SMART: Situationally-Aware Multi-Agent Reinforcement Learning-Based Transmissions

Zhiyuan Jiang, Yan Liu, Jernej Hribar, Luiz A. DaSilva, *Fellow, IEEE*, Sheng Zhou, and Zhisheng Niu, *Fellow, IEEE*

Abstract

In future wireless systems, latency of information needs to be minimized to satisfy the requirements of many mission-critical applications. Meanwhile, not all terminals carry equally-urgent packets given their distinct situations, e.g., status freshness. Leveraging this feature, we propose an on-demand Medium Access Control (MAC) scheme, whereby each terminal transmits with dynamically adjusted aggressiveness based on its situations which are modeled as Markov states. A Multi-Agent Reinforcement Learning (MARL) framework is utilized and each agent is trained with a Deep Deterministic Policy Gradient (DDPG) network. A notorious issue for MARL is slow and non-scalable convergence – to address this, a new Situationally-aware MARL-based Transmissions (SMART) scheme is proposed. It is shown that SMART can significantly shorten the convergence time and the converged performance is also dramatically improved compared with state-of-the-art DDPG-based MARL schemes, at the expense of an additional offline training stage. SMART also outperforms conventional MAC schemes significantly, e.g., Carrier Sensing and Multiple Access (CSMA), in terms of average and peak Age of Information (AoI). In addition, SMART also has the advantage of versatility – different Quality-of-Service (QoS) metrics and hence various state space definitions are tested in extensive simulations, where SMART shows robustness and scalability in all considered scenarios.

Z. Jiang and Y. Liu are with Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China. Emails: {jiangzhiyuan,lyann}@shu.edu.cn.

J. Hribar is with CONNECT, Trinity College Dublin, Dublin, Ireland. Email: jhribar@tcd.ie.

L. A. DaSilva is with the Commonwealth Cyber Initiative, Virginia Tech, USA. Email: ldasilva@vt.edu.

S. Zhou and Z. Niu are with Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. Emails: {sheng.zhou, niuzhs}@tsinghua.edu.cn.

This work was supported by the National Key R&D Program of China (No. 2019YFE0196600), the program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning.

Part of this work has been presented at *IEEE SPAWC* 2019 [1]. The corresponding author is Sheng Zhou.

Index Terms

Internet-of-Things, medium access control, multi-agent reinforcement learning, contention-based random access, Markov decision process

I. Introduction

In the last few years, we have witnessed a shift in wireless communications from human-based communications, e.g., voice and data, to Machine-Type Communications (MTC). Massively interconnected devices enable new applications such as factory automation, intelligent transportation system, tactile internet [2], etc. Unfortunately, current networks are not optimized to support a large number of devices per cell. In many cases, MTC devices transmit small data packets coupled with low data rates which leads to a high volume of control signaling in the network. Another challenge for the network is that devices transmit seldom and randomly. As a consequence of such massive MTC (mMTC) characteristics, the design of new solutions on how can devices access the wireless transmission medium is a necessity.

In this paper, we focus on enhancing Medium Access Control (MAC) by providing devices, which from now on we refer to as terminals, with an ability to cooperate when accessing the shared transmission medium. For terminals to successfully transmit a data packet, the terminals have to, by relying on the MAC layer, gain access to time/frequency resources each terminal requires. However, due to the large number of terminals allocating resource may lead to high latencies. For example, the uplink MAC layer latency in cellular networks (based on 3GPP Release 15) can be up to 10 ms, due to the fact that a terminal with a packet to be transmitted in the uplink must request a downlink grant, which indicates the allocated resources, before sending the packet. Note, that in our work, we consider only user-plane latency, which is defined as the time between the first transmission of a data packet and the reception of a physical layer acknowledgment. Enhancements such as grant-free schemes have been proposed to reduce the MAC uplink latency [3]. However, such solutions still do not fully resolve one of the main issues in mMTC; the high volume of control signaling. To that end, we propose the use of Deep Reinforcement Learning (DRL) to enable the terminal to learn when to take the wireless transmission medium without any additional control signaling messages from the Base Station (BS).

A. Contributions and Outline

By letting the terminals be situationally-aware, i.e., comprehend their urgency to transmit, the uplink MAC can be made more efficient, and this improved efficiency can be captured through application-oriented, flow-level metrics such as the Age of Information (AoI) [4]. In this paper, we leverage Multi-Agent Reinforcement Learning (MARL) to design a decentralized medium access solution that can be applied to latency-critical services in 5G, such as data collection from sensors in closed-loop control for Industrial Internet-of-Things (IIoT). Our main contributions include the following:

1. We propose a Situationally-aware MARL-based Transmissions (SMART) solution to address the notorious convergence issue in MARL for the decentralized access procedure. Based on SMART, terminals observe their local states and receive a common transmission tax signal from the central controller to determine their transmit probabilities, such that much of the training can be taken offline to improve robustness. This approach is shown to shorten the convergence time and improve the performance significantly compared with state-of-the-art MARL schemes, at the expense of an additional offline training stage wherein each terminal is trained and pre-stores its set of Deep Deterministic Policy Gradient (DDPG) parameters.

2. We conduct extensive investigations of our proposed scheme. We compare SMART against a conventional MARL scheme, which we implemented using a DDPG approach to highlight the advantages of SMART, as well as the conventional Carrier Sensing Multiple Access (CSMA) scheme, under various system parameters and Quality-of-Service (QoS) metrics. We show that the MARL-based scheme outperforms CSMA significantly when considering metrics such as AoI. In particular, SMART achieves the best performance and has the lowest complexity.

3. We provide detailed analysis of computation complexity, storage space, convergence time and runtime, to support a thorough comparison between the proposed and conventional schemes.

The remainder of the paper is organized as follows. In Section II, we introduce the system model and formulate the problem of utility maximization. For clarity, we present our main results here and detailed proofs and explanations are conveyed in the subsequent sections. In Section III, we present the proposed SMART scheme in detail. Section IV presents the main algorithm used for comparison and simulation results. Finally, in Section VI, we discuss our conclusions and directions for future work.

B. Related Works

The MAC layer operation is a key component of wireless link performance. Most work in this area has focused on throughput optimization – in particular, the notion of throughput optimality and its corresponding achievability has been investigated extensively [5]–[7]. Recently, with the advent of more delay-sensitive services, the focus has shifted to the issue of latency. It is well known that contention-based random access schemes, including Aloha and all variants of CSMA, suffer significant latency performance degradation under heavy traffic loads. To address this issue, extensive research efforts have been made, either aiming to prioritize the access [8]–[10], or to enhance the physical access capabilities [3], [11]. For the purpose of access prioritization, one of the techniques proposed is called Extended Access Barring (EAB) [9]. This technique randomly selects a certain set of terminals to transmit by broadcasting a threshold by the access point; each terminal generates a random number that it compares against that threshold. IEEE 802.11e supports QoS enhancements by pre-defining four channel access categories, each with a different priority in accessing the medium, achieved by adopting different contention window sizes [10]. Sparse-Code Multiple-Access (SCMA) [3] and beamforming [11] techniques, in turn, can be regarded as aiming to increase the number of concurrent terminals that can be supported by strengthening the receiver capability in the code and spatial domains, respectively. In IIoT, WISA [12] is one of the earliest wireless technologies that is based on a contention MAC. It is built upon the IEEE 802.15.1 physical layer and modifies the MAC layer such that the latency can be within 2 ms (without retransmissions). WISA supports up to 120 devices, but the reliability is insufficient to support closed-loop control applications. Based upon WISA, the Wireless Sensor Actuator Network for Factory Automation (WSAN-FA) standard [13], [14] was proposed for the PROFIBUS and PROFINET automation protocols. WSAN-FA achieves sufficient latency and reliability requirements, but with limited scalability of up to 40 devices.

On the other hand, cellular networks adopt a grant-based approach to avoid the collisions introduced by contention-based random access. That is, the BS transmits a downlink grant and allocates the resources centrally after it receives the Scheduling Requests (SRs) from terminals with packets to send. Since the SR interval is typically set to 10 ms [15], this procedure introduces latency that is unacceptable for mmMTC. Two approaches have been proposed to remedy this issue. The first is for the BS to pre-allocate uplink resources to certain terminals, which is referred to as fast uplink access [15]. This approach is based on predicting the uplink traffic

using, e.g., machine learning techniques, and hence may suffer from prediction inaccuracy. The other approach is grant-free uplink transmission [3], which is basically a contention-based scheme and hence has the same issue discussed in the previous paragraph.

Recently, RL has been widely applied to wireless communication systems, such as edge computing [16]–[18] and interference awareness [19]. Ref. [20] reviewed Deep RL-based methods that address the problems in communication networks, e.g., dynamic network access, data rate control, and wireless caching. Ref. [21] mainly applied DRL to solve the dynamic spectrum access problem—in particular, terminals access the wireless channels based on the DRL scheme with input being the previous feedback of transmission acknowledgments. In Ref. [22], authors have developed the deep reinforcement learning multiple access protocol for heterogeneous networks, capable of learning to achieve a global objective such as maximizing the total throughput or maximizing α-fairness among all terminals. However, none of the existing work has considered the situation of the terminals themselves, which can be taken as states in the DRL framework, allowing the terminals to adjust their transmission strategies accordingly. In many applications, the situations of terminals reflect the transmission urgency of the messages, which can be utilized to enhance the QoS. In [23], the authors reduce the duty cycle by compressing the activity time in continuous intervals results in lower power consumption. In essence, it learns about traffic trends, but it is not possible to make predictions about traffic in a real system. In [24], the ALOHA-Q protocol uses the Q learning mechanism as an intelligent policy for slot selection in frame-based ALOHAs to avoid conflicts with minimal additional overhead. However, in our paper the distributed system guarantees low signaling overhead and we focus on optimizing the user's QoS rather than avoiding conflicts.

The approach we take in this paper relies on situational-awareness [25], a concept related to the recent work on AoI [4], as well as Age of Synchronization (AoS) [26], Age upon Decision (AuD) [27] and Inter-Delivery-Time (IDT) [28]. In this line of work, a terminal has a time-varying sense of its state, which can reflect the value of its yet-to-be-sent packet; in contrast, traditional CSMA-type schemes assume state-less terminals. In our previous works [29]–[32], we have derived closed-form Whittle's index-based policies and decentralized scheduling schemes exclusively for AoI optimization. The scheduling problem for AoI optimization was also investigated in [33]–[37], however not considering the distributed scheduling nature of the uplink MAC. In this work, we build on our previous work [1]. Whereas the previous algorithm uses Deep Q-Network (DQN), while our new proposed SMART framework uses a DDPG network,

TABLE I
NOMENCLATURE

Notation	Description	Notation	Description	Notation	Description
$p_n(t)$	Transmission probability	ω_n	Terminal weight	$S_n(t)$	Markov state
$\bar{U}_n$	Average utility function	$h_{\text{AoI},n}(t)$	AoI at the FC	T	Contention process time
$\mu_{\text{AoS}}(t)$	Time passed since FC received a new update from the terminal	r_k	Arrival time of the k-th packet	ν_k	Duration of the valid update
$\mu_{\text{AoI}}(t)$	Time passed since terminal generated new packet	η	Target network update coefficient	τ'	Policy function of the target network
$g_n(t)$	AoI of the most up-to-date packets at the terminal queue	$A_{p,k}$	Peak age of the k-th update	Θ^Q, Θ^τ	Parameterize function approximators
$\mathcal{R}_{S_n}^{(0)}, \mathcal{R}_{S_n}^{(1)}$	Expected reward function for the transmit and silence	$a_{t/i}, s_{t/i}, r_{t/i}$	Action, state, reward	N_{target}	Number of contending terminals
p_{tx}	Transmission probability	$\hat{J}^*$	Average reward	D_k	IDT of the k-th update
$s_{m/f}$	Utility at the terminal/FC	R	Batch size	ι_k	Transmission interval

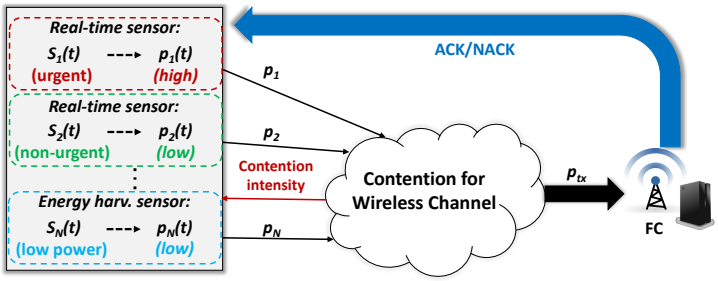


Fig. 1. Illustration of SMART framework, wherein heterogeneous QoS requirements can be simultaneously satisfied by learning-based access strategies.

which is more stable due to the adoption of soft updates and actor-critic ideas. The SMART solution we propose still adopts a distributed architecture, most of the training can be carried out offline, and results in better convergence performance. This paper goes beyond AoI optimization, considering arbitrary state definitions and QoS requirements of terminals by using a model-free MARL framework. Tab. I illustrates the notation used in the paper.

II. System Model and Problem Formulation

The SMART framework is illustrated in Fig. 1. In our system, the goal of each terminal is to learn the probability $p_n(t)$ with which it should try to transmit status updates. Terminals compete for the wireless channel and only one terminal in the time-slot can successfully transmit a status update to the Fusion Center (FC) for collection. A multi-access network is considered, wherein

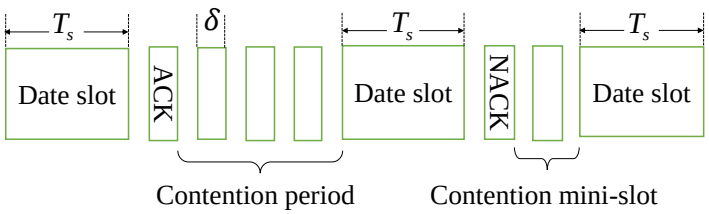


Fig. 2. Illustration of the transmission frame. Each big block represents a data slot, and each small block represents a mini-slot.

an information FC collects information from N distributed terminals with possibly heterogeneous QoS requirements. Time is slotted and we assume the terminals are synchronized – for instance, the terminals can maintain synchronization by receiving the primary synchronization signal from the FC but with no scheduling grants. The transmission model in the uplink is collision-based. As shown in Fig. 2, a transmission frame consists of data slots and several contention mini-slots. A data slot (of length $T_{\rm s}$) is preceded by several contention mini-slots (of length δ). In 5G New Radio (NR), the concept of mini-slots [11] is introduced: this is the minimum scheduling time unit, occupying as little as a single Orthogonal Frequency Division Multiplexing (OFDM) symbol. Given the scalable numerology of NR, wherein one slot, consisting of 14 OFDM symbols, can be 0.125 ms with 120 kHz Subcarrier Spacing (SCS), each mini-slot can be quite short ($\delta = 1/56$ ms or lower with larger SCS).

We assume a p-persistent CSMA framework [38], whereby terminal n transmits with a probability p_n in each contention mini-slot when it senses the channel as idle; otherwise, it stays silent. Note that, different from homogeneous p-persistent CSMA, the persistence levels of terminals can be different, i.e., p_n differs among terminals. In this way, the terminals can be situationally aware and thereby choose the appropriate p_n. We note that in the Q-CSMA scheme [6], p_n is determined by the queue length of each terminal, which however only applies to throughput optimization. Based on the definition of p-persistent CSMA, the terminal that has won the contention transmits in the following data slot and the others sense that the channel is busy and stay silent. After a data slot, the FC feeds back an acknowledgment (ACK) packet indicating successful reception; otherwise, a Negative-ACK (NACK) packet is fed back. Note that a p-persistent CSMA protocol closely approximates the IEEE 802.11 CSMA protocol, which employs uniform backoff counters and binary exponential backoff, if p and the backoff window size are chosen such that the average backoff intervals of the two protocols are identical.

In this work, we only consider access to a single channel. By using e.g., CSMA, beamforming and multiple frequency sub-channels, it is certainly possible to enable multiple packet reception

simultaneously. A straightforward approach to extend from single-channel to multiple packet reception is to let the terminals choose one channel randomly or to uniformly pre-allocate the channels.

Problem Formulation: The objective is to minimize the overall utility of all terminals over time, i.e.,

$$\min_{p_n(t),n=1,\cdots,N} \sum_{n=1}^{N} \omega_n \overline{U}_n, \tag{1}$$

where ω_n denotes the weight associated with the terminal n. $p_n(t)$ only depends on Markov state $S_n(t)$ and the average utility function $\overline{U}_n$, which is expressed as follows,

$$\overline{U}_n \triangleq \liminf_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} U_n(t), \tag{2}$$

where T denotes the total time of the contention process and the utility function of terminal n at time t is denoted by $U_n(t)$ which reflects the terminal's QoS requirements and will be further developed in the following section. In each time slot, the terminals choose their transition probabilities $p_n(t)$, depending only on their own situation, i.e., Markov states $S_n(t)$. They do so using learning-based approaches that will be specified later – this decentralized approach is especially important in future massive IoT systems, in order to avoid the prohibitively high signaling overhead of centralized scheduling methods.

General Situation (Markov State) Characterizations: The situation that terminal n is in at time t is denoted by its Markov state $S_n(t)$, which is a real-valued vector. The definition is quite general, encompassing arbitrary state space and transition dynamics. Note that even for non-Markov states, we can define the state as the concatenation of several historical states as approximated by Markov states.

Long-term Average AoI: The recently proposed concept of AoI [4] can be used to measure the information delay at the destination, so as to describe the freshness of information. It is formally defined as the time elapsed since the last-updated packet's generation. This definition takes into account the delays introduced by sampling the information source and data communication. The state of terminal n is defined as $S_{\text{AoI},n}(t) = (g_n(t), h_{\text{AoI},n}(t))$, where $g_n(t)$ is the age of the most up-to-date packet at the terminal queue, and $h_{\text{AoI},n}(t)$ denotes the AoI at the destination, which is known to the terminal by keeping track of the receiver feedback. The AoI is formally defined as

$$h_{\text{AoI},n}(t) \triangleq t - \mu_{\text{AoI}}(t), \tag{3}$$

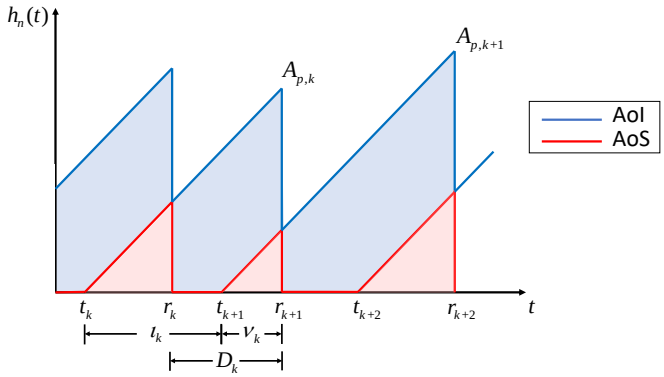


Fig. 3. An example that illustrates the evolution of AoI, AoS, IDT and PAoI. The transmission time of the k-th packet is t_{k-1} and the reception time of the k-th packet is r_k. ι_k is the transmission interval, and ν_k is the effective update time. Here, $A_{\mathrm{p},k}$ denotes the k-th peak of age, D_k represents the IDT of the k-th packet and the change processes of AoI and AoS are represented by the blue line and red line respectively.

where $\mu_{\mathrm{AoI}}(t)$ is the generation time of the most up-to-date packet at the receiver side up to time t. The AoI is helpful in control systems in which the control action stringently depends on the timely arrival of status updates from sensors. However, a long-term average AoI better characterizes the data freshness in the system. Therefore, the objective for each terminal is to minimize the long-term average AoI, defined as follows:

$$\bar{U}_{\mathrm{AoI}} \triangleq \limsup_{\tau\to\infty} \frac{1}{\tau}\sum_{t=1}^{\tau} \mathbb{E}[h_{\mathrm{AoI},n}(t)]. \tag{4}$$

The access probability $p_n(t)$ is consistent with the definition in the p-persistent CSMA scheme [38]. Conceptually, when the AoI of the terminal is high, meaning that the terminal has an aged status update at the FC, the terminal should be eager to transmit and hence $p_n(t)$ should be higher.

Long-term Average AoS: Similar to AoI, AoS is defined as the time difference between the current time and when the terminal became unsynchronized with the FC, expressed as

$$h_{\mathrm{AoS},n}(t) \triangleq t - \mu_{\mathrm{AoS}}(t), \tag{5}$$

where $\mu_{\mathrm{AoS}}(t)$ denotes the earliest time the FC received a packet since the last refresh of the terminal. The concept of AoS is also graphically illustrated in Fig. 3. Minimizing long-term average AoS is a QoS objective for some terminals, where the long-term average AoS is

$$\bar{U}_{\mathrm{AoS}} \triangleq \limsup_{\tau\to\infty} \frac{1}{\tau}\sum_{t=1}^{\tau} \mathbb{E}[h_{\mathrm{AoS},n}(t)]. \tag{6}$$

Long-term Average PAoI: We also consider the Peak Age of Information (PAoI) [39], which provides information about the maximum age value reached before an update was received. As shown in Fig. 3, consider the k-th packet's PAoI, represented by $A_{\mathrm{p},k}$. The peak age of the k-th update is

$$A_{\mathrm{p},k} = \iota_k + \nu_k, \tag{7}$$

where ι_k is the transmission interval and ν_k is the duration of the valid update. Therefore, the long-term average PAoI [40] is defined as

$$\bar{U}_{\mathrm{PAoI}} \triangleq \lim_{\kappa\to\infty} \frac{1}{\kappa} \sum_{k=1}^{\kappa} A_{\mathrm{p},k}. \tag{8}$$

PAoI is closely related to the AoI previously considered, has a simple formula, and characterizes the maximum age of information before receiving updates, which can be relevant in the context of multiple applications.

Long-term Average IDT: In this work, we are interested in treating inter-delivery time as a performance metric. IDT refers to the time between the kth and the $(k+1)$th packet of terminal n is denoted by D_k.

$$D_k = r_{k+1} - r_k. \tag{9}$$

where r_k is the arrival time of the kth packet. The terminal's QoS objective is to minimize the long-term average IDT with an energy constraint, i.e., the terminal can only transmit when its energy buffer is non-empty. The long-term average IDT is defined as

$$\bar{U}_{\mathrm{IDT}} \triangleq \lim_{\kappa\to\infty} \frac{1}{\kappa} \sum_{k=1}^{\kappa} D_k. \tag{10}$$

Our approach to enhancing the MAC layer for mmMTC is by noticing the fact that in many applications, e.g., sensors in CPS and autonomous driving vehicles, terminals have situational awareness (represented as states), and hence can use this awareness when accessing the wireless medium on-demand. In this work, the situation of a terminal is represented by a real-valued vector, which can incorporate heterogeneous types of terminals and their QoS requirements. An exemplary illustration is given in Fig. 1. For simplicity and concreteness, consider three terminals contending for access to the information FC. One of the terminals is a real-time sensor that updates its status and aims for AoI minimization [4]. In contrast, the second terminal is also a real-time sensor with the objective of minimizing AoI but has lower AoI at the moment. Therefore, its access probability $p_2(t)$ should be lower than $p_1(t)$. The third terminal is an

energy harvesting sensor, which is envisioned to be ubiquitous in future IoT systems. This poses a challenge for MAC for that not only the transmission urgency of each terminal should be considered as in the AoI minimization case, the transmission capability should also be taken into account. Fortunately, our general model is able to incorporate these considerations. We introduce our proposed schemes in the next section.

III. Proposed Situationally-Aware MARL-Based Transmissions Access Scheme

When considering the general Markov state definition in the previous section, it is quite challenging to find a universal analytical solution for various types of states and QoS requirements. For example, our previous works [30]–[32] found that, even for AoI optimizations alone, this can be quite challenging, let alone for the co-existence of diversified types of states. In view of this, we resort to the MARL framework.

RL [41] is a model-free control mechanism and therefore is applicable to arbitrary types of states and state transitions. The unique property of this problem is that each terminal, or agent in RL terms, can only observe its own states in order to make decisions – this is referred to as a *partially observable identical payoff stochastic game* (POIPSG) [42], which is used to model a problem wherein multiple agents learn simultaneously with a single objective (total utility functions) and observations of only local state information. The game terminology reflects the interplay among terminals, which is quite different from the conventional static environment setting in RL, as agents can interfere with each other while learning, and is thus named MARL.

The MARL nature of the problem poses significant challenges to find a scalable and stable solution. The challenging part, which is well-known for MARL problems, is that each agent only observes its own states' evolution over time, and its environment involves the actions of other agents, thus making it non-stationary – like trying to learn from a moving target. At the same time, all agents try to learn an optimal policy, which is the one that yields the highest long-term reward. This is why it is very hard to design a stable and scalable (massive number of terminals are common in IIoT) solution. In order to solve the above-mentioned problems, we have experimented with several approaches, as described in the following sections, including the state-of-the-art SMART scheme which combines the idea of Whittle's index and RL. We consider the set of stochastic reactive policies wherein the action of the terminal only depends on the current states, and the dependence can be probabilistic. We show that our approach achieves near-optimal performance consistently in various scenarios.

A. Situationally-Aware MARL-Based Transmissions

The newly proposed SMART approach builds on our previously proposed Transmission Tax (TT) based decoupled MARL approach [1]. TT is based on the idea of decoupled RL training to avoid the convergence issue introduced by the interplay among agents in MARL. The challenge the system is facing is to ensure that the trained policy using decoupled RL training also works well under the multi-agent setting. In particular, if we naively trained each agent separately with the objective of optimizing its own utility, then all agents would become selfish, causing the channel to be jammed all the time because no agents are trained to cooperate with others. We resolve this issue by introducing a universal transmission tax for all terminals when trained separately. That is, when an agent is trained, a transmission tax (i.e., a cost m) is added whenever the agent chooses to transmit; when it chooses to stay silent, no tax is added. By doing this, agents are trained to be less selfish, and more conservative in transmissions, i.e., only when an agent is in a situation where it has a high-value packet would it actually transmit, or else the transmission tax would surpass the value of the transmission. This approach works well in various scenarios. In fact, the transmission tax signal is broadcast from the FC at a very low frequency. The terminal, whether it transmits or not, is determined by itself. Therefore, it is still a distributed algorithm in general considering the signaling overhead. In the scenario of this paper, compared with the centralized method, the distributed method has a small increase in signaling overhead. This is caused by the difference between the signal sending cycle and the transmission tax signaling sending cycle. In a period of time, the adjustment period of the transmission tax is greater than the medium access period, so the decentralized method is especially important when there are a large number of terminals, to avoid the high signaling overhead of the centralized scheduling method. TT, while a very simple heuristic, is in fact based on Whittle's index, which is widely known to result in a near-optimal approach for this kind of problems, specifically restless multi-armed bandit.

The connection with the Whittle's index approach is illustrated as follows. In that approach, the utility maximization scheduling problem is decomposed into N sub-problems, where each subproblem can be formulated based on the Bellman optimality equations (average cost with infinite-horizon and relative cost-to-go functions [43]) as

$$f(S_n)+\hat{J}^*=\min\left\{\begin{array}{l}\mathcal{R}^{(0)}_{S_n}+\sum_{S'_n}\mathcal{P}^{(0)}_{S_nS'_n}f(S'_n),\\ m+\mathcal{R}^{(1)}_{S_n}+\sum_{S'_n}\mathcal{P}^{(1)}_{S_nS'_n}f(S'_n)\end{array}\right\}, \tag{11}$$

wherein the top and bottom terms in the minimization operator represent the cost-to-go from state S_n onward with the actions of remaining silent and transmitting, respectively. The expected reward functions are denoted by $\mathcal{R}_{S_n}^{(0)}$ and $\mathcal{R}_{S_n}^{(1)}$ respectively for the two actions; the transition matrices are denoted likewise. The relative cost-to-go function of state S_n and the average reward (i.e., time average utility) are denoted by $f(S_n)$ and $\hat{J}^*$ respectively. The terminal index is omitted for simplicity while one should note that the reward functions, transition matrices, and cost-to-go functions can all be different among terminals to reflect heterogeneous states and QoS, except for the transmission tax m, which is identical for all terminals.

There are two differences between TT and an exact Whittle's index policy. First, the scheduling decisions are centralized and deterministic in the Whittle's index policy, i.e., the index policy solves for the equivalent transmission tax for each state that makes the scheduling options of (11) equally good, and compares among terminals to find the one with the largest index. In contrast, the decentralized transmission strategy considered in our formulation is stochastic, which is necessary in distributed settings. Second, the Whittle's index approach seeks the maximum index (equivalent transmission tax) among terminals, while our approach lets all terminals share an identical m.

We also found in our experiments that when using the TT algorithm framework, the corresponding network needs to be trained for each transmission tax of each agent, and then tested in a multi-agent environment after training, to evaluate the reward resulting from the current m and finally seek out the suitable m through the golden search method. The downside of this is that as the number of terminals increases, the cost of training increases dramatically, so the training speed is extremely slow. At the same time, a serious consequence of both training and searching is that the final convergence results are unstable. The above two points indicate that the algorithm must be improved so as to make it more robust and adaptable to the multi-agent environment.

Specifically, the SMART scheme is divided into two parts: one is the generation of the database, and the other is the adjustment of transmission tax m. In stage 1, we input m and the state, and get an optimal transmission policy under the current transmission tax after training. At this stage we made an offline database, which trained for each m to obtain and save the corresponding network parameters Θ of DDPG, where the range of transmission tax m is chosen based on the empirical values if obtained in our previous work [1]. Since stage 1 had to trains M episodes for each m value, the formation of this database takes a long time. However, the benefit

is that it saves a lot of time for stage 2 training. In stage 2, we first set an initial transmission tax m_0 and read the corresponding network parameters Θ from the database prepared in stage 1. After that, the contention process is simulated and we seek out an optimal m. Specifically, all terminals deploy the current model parameters to participate in the multi-agent training phase after single-agent training, wherein each terminal would transmit with a probability calculated based on [38] if it senses the channel as idle and its instantaneous DDPG output is to transmit based on its current state. The FC feeds back an ACK/NACK after each data slot; the FC calculates the average reward (i.e., time-average utility) in each iteration and updates the transmission tax accordingly. If there are more than q_{fail} consecutive collisions, the transmission tax will be greatly increased; otherwise, it will be slightly reduced if there are more than q_{idle} consecutive idle times. The adjustment process will continue for a while until the average AoI is stable. In this paper, the terminal can only observe its own state – the degree of urgency, and determine the optimal strategy under the current state according to the transmission tax system. In our strategy, we want to maximize the expectation of future returns and predict not the flow of the system but the evolution of the state. We show the overall process in Alg. 1, wherein initialization part of the algorithm, N is the total number of terminals, ρ controls how many terminals contend in each iteration, and the final N_{target} is the number of contending terminals.

1) Details about state/action space: In RL, after an agent acts, the system will transit to a new state and give a reward. Subsequently, on the basis of the new state and the reward, the agent carries out a new action according to a certain strategy that is determined by the RL algorithm. The state space of the algorithm is defined as $\mathcal{S} = \{s_m, s_f\}$, where s_m reflects the utility at the terminal and s_f is the utility at the FC. The action output of DDPG is whether the terminal needs the contention channel to transmit information or not. Of particular note is that since its action space is continuous and one-dimensional, we define it as transmitting when the absolute value of action output is greater than 0.5 and remaining silent when it is less than 0.5. The reward indicates the cumulative utility of each terminal at the FC. Since one of the problems to be solved is to minimize the cumulative utility at the FC, the reward is designed as above. The entire framework is decentralized: as a consequence, we map the tradeoff between the utility at the terminal and the utility at the FC, which reflects the system state definition, to the transmit actions.

2) DDPG network framework and procedure: DDPG algorithm is an actor-critic algorithm [44], combining a policy-based approach with a value-based approach. In a policy-based ap-

Algorithm 1: SMART

Stage1: Decoupled Single-Agent Training

Initialization:

Terminals: Initialize model parameters $\mathbf{\Theta}_n$ $(n = 1, \cdots, N)$ following the normal distribution.

FC: Use m_{max} and m_{min} to denote the maximum and minimum transmission taxes, respectively. Set $m_{\text{min}} = 500$, $m_{\text{max}} = 10000$, $N_{\text{target}} = \rho N$, and $p_{\text{tx}} = \min\left\{\sqrt{\frac{2\delta}{T_s N_{\text{target}}^2}}, \frac{1}{N_{\text{target}}}\right\}$.

for $q = 1 : M$ **do**

  for $n = 1 : N$ **do**

    a) DDPG training for terminal-n to solve the MDP expressed in (11) with given m to update their model parameters $\mathbf{\Theta}_n$. The terminal's action can only be to transmit or not (stay silent).

    b) Save the mapping relationship between m and $\mathbf{\Theta}_n$ in the database.

  $m = m + m_{\text{i}}$

Stage2: Multi-Agent Training

Initialization:

Terminals: Initialize model parameters $\mathbf{\Theta}_n (n = 1, ..., N)$ $(n = 1, \cdots, N)$ corresponding to m_0 of stage1 training.

FC: Utilize m_0 to denote the initial transmission tax.

for $episode = 1 : T$ **do**

  for $n = 1 : N$ **do**

    if *Terminal-n senses the channel is idle and DDPG of terminal-n outputs transmit* **then**

      Terminal-n transmits with probability p_{tx} in this time slot.

    else

      Terminal-n stays silent in this time slot.

  $R_k =$ average utility for all terminals.

  if *Consecutive transmission attempts fail q_{fail} times* **then**

    $m = m + m_{\text{u}}$

  else if *The idle channel is sensed q_{idle} times in succession* **then**

    $m = m - m_{\text{d}}$

  else

    m remains unchanged and continues to the next episode.

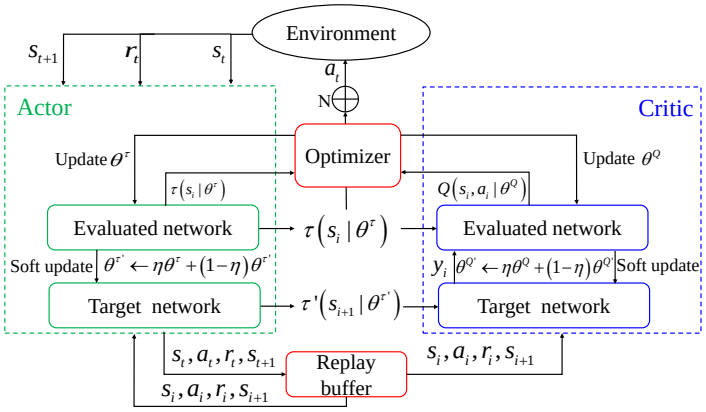


Fig. 4. Illustration of DDPG network framework and process. It is summarized as the process of iteratively training the network through the interaction of environment, actor network and critic network under the condition of a cyclic episode.

proach, the agent learns by interacting with the environment and directly adjusts its policy. In contrast, an agent adopting a value-based approach learns based on Q-functions and state values. As such, the actor-critic design allows the actor's neural network and the critic's neural network to learn according to their respective objective functions, resulting in faster convergence in comparison to other RL algorithms. The critic's neural network role is to approximate the value functions, while the actor's neural network's task is to approximate the policy function. We use θ^Q and θ^τ to parameterize function approximators. During the execution of the action, the actor network θ^τ applies the action a_t to the environment and gets an observation (state) s_t. During the training process, for each agent, the target network $\theta^{\tau'}$ of the actor network θ^τ will generate an action a_{t+1} and obtain the observation value s_{t+1}. Then, the critic network θ^Q is updated by minimizing the loss function as follows:

$$L(\theta^Q) = \frac{1}{R}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2, \tag{12}$$

where R is the batch size and $Q(.)$ is the Q function of the evaluated network. The target value is defined as

$$y_i = r_i + \gamma Q'(s_{i+1}, \tau'(s_{i+1}|\theta^{\tau'})|\theta^Q)^2, \tag{13}$$

where r_i denotes the cumulative utility of each terminal at the FC, γ is the discount factor, whose value range is $[0, 1]$, τ' and $Q'(.)$ are the policy function and the Q function of target network, respectively. At the same time, we optimize the actor network by maximizing the policy objective function J:

$$\nabla_{\theta^\tau} J \approx \frac{1}{R}\sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\tau(s_i)} \nabla_{\theta^\tau}\tau(s|\theta^\tau)|_{s_i}. \tag{14}$$

TABLE II
SMART NEURAL NETWORK COMPOSITION

Layers	Input	Hidden 1	Hidden 2	Output
Actor's network size	W_{s}	G_{s}	G_{s}	W_{a}
Critic's network size	$W_{\mathrm{s}}+W_{\mathrm{a}}$	G_{s}	/	1
Actor's network activation function	ReLU	ReLU	Tanh	ReLU
Critic's network activation function	ReLU	ReLU	/	ReLU

Then, the running average method is utilized to soft update the parameters of the evaluated network to the target network:

$$\theta^{\tau'} \leftarrow \eta\theta^{\tau} + (1-\eta)\theta^{\tau'}, \tag{15}$$

$$\theta^{Q'} \leftarrow \eta\theta^{Q} + (1-\eta)\theta^{Q'}. \tag{16}$$

In Fig. 4 we illustrate the data flow between the actor and the critic in the DDPG. DDPG is a deep learning algorithm and takes advantage of techniques introduced in the work proposed in the design of DQN [45]. More specifically, DDPG is off-policy, meaning that it is trained using a randomly selected batch R of past experiences saved in its memory, referred to as the replay buffer. Additionally, DDPG relies on the use of target and evaluation neural networks combined with batch normalization to stabilize the learning process.

3) Network architecture: For illustration purposes, we describe the network architecture in Tab. II, where G_{s} denotes the number of neurons in a hidden layer, W_{s} is the dimension of the state space, and W_{a} represents the dimension of action space. The actor network contains two fully-connected hidden layers, and the critic network contains one fully-connected layer. The actor outputs a specific action, while the critic network outputs a specific Q value.

IV. SIMULATION RESULTS

In this section, we test the performance of our proposed SMART scheme using simulation. We compare SMART against conventional MAC schemes (TDMA and CSMA based) and a MARL approach in a multi-terminal scenario, all presented in the first subsection. In the following subsection, we focus on evaluating SMART performance using the AoI metric. In the third subsection, we analyze the trade-offs between SMART and a conventional MARL approach in the aspects of complexity, convergence speed, runtime, variance and mean value performance.

TABLE III

SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
Number of agents N	30	Tax interval m_{i}	20
Packet arrival rate λ	0.1 [packet/ms]	Initial tax m_0	1500
Energy arrival rate λ_{e}	0.2 [packet/ms]	Magnitude of increase m_{u}	220
Episodes in stage1 M	300000	Magnitude of decrease m_{d}	40
Episodes in stage2 T	2000	Threshold of fail times q_{fail}	5
Length of a data slot T_{s}	1.5 [ms]	Threshold of idle times q_{idle}	3
Length of a mini-slot δ	0.01 [ms]	Number of neurons in SMART network G_{s}	30
Control contention ρ	0.03	Number of neurons in conventional MARL network G_{m}	75

Finally, in the last, fourth subsection, we assess the performance of SMART in a scenario with multiple terminals with different QoS demands.

The empirical parameters we use throughout our simulation are shown in Tab. III. In Tab. IV we list hyperparameters in the neural network and the test environment we used to implement the DDPG. The computational complexity of the DDPG algorithm is an important issue, and in our actual experiment, we also find that different library versions could affect the results. Therefore, the test environment must be consistent when comparing the performance of various algorithms. We train the SMART in two stages. In stage-1, we train 3×10^5 episodes to prepare the database, and test with 2000 episodes in stage-2. The packet arrival rate is 0.1 packets/ms and the length of a mini-slot is 0.01 ms. Assuming that the size of the energy buffer is 1, the energy arrival rate is 0.2 packets/ms and the arrival follows the Poisson process.

A. Conventional MAC and MARL Schemes for Comparisons

RR-ONE: In our previous work [31], we have demonstrated that in the set of Arrival-Independent and Renewal (AIR) policies, a round-robin policy with one-packet (latest packet only and others are dropped) buffers (RR-ONE) is the optimal strategy for minimizing time-average AoI. Such an RR-ONE scheme is essentially a TDMA scheme, i.e., each terminal occupies a dedicated time slot and takes turns in sending an update. In addition, RR-ONE is asymptotically optimal in all strategies, in the sense that it can achieve the optimal scaling factor in the regime of a large number of terminals. The average AoI achieved by RR-ONE can be

TABLE IV

DDPG HYPERPARAMETERS AND TESTING ENVIRONMENT

	Hyperparameter / Item	Value / Version	Hyperparameter / Item	Value / Version
SMART	Learning rate for actor L_{A}	10^{-3}	Learning rate for critic L_{C}	2×10^{-3}
	Discount factor γ	0.9	Soft replacement factor η	10^{-2}
	Memory capacity B	10^4	Batch size R	32
	Optimizer	Adam		
Conventional MARL	Learning rate for actor L_{A}	10^{-3}	Learning rate for critic L_{C}	10^{-3}
	Discount factor γ	0.99	Soft replacement factor η	10^{-3}
	Memory capacity B	10^5	Batch size R	512
	Optimizer	Adam		
Test Environments	Python	v3.6.9	Pytorch	v0.4.1
	CPU	Intel(R) Core(TM) i5-7500 CPU @3.40 GHz	Tensorflow	v1.14.0

derived in closed-form as

$$\bar{h}_{\mathrm{AoI,RR}}^{(\infty,N)} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{\lambda_n} + \frac{N-1}{2}, \tag{17}$$

where N is the number of terminals and λ_n is the status packet arrival rates of terminal-n.

p-persistent CSMA: In p-persistent CSMA, when a terminal n wishes to transmit data, it first listens for the channel. If the channel is busy, the terminal keeps listening to the next contention mini-slot. If the channel is free, it transmits with probability p_n (selected to be $p_n = \frac{1}{N}$ in this work [38]) and delays to the next contention mini-slot with probability $1 - p_n$.

We also designed a ***conventional MARL*** approach as a baseline against which to assess the benefits of employing our proposed SMART solution. We designed the conventional MARL approach using the same RL algorithm (DDPG [44]) as in SMART to make the comparison fair. The selected state space consists of the three most important aspects of the environment for the terminal. The first state is the AoI value of the terminal's status update (the time since the terminal generated a new status update). The second state is the AoI value of the terminal's status update in the FC (the time elapsed since the terminal's successfully transmitted status update was generated). The third state captures the success of the last transmission attempt by the terminal, i.e., a collision flag. The third state indirectly represents the terminal interplay with other terminals in the environment. The action the terminal takes represents the transmission probability at a given time instant and is a real value $p_n \in [0, 1]$. The terminal can successfully

TABLE V
CONVENTIONAL MARL NEURAL NETWORK COMPOSITION

Layer	Input	Hidden 1	Hidden 2	Hidden 3	Hidden 4	Output
Actor's network size	W_{ms}	G_{m}	G_{m}	G_{m}	$1/3G_{\text{m}}$	W_{ma}
Actor's activation function	ReLU	ReLU	ReLU	ReLU	ReLU	Tanh
Critic's network size	W_{ms} + W_{ma}	G_{m}	G_{m}	G_{m}	$1/3G_{\text{m}}$	1
Critic's activation function	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU

transmit a status update only when it is the only terminal in the system to do so. Terminals have to learn to cooperate and avoid acting greedily, i.e., trying to transmit a new update at every opportunity. The latter turns out to be particularly problematic as the number of agents present in the system increases, as we verified during our simulations. The terminal obtains the reward based on the state space. More specifically, the reward depends on the AoI value of the terminal's status update stored in the FC. The higher the AoI at the FC, the lower the reward is.

Deep-reinforcement Learning Multiple Access (DLMA): Finally, we also compared the method proposed in the [22], and made appropriate modifications to the environment. The framework is similar to the conventional MARL approach, where the network changes from a traditional full connection to a Q neural network (QNN) with the first two hidden layers fully connected, followed by two ResNet blocks. The setting of parameters is consistent with the work of [22].

We implemented this conventional MARL approach using PyTorch [46], a Python library for deep learning. In Tab. V, we list details of the neural network structure we used in our design. Note that W_{ma} denotes the dimension of the agent's action, W_{ms} denotes its state dimension, and G_{m} represents the number of neurons in the layer. Additionally, between each hidden layer, we applied a dropout layer, to prevent over-fitting. We also added batch normalization after activation for the first hidden layer to improve the learning speed. As we demonstrate in Section IV, terminals employing the conventional MARL approach can learn to cooperate to decrease the AoI of status updates collected at the FC. Unfortunately, the policy agents' learning is far from optimal.

Remark 1 (Comparison of runtime): SMART requires significantly less computational power to operate than conventional MARL. In SMART, a terminal has to adjust only m, which it obtains from the database generated offline, i.e., Stage 1 in Alg. 1. In the conventional MARL

scheme, the terminal is required to train continually; otherwise, its performance will deteriorate over-time. Consequently, the simulation runtime is much shorter for SMART. In contrast, RR-ONE and CSMA schemes require no learning processes and thus have much lower computational overhead.

Remark 2 (Comparison of stability): In most cases, when using SMART, the terminals achieve lower average AoI than in the conventional MARL scheme. However, the variation in the obtained average AoI per episode is smaller in the case of conventional MARL. This drawback of SMART stems from the fact that the offline training of SMART is more sensitive to online system changes. The variation comparison indicates that conventional MARL is more stable than SMART. It appears that when terminals adopt the conventional MARL scheme, they sacrifice performance to achieve better stability.

Remark 3 (Comparison of scalability): Due to the fact that SMART adopts an offline training stage that pre-trains the DDPG networks such that they are stable models in the online training stage, it can maintain good performance even when there are many terminals and traditional MARL methods have difficulty in converging. SMART is more scalable and, as we show in the next section, with an increase in the number of agents, the performance degrades more slowly than that of the conventional MARL approach.

It is worth stating that we believe that these three can represent the state-of-the-art MAC schemes. In our previous work [31], we have shown that among a set of independent arrival and update (AIR) strategies, the RR-ONE scheme is the best strategy to minimize the time-averaged AoI. The CSMA protocol represents a traditional algorithm that improves on ALOHA by adding carrier sense. At the same time, p-persistent CSMA mediates a compromise between reducing conflicts such as non-persistent CSMA and reducing channel idle time with 1-persistent CSMA. Finally, the conventional MARL scheme is used in the field of reinforcement learning, which converges quickly and smoothly.

B. Comparison of SMART, Conventional MARL, RR-ONE, and CSMA Schemes in average AoI

To assess the performance of SMART, we first test it in a scenario of 30 terminals and vary terminals' packet arrival rate. As we show in Fig. 5, we observe that both the SMART and conventional MARL schemes proposed are superior to DLMA algorithm and traditional p-persistent CSMA. Overall, SMART performs better than conventional MARL in terms of average AoI, and conventional MARL is still far from reaching the asymptotically optimal lower bound

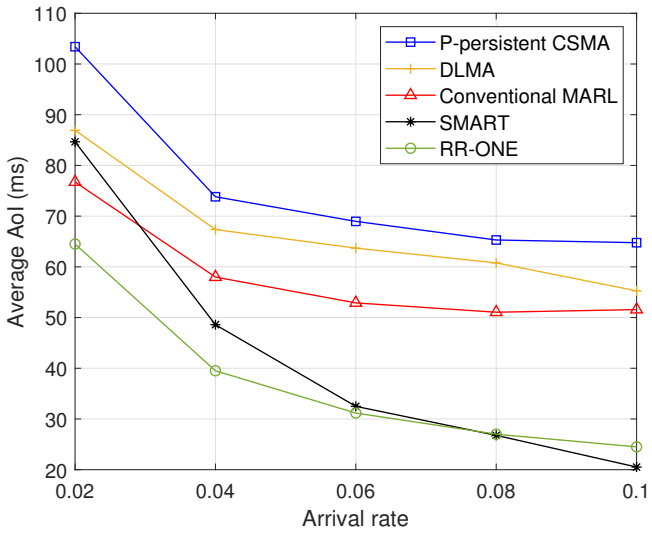
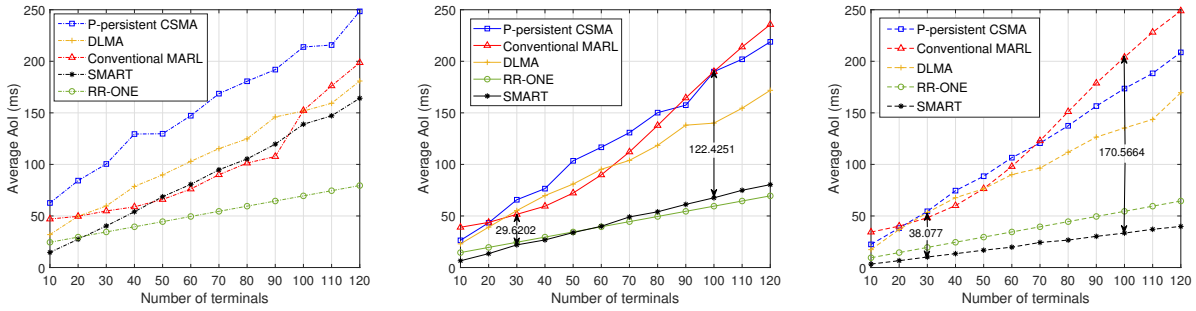


Fig. 5. Performance evaluations for AoI with different arrival rates.



(a) $\lambda = 0.05$ (b) $\lambda = 0.1$ (c) $\lambda = 0.2$

Fig. 6. Average AoI as a function of the number of terminals when (a) the arrival rate is 0.05 packets/ms; (b) the arrival rate is 0.1 packets/ms; (c) the arrival rate is 0.2 packets/ms.

achieved by RR-ONE. It is of special note that when the arrival rate is low, there is a gap of about 20 ms between the proposed SMART scheme and the asymptotically optimal RR-ONE. As the arrival rate gradually increases, the gap shrinks, and eventually SMART outperforms RR-ONE in terms of average AoI.

Fig. 6 presents the average AoI as a function of the number of terminals, for different arrival rates. The three subfigures from left to right correspond to the results for arrival rates of 0.05 packets/ms, 0.1 packets/ms and 0.2 packets/ms, respectively. As can be seen from Fig. 6(b), for a low number of terminals, the average AoI achieved by the traditional CSMA scheme, DLMA scheme and conventional MARL scheme is similar to the proposed scheme, but as the number of terminals increases, the performance deteriorates for all. This also highlights the disadvantages of traditional solutions in high-density scenarios. The conventional MARL scheme

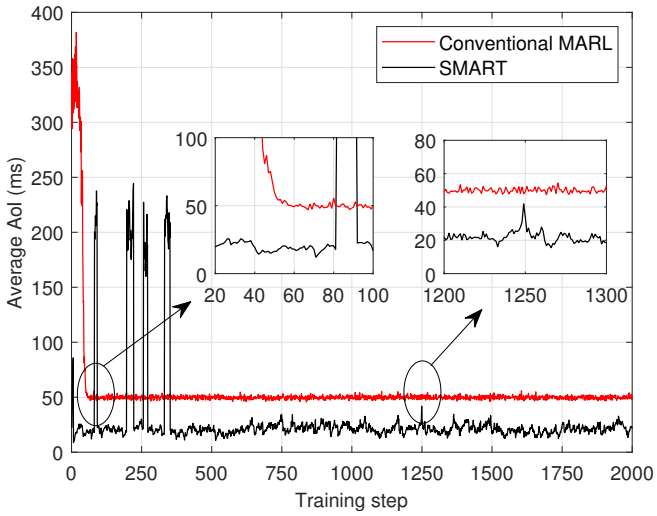


Fig. 7. Training results of the SMART and conventional MARL schemes. The number of agents is 30, the packet arrival rate is 0.1 packets/ms, and the length of a mini-slot is 0.01 ms.

and DLMA schemes perform better than the traditional p-persistent CSMA scheme for a low arrival rate, but their performance is comparable when the arrival rate increases. For arrival rates of $\lambda = 0.1$ and above, SMART outperforms RR-ONE. Such a result is expected because while RR-ONE is an asymptotically optimal strategy for minimizing the average AoI in AIR policies, but it is not optimal under other policies or QoS. An additional limitation of RR-ONE is that terminals can only access a slot in a cyclical order, which means, that terminals can't prioritize access. In contrast, SMART can take advantage of terminals' situation; thus, it can deliver better performance in the conditions presented in our work.

The long-time average AoI of SMART is tested in comparison with the conventional MARL approach which utilizes DDPG to train and feedback the entire contention process. The packet arrival rate is 0.1 packets/ms, the length of a mini-slot is 0.01 ms, and the number of agents is 30 in Fig. 7, and the plot shows the running average AoI over time. It is observed that the conventional MARL scheme converges rapidly, and the average AoI reaches about 50 ms after 50 training steps. SMART requires more training steps for convergence, and the AoI has a larger variation after convergence. The tradeoff is that the average AoI after convergence is about 20 ms for SMART, about 30 ms lower than that achieved by a conventional MARL approach.

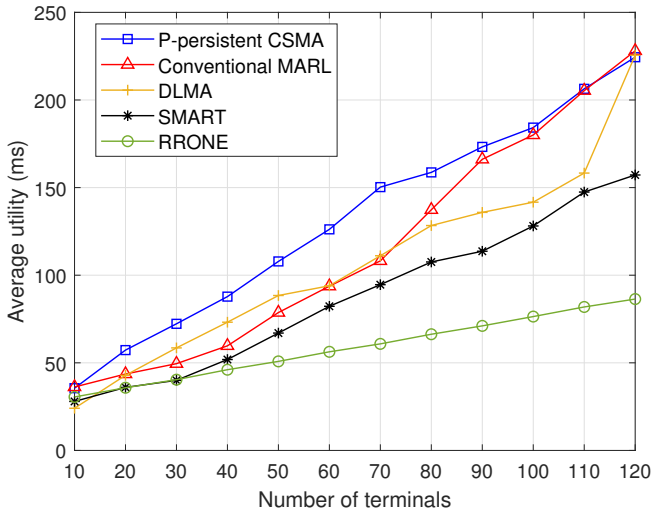


Fig. 8. Average utility when terminals have variable arrival rates.

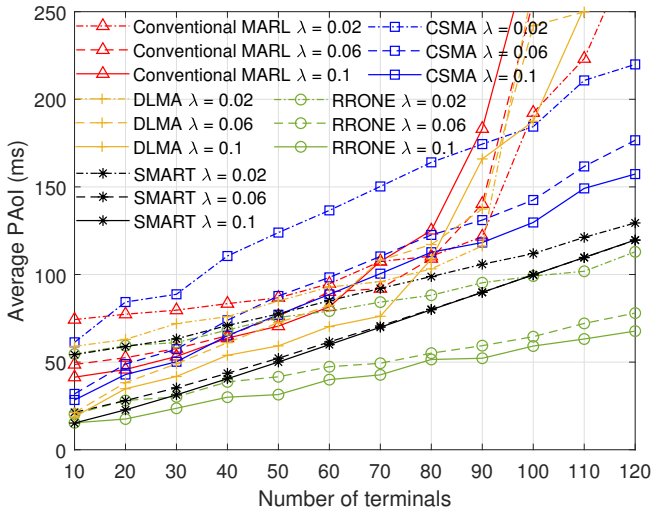


Fig. 9. Average PAoI with a different number of terminals and different arrival rates.

C. Scenarios with Variable Arrival Rates and Different QoS Requirements

We apply the schemes to a scenario with variable arrival rates for each terminal, whose value is drawn from a uniform distribution in $[0, 0.1]$. Fig. 8 shows that the three schemes that employ RL can learn and adapt to the arrival rate variation among terminals by employing the deep neural network. The overall performance of the three is better than p-persistent CSMA, and SMART outperforms the conventional MARL scheme and DLMA scheme. In this case, RR-ONE outperforms due to the fact that RR-ONE utilizes every time slot, while SMART has more idle slots at low arrival rates.

We show the impact of the different number of terminals and different arrival rates on

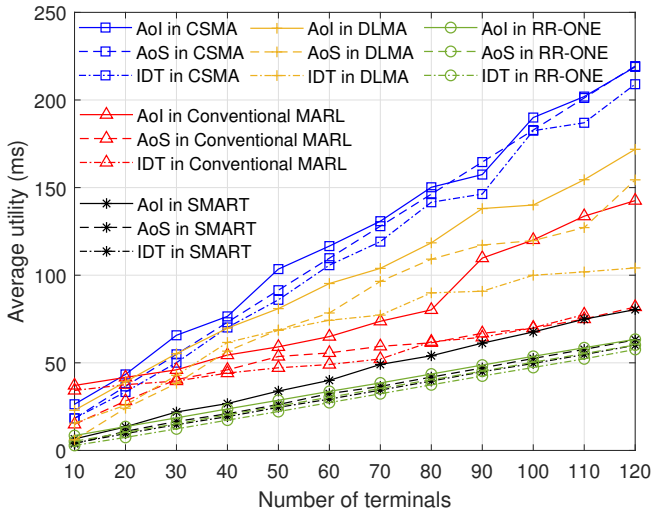


Fig. 10. Performance evaluations for diversified QoS requirements. It illustrates that the proposed SMART scheme is adaptable for heterogeneous QoS requirements.

the average PAoI for various schemes in Fig. 9. The results show that SMART yields the lowest minimal PAoI compared to the conventional MARL, DLMA and p-persistent CSMA schemes. RR-ONE also exhibits optimality in this case. Additionally, SMART is more robust than conventional MARL and DLMA as the conventional MARL and DLMA performance deteriorates significantly once more than 80 terminals compete for the same channel. The low PAoI indicates that the proposed SMART approach enables all terminals equal access to the transmission channel. An important quality to enable timely control to services relying on the collected information.

In Fig. 10, the average utility—which in this case includes AoI, AoS and Energy Harvesting (EH) sensors that are optimizing the IDT (described in details in Section II) and each type contains the same number of terminals—shows that the proposed SMART scheme is scalable and applicable to heterogeneous QoS requirements. The arrival rate is 0.1 packets/ms, the energy arrival rate is 0.2 packets/ms and the energy buffer size is 1 for the EH sensor. The traditional p-persistent CSMA, DLMA and conventional MARL schemes do not perform well compared to SMART. SMART outperforms all, while the DLMA is superior to the conventional MARL and the conventional MARL is better than p-persistent CSMA when the number of terminals is low.

In summary, SMART and conventional MARL virtually have the same computational complexity and SMART can achieve better average performance in general, whereas conventional MARL performs better in terms of convergence speed, stability and storage space. However, the

TABLE VI

COMPARISON OF RUNTIME AND STORAGE SPACE

	Runtime	Storage space
SMART	22.908 s	50.9 MB
Conventional MARL	6265.923 s	55.7 KB

1. Runtime simulation is completed under the condition that the number of terminals is 30, the arrival rate is 0.1 packet/ms, and the episode is 2000.

2. Occupied memory by the scheme contains the basic code and the database, where $m \in [500, 10000], m = 520, 540, 560, ..., 10000$.

TABLE VII

COMPARISON OF COMPLEXITY

	Additions	Multiplications	Total
SMART	$2MNG_{\mathrm{s}} + TNG_{\mathrm{s}} + M + T + 1$	$MN(W_{\mathrm{s}}G_{\mathrm{s}} + G_{\mathrm{s}}^2 + W_{\mathrm{a}}G_{\mathrm{s}}) + 2TNG_{\mathrm{s}}$	$\mathcal{O}(MNG_{\mathrm{s}}^2)$
Conventional MARL	$\frac{10}{3}L_{\mathrm{o}}L_{\mathrm{i}}G_{\mathrm{m}} + L_{\mathrm{i}} + 1$	$L_{\mathrm{o}}L_{\mathrm{i}}(W_{\mathrm{ms}}G_{\mathrm{m}} + \frac{8}{3}G_{\mathrm{m}}^2 + \frac{1}{3}N_{\mathrm{m}}W_{\mathrm{ma}}) + L_{\mathrm{o}}$	$\mathcal{O}(L_{\mathrm{o}}L_{\mathrm{i}}G_{\mathrm{m}}^2)$

longer training process for SMART is a fair trade-off as it outperforms conventional MARL in runtime performance and scalability.

V. ANALYSIS RESULT

In this section, we analyze the runtime, complexity, and variance of the algorithms for a more comprehensive comparison.

Runtime: In order to evaluate the difference in the runtime, we conduct an experiment on the time required by the two schemes to run 2000 episodes. We set the same environment parameters for both schemes, i.e., the number of terminals is 30 and the arrival rate to 0.1 packet/ms. From Tab. VI, we can observe that under the same condition, the runtime of conventional MARL is about 273 times that of the SMART. This is due to the fact that in SMART, Stage 1 of the database has been completed for most of the network training tasks. In order to store a database with $m \in [500, 10000]$, a large amount of storage space is required, and during the memory-reading process, there is memory consumption. Therefore SMART occupies about $1,000$ times as much storage as conventional MARL. In other words, SMART trains the majority of its neural network offline, which sacrifices a part of the storage capacity in exchange for faster runtime.

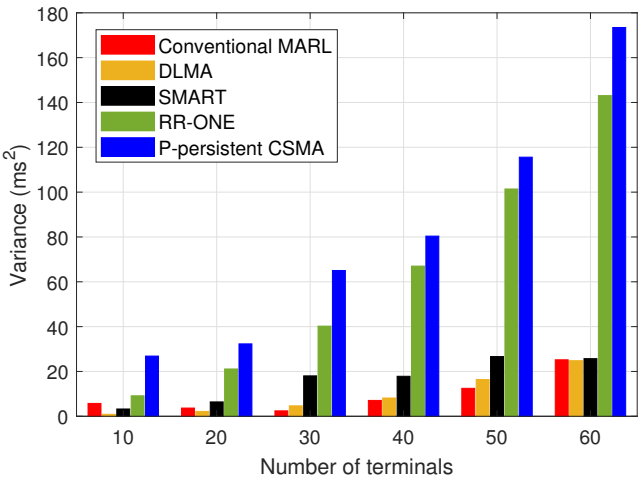


Fig. 11. Variance of AoI for different schemes.

Complexity: We analyze the algorithmic complexity of the two schemes. In a fully-connected network, the output of a neuron can be formulated as

$$f\left(\sum_{j}\omega_j x_j + b\right), \tag{18}$$

where f is the activation function, ω_j is the weight and x_j is the input of the neuron-j respectively. The bias is represented by b. We calculate the complexity of addition and multiplication separately and get a relatively simple mathematical expression. In Section III, Tab. II specifies the network architecture of DDPG in SMART, and Alg. 1 describes the overall process of the algorithm. The network structure for conventional MARL is illustrated in Tab. V. We represent the number of outer loops and inner loops of the conventional MARL code as L_{o} and L_{i}, respectively. From the above, we summarize the complexity of the two schemes in Tab. VII. As a result, the two schemes are comparable in terms of algorithm complexity because the two schemes every computational task use the same DRL framework.

Variance: Next, we analyze the variance of the converged values of AoI under the different tested schemes and plotted in Fig. 11. As the number of terminals increases, the variance of the AoI achieved by the traditional p-persistent CSMA scheme and the RR-ONE scheme increases significantly. At the same time, the variances of the AoI for the conventional MARL, DLMA and SMART schemes fluctuate within a much smaller range.

VI. Conclusions and Future Work

In this paper, we have proposed a scalable and robust MARL-based framework for decentralized wireless access with general system states and QoS metrics. The framework is applicable to many practical scenarios that demand situationally-aware status-based packet transmissions. A new, common feedback-based distributed learning scheme, SMART, is designed to tackle the slow and non-scalable convergence issue in MARL. Compared with a state-of-the-art DDPG-based MARL scheme, SMART achieves consistently stable convergence to a better solution with hundreds of terminals (in a single channel), at the expense of increased storage space to store the DDPG network parameters. Based on extensive simulations, it is shown that SMART outperforms conventional CSMA and state-of-the-art MARL schemes significantly when metrics involving status-based packet transmissions are considered.

The proposed scheme is versatile, since the state space and reward function, i.e., QoS metrics under consideration, are quite general. However, the design with a common feedback signal, which is vital in our design to stabilize the MARL convergence, puts restrictions on the action space, that is the action should be binary, e.g., transmit or not in this paper. It remains interesting to design novel solutions for MARL with a more complicated action space. This is left for future work.

References

[1] Z. Jiang, A. Marinescu, L. DaSilva, S. Zhou, and Z. Niu, "Scalable multi-agent learning for situationally-aware multiple-access and grant-free transmissions," in *IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2019, pp. 1–6.

[2] G. P. Fettweis, "The Tactile Internet: Applications and challenges," *IEEE Veh. Tech. Mag.*, vol. 9, no. 1, pp. 64–70, Mar 2014.

[3] J. Zhang, L. Lu, Y. Sun, Y. Chen, J. Liang, J. Liu, H. Yang, S. Xing, Y. Wu, J. Ma, I. B. F. Murias, and F. J. L. Hernando, "PoC of SCMA-based uplink grant-free transmission in UCNC for 5G," *IEEE J. Select. Areas Commun.*, vol. 35, no. 6, pp. 1353–1362, Jun 2017.

[4] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *IEEE INFOCOM*, Mar 2012, pp. 2731–2735.

[5] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *Proc. 29th IEEE Conf. Decision Control*, Dec 1990, pp. 2130–2132 vol.4.

[6] J. Ni, B. Tan, and R. Srikant, "Q-CSMA: Queue-length-based CSMA/CA algorithms for achieving maximum throughput and low delay in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 825–836, Jun 2012.

[7] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[8] J. Cheng, C. Lee, and T. Lin, "Prioritized random access with dynamic access barring for RAN overload in 3GPP LTE-A networks," in *Proc. IEEE GLOBECOM Workshops*, Dec 2011, pp. 368–372.

[9] R. Cheng, J. Chen, D. Chen, and C. Wei, "Modeling and analysis of an extended access barring algorithm for machine-type communications in LTE-A networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 2956–2968, Jun 2015.

[10] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor, "IEEE 802.11e wireless LAN for quality of service," in *Proc. European Wireless*, vol. 2, 2002, pp. 32–39.

[11] S. Lien, S. Shieh, Y. Huang, B. Su, Y. Hsu, and H. Wei, "5G new radio: Waveform, frame structure, multiple access, and initial access," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 64–71, Jun 2017.

[12] G. Scheible, D. Dzung, J. Endresen, and J. E. Frey, "Unplugged but connected [design and implementation of a truly wireless real-time sensor/actuator interface]," *IEEE Ind. Electron. Mag.*, vol. 1, no. 2, pp. 25–34, Summer 2007.

[13] PNO. (2012). WSAN Air Interface Specification, [Online].Available: http://www.profibus.com/nc/download/installation-guide/downloads/wsan-air-interface-specification/display/.

[14] IO-Link Community, Sep. 2017. IO-Link WirelessSystem Extensions, Specification 10.112, [Online].Available: http://www.io-link.com.

[15] 3GPP TR 36.881 v0.6.0, "Evolved Universal Terrestrial Radio Access (E-UTRA); Study on Latency Reduction Techniques for LTE (Release 13).

[16] Z. Ning, K. Zhang, X. Wang, L. Guo, X. Hu, J. Huang, B. Hu, and R. Y. K. Kwok, "Intelligent edge computing in internet of vehicles: A joint computation offloading and caching solution," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.

[17] Z. Ning, K. Zhang, X. Wang, M. S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, B. Sadoun, and R. Y. K. Kwok, "Joint computing and caching in 5g-envisioned internet of vehicles: A deep reinforcement learning-based traffic control system," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[18] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[19] N. Abuzainab, T. Erpek, K. Davaslioglu, Y. E. Sagduyu, Y. Shi, S. J. Mackey, M. P. Patel, F. Panettieri, M. A. Qureshi, V. Isler, and A. Yener, "Qos and jamming-aware wireless networking using deep reinforcement learning," *CoRR*, vol. abs/1910.05766, 2019. [Online]. Available: http://arxiv.org/abs/1910.05766

[20] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, Fourthquarter 2019.

[21] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, Jan 2019.

[22] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1277–1290, June 2019.

[23] Zhenzhen Liu and I. Elhanany, "Rl-mac: A qos-aware reinforcement learning based mac protocol for wireless sensor networks," in *2006 IEEE International Conference on Networking, Sensing and Control*, 2006, pp. 768–773.

[24] Y. Chu, S. Kosunalp, P. D. Mitchell, D. Grace, and T. Clarke, "Application of reinforcement learning to medium access control for wireless sensor networks," *Engineering Applications of Artificial Intelligence*, vol. 46, pp. 23 – 32, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0952197615001827

[25] Z. Jiang, S. Fu, S. Zhou, Z. Niu, S. Zhang, and S. Xu, "AI-assisted low information latency wireless networking," *IEEE Wireless Commun.*, 2020.

[26] J. Zhong, R. D. Yates, and E. Soljanin, "Two freshness metrics for local cache refresh," in *IEEE Int'l Symp. Info. Theory*, Jun 2018, pp. 1924–1928.

[27] B. Yin, S. Zhang, Y. Cheng, L. X. Cai, Z. Jiang, S. Zhou, and Z. Niu, "Only those requested count: Proactive scheduling policies for minimizing effective age-of-information," in *IEEE INFOCOM*, April 2019.

[28] X. Guo, R. Singh, P. R. Kumar, and Z. Niu, "A risk-sensitive approach for packet inter-delivery time optimization in networked cyber-physical systems," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1976–1989, Aug. 2018.

[29] Z. Jiang, B. Krishnamachari, S. Zhou, and Z. Niu, "Can decentralized status update achieve universally near-optimal age-of-information in wireless multiaccess channels?" in *Proc. Int. Teletraffic Congr. ITC*, Sep 2018.

[30] Z. Jiang, B. Krishnamachari, X. Zheng, S. Zhou, and Z. Niu, "Decentralized status update for age-of-information optimization in wireless multiaccess channels," in *IEEE Int'l Symp. Info. Theory*, 2018.

[31] Z. Jiang, B. Krishnamachari, X. Zheng, S. Zhou, and Z. Niu, "Timely status update in wireless uplinks: Analytical solutions with asymptotic optimality," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3885–3898, April 2019.

[32] Z. Jiang, S. Zhou, Z. Niu, and Y. Cheng, "A unified sampling and scheduling approach for status update in wireless multiaccess networks," in *IEEE INFOCOM*, April 2019, pp. 1–9.

[33] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *arXiv preprint arXiv:1801.01803*, 2018.

[34] Y.-P. Hsu, "Age of information: Whittle index for scheduling stochastic arrivals," in *IEEE Int'l Symp. Info. Theory*, 2018.

[35] Q. He, D. Yuan, and A. Ephremides, "Optimal link scheduling for age minimization in wireless systems," *IEEE Trans. Inform. Theory*, in press.

[36] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff, "Age-optimal sampling and transmission scheduling in multi-source systems," *arXiv preprint arXiv:1812.09463*, 2018.

[37] R. D. Yates and S. K. Kaul, "The age of information: Real-time status updating by multiple sources," *IEEE Trans. Inform. Theory*, pp. 1–1, 2018.

[38] Y. Gai, S. Ganesan, and B. Krishnamachari, "The saturation throughput region of p-persistent CSMA," in *Proc. ITA Workshop*, Feb 2011, pp. 1–4.

[39] M. Costa, M. Codreanu, and A. Ephremides, "On the age of information in status update systems with packet management," *IEEE Trans. Inform. Theory*, vol. 62, no. 4, pp. 1897–1910, April 2016.

[40] L. Huang and E. Modiano, "Optimizing age-of-information in a multi-class queueing system," in *IEEE Int'l Symp. Info. Theory*, Jun 2015, pp. 1681–1685.

[41] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[42] L. Peshkin, K.-E. Kim, N. Meuleau, and L. P. Kaelbling, "Learning to cooperate via policy search," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2000, pp. 489–496.

[43] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1.

[44] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[45] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari With Deep Reinforcement Learning," *arXiv preprint arXiv:1312.5602*, Dec. 2013.

[46] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.