_1$, $\mathbb{A}_2$ taken by u_1 and u_2. The game Γ is executed stage by stage. At each time stage, node u_i chooses an action based on the current state s, $s \in \mathbb{S}$, then receives a payoff determined by the joint action of two players and the current state s, $s \in \mathbb{S}$. Each node attempts to maximize the expected sum of payoffs with the time discount effect.

State space $\mathbb{S}$. The state space is defined as the average AoI and the channel quality of u_1 and u_2, i.e., the state at time slot n is $s(n) = \{f_1(n), f_2(n), q_1(n), q_2(n)\}$. The channel quality is

measured by channel state information in wireless communications, and represented generally by **Channel Quality Indication (CQI)**. The AoI of a node is computed as the average duration from the moment that packets are generated to the time when they have been received. As a node adjusts the packet generation rate at different time slots, its AoI will be updated over time. The channel quality affects packet transmission and will determine the number of packets that can be successfully delivered. The control center offers reward to the node depending on the number of received packets and the AoI of the contents, i.e., the node's payoff is decided by channel quality and AoI. Therefore, a node needs to adapt its packet generation rate based on the channel quality and the current AoI of both nodes. The channel quality of the rival node can be observed by channel estimation, and the AoI of the rival node can be deduced from the reward distribution.

Actions $\mathbb{A}_1, \mathbb{A}_2$. The action of node u_i is defined as the packet generation rate chosen by node u_i, $i \in \{1, 2\}$, i.e., the actions taken by u_1 and u_2 in time slot n are $a(n) = \{a_1(n), a_2(n)\}$, where $a_i(n)$, $i \in \{1, 2\}$ is the number of packets node u_i generates in time slot n. Each node decides its action for the current time slot based on the current state $s(n) = \{f_1(n), f_2(n), q_1(n), q_2(n)\}$. With packet generation rate $a_i(n)$, we assume that node u_i can generate a total number of $J_i(n) = a_i(n) \times \Delta t$ packets, where Δt is the duration of a time slot. Each generated packet j is stamped with the time of generation, the order of generation, and the aggregate number of packets generated in this time slot, denoted as $\{\tau_{i,j}, j, J_i(n)\}$.

Stage payoff r_1, r_2. The payoff at stage n depends on $s(n)$ and $a(n)$ of u_1 as well as u_2, where $s(n)$ is the current state, and $a(n)$ is actions taken by two nodes. To be more specific, the payoff of node u_i is determined by its average AoI, the number of packets transmitted successfully, and the cost of packet generation and transmission. The payoff of u_1 in time slot n can be computed as

$$r_1[s(n), a(n)] = PoC_1[s(n), a(n)] \times W(n) - c_1 a_1(n), \tag{1}$$

where $PoC_1[s(n), a(n)]$ is the proportion of u_1's contribution to the platform, determined by the current state and actions of both nodes. $W(n)$ is the total reward offered by the center in time slot n. c_1 is the unit cost of packet generation and transmission of node u_1.

To enhance the freshness and richness of information, the control center distributes rewards based on the average AoI and the number of received packets of nodes in each time slot. Thus, we have

$$PoC_1[s(n), a(n)] = \frac{\psi_1\big(s(n), a(n)\big)}{\psi_1\big(s(n), a(n)\big) + \psi_2\big(s(n), a(n)\big)}, \tag{2}$$

where $\psi_i(s(n), a(n)), i \in \{1, 2\}$ is the incentive mechanism adopted by the control center, which considers both the quality and quantity of information contributed by node u_i, and aims to motivate nodes to provide more information to refresh the AoI. Therefore, we define the reward function as

$$\psi_i\big(s(n), a(n)\big) = \frac{m_i(n)}{f_i(n)}, \tag{3}$$

where $m_i(n)$ is the number of packets that transmitted successfully by node u_i given the packet generation rate $a_i(n)$ and the channel quality $q_i(n)$, and $f_i(n)$ is the average AoI of node u_i, $i \in \{1, 2\}$, in time slot n.

In the same way, $r_2[s(n), a(n)]$ can be calculated as

$$r_2[s(n), a(n)] = PoC_2[s(n), a(n)] \times W(n) - c_2 a_2(n), \tag{4}$$

where $PoC_2[s(n), a(n)]$ is

$$PoC_2[s(n), a(n)] = \frac{\psi_2\big(s(n), a(n)\big)}{\psi_1\big(s(n), a(n)\big) + \psi_2\big(s(n), a(n)\big)}. \tag{5}$$

State transition probability. The state transition includes the transition of channel quality and the transition of AoI of nodes, which are independent from each other. Furthermore, the AoI transition of node u_i, $i \in \{1, 2\}$, only depends on the current AoI and the action of u_i, $i \in \{1, 2\}$, the channel quality follows certain random distribution that is independent of node actions. Therefore, we can decouple the state transition function as

$$\begin{aligned} &P[s(n+1)|s(n), a(n)] = P[f_1(n+1)|f_1(n), a_1(n)] \\ &\times P[f_2(n+1)|f_2(n), a_2(n)] \times P[q_1(n+1)] \times P[q_2(n+1)], \end{aligned} \tag{6}$$

where $P[q_1(n+1)]$ and $P[q_2(n+1)]$ can be estimated according to radio propagation models, evaluation and transition of channel quality are outside the scope of the study. We explain how to derive $P[f_1(n+1)|f_1(n), a_1(n)]$ and $P[f_2(n+1)|f_2(n), a_2(n)]$ in the following context.

Given the channel quality $q_i(n)$ in time slot n, the probability that packets of node u_i can be transmitted successfully to the control center is $p_i(n) \in [0, 1]$, which follows a Gaussian distribution $p_i \sim N(1, \frac{1}{2q_i})$ [4]. If at least one packet is transmitted successfully, then the platform can derive the AoI of each packet j of u_i as follows:

$$\Delta_{i,j}(n) = \begin{cases} \alpha_{i,j} - \tau_{i,j}, & \text{if packet } j \text{ is received,} \\ \Delta_{i,l}(n-1) + 1, & \text{otherwise,} \end{cases} \tag{7}$$

where $\alpha_{i,j}$ is the time that packet j successfully arrives at the control center, $\tau_{i,j}$ is the generation time of packet j, and $\Delta_{i,l}(n-1)$ is the AoI of the last received packet l in the previous time slot $n-1$. The control center can infer that packet j is lost based on the information carried by successfully transmitted packets. The AoI of a lost packet is set as the AoI of the last received packet l in the previous time slot plus 1.

The average AoI considering all packets generated by node u_i in time slot n is

$$\Delta_i(n) = \begin{cases} \frac{\sum_{j=1}^{J_i(n)} \Delta_{i,j}(n)}{J_i(n)}, & \text{with probability } 1 - \mathcal{A}, \\ \Delta_{i,l}(n-1) + 1, & \text{with probability } \mathcal{A}, \end{cases}$$

where $\mathcal{A} = [1 - p_i(n)]^{J_i(n)}$, $p_i(n)$ is the transmission success probability of node u_i, and $J_i(n)$ is the total number of generated packets of node u_i in time slot n. If at least one packet is successfully delivered to the control center (the probability is $1 - [1 - p_i(n)]^{J_i(n)}$), then the average AoI of node u_i will be updated according to the information of the received packets. If no packet arrives at the control center (the probability is $[1 - p_i(n)]^{J_i(n)}$), then the average AoI of node u_i will be updated as the AoI of the last received packet in the previous time slot plus 1.

To better understand the update of AoI, we provide a toy example to illustrate how the platform calculates the AoI of nodes. Suppose that node u_i takes action after observing the state by choosing packet generation rate of 4. We assume that $\Delta t = 1$ so that 4 packets are generated in the time slot, and each packet is stamped with the information for calculating AoI. Given different channel qualities, the transmission success probability is different.

- With a high channel quality, we assume that the transmission success probability is $p_i(n) = 1$, i.e., all packets can be received by control center. The information carried by the four packets is $(0, 1, 4)$, $(0.25, 2, 4)$, $(0.5, 3, 4)$, and $(0.75, 4, 4)$. We assume that the arrival time of the four packets is 0.25, 0.5, 0.75, and 1. Hence, the AoI of each packet is 0.25. and the average AoI of the node in this time slot is 0.25.
- With a moderate channel quality, we assume that the transmission success probability is $p_i(n) = 0.75$ and the second packet is lost. The center receives three packets $(0, 1, 4)$, $(0.5, 3, 4)$, and $(0.75, 4, 4)$ with the arrival time 0.25, 0.75, and 1. The AoI of each received

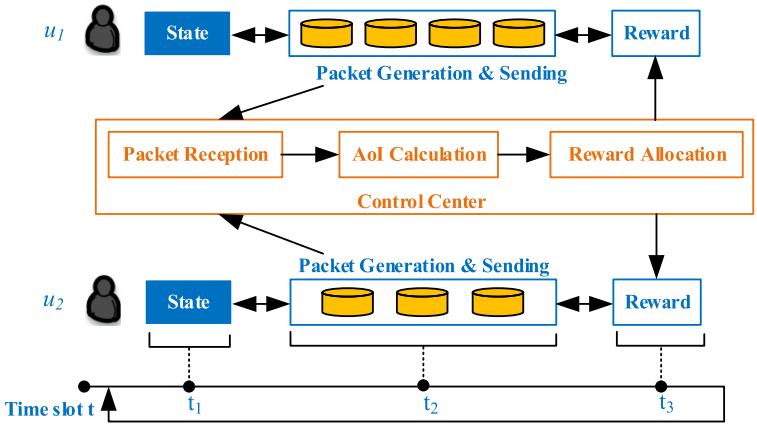


Fig. 2. Illustration of time slot n of the stochastic game.

packet is 0.25. The center can infer that packet 2 is lost, and calculate its AoI as $0.4 + 1 = 1.4$, where 0.4 is the AoI of the last received packet in the previous time slot. Thus, the AoI of the node in this time slot is 0.5375.

- With a poor channel quality, it may happen that no packet is transmitted successfully. The platform will calculate the AoI of the node as 1.4.

Based on AoI of node u_i in the current time slot, the average AoI of u_i is updated as

$$f_i(n+1) = \frac{(n-1)f_i(n) + \Delta_i(n)}{n}. \tag{8}$$

We can compute the transition probability of the AoI as

$$P[f_i(n+1)|f_i(n), a(n)] = \begin{cases} 1-[1-p_i(n)]^{J_i(n)}, & \text{if } f_i(n+1) = \frac{(n-1)f_i(n)+\frac{\sum_{j=1}^{J_i(n)}\Delta_{i,j}(n)}{J_i(n)}}{n}, \\ [1-p_i(n)]^{J_i(n)}, & \text{if } f_i(n+1) = \frac{(n-1)f_i(n)+\Delta_{i,1}(n-1)+1}{n}, \\ 0, & \text{otherwise.} \end{cases}$$

As illustrated in Figure 2, we divide each time slot into three phases to better explain the flow of events in the stochastic game. At the initial phase n_1 of time slot n, the nodes first observe the current state, i.e., the average AoI and channel quality $\{f_1(n), f_2(n), q_1(n), q_2(n)\}$, based on which they choose the packet generation rate $a_1(n)$ and $a_2(n)$ according to their learned strategies. At phase n_2, the two nodes generate packets and transmit the packets to the control center. At phase n_3, the center allocates the rewards to nodes according to their contributions considering the average AoI and the number of successfully transmitted packets.

As shown in Table 1, according to the proposed scheme, we illustrate the state transition over 10 rounds by using the given toy example. Here, we assume that the unit cost per packet is 0.02, and the transmission success ratio of the two nodes is fixed as $p_1 = 0.3, p_2 = 0.6$. We can observe that the payoff of nodes depend on the actions of both nodes, and the payoff of a certain time slot may be negative, since the rewards from the control center is lower than the cost of packet generation and transmission. Due to fierce competition, it is critical for nodes to learn the optimal strategy to obtain a high long-term payoff.

Table 1. Toy Example of the Stochastic Game

Time slot n	State $\{f_1(n), f_2(n), q_1(n), q_2(n)\}$	Action $\{a_1(n), a_2(n)\}$	Payoff $\{r_1(n), r_2(n)\}$	Next State $\{f_1(n+1), f_2(n+1), q_1(n+1), q_2(n+1)\}$	Average payoff $\{\bar{r}_1(n), \bar{r}_2(n)\}$
1	{0.200, 0.400, 0.3, 0.6}	{6, 5}	{0.325, 0.456}	{0.184, 0.300, 0.4, 0.6}	{0.325, 0.456}
2	{0.184, 0.300, 0.3, 0.6}	{4, 10}	{0.206, 0.515}	{0.206, 0.233, 0.4, 0.6}	{0.266, 0.485}
3	{0.206, 0.233, 0.3, 0.6}	{7, 2}	{0.610, 0.210}	{0.190, 0.300, 0.4, 0.6}	{0.381, 0.394}
4	{0.190, 0.300, 0.3, 0.6}	{5, 10}	{0.150, 0.550}	{0.192, 0.260, 0.4, 0.6}	{0.323, 0.433}
5	{0.192, 0.260, 0.3, 0.6}	{3, 9}	{−0.600, 0.820}	{0.327, 0.236, 0.4, 0.6}	{0.247, 0.510}
6	{0.327, 0.236, 0.3, 0.6}	{4, 2}	{0.520, 0.360}	{0.316, 0.273, 0.4, 0.6}	{0.292, 0.485}
7	{0.316, 0.273, 0.3, 0.6}	{7, 7}	{0.360, 0.360}	{0.294, 0.257, 0.4, 0.6}	{0.302, 0.468}
8	{0.294, 0.257, 0.3, 0.6}	{5, 1}	{0.700, 0.180}	{0.284, 0.329, 0.4, 0.6}	{0.352, 0.432}
9	{0.284, 0.329, 0.3, 0.6}	{9, 6}	{0.392, 0.309}	{0.267, 0.313, 0.4, 0.6}	{0.326, 0.418}
10	{0.267, 0.313, 0.3, 0.6}	{7, 1}	{0.8600, −0.02}	{0.256, 0.375, 0.4, 0.6}	{0.407, 0.374}

4.2 Problem Statement

We execute a stochastic game between the two nodes for an infinite number of time slots. Furthermore, we take the effect of time discount into consideration. In other words, payoffs in the current time slot should be worth more than those in future time slots. The total long-term utility of $u_i, i = 1, 2$ can be denoted as the expected sum of discounted payoffs,

$$U_i = \mathbb{E}\left\{\sum_{n=0}^{\infty} \gamma(n) r_i[s(n), a(n)]\right\}, \tag{9}$$

let γ denote the discount factor, $\gamma < 1$. Our objective for a node is to learn an optimal strategy to gain maximum expected long-term utility.

5 LEARNING THE OPTIMAL STRATEGY

In this section, we describe how to obtain the optimal strategy for sensor nodes in IoTs, based on the stochastic game formulated in the previous section.

5.1 Nash Equilibrium

For stochastic games, the strategy is a probability distribution over all possible actions for the set of states. In this article, we study the stationary strategy, which means that the strategy is independent of time, i.e., $\pi(n) = \pi$ for all n.

We express the strategy of node u_i as $\pi_i : \mathbb{S} \mapsto P(\mathbb{A}_i), i \in \{1, 2\}$, let $\mathbb{S}$ denote the state space, let $P(\mathbb{A}_i)$ denote the probability distribution over action space $\mathbb{A}_i$. Given strategies π_1, π_2 and the current state $s \in \mathbb{S}$, the value of the state s is $V_i^{\pi}(s)$ for $u_i, i \in \{1, 2\}$, can be computed as

$$\begin{aligned} V_i^{\pi}(s) &= \sum_{n=0}^{\infty} \gamma(t) \mathbb{E}\{r_i[s(n), a(n)] | \pi_1, \pi_2, s(0) = s\} \\ &= r_i(s, a_1^{\pi}, a_2^{\pi}) + \gamma \sum_{s' \neq s} Pr[s' | s, a_1^{\pi}, a_2^{\pi}] V_i^{\pi}(s'), \end{aligned} \tag{10}$$

where a_1^{π} and a_2^{π} are the actions of u_1 and u_2, which are decided by strategies π_1 and π_2, and $s(0)$ is the initial state in time slot 0.

Let $\pi^* = \{\pi_1^*, \pi_2^*\}$ denote the optimal strategies for u_1 and u_2. Due to the nature of the general-sum stochastic game, the interactions between u_1 and u_2 will converge to an optimal strategy pair, and achieve a Nash Equilibrium finally. A Nash equilibrium [7] is a collection of strategies where each node's strategy is a best-response to other players' strategies, defined as follow.

Definition 1 (Nash Equilibrium). The existence of Nash Equilibrium implies that every player has the best strategy. Furthermore, players have no motivation to change the current state when

ALGORITHM 1: ϵ-Nash Learning Algorithm

Input: $s(0) = \{f_1(0), f_2(0), q_1(0), q_2(0)\}$, and game Γ.
Output: π^{ϵ}.

$t \leftarrow 0$, $f_1(n) = f_1(0)$, $q_1(n) = q_1(0)$, $f_2(n) = f_2(0)$, $q_2(n) = q_2(0)$.
$V_1[s_1(n), s_2(n)] = 0$, $V_2[s_1(n), s_2(n)] = 0$, $\forall s_1, s_2 \in \mathbb{S}$.
Initialize $\pi(n) : a_1(n) = 5, a_2(n) = 5$.
Repeat
Compute θ using Equation (18).
Choose an action pair $a_1(n), a_2(n)$ based on $\pi(n)$ with probability $1 - \theta$.
Choose randomly an action pair $a_1(n), a_2(n)$ with probability θ.
Compute $f_1(n+1)$, $f_2(n+1)$ after u_1 and u_2 take their actions $a_1(n), a_2(n)$.
Compute optimal strategies $\pi_1(n+1)$, $\pi_2(n+1)$ for u_1 and u_2 using Equation (16).
Compute $V_1(n+1)$, $V_2(n+1)$ using Equation (17).
$n \leftarrow n + 1$.
until convergence.
Return π^{ϵ}.

the best strategy is chosen by every player. Given $\pi^* = \{\pi_1^*, \pi_2^*\}$ for two players, for any possible state $s \in \mathbb{S}$, we have

$$V_1^{\pi^*}(s) \geq V_1^{(\pi_1, \pi_2^*)}(s), \tag{11}$$

and

$$V_2^{\pi^*}(s) \geq V_2^{(\pi_1^*, \pi_2)}(s). \tag{12}$$

For a specific data collection task, the objective of node u_1 is to attain the optimal strategy π_1^* to maximize $V_1^{\pi}(s)$ for any possible state, while node u_2 has the same goal to maximize $V_2^{\pi}(s)$ for any possible state. However, a pure tactical Nash Equilibrium may not exist in this bimatrix game. We demonstrate a game playing process of 0.03-Nash Equilibrium, as shown in Figure 3.

The current state is $\{0.7, 0.8, 0.13, 0.13\}$, and the transmission success ratio is 0.13 for both nodes. As shown in Figure 3, the reward achieved by the two nodes when taking different actions, e.g., if node u_1 and node u_2 choose action $\{2, 3\}$, their rewards are 0.39 and 0.21, respectively. More specifically, given that node u_1 takes action 2, the dominant strategy of node u_2 is 5 that produces the highest reward 0.52. Then, u_1 chooses the new dominant strategy 3 and gains a reward of 0.45. After this, u_2 takes a new dominant strategy 3 with a reward of 0.45. In turn, u_1's new dominant strategy is 2. This search cycle goes on forever and there is no pure tactical Nash Equilibrium for the given game.

Accordingly, we define ϵ-Nash Equilibrium [3] as follows instead.

Definition 2 (ϵ-Nash Equilibrium). For stochastic game Γ, an **ϵ-Nash Equilibrium (ϵ-NE)** is an optimal strategy pair $\pi^{\epsilon} = \{\pi_1^{\epsilon}, \pi_2^{\epsilon}\}$, for all state s, $s \in \mathbb{S}$, we have

$$V_1^{\pi^{\epsilon}}(s) \geq V_1^{\pi^2}(s) - \epsilon, \tag{13}$$

and

$$V_2^{\pi^{\epsilon}}(s) \geq V_2^{\pi^1}(s) - \epsilon, \tag{14}$$

where $\pi^2 = (\pi_1, \pi_2^{\epsilon})$, $\forall \pi_1$, and $\pi^1 = (\pi_1^{\epsilon}, \pi_2)$, $\forall \pi_2$, and $\epsilon > 0$.

As shown in Figure 3, the bimatrix game converges to a Nash Equilibrium with $\epsilon = 0.03$ after three rounds of execution. When node u_2 selects action 5, node u_1 changes the action from 2 to 3 with a reward of 0.45. If node u_1 continues to choose action 2, then it will get a reward of 0.42. Since $0.42 = 0.45 - 0.03$, according to the definition of ϵ-Nash equilibrum, u_1 will not change its action. If no node can alter its action to attain a growth of reward that exceeds 0.03,

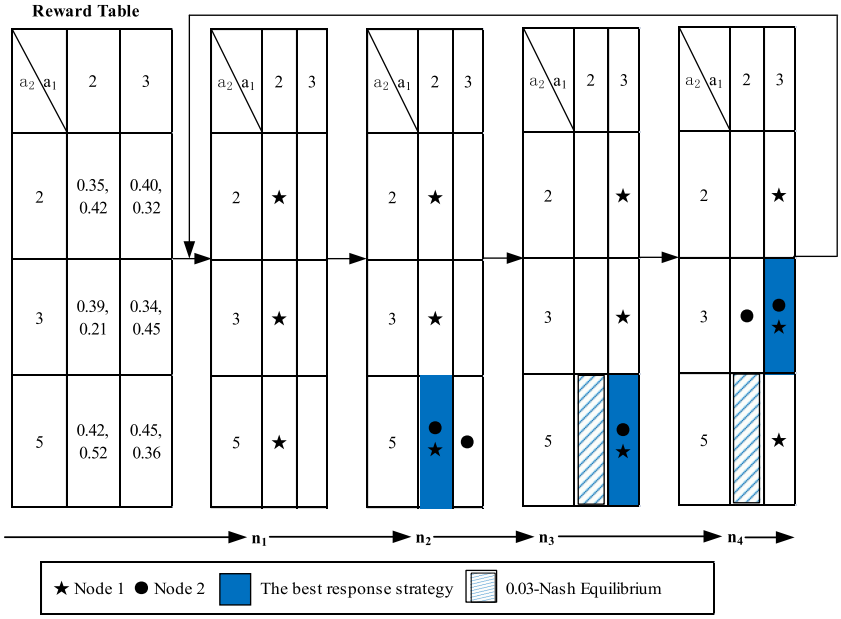


Fig. 3. Illustration of a bimatrix game. It can be observed there is no pure tactical Nash equilibrium for this game, but a 0.03-Nash equilibrium exists.

then the 0.03 Nash Equilibrium is achieved, and $\{a_1, a_2\} = \{2, 5\}$ is a 0.03-approximate Nash equilibrium.

We update the state value of u_i, $i \in \{1, 2\}$ as

$$V_i^{\pi^\epsilon}(s) = r_i(s, a_1^{\pi^\epsilon}, a_2^{\pi^\epsilon}) + \gamma \sum_{s' \neq s} Pr[s'|s, a_1^{\pi^\epsilon}, a_2^{\pi^\epsilon}] V_i^{\pi^\epsilon}(s'), \tag{15}$$

where ϵ is an arbitrarily small value, such that the reward obtain in the ϵ-Nash equilibrium can approach asymptotically that in the Nash equilibrium. We set ϵ as 0.001 in our algorithm.

The target problem is to learn the best strategy π_i^ϵ in $\pi^\epsilon = \{\pi_1^\epsilon, \pi_2^\epsilon\}$ for two nodes,

$$\pi_i^\epsilon = \arg\max_{\pi_i} r_i(s, a^{\pi^\epsilon}) + \gamma \sum_{s' \neq s} Pr[s'|s, a^{\pi^\epsilon}] V_i^{\pi^\epsilon}(s'). \tag{16}$$

5.2 Learning Algorithm for ϵ-Nash Equilibrium with θ-Greedy Exploration

To achieve the ϵ-Nash Equilibrium, we present a learning algorithm with θ-greedy exploration based on Q-learning. According to the update rules of Q-learning [6], we can obtain π^ϵ by learning $V_i^{\pi^\epsilon}$, $i = 1, 2$.

$$\begin{aligned} V_i[s(n+1)] = {} & [1 - \eta(n+1)] V_i[s(n)] \\ & + \eta(n+1) r_i[s, a_1(n+1), a_2(n+1)] + \gamma V_i[s'(n)], \end{aligned} \tag{17}$$

where let $\eta(n) \in [0, 1)$ denote the rate of learning, our algorithm will converge when $\eta(n)$ declines over time. In the meantime, $\eta(n)$ is set as $\frac{1}{n^{1/3}}$, and we use $V_i[s(n+1)]$ as the approximate reference of $V_i^{\pi^\epsilon}$. The state values of u_1 and u_2 are calculated iteratively based on Equation (17) until convergence.

During the learning process, each node has two choices: exploit the action chosen by the currently learned strategy or explore a new action. More specifically, in the exploitation process, the node chooses the optimal action determined by its strategy, and in the exploration process, the node randomly selects an action. We adopt a θ-greedy method to balance exploration and exploitation, i.e., there is a $1-\theta$ probability that node u_i will choose the action, which is determined based on the current strategy $\pi(n)$, as well as will explore a new action randomly with probability θ. While exploration has potential to find the optimal action, exploitation can achieve the best reward based on the result of exploration. Therefore, more explorations first and then exploitations are executed in our algorithm; θ should decrease over time as the algorithm converges. Therefore, θ can be updated as

$$\theta = \frac{\theta_0}{n^{\frac{1}{2}} + 1}, \tag{18}$$

where θ_0 is the initial value of θ.

The learning algorithm for ϵ-optimal strategy is described in Algorithm 1. We set the initial state as 0, the strategy pair as $\pi(0) = \{5, 5\}$, and γ as 0.8. Then, the strategy pair is calculated iteratively based on Equations (16) and (17) until the pair of the best strategy is achieved.

6 EVALUATIONS

In this section, we evaluate the performance of the proposed scheme, i.e., ϵ-Nash learning algorithm, on a server with an Intel Xeon processor at 2.80 GHz and 24 GB RAM. We conduct a set of simulations to pay close attention to the utility of u_1, and compare the average payoffs as well as the aggregate discounted payoffs of u_1 when u_1 adopts different strategies.

6.1 Experiment Setup

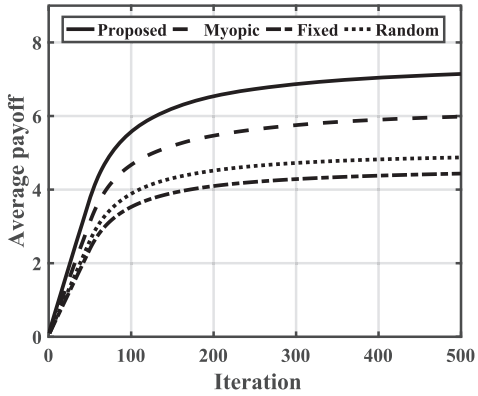
6.1.1 Parameter Settings. The parameter settings by default in our experiments are as follows. The transmission success ratio p_i follow the Gaussian distribution $p_i \sim N(1, \frac{1}{2q_i})$ [4]. The time discount factor γ is set as 0.8, and the unit cost per packet is set as 0.1. To closely approximate the Nash equilibrium, we set ϵ as 0.001.

6.1.2 Baselines. We assume that the rival node u_2 adopts the proposed strategy. We compare the utility of node u_1 when utilizing the proposed algorithm and the following baselines.

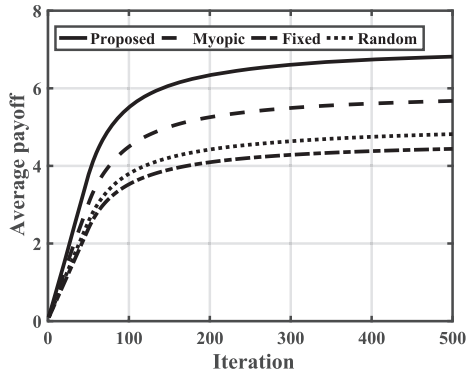
- *Random strategy.* Node u_1 randomly chooses an action at each time slot.
- *Fixed strategy.* Node u_1 chooses a fixed action at each time slot no matter what the state is.
- *Myopic strategy.* Node u_1 uses the optimal strategy, which is derived by myopic learning. In other words, we ignore the impact of the time discount factor, i.e., $\gamma = 0$.

6.2 Performance Analysis

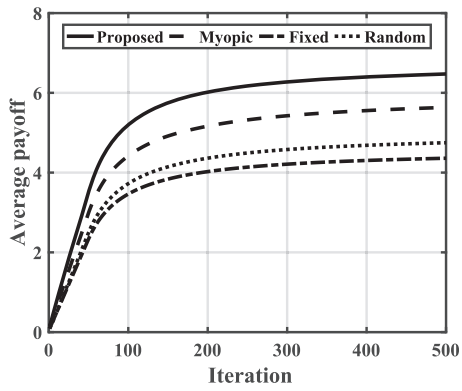
6.2.1 Payoff. We compare the aggregate average payoff and accumulated discounted payoff when different strategies are employed at time slot n under different $s(0)$ with the fixed transmission success ratio. As shown in Figures 4 and 5, we can observe that our strategy as well as the myopic strategy obtain higher payoff than random strategy as well as fixed strategy, since the two former strategies are both learning-based and dynamically updated according to the state. The strategy proposed in this article attains both the highest accumulated average payoff and the highest aggregate discounted payoff no matter what the initial state is, since the ϵ-Nash learning algorithm tries to find the best strategy for the current state by also taking the future payoff into account. This confirms that for nodes competing in IoTs that consider AoI as a contributing factor, taking the result learned from the ϵ-Nash learning algorithm, which considers both current payoffs and long-term payoffs, as their own strategies can always achieve the highest payoffs.



(a) {0.6, 0.8, 0.8, 0.8}



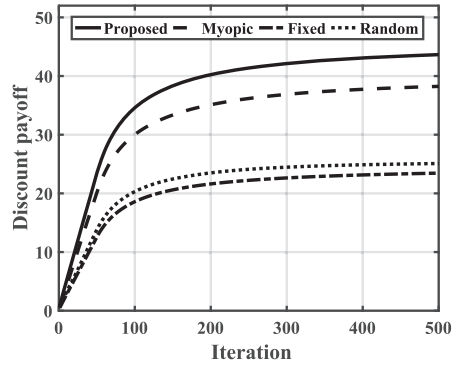
(b) {0.8, 0.8, 0.8, 0.8}



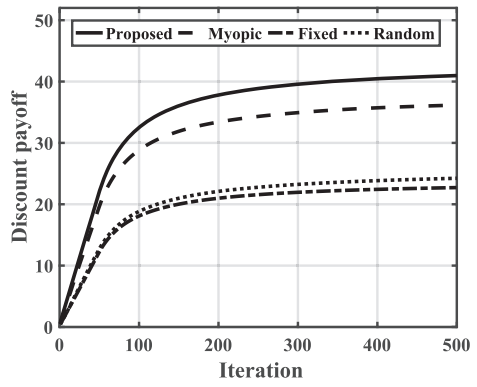
(c) {1, 0.8, 0.8, 0.8}

Fig. 4. Average payoff versus initial states.

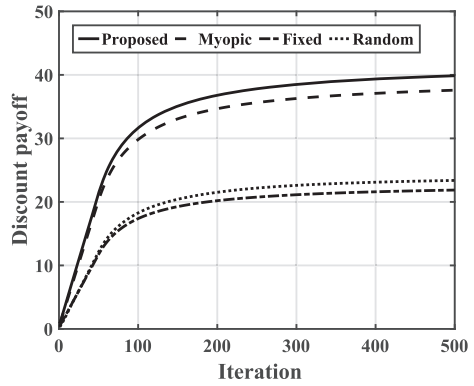
Figures 6 and 7 show the accumulated average payoff and aggregate discounted payoff of the four strategies with different channel qualities, given initial average AoIs. We can find that the strategy proposed in this article can obtain the best payoff, and the payoff of nodes increases with the channel quality, since a larger fraction of generated packets can be successfully transmitted,



(a) {0.6, 0.8, 0.8, 0.8}



(b) {0.8, 0.8, 0.8, 0.8}

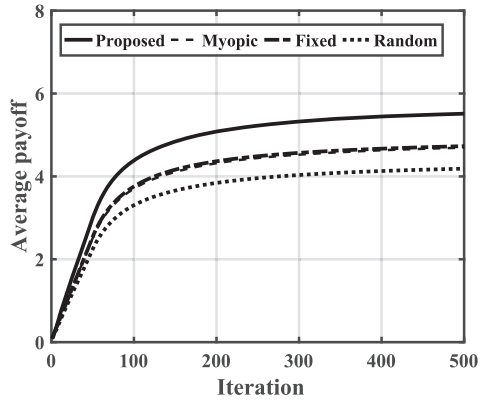


(c) {1, 0.8, 0.8, 0.8}

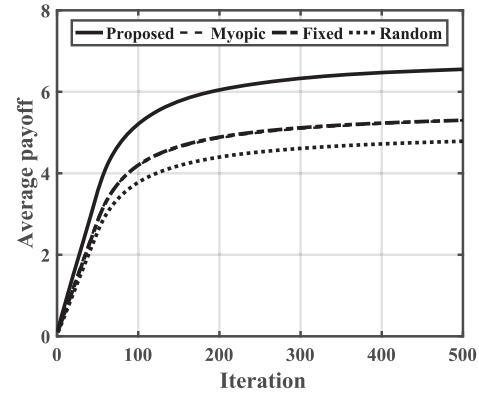
Fig. 5. Discounted payoff versus initial states.

which boosts the payoff of the nodes. In addition, all curves in Figures 4–7 gradually become stable, which shows that the proposed learning algorithm has converged to an ϵ-Nash equilibrium.

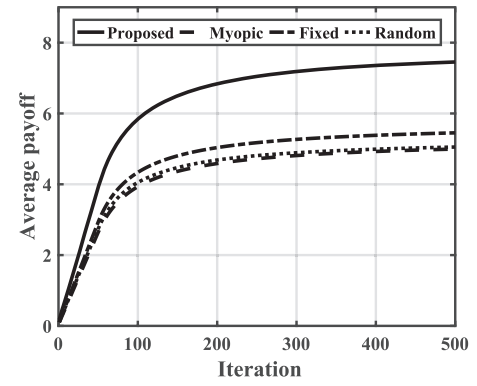
6.2.2 Impacts of State. In Figures 8 and 9, we analyze the impact of state on generation rate selection. Since the channel quality determines the transmission success ratio and introduces uncertainty to AoI, we compare the sum of generation rate over a long term.



(a) {0.5, 0.5, 0.6, 0.8}



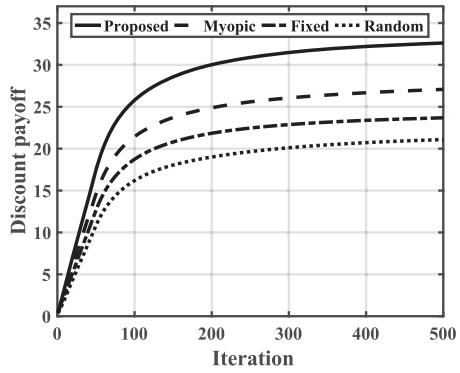
(b) {0.5, 0.5, 0.8, 0.8}



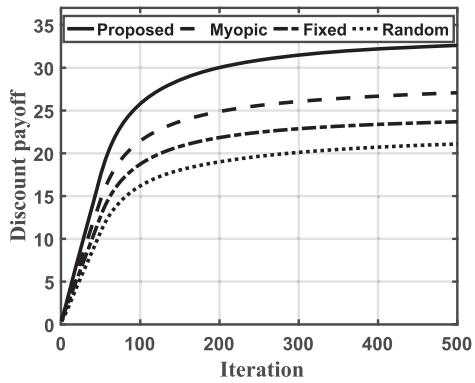
(c) {0.5, 0.5, 1, 0.8}

Fig. 6. Average payoff versus channel qualities.

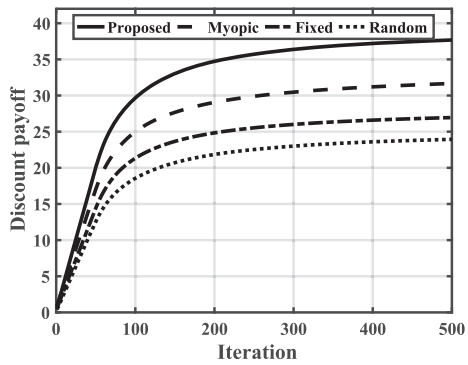
In Figure 8, we show the impact of initial average AoI on generation rate selection with transmission success ratio {0.8, 0.8}. In general, the sum of generation rate increases as the average AoI increases. Further, we analyze the generation rate selection when node u_1's initial average AoI is 0.6 with different initial state of u_2. It is observed that the generation rate selection of two nodes



(a) {0.5, 0.3, 0.6, 0.8}



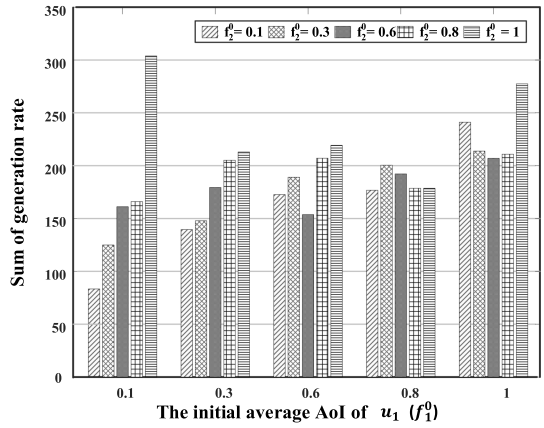
(b) {0.5, 0.5, 0.8, 0.8}



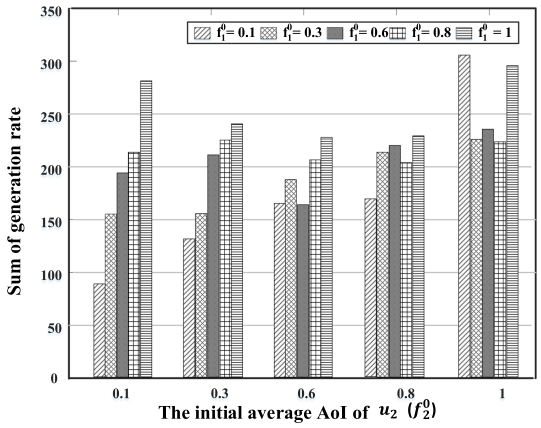
(c) {0.5, 0.7, 1, 0.8}

Fig. 7. Total discounted payoffs of different strategies with different channel qualities.

is lower when their initial average AoI is the same. This is because in an IoT, it is ideal to allocate the total reward evenly among sensor nodes to maximize the payoff of each node. Therefore, each node obtains the same reward in one time slot when the states (average AoI and channel quality) are the same, and applies a relatively small generation rate to reduce the cost for a higher payoff. Moreover, to minimize the average AoI, the control center can dynamically change the total reward according to the average AoI at each time slot to motivate nodes to provide more updated



(a) $\{f_1^0, f_2^0, 0.8, 0.8\}$



(b) $\{f_1^0, f_2^0, 0.8, 0.8\}$

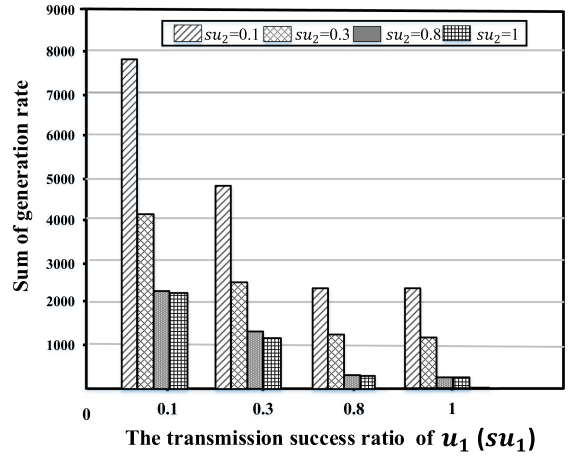
Fig. 8. Generation rate selection of nodes with different initial average AoI.

information. In this article, we assume that the total reward distributed to each node in each slot is proportional to the number of packets it successfully transmits and its corresponding average AoI, and we will explore other reward allocation schemes in our future work.

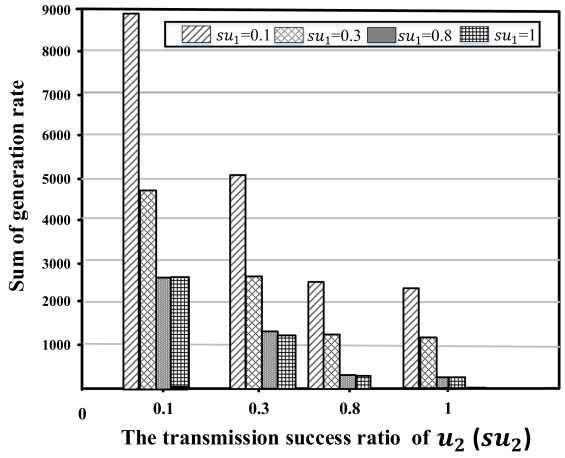
In Figure 9, we demonstrate the impact of channel quality on generation rate selection under different channel qualities with initial average AoI $\{0.5, 0.5\}$. It is shown that the sum of generation rate decreases as the channel quality improves. This indicates that when the channel quality is ideal, nodes can reduce the generation rate to lower the costs and obtain a high payoff.

6.2.3 Impacts of System Parameters. In this set of simulations, we analyze the convergence properties of the proposed algorithm and the impact of system parameters (i.e., channel quality and ϵ) on our proposed scheme.

As shown in Figure 10(a), the running time of the four strategies decreases with improvement of channel quality, where the initial average AoI is $\{0.5, 0.5\}$, and ϵ is 0.001. We noticed that Random strategy has the longest running time, and our algorithm as well as Myopic strategy can achieve shorter running time. Compared with Myopic strategy, the proposed algorithm always find the



(a) {0.5, 0.5, su_1, su_2}



(b) {0.5, 0.5, su_1, su_2}

Fig. 9. Generation rate selection of nodes with different channel qualities.

ϵ-optimal strategy as the Nash equilibrium solution, and thus require a slightly longer running time than Myopic strategy.

Given the initial average AoI $\{0.5, 0.5\}$, with better channel quality, the running time of the proposed algorithm generally shows a decreasing trend, as shown in Figure 10(b). Furthermore, larger ϵ corresponds to lower running time. This is mainly because the game between nodes will experience fewer rounds for bigger ϵ. More specifically, nodes will take the next move only when they have sufficient reward increment, which is consistent with the nature of ϵ-Nash Equilibrium.

The impact of ϵ on the profit of nodes is illustrated in Figure 11, where the initial average AoI is 0.5,0.5. Naturally, payoff increases with channel quality. Both average and discount payoff are low for bad channel quality but will increase rapidly with the increase of channel quality, as shown in Figures 11(a) and 11(b). At the same time, there is a rough trend that smaller ϵ corresponds to a higher reward. ϵ-Nash Equilibrium implies that the game target is to attain a growth of profit that exceeds ϵ. Thus, smaller ϵ will strive for even a little profit and thus have a larger reward. Our

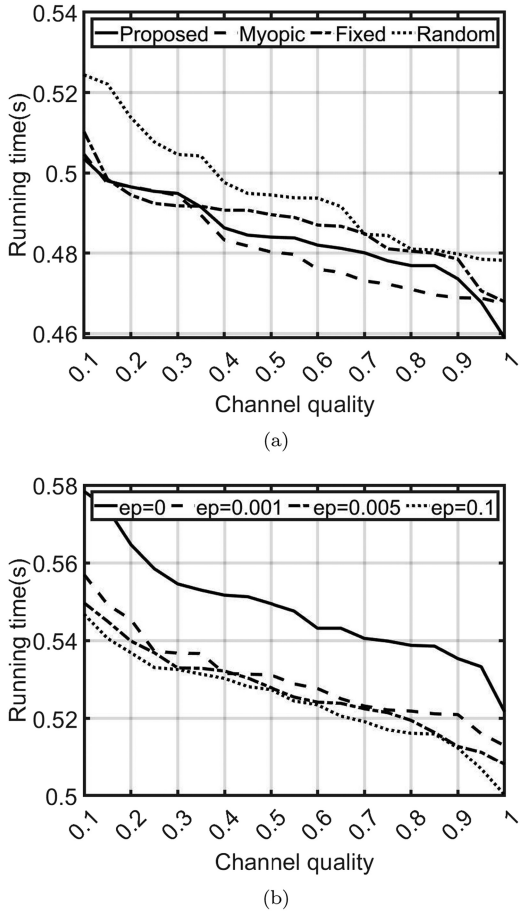


Fig. 10. Running time comparison with different channel quality.

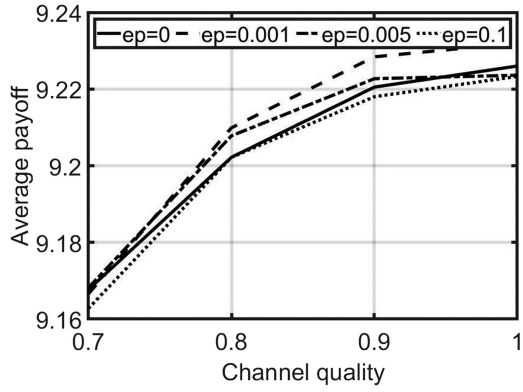
model will deteriorate into a general stochastic game when $\epsilon = 0$. Thus, its payoff is between the best and the worst. Combined with results in Figure 10(b), it is vital to choose ϵ cautiously to trade off between the running time and profit.

6.2.4 AoI. In Figure 12, we compare the average AoI under four strategies during 50 time slots. Specifically, we show the change process of average AoI with initial states $\{0.5, 0.5, 0.6, 0.8\}$. It is observed that our proposed strategy attains the minimum average AoI. In addition, we have demonstrated the change process of average AoI with different initial states and different channel quality, and all results indicate that our proposed strategy can achieve the minimum average AoI.

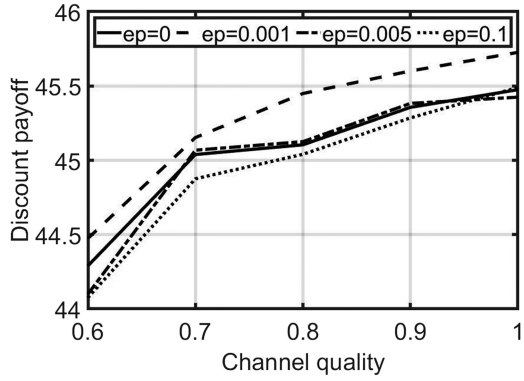
Based on the analysis above, we can find that our proposed strategy is superior to the baseline strategies. Our proposed strategy can obtain the highest payoff for nodes, while keeping the average AoI of platform minimum.

7 DISCUSSIONS

In this section, we discuss the limitation and guiding principles for control center administrators to consider.



(a)



(b)

Fig. 11. Payoff versus channel quality with different ϵ.

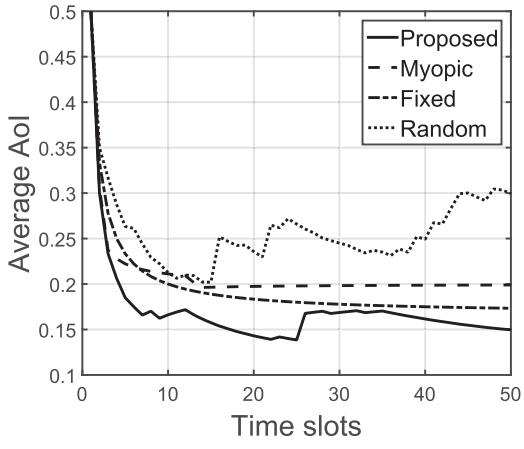


Fig. 12. Change process of AoI during 50 time slots.

7.1 System State Evaluation

The accurate evaluation of the system state determines the performance of the system. In our formulation, the system state $\mathbb{S}$ includes the average AoI and the channel quality. The average AoI of node n_i, $i \in \{1, 2\}$, describes the freshness of collected information provided by this node, and can be calculated by Equation (8). In our algorithm, we assume that all the packets sent in a time slot will be all received in the same time slot, and the calculation of the average AoI is based on time slots. However, due to uncertain reasons such as retransmission caused by low channel quality, packets sent in a time slot may not be received until the next time slot. In this case, packets take a long time to transmit, so they can no longer meet the information freshness requirements of the control center with high probability. Therefore, such packets can be regarded as being lost. Because of the challenges faced by wireless data transmission in IoTs, such as dynamic topology changes and time-varying communication environment, channel quality changes dynamically. Fortunately, there are some available general methods to estimate the channel quality. In general, channel quality is evaluated by CQI, which characterizes the overall influence of various effects, including scattering, fading, and power decay, on the propagation of wireless signals. Furthermore, our goal is to find the optimal data generation rate for nodes to achieve the best data freshness. The evaluation of channel quality is outside the scope of our study. Therefore, we assume that the channel quality of the node is given, and the node can deduce the channel quality of the rival node.

7.2 Penalty Term of the Game

In our two-player stochastic game, the payoff depends on the joint actions of two players. More specifically, the reward of each player relies not only on its average AoI and the number of packets successfully transmitted but also on the penalty term. With regard to the penalty term of the game, both the cost of packet generation and the overhead of transmission (e.g., power consumption) should be taken into account. Furthermore, packets transmitted by nodes may conflict, leading to packet loss. At the same time, if the transmission rate of the sending node exceeds the receiving capacity of the receiver, it will also lead to packet loss. According to the reliable transmission protocol, packet loss will cause retransmission. In principle, the resources wasted due to packet loss and the resources consumed by retransmission should be regarded as overhead, corresponding to the penalty term in our model. However, packet loss and resulting retransmission are unpredictable, and it is impossible to determine the accurate calculation formula of the cost. Therefore, we do not include the cost of packet loss and retransmission into the model but take the cost of data generation and only one-time data transmission as a penalty.

7.3 Extension to N-Player Stochastic Games

In the wireless data gathering applications, a node may have multiple rival nodes who employ diverse game policies. Accordingly, the proposed stochastic game formulation for two competing nodes should be extended to an N-player one. More specifically, an N-player stochastic data gathering game can be formulated as a tuple $\Gamma^N =< \mathbb{S}, \mathbb{A}_1, \ldots, \mathbb{A}_N, r_1, \ldots, r_N, P >$, where (1) $\mathbb{S}$ is the state space, and the system state is characterized by the average AoI and the channel quality, $\mathbb{S} = \{s(1), \ldots, s(T)\}$, $s(T) = \{f_1(T), \ldots, f_N(T), q_1(T), \ldots, q_N(T)\}$, where $s(T)$ is the state at time slot T, $f_i(T)$ is the average AoI of node u_i, $i \in \{1, \ldots, N\}$, at time slot T, and $q_i(T)$ is the channel quality of node u_i, $i \in \{1, \ldots, N\}$, at time slot T, (2) $\mathbb{A}_i$ is the action space that contains all possible actions that node u_i can take, $i = 1, 2, \ldots, N$, (3) $r_i : \mathbb{S} \times \mathbb{A}_1 \times \cdots \times \mathbb{A}_N \mapsto r_i$ is the payoff for u_i, where $r_i \in \mathbb{R}, \mathbb{R}$ is denoted as the set of real numbers, (4) $P : \mathbb{S} \times \mathbb{A}_1 \times \cdots \times \mathbb{A}_N \mapsto \Delta(\mathbb{S})$ is described as the transition probability function of $\mathbb{S}$ and actions, i.e., $\mathbb{A}_1$, …, $\mathbb{A}_N$ taken by u_i, $i = 1, 2, \ldots, N$. The two-player stochastic game formulation of the system states that actions to

choose the generation rate, stage payoff, and state transition probability detailed in Section 4 can be extended to the multiplayer stochastic game without difficulty, and the learning algorithm can achieve the maximum sum of payoffs for multiplayer.

8 CONCLUSION

In this article, we present a general-sum stochastic game model to analyze the AoI update for IoTs, which characterizes the interactions and competitions among sensor nodes. To derive the best long-term payoff, we investigate a Nash learning algorithm for nodes to dynamically adjust their strategies. Our extensive experimental results and performance analysis verify that our proposed algorithm outperforms baseline algorithms. We believe that the proposed algorithm can provide practical guidelines for nodes to get more payoff in highly competitive IoTs.

REFERENCES

[1] Ahmed M. Bedewy, Yin Sun, and Ness B. Shroff. 2016. Optimizing data freshness, throughput, and delay in multi-server information-update systems. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT'16)*. 2569–2573.

[2] Yanjiao Chen, Fan Zhang, Kaishun Wu, and Qian Zhang. 2015. QoE-aware dynamic video rate adaptation. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM'15)*. 1–6.

[3] Steve Chien and Alistair Sinclair. 2011. Convergence to approximate nash equilibria in congestion games. *Games Econ. Behav.* 71, 2 (2011), 315–327.

[4] Alessandro Chiumento, Mehdi Bennis, Claude Desset, Andre Bourdoux, Liesbet Van Der Perre, and Sofie Pollin. 2015. Gaussian process regression for CSI and feedback estimation in LTE. In *Proceedings of the IEEE International Conference on Communication Workshop*.

[5] Shugang Hao and Lingjie Duan. 2019. Economics of age of information management under network externalities. In *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'19)*. 131–140.

[6] Junling Hu and Michael P. Wellman. 1998. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the International Conference on Machine Learning (ICML'98)*, Vol. 98. Citeseer, 242–250.

[7] Junling Hu and Michael P. Wellman. 2003. Nash Q-learning for general-sum stochastic games. *J. Mach. Learn. Res.* 4 (Nov. 2003), 1039–1069.

[8] Igor Kadota, Abhishek Sinha, and Eytan Modiano. 2018. Optimizing age of information in wireless networks with throughput constraints. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'18)*. 1844–1852.

[9] Sanjit Kaul, Roy Yates, and Marco Gruteser. 2012. Real-time status: How often should one update. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'12)*. 2731–2735.

[10] Chengzhang Li, Shaoran Li, and Y. Thomas Hou. 2019. A general model for minimizing age of information at network edge. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'19)*. 118–126.

[11] Xiaomin Lin, Stephen C. Adams, and Peter A. Beling. 2018. Multi-agent Inverse Reinforcement Learning for General-sum Stochastic Games. arXiv preprint arXiv:1806.09795 (2018).

[12] Michael L. Littman. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*. Elsevier, 157–163.

[13] Abraham Neyman, Sylvain Sorin, and S. Sorin. 2003. *Stochastic Games and Applications*, Vol. 570. Springer Science & Business Media.

[14] H. L. Prasad, L. A. Prashanth, and Shalabh Bhatnagar. 2015. Two-timescale algorithms for learning Nash equilibria in general-sum stochastic games. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1371–1379.

[15] Lloyd S. Shapley. 1953. Stochastic games. *Proc. Natl. Acad. Sci. U.S.A.* 39, 10 (1953), 1095–1100.

[16] Beibei Wang, Yongle Wu, K. J. Ray Liu, and T. Charles Clancy. 2011. An anti-jamming stochastic game for cognitive radio networks. *IEEE J. Select. Areas Commun.* 29, 4 (2011), 877–889.

[17] Wei Wang and Qian Zhang. 2014. A stochastic game for privacy preserving context sensing on mobile phone. In *Proceedings of the IEEE Conference on Computer Communications (INFOCOM'14)*. 2328–2336.

[18] Xuehe Wang and Lingjie Duan. 2019. Dynamic pricing for controlling age of information. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT'19)*.

[19] Shuangke Wu, Yanjiao Chen, Minghui Li, Xiangyang Luo, Zhe Liu, and Lan Liu. 2020. Survive and thrive: A stochastic game for DDoS attacks in bitcoin mining pools. *IEEE/ACM Trans. Netw.* 28, 2 (2020), 874–887.

[20] Sun Yin, Elif Uysal-Biyikoglu, Roy D. Yates, C. Emre Koksal, and Ness B. Shroff. 2017. Update or wait: How to keep your data fresh. *IEEE Trans. Info. Theory* 63, 11 (2017), 7492–7508.
[21] Meng Zhang, Ahmed Arafa, Jianwei Huang, and H. Vincent Poor. 2019. How to price fresh data. In *Proceedings of the International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT'19)*.

Received 6 January 2022; revised 24 August 2022; accepted 21 September 2022