A Distributed Abstract MAC Layer for Cooperative Learning on Internet of Vehicles

Yifei Zou, *Member, IEEE*, Zuyuan Zhang, Congwei Zhang, Yanwei Zheng, *Member, IEEE*, Dongxiao Yu, *Senior Member, IEEE*, and Jiguo Yu, *Fellow, IEEE*

***Abstract*— This paper addresses the problem of reliable communications for cooperative learning on Internet-of-Vehicles, where a large amount of data from users and services needs to be processed. Previous works have proposed various cooperative learning schemes, but they often assume that the communications between vehicles are reliable, without considering how to achieve this in an Internet-of-Vehicles network. This paper is the first one that implements an abstract MAC layer using a distributed deep reinforcement learning scheme, which can directly meet the reliable communication requirements of cooperative learning in previous works. Our abstract MAC layer performs two operations: *acknowledgement*, which makes sure that all vehicles can successfully broadcast their messages to all of their neighbors, and *progress*, which ensures that each vehicle can receive at least one message from its neighbors. These operations facilitate vehicles to exchange and update their training models in a cooperative learning service. Our simulation results show the efficiency and fairness of our deep reinforcement learning abstract MAC layer.**

***Index Terms*— Abstract MAC layer, cooperative learning, Internet of Vehicles, deep reinforcement learning.**

I. Introduction

EDGE intelligence has attracted lots of research attention in recent years, and many cooperative learning methods have been developed to support various applications in the Internet-of-Vehicles (IoV) networks [1], [2], most of which rely on reliable vehicle-to-vehicle communications, such as the device-to-device communication model seen in [3]. However, these approaches overlook latency and bandwidth constraints, which clash with the often unreliable communication environment in wireless IoV networks. For example, the algorithm in [4] optimizes federated edge learning via joint data selection and resource allocation, assuming instantaneous and reliable communication. Similarly, edge-assisted federated learning in [5] reduces the burden and delay of federated learning, assuming stable communication without delays. These cooperative learning algorithms heavily lean on dependable communication assumptions. This raises a crucial question for IoV cooperative learning: *how can we ensure stable and efficient communications in IoV networks?* This way, the current cooperative learning methods could be directly applicable.

In this paper, we address this question by introducing an abstract MAC (absMAC) layer tailored for cooperative learning within IoV. The concept of the absMAC layer was initially introduced by Kuhn et al. in 2009 [6], serving as a means of ensuring reliable communications among devices. This layer encompasses two crucial operations: *acknowledgement* and *progress*. The acknowledgement (ack. for short) operation signifies the successful broadcast of messages to all neighboring devices, while the progress (prog. for short) operation ensures that all nodes receive at least one message from their neighboring devices. The parameters f_{ack} and f_{prog} denote the timing constraints for accomplishing these ack. and prog. operations, respectively. The integration of the absMAC layer offers significant benefits for devising and analyzing diverse cooperative learning strategies. This framework facilitates efficient exchange of learning parameters among agents, enabling them to locally update their training models. With the inclusion of the prog. operation, each learning agent receives at least one message from its neighbors, leading to the update of its local parameters at least once. On the other hand, the ack. operation guarantees that each agent's local parameters are shared with all neighboring agents, thereby mitigating the challenge of isolated data islands.

Designing an efficient absMAC layer for IoV networks involves addressing two critical challenges. Firstly, IoV networks are inherently distributed, while optimizing the absMAC layer efficiency requires a global perspective. Overcoming this challenge within a limited vehicle knowledge framework is the primary concern. Secondly, diverse scenarios bring variations in data distribution, high-level application information scheduling requirements, and the communication model of cooperative learning. As a result, the absMAC layer must exhibit the necessary flexibility. Notably, existing works on absMAC layer implementation in wireless settings tend to be centralized and lack specific tailoring to ensure dependable communication for IoV's cooperative learning.

In this paper, we address the above two issues by designing a distributed deep reinforcement learning (DRL) method after

Manuscript received 15 July 2022; revised 19 August 2023; accepted 18 January 2024. Date of publication 19 February 2024; date of current version 1 August 2024. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62102232, Grant 62122042, and Grant 61971269; and in part by the Shandong Science Fund for Excellent Young Scholars under Grant 2023HWYQ-007. The Associate Editor for this article was Z. Xiao. *(Corresponding author: Yanwei Zheng.)*

Yifei Zou, Zuyuan Zhang, Congwei Zhang, Yanwei Zheng, and Dongxiao Yu are with the Institute of Intelligent Computing, School of Computer Science and Technology, Shandong University, Qingdao 266237, China (e-mail: yfzou@sdu.edu.cn; zyzhang2000@mail.sdu.edu.cn; xczhw@mail.sdu.edu.cn; zhengyw@sdu.edu.cn; dxyu@sdu.edu.cn).

Jiguo Yu is with the Big Data Institute, Qilu University of Technology (Shandong Academy of Sciences), Jinan, Shandong 250353, China, and also with the Shandong Laboratory of Computer Networks, Jinan 250014, China (e-mail: jiguoyu@sina.com).

Digital Object Identifier 10.1109/TITS.2024.3362909

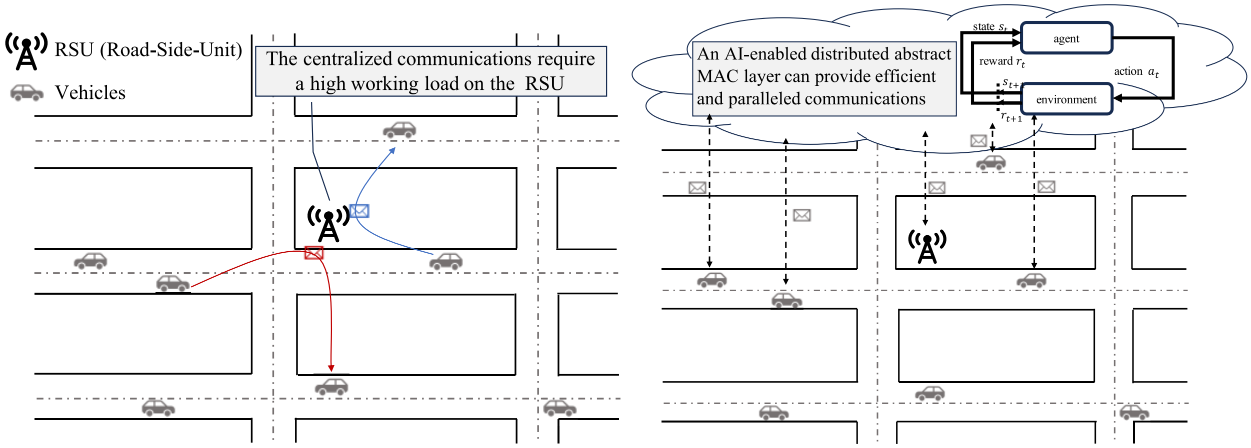


Fig. 1. The classical communication mode (left side) and our AI-enabled abstract MAC layer (right side).

fully understanding the efficiency and flexibility features of the communications in IoV networks. Then, based on the above DRL scheme, a distributed algorithm is presented to implement our absMAC layer for cooperative learning. In our algorithm, each vehicle independently makes decisions according to its local information while finally an absMAC layer with global acknowledgement and progress operations are obtained. Compared with the classical communication mode that relies on the Road-Side-Unit to provide reliable communications, our AI-enabled absMAC layer can provide efficient and paralleled reliable communications, as is illustrated in Fig. 1. Besides, different from the previous absMAC layer with inflexible parameters, by setting the states and rewards closely related to the efficiency and priority of the communications, the newly designed deep reinforcement learning method dramatically strengthens the efficiency and flexibility of our algorithm when facing various scenarios in reality. The detailed contributions of our paper are summarized in the following:

A. Contribution

To the best of our knowledge, this paper is the first one implementing the absMAC layer specifically for cooperative learning in the distributed Internet-of-Vehicles networks. By designing a deep reinforcement learning scheme, in which the states and rewards are closely related to the efficiency and priority of the communications in the wireless channel, our absMAC layer is more efficient and flexible than the previous solutions. Besides, an application of our absMAC layer for cooperative learning in IoV is presented. Extensive simulations are conducted to evaluate the performance of our algorithm and the comparisons with the previous works show that our absMAC layer is at least 5 times faster in providing reliable communications.

B. Organization

The rest of the paper is organized as follows. In section II, the related work is introduced. In section III, we present the network model, problem definition, and some basic assumptions. In section IV and V, we show the implementation algorithm of absMAC layer and the application of absMAC layer on cooperative learning in Internet-of-Vehicles, respectively. In section VI, the simulation results are presented. In section VII, We conclude our paper.

II. Related Work

A. Cooperative Learning in IoV Networks

Recently, cooperative learning has become a popular way to handle the massive data of users and to support high-level applications in IoV networks and some relative areas. For example, Kong et al. in [1] study how federated learning could be used for license plate recognition in 5G networks; Manias and Shami in [2] focus on how to solve the scalability, high availability, and data privacy problems in intelligent transportation systems; Zhang et al. in [7] propose a new RL algorithm for Partially Detected Intelligent Traffic Signal Control (PD-ITSC) systems to alleviate traffic congestion; Liu et al. develop FedCPF, an efficient-communication approach for Customized, Partial and Flexible in [8]; Xu et al. in [9] combine Monte Carlo tree search (MCTS) and some heuristic rules to find a nearly global-optimal passing order (leaf node) within a very short planning time; Similar works in the relative areas can also be found in [10], [11], and [12]. Note that the above works are all based on some reliable communication assumptions, e.g., the device-to-device communication mode without any delay. There are also some works based on more realistic communication models, in which the communication is reliable but with a bounded delay. For instance, the federated learning applications in [13] require that clients should complete local calculations and model uploads within a defined delay; Zhou et al. in [14] take the impact of delay into account on system performance and provide a quantitative standard. However, none of them have discussed how the delays of the communications in their work are bounded.

B. Reliable Communications in IoV Networks

In general, the research on vehicular communications can be divided into two categories according to their reliable/unreliable communication modes. By assuming that the vehicles have reliable communications with each other, the works in the first category mainly focus on designing efficient algorithms for high-level applications, such as Quality of Service in [15] and [16], and Cloud-Based framework in [17] and [18]. In the second category in which wireless communications are unreliable, a series of algorithms have been proposed based on the new and complex hardware techniques, to improve the reliability of the communications and the throughput of the networks. For example, the MIMO in [19] and [20], the light communications in [21]. Considering the fact that most of the existing vehicles prefer the light-weight transceiver for communications due to the limited space and low cost, this paper no longer relies on the complex hardware techniques that require large space and high cost, but try to design some efficient algorithms (the absMAC layer algorithm) based on the simple and light-weight transceiver in an unreliable wireless network. In other words, the existing solutions in IoV with unreliable communication mainly rely on some complex and powerful hardware techniques, which are high-cost and require large space. Whereas, our work ensures reliable and efficient communications by designing an efficient absMAC layer algorithm based on the lightweight transceiver.

C. The Development of absMAC Layer

As a useful method to provide communications with bounded delay, the absMAC lay was first proposed by Kuhn et al. in [6] in 2009. Later, a variety of works about the implementation and the usage of the absMAC layer have been presented under the graph-based interference model [22], [23], [24], [25], [26] and the physical interference model [27], [28], [29], [30]. Specifically, in [31], C. Newport studies distributed consensus with an abstract MAC layer in the radio network setting and produces new upper and lower bounds for the consensus problem. Later, this work is extended to a fault-tolerant one in [23]. In [24], Lynch et al. show the usage of an abstract MAC layer for the asynchronous leader election and MIS (maximal independent set) construction problems, in which only constant messages are required for each node to complete the leader election and MIS construction. Similarly, Khabbazian et al. in [25] and Ghaffari et al. in [26] investigate the broadcast and multiple-message broadcast problem in wireless networks, both of which get the efficient time complexity. As for the implementation and usage of the abstract MAC layer in the physical interference model, Yu et al. show some efficient implementations of the absMAC layer via physical carrier sensing and inductive coloring in [27] and [28], respectively. Later, the implementation of the absMAC layer on the dynamic network with/without physical carrier sensing is presented in [29] and [30], which also discuss the time complexity of the consensus and local/global broadcast problem by using their implemented absMAC layer. However, to the best of our knowledge, previous works rarely consider how to implement an abstract MAC layer through a cooperative learning scheme or how the abstract MAC layer facilitates cooperative learning. Different from the previous works, our paper is the first one implementing the abstract MAC layer with a distributed DRL scheme and is specifically designed to support high-level cooperative learning services.

III. Model and Problem Definition

We consider such a cooperative learning framework on the Internet-of-Vehicles, in which n nodes are arbitrarily deployed within a multi-hop wireless network. Each node in IoV can be a vehicle, a roadside unit, or any device in the smart city system that connects to the IoV networks and is equipped with some computing units. When a cooperative learning task is deployed on the IoV, each node can update its learning model locally by exchanging the training parameters with its neighbors via the wireless channel. Then, we formulate the parameter exchanging process in cooperative learning on the Internet of Vehicles in the following network and communication models.

A. Network and Communication Models

We consider a multi-hop wireless network with n nodes arbitrarily deployed on a two-dimensional Euclidean space. The running time of nodes in this model is divided into equal-length synchronization rounds, each of which is sufficient for nodes to send/receive a message. Each node is equipped with a half-duplex transceiver, which means that in each round, a node can only transmit or listen in the wireless channel, but cannot do both simultaneously. R is the communication range of nodes, and we assume that nodes within the distance R can communicate with each other by transmitting/receiving signals via the wireless channel in a synchronized round. We use the following SINR (signal-to-interference-plus-noise-ratio) communication model to depict the process that whether a signal can be decoded by a listening node.

In each round, the nodes who choose to transmit/receive are termed as transmitters/receivers for short in our paper. The signal from a transmitter can be regarded as a vector, consisting of the strength and phase knowledge of this signal. The strength of a signal gets weaker with a longer distance from the transmitter. When the signals from multiple transmitters accumulate at a receiver, the process can be regarded as the sum of vectors. For a signal from the transmitter u to the receiver v, it is denoted by the vector $\vec{S}_{u,v}$, which not only includes the strength but also the phase knowledge of the signal. Whether the signal $\vec{S}_{u,v}$ can be decoded by the receiver v is formulated by the following SINR equations.

$$|\vec{S}_{u,v}| = P_u \cdot d(u,v)^{-\alpha}, \quad |\vec{S}_{W,v}| = |\sum_{u\in W} \vec{S}_{u,v}|,$$
$$SINR(u,v) = |\vec{S}_{u,v}|/(|\vec{S}_{W\setminus\{u\},v}| + N). \tag{1}$$

In the above SINR equations, P_u is the transmission power of signal $\vec{S}_{u,v}$, W is the set of transmitters in the current round, N is the ambient noise determined by the environment, and $d(u,v)$ is the Euclidean distance between u and v. $|\vec{S}_{u,v}|$ is defined as the strength of the signal $\vec{S}_{u,v}$, which gets weaker with distance and is also determined by the path-loss exponent α, a constant determined by the wireless medium and within 2 to 6 in usual. $\vec{S}_{W,v} = \sum_{u\in W} \vec{S}_{u,v}$ is used to depict the accumulation of signals on the receiver v, and $|\vec{S}_{W,v}|$ is the strength of the mixed-signal sensed by the node v. As a sufficient condition, we say when the strength of a signal from

u is β times larger than that of the others plus the ambient noise at v, i.e. $SINR(u, v) \geq \beta$, v can decode the signal $\vec{S}_{u,v}$, where constant β is a hardware-determined threshold larger than 1.

B. Problem Formulation

Based on the above network model and communication model, we implement the following absMAC layer, to provide reliable message exchange for various cooperative learning services. Specifically, there are two operations in our absMAC layer: *acknowledgement* and *progress*. In an acknowledgement operation, each node has its message broadcast to all of its neighbors; and in a progress operation, each node at least has one message received from its neighbors. The f_{ack} and f_{prog} are used to denote the timing bounds to complete the acknowledgement and progress operations, respectively.

Our implementation of the absMAC layer can strongly support parameter exchange in cooperative learning. Let $\mathcal{M}_v(t)$ be the parameter of vehicle v to be shared with its neighbors in round t. Then, in the round t, vehicle v executes the abstract MAC layer protocol to broadcast its parameter $\mathcal{M}_v(t)$. Obviously, with our absMAC layer implemented, each vehicle in IoV at least updates its local learning model once within f_{prog} rounds; and within f_{ack} rounds, a vehicle knows all parameters from its neighbors, according to our definition for the ack. and prog. operations in the absMAC layer.

To implement an absMAC layer, the policy of the vehicles in an interval includes their actions (transmit or listen) in each round of the interval. As has been discussed in our communication model, on one hand, too many nodes transmitting synchronously will cause heavy contention and interference in the wireless channel, which results in the failures of communications. On the other hand, if the number of nodes transmitting in each round is not enough, it will take a long time to complete the ack. and prog. operation, which reduces the efficiency of our absMAC layer. Thus, an efficient policy should control the contention and reduce the interference of the communications in each round, and minimize the running time of ack. and prog. operations.

For each node v, boolean variable $a(v, t)$ is used to denote its action in the round t. If v transmits, $a(v, t) = 1$; otherwise, $a(v, t) = 0$. Combining with the SINR model that depicts the accumulative contention and interference, our problem can be formulated as minimizing the length of the interval with the constraints on ack. and prog., given in the following.

$$\begin{aligned}
&\text{minimize } |I| \\
&s.t.\ a(v,t) = 1, a(u,t) = 0, \text{and } SINR(v,u) \geq \beta, \\
&\qquad \forall v \in V, \forall u \in \text{Neighbor}(v), \exists t \in I \\
&\qquad a(u,t) = 1, a(v,t) = 0, \text{and } SINR(u,v) \geq \beta, \\
&\qquad \forall v \in V, \exists u \in \text{Neighbor}(v), \exists t \in I \\
&\qquad \text{with } \ SINR(v,u) = \frac{|a(v,t) \cdot \vec{S}_{v,u}|}{(|\sum_{w \in V \setminus \{v\}} a(w,t) \cdot \vec{S}_{w,u}| + N)}, \\
&\qquad \forall v \in V, \quad \forall u \in \text{Neighbor}(v), \ \forall t \in I
\end{aligned} \tag{2}$$

In the above equation, $Neighbor(v)$ is the set of nodes that are within the transmission range of node v. The first and the second constraints require that the ack. and prog. operation for all nodes should be completed within the interval I. Let $a(V, t) = \{\cup_{v \in V} a(v, t)\}$ be the action set of all the nodes in the network, and $a(V, I) = \{\cup_{t \in I} a(V, t)\}$ be the action set of all the nodes in the interval I, which is also termed as the policy of nodes in interval I. In this paper, our objective is to design a distributed algorithm, by running which an efficient policy $a(V, I)$ can be generated for the absMAC layer.

1) Knowledge and Capability of Nodes: All nodes synchronously wake up at the beginning of our protocol, and have a uniform transmission power assumption. In each transmission, nodes can only get information about the environment by sensing the channel with the physical carrier sensing and from its neighbors if the received signal can be decoded. The number of nodes n, SINR parameters α, β, and N are unknown for nodes.

IV. Distributed Implementation of AbsMac Layer

In this section, we show how our abstract MAC layer is implemented with a distributed deep reinforcement learning scheme. Specifically, each of the nodes is deployed with an independent deep reinforcement learning agent. The agent gets the necessary information from the environment by sensing the channel and from its neighbors by decoding the received signal, to carve its states and rewards for deep reinforcement learning. Then, the agent chooses an approximate action to adjust its transmission probability, to make sure the efficiency and fairness of the communications in our absMAC layer protocol.

A. Challenges and Solutions

Implementing an abstract MAC layer over an open-access wireless channel in a distributed manner presents significant challenges. The primary hurdle is enhancing communication efficiency among nodes. This is crucial because the efficiency of communications directly impacts the timing constraints for acknowledgement and progress operations. Efficient communication hinges on accurately estimating contention in the wireless channel. When numerous nodes transmit simultaneously, excessive contention leads to message collisions. Conversely, low contention results in minimal successful transmissions. While existing methods employ back-off strategies to adjust contention exponentially, their flexibility in choosing optimal parameters for this process is limited. Thus, our initial algorithmic challenge revolves around each node accurately estimating wireless channel contention and selecting the most suitable transmission probability based solely on local information.

The second challenge is how to guarantee the fairness of communications in the absMAC layer. Lacking fairness will result in some worst cases, in which a fraction of nodes transmit too many times while the others always keep silent. The timing bounds f_{ack} and f_{prog} in those worst cases will also be terrible. An intuitive approach is to adjust the transmission probability of nodes according to their priorities. In other words, if a node has not transmitted for a long time, its priority on transmitting should increase. To the best of our knowledge, most of the existing works adjust their transmission probability according to the contention in the wireless channel but ignore the priorities of transmitters. For example, the algorithms in [32] chooses a uniform transmission probability for all nodes, regardless of their priorities. How to build a mapping from the priority of a node to its transmission probability with the local information in a distributed environment is the second challenge.

Note that both of the challenges mentioned above have special requirements for tuning the transmission probability, and simultaneously maintaining the efficiency and the fairness of the communications in our absMAC layer dramatically increasing the hardness of our algorithm design. Fortunately, the deep reinforcement learning scheme has its advantage in selecting the most appropriate answer from a complex scenario. We find that by the well-designed states, rewards, and actions, it is possible to adopt a deep reinforcement learning method to help each node choose the most appropriate transmission probability. Specifically, by sensing the channel, each node has a rough estimation of the contention; and with a reply from its neighbors, each node updates (increases or decreases) its priority. Then, the states in our DRL method include the above contention and the priority information; and the reward in our DRL is determined by the efficiency and the fairness of communications observed by the node itself from the environment. With such states and rewards, the DRL on each node can choose the appropriate action to adjust its transmission probability. Finally, due to the convergence of DRL, the transmission probability of each node gets close to its optimal one.

Algorithm 1 Implementation of absMAC Layer

In each phase, node v with $\mathcal{M}_v$ and $prio_v$ does:

$\{\mathcal{M}'_v,\ X_v,\ Signal_v,\ e_v\} \leftarrow$ RC-stage ($\mathcal{M}_v$);
    `// Regular Communication Stage`
$\{prio_v, p_v\} \leftarrow$ CC-stage ($\mathcal{M}'_v,\ \mathcal{M}_v$);
    `// Cooperative Communication stage`
$\{p_v\} \leftarrow$ CT-stage ($X_v, Signal_v, e_v, prio_v,\ p_v$);
    `// Computing stage`

RC-stage ($\mathcal{M}_v$)

Let $X_v \leftarrow 1$ or 0 with probability p_v and $1 - p_v$;
if $X_v = 1$ **then**
    Transmit the message $\mathcal{M}_v$;
    $s_v \leftarrow \phi$ and $\mathcal{M}'_v \leftarrow \phi$;
else
    Listen to the channel and $s_v \leftarrow Signal_v$;
    if *Received a message* $\mathcal{M}_u$ **then**
        $e_v = 1$ and $\mathcal{M}'_v \leftarrow \mathcal{M}_v$;
    else
        $e_v = 0$ and $\mathcal{M}'_v \leftarrow \phi$;
Output $\{\mathcal{M}'_v,\ X_v,\ Signal_v,\ e_v\}$;

CC-stage ($\mathcal{M}'_v$, $\mathcal{M}_v$)

Let $Y_v \leftarrow 1$ or 0 with probability p_v and $1 - p_v$;
if $\mathcal{M}'_v \neq \phi$ *and* $Y_v = 1$ **then**
    Transmit $\mathcal{M}'_v$;
else
    listen to the channel;
    if *Received a message* $\mathcal{M}'_u$ *which is the same as* $\mathcal{M}_v$ **then**
        $prio \leftarrow 1$;
        $\mathcal{M}_v \leftarrow$ a new message from its queue F_v;
    else
        $prio_v \leftarrow prio_v + 1$;
Output $\{prio_v,\ p_v\}$;

CT-stage (X_v, $Signal_v$, e_v, $prio_v$, p_v)

$State_v \leftarrow [X_v,\ Signal_v,\ e_v,\ prio_v,\ p_v]$;
$Reward_v \leftarrow e_v / prio_v$;
$k_v \leftarrow DRL-agent(State_v,\ Reward_v)$;
Output $\{k_v \cdot p_v\}$;

B. Detailed Description

We implement our abstract MAC layer by the following algorithm, which consists of successive phases. Each phase contains two communication stages and one computing stage. In the first stage, each node v broadcasts its message with the probability p_v. If a node v once broadcasts a message $\mathcal{M}_v$ in the first stage of a phase and then gets a reply for the message $\mathcal{M}_v$ from its neighbors in the second stage of the following phases, it can be proved that all of its neighbors must have received its message [27]. Then, the node v chooses a new message from F_v, which is a queue to store the messages of v. Additionally, as is discussed in the challenges and solutions part, the states and rewards of our DRL scheme highly rely on the contention of the wireless channel and the priority of each node. Thus, the first and second communication stages are also used for nodes to get relative information about the contention and the priority. Then, the DRL algorithm is embedded in the computing stage, which helps the node choose its appropriate transmission probability. By executing our algorithm, each node updates its transmission probability phase by phase. Finally, all nodes will choose their optimal transmission probability. The pseudocode of our algorithm is presented in Algorithm 1.

In the regular communication stage (the first stage, also termed as RC-stage for short) of a phase, each node v sets its parameter $X_v = 1$ or 0 with probability p_v and $1 - p_v$, respectively. Initially, we set $p_v = 1$ for each node v, and our DRL scheme will adjust the value of p_v phase by phase to an appropriate value. If $X_v = 1$, v transmits its message $\mathcal{M}_v$ with a probability p_v; Otherwise, when $X_v = 0$, it listens in the channel to sense the strength of the signal, which is recorded by the parameter $Signal_v$. Additionally, if v decodes a message $\mathcal{M}_u$ from the node u, it sets its message $\mathcal{M}'_v = \mathcal{M}_u$, and set $e_v = 1$; Otherwise $e_v = 0$; In the above process, X_v denotes the action of v in regular communication stage, i.e., whether v transmits or listens; $Signal_v$ denotes the contention of the channel if node v chooses to listen; e_v denotes whether there is a communication succeeded in the current stage. The output of the regular communication stage is the parameters M'_v, X_v, $Signal_v$, and e_v.

In the following cooperative communication stage (CC-stage), each node v sets its parameter $Y_v = 1$ or 0 with probability p_v and $1 - p_v$, respectively. If $Y_v = 1$, v transmits its message $\mathcal{M}'_v$ with a probability p_v; Otherwise when $Y_v = 0$, it listens in the channel. If it receives the message $\mathcal{M}'_u$ that is the same as $\mathcal{M}_v$, we can get the fact that there is once a time v transmits and its message $\mathcal{M}_v$ was received by the node u in regular communication stage previously. Thus, v update its priority $prio_v$ to 1, and select a new message from its queue F_v to transmit in the following. Otherwise, the priority of v increases by 1. Note that the priority of each node is set to be 1 initially. The output of the cooperative communication stage is the parameters $prio_v$ and p_v.

In both the regular communication stage and the cooperative communication stage, the actions of nodes are transmitted or listened once. So, the running time for two communication stages is 1 round. In the following computing stage (CT-stage), each node adjusts its transmission probability by executing the following DRL algorithm.

1) State Space: As the input of the DRL, the selection of state is of much importance for the convergence and efficiency of the DRL process. Specifically, a well-designed state space should not only contain all the information that the agent needed to take appropriate actions but also be as concise as possible. A piece of important information missing in the state may result in the failure of the convergence process, while too much information dilutes the relationship between the state and the action and is hard to satisfy in a distributed framework. As mentioned above, the contention of the channel and priority of nodes is important for nodes to take the following actions. When our algorithm is designed, the parameters $Signal_v$ and $prio_v$ are used to record the contention sensed by v and the priority of v. Thus, our state includes the parameters $Signal_v$ and $prio_v$. Additionally, the state of v also contains the parameters X_v, e_v, and p_v, i.e., some information that can be obtained by v easily. The formulation for the state of v is given in Eqt. 3

$$State_v = [X_v, Signal_v, e_v, prio_v, p_v] \tag{3}$$

In the above equation, $X_v = 0$ or 1 indicates that whether v transmits or listens in the RC-stage; $Signal_v$ is the strength of the signal sensed by the node v in RC-stage; e_v indicates that whether there is a communication succeeded in the RC-stage; $prio_v$ is the priority of node v; and p_v is the transmission probability of node v;

2) Action Space: The agent interacts with the environment by selecting an action from the action space. For each agent, its behavior in each phase mainly depends on its state in the current phase and the learning experience so far. In this paper, we define $Action_v$ as the action vector of the agent in device v in each phase, which can be expressed by Eq.4

$$Action_v = [a, a + \delta, a + 2\delta, a + 3\delta, \ldots, b] \tag{4}$$

In the above Equation, $0 < a < 1$ and $b > 1$ are the lower bound and upper bound of our action space and δ is the gap between the neighboring actions.

3) Reward: After the agent selects an action, a reward is returned from the environment. From a statistical view, when an agent made a good decision, the environment should reward it with a positive value; on the contrary, when a bad action was made by the agent, a negative reward will be returned. Thus, the agents can be motivated to perform good behaviors in order to maximize the rewards returned from the environment, which makes the learning process effective.

As mentioned above, efficient and fair communications are the key point for the implementation of our abstract MAC layer. And we hope that in each RC stage, there is communication success and nodes prefer to transmit when their priorities are small, to avoid the case that a node always keeps listening for a long time. With such a target, we give the formal definition for the reward function in Eqt.5.

$$Reward_v = e_v / prio_v \tag{5}$$

In the above equation, $e_v = 0$ or 1 indicates whether there is a communication succeeded, and $prio_v$ is the priority of the node v which is set to 1 initially, constantly increases in each phase, and reset to 1 if the message $\mathcal{M}_v$ is received by other nodes.

C. Details of Our Learning Part

The agent in our Algorithm 1 is implemented by the double deep Q-network (DDQN) method, which is an evolution of the DQN, to efficiently handle the massive states in our learning algorithm. In this part, for a brief description, we use $s(t)$, $a(t)$, and $r(t)$ to denote the state $State_v$, the action $Action_v$, and the reward $Reward_v$ of a node v in phase t. Then, the detailed expression of our DDQN is given in the following:

$$\begin{aligned} & q'(s(t), a(t) \\ & \quad \mid \boldsymbol{\theta}) \leftarrow \left(r(t) + \gamma \max_{a'} q\left(s(t+1), a' \mid \boldsymbol{\theta}\right) \right) \\ & q(s(t), a(t) \\ & \quad \mid \boldsymbol{\theta}) \leftarrow (1-\alpha) q(s(t), a(t) \mid \boldsymbol{\theta}) + \alpha q'(s(t), a(t) \mid \boldsymbol{\theta}), \end{aligned} \tag{6}$$

in which θ is the DDQN parameters. The parameter $\alpha \in [0, 1]$ is the learning rate of RL, and the parameter $\gamma \in [0, 1]$ is the discount factor of Deep learning. To update the values of θ at the phase t, the DQN-based algorithm combining Q-learning and DNNs has two Q-functions, i.e., a Q-target network $q(s_t, a_t|\theta_{target})$ and a Q-train network $q(s_t, a_t|\theta_{train})$. The loss error between the two Q-functions is minimized following the experience replay, where a loss function is defined as Eqt.7.

$$\mathcal{L}(\boldsymbol{\theta}_{\text{train}}) = \sum_{(s(t), a(t), r(t), s(t+1))} \left(y_{\text{target}} - q\left(s(t), a(t) \mid \boldsymbol{\theta}_{\text{train}}\right) \right), \tag{7}$$

with $y_{\text{target}} = r(t) + \gamma \max_{a'} q\left(s(t+1), a' \mid \boldsymbol{\theta}_{\text{target}}\right)$.

However, the DQN-based algorithm may cause a large deviation in its model due to an overestimation of the values of the Q-target. To solve this problem, the DDQN algorithm

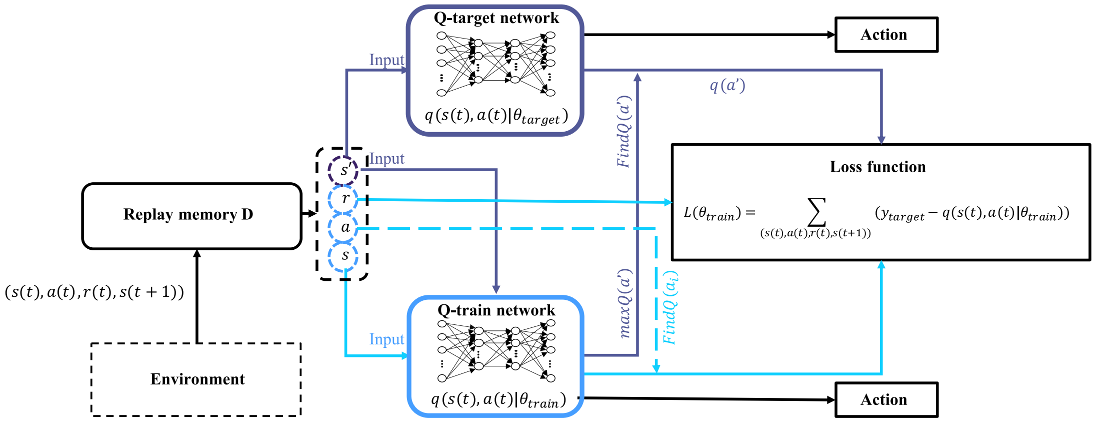


Fig. 2. The framework of our double deep Q-network.

decouples the action selection and calculation of the value of the Q-target, as shown in Figure 2. In the DDQN, the value of Q-target is calculated by Eqt.8.

$$\hat{q} = q\left(s(t+1), \max_{a'} q\left(s(t+1), a' \mid \boldsymbol{\theta}_{\text{target}}\right) \mid \boldsymbol{\theta}^{-}_{\text{target}}\right),$$
$$y^{\text{DDQN}}_{\text{target}} = r(t) + \gamma \max_{a'} \hat{q}, \tag{8}$$

where θ_{target} and θ^{-}_{target} denote the DNN's parameters of action selection in the current phase and the last phase.

To balance the exploitation and exploration in our DDQN, we adopt an ε-greedy scheme, in which an agent randomly selects an action with probability ε or chooses the action with the largest reward with probability $1-\varepsilon$. By setting the ε-greedy scheme, our algorithm can avoid falling into a local optimal solution and make a timely response to network changes such as the mobility of nodes. In the training process, our algorithm makes decisions according to the data and modified its network parameters in an online mode.

V. absMAC Layer for Cooperative Learning

In this section, we show how our absMAC Layer can be applied to support collaborative learning in Internet-of-Vehicles by giving the following example. Consider a scenario that n vehicles cooperatively train a learning model for high-level applications. Each vehicle has its own data set and updates its training parameters from the learning process of its own data set and receiving new parameters from its neighbors. By implementing our abstract MAC layer, as illustrated in Figure 3, each vehicle v broadcasts its parameters by the acknowledgement operation and receives the parameters of other vehicles by the progress operation, periodically. Gradually, when more and more local parameters of vehicles are exchanged by our absMAC layer, the learning model on each vehicle becomes more and more accurate.

In the following, we give a detailed description of the application of our abstract MAC layer on each node v with its data set D_s. The parameter $\vec{S}_0(v)$ is the parameter in its final learning model, and $w_0(v)$ is the weight of $\vec{S}_0(v)$. The parameter $\vec{S}_1(v)$ is the local parameter periodically learned by v from its local data set, and $w_1(v)$ is the corresponding weight; $\mathcal{M}_v$ is the message of v, which will be broadcast by the acknowledgement operation. Each time when there is a local parameter $\langle \vec{S}_1(v), w_1(v)\rangle$ learned, v updates its own learning model by the Eqt. 9

$$\langle \vec{S}_0(v)|w_0(v)\rangle = \langle \frac{\vec{S}_0(v)w_0(v)+\vec{S}_1(v)w_1(v)}{w_0(v)+w_1(v)}|w_0(v)+w_1(v)\rangle, \tag{9}$$

and adds the element $\langle \vec{S}_1(v), w_1(v)\rangle$ to the queue F. Note that F is a first-in-first-out queue, from which the acknowledgement operation selects a message to transmit periodically if F is not empty.

With an acknowledgement operation, each vehicle can have its local parameters broadcast to all of its neighbors once; and with a progress operation, each vehicle will receive the local parameters from one of its neighbors. When receiving a message $\mathcal{M}_u = \langle \vec{S}_1(u), w_1(u)\rangle$, v updates its own learning model by the Eqt. 10

$$\langle \vec{S}_0(v)|w_0(v)\rangle = \langle \frac{\vec{S}_0(v)w_0(v)+\vec{S}_1(u)w_1(u)}{w_0(u)+w_1(u)}|w_0(u)+w_1(u)\rangle \tag{10}$$

With such a framework illustrated in Figure 3, we show how our abstract MAC layer helps on providing reliable communications for cooperative learning on the Internet of Vehicles.

VI. Simulation Results

In this section, we investigate the performance of our absMAC layer with the network parameter varying. Specifically, we mainly observe the maximum and average time used for the ack. and prog. operations, because the average time used for the ack. and prog. operations are the most straightforward and prime metrics that can be observed in the simulation. A smaller average time indicates that the vehicles can exchange their messages faster (i.e., within a shorter time) through the absMAC layer. The maximum time is observed to make a worst-case guarantee from a statistical view. Then, we use the ratio and the waiting time of the ack. operation as the metrics to evaluate the efficiency and fairness of our absMAC layer algorithm. Specifically, the ratio of the ack.

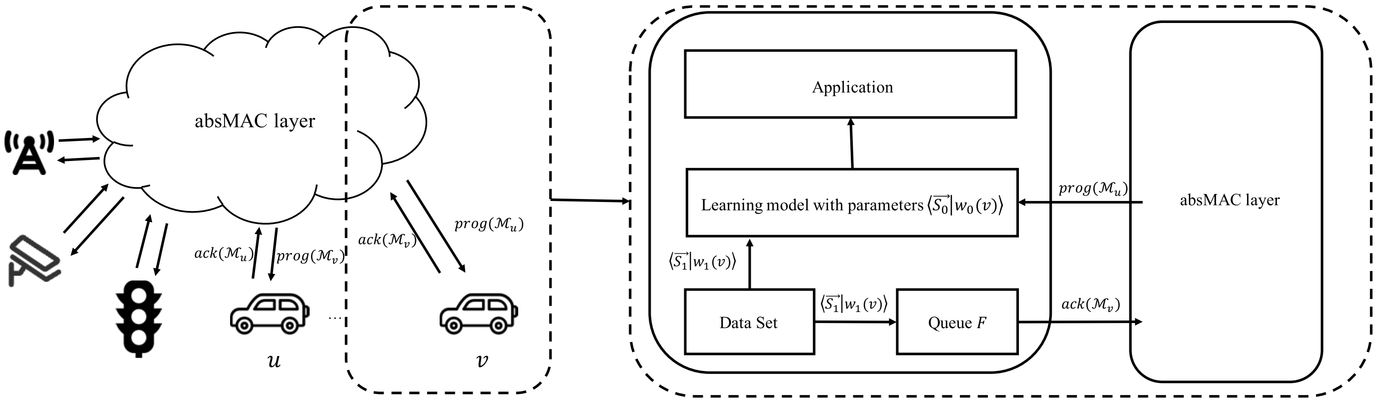


Fig. 3. An application of our absMAC layer for cooperative learning in Internet of Vehicles.

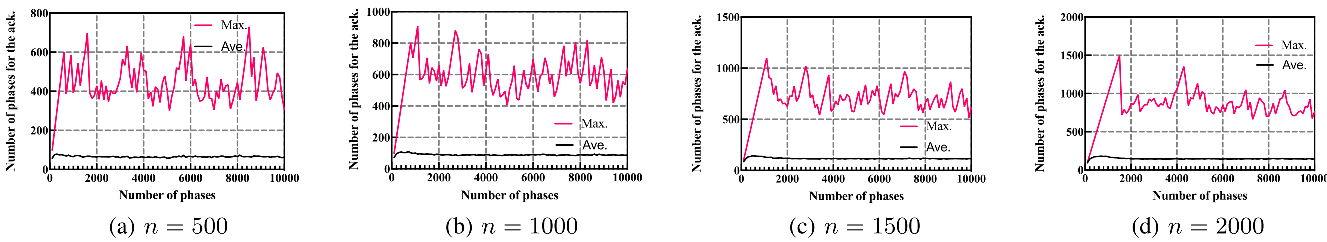


Fig. 4. Maximum/average time for the acknowledgement operation when n varies.

for a node v is defined as the ratio between the number of ack. that v has successfully completed and the number of phases that our algorithm has been executed by v; the waiting time of v is defined as the time elapsed since v's last successful ack. operation in our algorithm. Considering the fact that the ack. operation is more complex than the progress operation, we choose the ratio and the waiting time of the ack. operation to reflect the efficiency and fairness of the communications in our algorithm. Smaller the ratio and waiting time we observe, more efficient the absMAC layer algorithm is. Besides, when all the vehicles have a similar ratio and waiting time, we say the absMAC layer algorithm guarantees the fairness of communications among the vehicles.

A. Implementation of Simulations

We simulate an edge-computing system in a 300m*300m 2-dimensional Euclidean space, in which n nodes are randomly and uniformly deployed. $n \in [500, 2000]$ in our simulation. The transmission range R of each node is 30m, to ensure a multi-hop network environment. In each phase, the nodes select the message with the maximum age from its queue to transmit by executing our abstract MAC layer. In our simulation, γ, lr, ε are parameters in our deep reinforcement learning, where γ is the learning rate of reinforcement learning, lr is the learning rate of the neural network, and ε is the parameter of ε-greedy scheme. The detailed parameters in our simulation are listed in Table I.

Without loss of generality, over 50 runs of the simulation have been carried out for each reported result. All experiments are conducted on a Linux machine with Intel Xeon CPU E5-2670@2.60GHz and 128 GB main memory, implemented in python3 and compiled by a Python compiler.

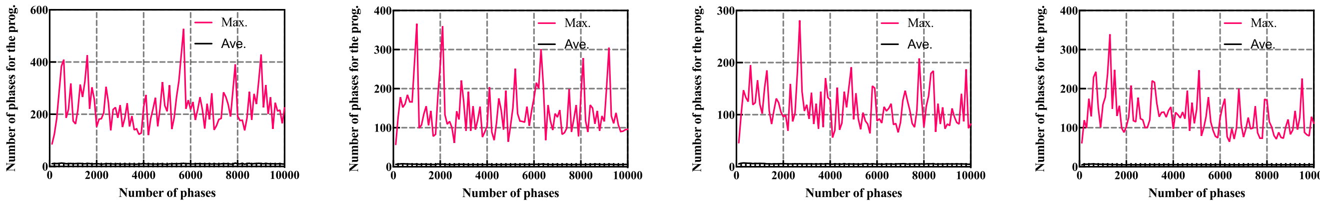
TABLE I
PARAMETER SETTING IN SIMULATION

Parameter	settings	Definitions
n	$[500, 2000]$	Number of nodes
R	$30m$	Transmission range
α	4	Parameter in SINR
β	2	Parameter in SINR
N	1	Parameter in SINR
γ	0.5	Parameter in DRL
lr	0.01	Parameter in DRL
ε	0.1	Parameter in DRL

B. Performance of Our Algorithm

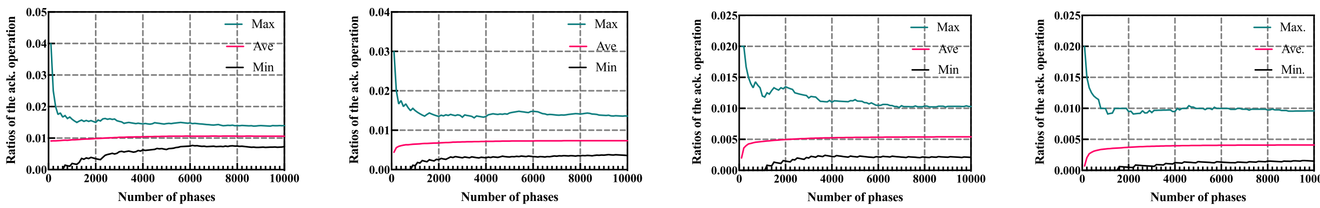
In this part, we present the time used to complete the ack. and prog. operations in Fig. 4 and Fig.5, respectively. Additionally, the efficiency and fairness of the communications in our algorithm are investigated in Fig. 6 and Fig. 7 by counting the ratio and the waiting time of the ack. operation on each node. A detailed description and analysis of the Fig. 4-7 are given in the following.

The time bound for the ack. operation is presented in Fig. 4, in which the x-axes represent the running time of our algorithm, and the y-axes represent the number of phases used to complete the ack. operation, and n varies from 500 to 2000, respectively. From the curves in each of the Fig. 4(a)-4(d), both the maximum and the average time for the ack. operation increase initially and keep stable at some low levels after our algorithm is executed by at least 500 phases, which directly shows that our algorithm can complete the ack. operation within a short time; by comparing the curves across the Fig. 4(a)-4(d), in which the number of nodes varies from 500 to 2000, we can see that the maximum and the average time used for an ack. operation increase when n gets larger.



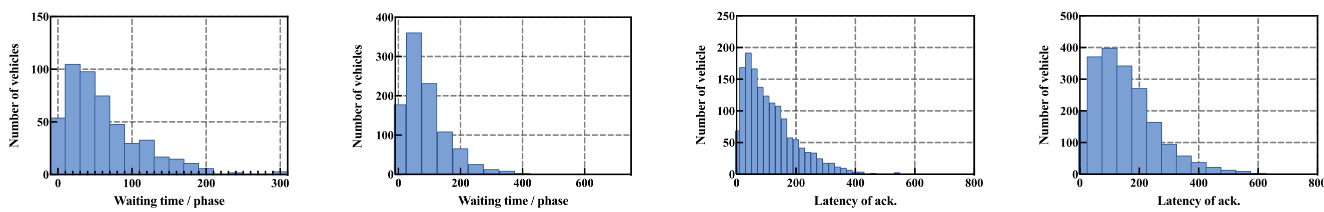
(a) $n = 500$ (b) $n = 1000$ (c) $n = 1500$ (d) $n = 2000$

Fig. 5. Maximum/average time for the progress operation when n varies.



(a) $n = 500$ (b) $n = 1000$ (c) $n = 1500$ (d) $n = 2000$

Fig. 6. Maximum/average/minimum ratio of the acknowledgement operation when n varies.



(a) $n = 500$ (b) $n = 1000$ (c) $n = 1500$ (d) $n = 2000$

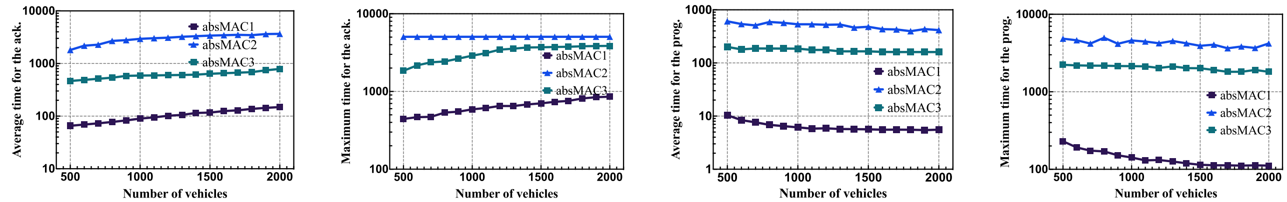
Fig. 7. The distribution of vehicles on their waiting time when v varies.

This is because when there are more vehicles, it becomes rarer for each of them to transmit in the wireless channel. Even in the worst case with $n = 2000$, we have the maximum/average time for the acknowledgement no larger than 150/1000 phases, respectively in Fig. 4(d), which is a competitive result on the time cost.

The time bound for the prog. operation is presented in Fig. 5, in which the x-axes represent the running time of our algorithm, the y-axes represent the number of phases used to complete a progress operation, and n varies from 500 to 2000, respectively. From the curves in each of the Fig. 5(a)-5(d), we can see that the average and the maximum time used for the prog. operation have a similar tendency to those of the ack. operation, but keep stable at some smaller values; By comparing the curves across the Fig. 5(a)-5(d), with n various from 500-2000, we can see that the average time for prog. decreases while the maximum time increases when n gets larger. This is because when there are more vehicles, it becomes easier for a vehicle to receive a message from its neighbours. While the maximum time indicates the worst case, which is more likely to happen when n gets larger.

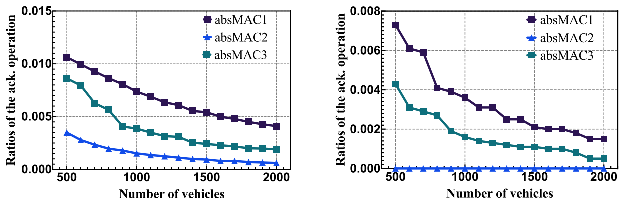
To show the efficiency and fairness of our algorithm on communications, we observe the maximum, average, and minimum ratios of the ack. operation in simulation, the results of which are presented in Fig. 6. Specifically, in Fig. 6, the x-axes represent the running time of our algorithm, y-axes represent the value of the ratios, and $n \in [500, 2000]$. From the curves in each of the Fig. Fig. 6(a)-6(d), the average ratio increases initially and keeps stable later. Even though the value of the average ratio for a single node is not large when it keeps stable, considering the total number of nodes, those values are strong enough to verify the efficiency of our algorithm on ack. operation; Additionally, the maximum and minimum ratios indicate the best and the worst cases for the ack. operations among nodes. From the curves in Fig. 6, we can see that the maximum and the minimum ratios differ a lot initially, and get close to each other gradually, which means the fairness of communications between nodes is gradually maintained when our algorithm is executed; Finally, by comparing the curves across the Fig. 6(a)-6(d), as the number of nodes increasing, the maximum, average, and minimum ratios of the ack. operation have a similar tendency, but their values decrease when n gets larger.

To better demonstrate the fairness of our algorithm, Fig. 7 shows the distribution of all the nodes on the waiting time, which is defined as the time elapsed since the last successful ack. operation of a node. In Fig. 7(a), we can see that at least 60% and 80% nodes have their waiting time smaller than 80 and 120 phases, respectively. Even though the waiting time in Fig. 7(b)-7(d) is very likely to increase when there are more nodes in the network, we still have the result that at least 60% and 80% nodes have their waiting time smaller than 150 and 200 phases with $n \leq 2000$. With the above histograms in



(a) Average time for the ack. (b) Maximum time for the ack. (c) Average time for the prog. (d) Maximum time for the prog.

Fig. 8. Comparison with previous works on the running time for ack. and prog. operations.



(a) Average ratio of the ack. (b) Minimum ratio of the ack.

Fig. 9. Comparison with previous works on the ack. ratio.

Fig. 7, we verify the fairness of our algorithm on the ack. operation.

C. Comparison With Previous Works

In this part, we compare our algorithm with two previous works in [27] and [30], respectively, in which a statistical back-off scheme and a leader election & broadcast scheme are designed to implement the absMAC layer, respectively. In the following, we compare our work with those two previous works in terms of the running time for ack. and prog. operations, and ratio of ack. operation. Our algorithm and those two previous algorithms in [27], [30] are abbreviated as *absMAC1*, *absMAC2*, and *absMAC3*, respectively.

Firstly, we show the comparison results on the running time for ack. and prog. operations in Fig. 8. Specifically, in Fig. 8(a) and 8(b), the x-axes are the number of nodes and the y-axes are the time used for the ack. operation. From the curves in Fig. 8(a) and 8(b), we can see that our algorithm is at least 10 and 5 times faster than *absMAC2* on the average and the maximum running time for ack. operation, and at least 5 and 3 times faster than *absMAC3* on the average and the maximum running time for ack. operation. Similarly, by comparing the curves in Fig. 8(c) and 8(d), we get the result that our algorithm is about 100 and 40 times faster than *absMAC2* on the average and maximum running time for prog. operation, and 20 and 10 times faster than *absMAC3* on the average and maximum running time for prog. operation.

Secondly, by observing the average/minimum ratios of ack. operation, we evaluate the efficiency of the communications in *absMAC1*, *absMAC2*, and *absMAC3* in Fig. 9, in which the x-axes are the number of nodes and the y-axes are the value of the average/minimum ratio of ack. operation. By comparing the curves in Fig. 9(a), it can be seen that the average ratio of ack. in our *absMAC1* is at least 2 and 1.2 times larger than that of the *absMAC2* and *absMAC3*, respectively. Additionally, for the minimum ack. ratio, our algorithm always has a larger value than that of *absMAC3*, and the minimum ack. ratio of *absMAC2* gets close to 0. With the above results observed, we believe that our algorithm has higher efficiency and fairness in communications than the previous works in [27] and [30].

D. Conclusion

In this section, by presenting the time cost for ack. and prog. operations, and comparing with two previous works, we evaluate the performance of our algorithm. In general, the timing bounds for ack. and prog. operations increase when there are more nodes deployed in IoV networks and are within 200 and 10 phases, respectively, when $n \leq 2000$. The observation of the ratio and the waiting time of ack. operation verifies the efficiency and fairness of the communications in our algorithm. By comparison with the work in [27] and [30], our algorithm is much faster on the ack. and prog. operations, respectively, with higher efficiency and fairness.

VII. Conclusion

In this paper, we implement an abstract MAC layer via a distributed deep reinforcement learning method, to satisfy the communication demand of cooperative learning on the Internet of Vehicles. Specifically, our designed abstract MAC layer can support the various cooperative learning services by providing the acknowledgement and progress operations, in which each vehicle shares its learning result with all of its neighbours, and receives at least one result from its neighbours to update its training model, respectively. When carving the states, rewards, and actions of our DRL scheme, the efficiency and priority of communications between vehicles are considered as important features, which results in the efficient acknowledgement and progress operations in our abstract MAC layer. We hope that our work can be a bridge between the current cooperative learning applications that highly rely on reliable communication assumptions and the open-access wireless channel in IoV in which the communications are inherently unreliable. Combining our scheme with some more efficient but complex communication techniques, such as the multi-channel and NOMA (Non-Orthogonal Multiple Access) technologies will be further investigated in our future research.

References

[1] X. Kong et al., "A federated learning-based license plate recognition scheme for 5G-enabled Internet of Vehicles," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8523–8530, Dec. 2021, doi: 10.1109/TII.2021.3067324.

[2] D. M. Manias and A. Shami, "Making a case for federated learning in the Internet of Vehicles and intelligent transportation systems," *IEEE Netw.*, vol. 35, no. 3, pp. 88–94, 2021, doi: 10.1109/MNET.011.2000552.

[3] I. Ioannou, C. Christophorou, V. Vassiliou, and A. Pitsillides, "A novel distributed AI framework with ML for D2D communication in 5G/6G networks," *Comput. Netw.*, vol. 211, Jul. 2022, Art. no. 108987, doi: 10.1016/j.comnet.2022.108987.

[4] Y. He, J. Ren, G. Yu, and J. Yuan, "Importance-aware data selection and resource allocation in federated edge learning system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13593–13605, Nov. 2020, doi: 10.1109/TVT.2020.3015268.

[5] Z. Ji, L. Chen, N. Zhao, Y. Chen, G. Wei, and F. R. Yu, "Computation offloading for edge-assisted federated learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9330–9344, Sep. 2021, doi: 10.1109/TVT.2021.3098022.

[6] F. Kuhn, N. A. Lynch, and C. C. Newport, "The abstract MAC layer," in *Proc. 23rd Int. Symp.* (Lecture Notes in Computer Science), vol. 5805, I. Keidar, Ed., Elche, Spain. Cham, Switzerland: Springer, Sep. 2009, pp. 48–62, doi: 10.1007/978-3-642-04355-0_9.

[7] R. Zhang, A. Ishikawa, W. Wang, B. Striner, and O. K. Tonguz, "Using reinforcement learning with partial vehicle detection for intelligent traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 404–415, Jan. 2021, doi: 10.1109/TITS.2019.2958859.

[8] S. Liu, J. Yu, X. Deng, and S. Wan, "FedCPF: An efficient-communication federated learning approach for vehicular edge computing in 6G communication networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1616–1629, Feb. 2022, doi: 10.1109/TITS.2021.3099368.

[9] H. Xu, Y. Zhang, L. Li, and W. Li, "Cooperative driving at unsignalized intersections using tree search," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4563–4571, Nov. 2020, doi: 10.1109/TITS.2019.2940641.

[10] T.-D. Cao, T. Truong-Huu, H. Tran, and K. Tran, "A federated deep learning framework for privacy preservation and communication efficiency," *J. Syst. Archit.*, vol. 124, Mar. 2022, Art. no. 102413, doi: 10.1016/j.sysarc.2022.102413.

[11] Z. Ma, M. Zhao, X. Cai, and Z. Jia, "Fast-convergent federated learning with class-weighted aggregation," *J. Syst. Archit.*, vol. 117, Aug. 2021, Art. no. 102125, doi: 10.1016/j.sysarc.2021.102125.

[12] T. Chu, J. Wang, L. Codeca, and Z. Li, "Multi-agent deep reinforcement learning for large-scale traffic signal control," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1086–1095, Mar. 2020, doi: 10.1109/TITS.2019.2901791.

[13] Z. Zhao, J. Xia, L. Fan, X. Lei, G. K. Karagiannidis, and A. Nallanathan, "System optimization of federated learning networks with a constrained latency," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 1095–1100, Jan. 2022, doi: 10.1109/TVT.2021.3128559.

[14] Y. Zhou et al., "The role of communication time in the convergence of federated edge learning," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 3241–3254, Mar. 2022, doi: 10.1109/TVT.2022.3144099.

[15] K. P. Thakur and B. Palit, "A QoS-aware joint uplink spectrum and power allocation with link adaptation for vehicular communications in 5G networks," 2023, *arXiv:2305.02667*.

[16] Á. Martín et al., "Adaptive QoS of WebRTC for vehicular media communications," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast.*, Bilbao, Spain, 2022, pp. 1–6, doi: 10.1109/BMSB55706.2022.9828782.

[17] Z. Yang, L. Li, F. Gu, and X. Ling, "Dependable and reliable cloud-based architectures for vehicular communications: A systematic literature review," *Int. J. Commun. Syst.*, vol. 36, no. 7, p. e5457, May 2023, doi: 10.1002/dac.5457.

[18] E. Vieira, J. Almeida, J. Ferreira, T. Dias, A. V. Silva, and L. Moura, "A roadside and cloud-based vehicular communications framework for the provision of C-ITS services," *Information*, vol. 14, no. 3, p. 153, Mar. 2023, doi: 10.3390/info14030153.

[19] X. Cheng, S. Gao, and L. Yang, *mmWave Massive MIMO Vehicular Communications* (Wireless Networks). Berlin, Germany: Springer, 2023, doi: 10.1007/978-3-030-97508-1.

[20] Q. Hamarsheh, O. R. Daoud, M. Al-Akaidi, A. Damati, and M. Bani Younis, "Robust vehicular communications using the fast-frequency-hopping-OFDM technology and the MIMO spatial multiplexing," *Int. J. Commun. Netw. Inf. Secur. (IJCNIS)*, vol. 14, no. 1, pp. 19–25, Apr. 2022, doi: 10.17762/ijcnis.v14i1.5216.

[21] C. Beguni, A.-M. Căilean, S.-A. Avătămănitei, A.-D. Potorac, E. Zadobrischi, and M. Dimian, "Increasing vehicular visible light communications range based on LED current overdriving and variable pulse position modulation: Concept and experimental validation," *Sensors*, vol. 23, no. 7, p. 3656, Mar. 2023, doi: 10.3390/s23073656.

[22] M. Khabbazian, F. Kuhn, N. Lynch, M. Médard, and A. ParandehGheibi, "MAC design for analog network coding," in *Proc. 7th ACM ACM SIGACT/SIGMOBILE Int. Workshop Found. Mobile Comput.*, A. Chaintreau and D. R. Kowalski, Eds. San Jose, CA, USA, Jun. 2011, pp. 42–51, doi: 10.1145/1998476.1998484.

[23] C. Newport and P. Robinson, "Fault-tolerant consensus with an abstract MAC layer," in *Proc. 32nd Int. Symp. Distrib. Comput.*, vol. 121, U. Schmid and J. Widder, Eds. New Orleans, LA, USA, Oct. 2018, pp. 1–20, doi: 10.4230/LIPIcs.DISC.2018.38.

[24] N. Lynch, T. Radeva, and S. Sastry, "Asynchronous leader election and MIS using abstract MAC layer," in *Proc. 8th Int. Workshop Found. Mobile Comput.*, Jul. 2012, p. 3, doi: 10.1145/2335470.2335473.

[25] M. Khabbazian, F. Kuhn, D. R. Kowalski, and N. Lynch, "Decomposing broadcast algorithms using abstract MAC layers," in *Proc. 6th Int. Workshop Found. Mobile Comput.*, Cambridge, MA, USA, A. W. Richa and T. Moscibroda, Eds. Sep. 2010, pp. 13–22, doi: 10.1145/1860684.1860690.

[26] M. Ghaffari, E. Kantor, N. Lynch, and C. Newport, "Multi-message broadcast with abstract MAC layers and unreliable links," in *Proc. ACM Symp. Princ. Distrib. Comput.*, Paris, France, Jul. 2014, pp. 56–65, doi: 10.1145/2611462.2611492.

[27] D. Yu, Y. Zhang, Y. Huang, H. Jin, J. Yu, and Q. Hua, "Exact implementation of abstract MAC layer via carrier sensing," in *Proc. IEEE Conf. Comput. Commun.*, Honolulu, HI, USA, Apr. 2018, pp. 1196–1204, doi: 10.1109/INFOCOM.2018.8486397.

[28] D. Yu, Y. Zou, Y. Wang, J. Yu, X. Cheng, and F. C. M. Lau, "Implementing the abstract MAC layer via inductive coloring under the Rayleigh-fading model," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 6167–6178, Sep. 2021, doi: 10.1109/TWC.2021.3072236.

[29] D. Yu, Y. Zou, Y. Zhang, H. Sheng, W. Lv, and X. Cheng, "An exact implementation of the abstract MAC layer via carrier sensing in dynamic networks," *IEEE/ACM Trans. Netw.*, vol. 29, no. 3, pp. 994–1007, Jun. 2021, doi: 10.1109/TNET.2021.3057890.

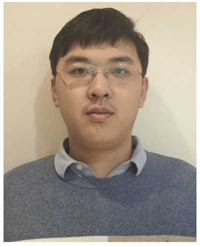
[30] D. Yu et al., "Implementing the abstract MAC layer in dynamic networks," *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 1832–1845, May 2021, doi: 10.1109/TMC.2020.2971599.

[31] C. Newport, "Consensus with an abstract MAC layer," in *Proc. ACM Symp. Princ. Distrib. Comput.*, M. M. Halldorsson and S. Dolev, Eds. Paris, France, Jul. 2014, pp. 66–75, doi: 10.1145/2611462.2611479.

[32] D. Yu et al., "Competitive age of information in dynamic IoT networks," *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15160–15169, Oct. 2021, doi: 10.1109/JIOT.2020.3038595.

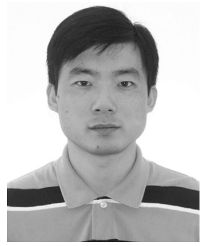
Yifei Zou (Member, IEEE) received the B.E. degree from the Computer School, Wuhan University, in 2016, and the Ph.D. degree from the Department of Computer Science, The University of Hong Kong, in 2020. He is currently an Assistant Professor with the School of Computer Science and Technology, Shandong University, Qingdao. His research interests include wireless networks, ad hoc networks, and distributed computing.

Zuyuan Zhang is currently pursuing the bachelor's degree with the Computer Science Program, Taishan College, Shandong University. His research interests include edge intelligence, wireless networks, and distributed computing.



Congwei Zhang received the B.E. degree from the Computer Science Program, Taishan College, Shandong University, in 2021. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Shandong University. His research interests include wireless networks, edge intelligence, and distributed computing.

Yanwei Zheng (Member, IEEE) received the Ph.D. degree from the School of Computer Science and Engineering, Beihang University, Beijing, in 2019. He is currently an Assistant Researcher with the School of Computer Science and Technology, Shandong University, Qingdao. His research interests include object navigation and computer vision.



Dongxiao Yu (Senior Member, IEEE) received the B.Sc. degree from the School of Mathematics, Shandong University, in 2006, and the Ph.D. degree from the Department of Computer Science, The University of Hong Kong, in 2014. He was an Associate Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, in 2016. He is currently a Professor with the School of Computer Science and Technology, Shandong University. His research interests include wireless networks, distributed computing, and graph algorithms.

Jiguo Yu (Fellow, IEEE) received the Ph.D. degree in mathematics from the School of Mathematics, Shandong University, Jinan, China, in 2004. In 2007, he was a Full Professor with the School of Computer Science, Qufu Normal University, Jining, China. He is currently a Full Professor with the Qilu University of Technology (Shandong Academy of Sciences), the Shandong Computer Science Center (National Supercomputer Center, Jinan), and the Shandong Laboratory of Computer Networks, Jinan. His research interests include privacy-aware computing, wireless networking, distributed algorithms, blockchain, and graph theory.