Deep Reinforcement Learning-based Joint Frame Length and Rate Adaption for WLAN Network

Lihong Zhou, Xuming Fang, Rong He*, and Huanrong Zhang
Key Laboratory of Information Coding and Transmission, Southwest Jiaotong University, Chengdu, China
z_lh@my.swjtu.edu.cn, xmfang@swjtu.edu.cn, rhe@swjtu.edu.cn, z_hr@my.swjtu.edu.cn

2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall) | 979-8-3503-2928-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/VTC2023-Fall60731.2023.10333585

***Abstract*—Frame aggregation and physical rate adaptation are the most important enhancement for Wi-Fi network. However, both of them involve certain tradeoffs between achieving higher throughput and facing a higher error rate. The gain suffers from the imperfect and highly dynamic channel condition. In addition, there is a certain coupling relationship between the aggregation frame length and the physical rate. That means the selection of physical rate may affects the optimal frame length, and vice versa. Therefore, a joint frame length and rate adaption scheme is needed. Moreover, the large number of all available frame lengths and rates makes the joint adaption more challenging. In this paper, we propose a joint frame length and rate adaption (JFRA) scheme based on Double Deep Q-learning (DDQN) algorithm. The proposed scheme can automatically explore the environment and learn the optimal frame length and rate from experience. We apply prioritized training and incorporate reward value into the computation of experience priority. It can improve learning efficiency and accelerate the convergence of JFRA. We implement and evaluate JFRA in ns3-ai framework and the simulation results show that JFRA can outperform the Minstrel HT and Thompson Sampling algorithm by up to 21.3% and 68.9% in various cases.**

***Index Terms*—Wi-Fi, deep reinforcement learning, rate adaption, frame aggregation, joint optimization**

I. Introduction

To meet the growing demand for higher wireless transmission rate, IEEE has continuously promoted 802.11 protocol extensions to 802.11be [1] and 802.11bn [2], also named Extremely High Throughput (EHT) and Ultra High Reliability (UHR) wireless networks. Several enhanced mechanisms in physical (PHY) layer and media access control (MAC) layer are introduced to improve the performance of Wi-Fi network, such as throughput, delay, reliability, etc. One of the key features in MAC layer is frame aggregation mechanism, including two basic methods: the aggregated MAC service data unit (A-MSDU) and the aggregated MAC protocol data unit (A-MPDU). Aggregation mechanism could merge preambles and headers altogether, and concatenate data frames in a single transmission. Thus, the transmission overhead can be reduced, and MAC efficiency can be significantly improved.

Furthermore, several PHY layer enhancements including channel bonding, short guard interval (GI), multiple input multiple output (MIMO) spatial streams, and more advanced modulation and coding schemes (MCS) are brought forward. These enhancements have significantly increased the PHY rate, which currently reaches the levels of up to Gigabits per second (Gbps), thereby leads to a higher throughput.

However, it is important to note that all of these features involve certain tradeoffs between achieving higher throughput and facing a higher error rate. More specifically, longer aggregated frames can reduce more transmission overhead, but also increase packet loss rate [3]–[5]. Particularly, only A-MSDU suffers in error-prone channel since it contains only one frame check sequence (FCS) for the whole aggregated packet. Similarly, tradeoff also exists in PHY rate selection [6], [7]. High PHY rate can lead to high throughput, but also increase bit error rate (BER). While low PHY rate may suffer from poor channel utilization and thus reduce throughput. In addition, there is a certain coupling relationship between the aggregation frame length and the physical rate, since both of them impact the packet error rate. The optimal frame length varies with the selection of rate [8], and vice versa.

Considering above, optimal frame length and rate need to be determined jointly to address the tradeoffs. While such large number of enhanced features and available sets makes finding the best configuration for frame length and rate an extremely difficult challenge for conventional algorithms. However, recent researches have disclosed that, machine learning algorithms are well-suited for addressing multi-parameter optimization problems. Moreover, IEEE has established a Topic Interest Group (TIG) for Artificial Intelligence and Machine Learning (AIML) in July 2022, which aims to apply ML technique to optimize network performance in future Wi-Fi standard.

Although there are many research works in using ML to optimize frame length or PHY rate, few have taken them into consideration jointly. Chen et al. [6] models rate adaption as a 3D maze problem, where the coordinates correspond to spatial stream, bandwidth and MCS respectively. Two Q-network, namely, DDQN and DQN, are applied to solve it. However the frame length is not involved. Karmakar et al. [7] proposed a multi-armed bandit (MAB) based online-learning mechanism to optimize link configurations. The authors further introduced an improved Thompson sampling algorithm in [9] and fuzzy logic algorithm in [10] to improve their MAB based scheme to solve link adaption. However, all of them focus on A-MPDU aggregation instead of A-MSDU. Coronado et al. [4], [5] applied two supervised learning (SL) models, namely,

The work was supported in part by NSFC under Grant 62071393, Sichuan Provincial Applied Basic Research Project under Grant 2020YJ0218, and Fundamental Research Funds for the Central Universities under Grant 2682023ZTPY058.

RFR and M5P, in frame size selection according to the MCS and channel features. Nevertheless, this approach requires a preliminary dataset acquisition and cleaning, which is not universally applicable to Wi-Fi networks.

Motivated by the above observations, in this paper, we propose a joint frame length and rate adaption (JFRA) scheme to optimize the link throughput. Our main contributions are as follows:

- We formulate the joint frame length and rate adaption problem as a Markov decision process and apply double DQN algorithm [11] to solve it. Our proposed scheme can automatically explore the environment and learn the optimal frame length and rate from experience.
- We incorporate the reward value into the experience priority calculation for prioritized training [12], which improves the learning efficiency and accelerates the convergence of JFRA. The reward function is formulated by scaling the throughput based on the channel quality and transmission time, thus remains the same importance in different environments.

The rest of the paper is organized as follows. In Section II, we present the system model and formulate the joint frame length and rate adaption as a Markov decision process with state, action and reward function. In Section III, we describe the detailed design of the proposed JFRA scheme. Section IV illustrates the performance evaluation of the proposed JFRA scheme comparing to Minstrel HT [13] and Thompson Sampling [14] in terms of throughput and packet delay. Finally, we summarize this paper in Section V.

II. System model

The system model is shown in Fig. 1. We consider a Wi-Fi network consisting of a transmitter and a receiver. The transmitter sends traffic to the receiver, while N_{BG} additional stations, referred as background (BG) STAs, generate background traffic to simulate more realistic scenarios. The BG STAs act as interference nodes, which take part in channel contention and data transmission, but their throughput is not counted in performance evaluation. The BG STAs remain stationary, and the distance between transmitter and receiver could be either fixed or variable. In this paper, we suppose AP is the transmitter and the the receiver is referred as target STA. Our goal is to design a joint frame length and rate adaption scheme that maximizes the link throughput from the transmitter to the receiver.

The joint frame length and rate adaption problem can be modeled as a Markov decision process which consists of an agent, states, actions, and rewards. The agent is deployed at the transmitter. For each time t, the agent observes the channel environment at a state $s_t \in S$, and then selects an action a_t to adapt the frame length and rate to the environment. After the action-selecting, the agent receives a reward r_t and the next state s_{t+1} at the beginning of next time period. The transition $e_t\left(s_t, a_t, r_t, s_{t+1}\right)$ is then stored in experience buffer B_e for the DDQN model training.

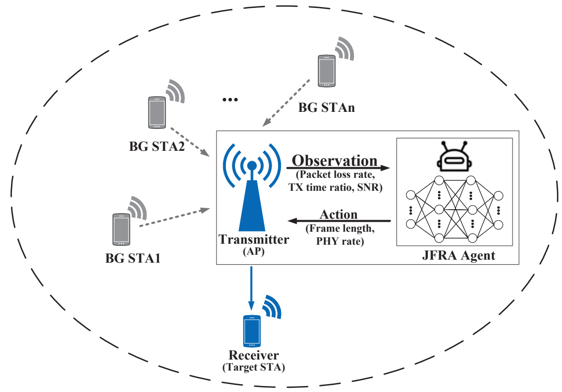


Fig. 1. System Model

The state space S describes the channel condition, where we use packet loss rate (PLR), signal-to-noise ratio (SNR) and transmission time ratio (TTR) as the indictors. The former two reflect the quality of the channel, while the latter reflects the contention level of the channel. Specifically, PLR is calculated via the number of transmitted frames n_{TX} and the number of received ACK frames n_{ACK} during the interaction time period as follows:

$$plr_t = 1 - n_{TX}/n_{ACK} \tag{1}$$

TTR represents the ratio of the transmission time to the interaction time period:

$$ttr_t = t_{TX}/t_{period} \tag{2}$$

where t_{TX} represents the transmitting duration to the receiver and t_{period} represents the time interval for each interaction. Then, the state can be formulated as: $s_t = (plr_t, snr_t, ttr_t)$.

The action space A maps directly to all of the possible combinations of available frame lengths and rates. The action for each time t can be formulated as: $a_t = (l_t, rate_t)$, where l_t represents the frame length and $rate_t$ represents the rate index. In this paper, we consider A-MSDU aggregation, thus the frame length l_t varies from $minMSDU$ to $maxMSDU$ in an interval of L_{seg}. While rate index $rate_t$ encompasses all the available physical rates determined by a tuple of PHY features:$\langle bw, gi, nss, mcs\rangle$, which corresponds to channel bandwidth, guard interval, number of spacial stream and mcs index respectively.

Reward function is designed considering the throughput performance. We normalize the throughput values to a real number between 0 and 1. The reward function at time t is formulated as:

$$r_t = \begin{cases} \frac{D_t}{rate_t^{ideal} \cdot ttr_t}, & \text{if } ttr_t > 0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where D_t represents the actual throughput and ttr_t represents the aforementioned transmission time ratio. The ideal rate $rate_t^{ideal}$ is determined by looking up the highest rate according to the current SNR and pre-set BER threshold in a BER-SNR table. This table is provided by IEEE P802.11 TGax

[15]. The ideal rate $rate_t^{ideal}$ represents the ideal throughput in different channel quality, while ttr_t adjusts the ideal throughput according to the channel contention level. Thus, the reward values can keep the same importance in different environments.

III. Design of JFRA Scheme

JFRA scheme includes two phases, namely, training phase and operating phase. In the training phase, JFRA agent is required to explore the action and learn from the experience. We add an additional noise to the selected action in order to enable exploration. The noise follows the normal distribution with the mean equal to zero and the standard deviation decreases as the number of training iterations increases. This is to balance the overhead of learning new experience and maximize the performance.

The large number of available frame lengths and rates set will increase the action space, thereby leading to a lengthy training process. Considering this, we apply prioritized training to improve sample efficiency and accelerate the algorithm convergence. Since the performance of Wi-Fi networks is mainly affected by the instant reward, we additionally incorporate the reward values in priority calculation. Besides, due to the consistent importance of the reward values, good actions can be prioritized for training regardless of the channel conditions. The priority value p_i of transition e_i is calculated as:

$$p_i = \alpha r_i + \left|Q_{target}\left(s_i, a_i\right) - Q_\theta\left(s_i, a_i\right)\right| + \varepsilon \tag{4}$$

where α is a discount factor of reward, the following absolute value is TD-error, and ε is a small real constant that prevents from never being sampled.

At every training step, a mini-batch b of transitions is sampled. The probability of each transition being sampled is proportional to its priority value:

$$\mathrm{P}_i = \frac{p_i}{\sum_k p_k} \tag{5}$$

where k is the total number of transitions in the experience buffer. The TD-error of each transition is recalculated to update its priority value after one round of training. In order to lower the time complexity of transitions retrieval to $O\left(\log n\right)$, the experience buffer is constructed as a sum-tree [12]. The pseudocode for the training phase of JFRA scheme is as Algorithm 1.

Once the training steps reach the threshold set by the user, JFRA scheme enters operating phase. Note that the agent will not perform exploration and updating during the operating phase. The action noise is set to zero and the agent always selects the best link configuration. It is worth remarking that the training phase can be restarted if significant changes occur in the environment and lead to a degradation in network performance.

Algorithm 1: The training process of JFRA scheme

```
Initialize experience buffer B_e constructed by a sum tree;
initialize Q-network with random weight θ and duplicate
to the target network θ⁻;
Define n_t and n_max as the current training steps and the
max training steps;
while t < n_max do
    Calculate snr_t, plr_t and ttr_t as s_t;
    sample noise_t ~ N(0, σ²);
    a_t ← argmax_{a_t} Q_θ(s_t, a_t) + noise_t;
    (l_t, rate_t) ← a_t;
    r_{t-1} ← Equation(3);
    if train then
        append (s_{t-1}, a_{t-1}, r_{t-1}, s_t) to B_e with max p;
        b ← prioritized samples from B_e;
        update θ ;
        soft update θ⁻;
        update p_i and B_e according to Equation(4);
    end
    σ ← σ(1 − n_t/n_max);
    s_{t-1} ← s_t;
    a_{t-1} ← a_t;
end
```

IV. Performance evaluation

We evaluate the performance of the proposed algorithm based on ns3-ai [16], a shared-memory based interface that provides efficient and high-speed data exchange between the Python-based AI frameworks and Network Simulator (ns-3). More specifically, we adopted ns-3.37 to simulate 802.11ax Wi-Fi network environment. JFRA scheme is implemented in PyTorch. Then, ns3-ai is used to connect the Wi-Fi network environment with JFRA scheme.

The network topology is shown in Fig. 1. All nodes are equipped with 802.11ax protocol. The main simulation parameters in ns-3 are shown in Table 1. Note that JFRA scheme can be easily extended to more rate features since it won't cause any qualitatively effect to our DDQN model.

TABLE I. ns-3 Simulation Parameters

Parameter	Value
Frequency	5 GHz
Channel bandwidth	20/40/80 MHz
Guard interval	3200 ns
Number of spatial streams	1
MCS index	HE-MCS 0 - HE-MCS 11
Max A-MSDU size	11398 Byte
Frame length interval L_{seg}	2000 Byte
payload size	1464 Byte
traffic source	UDP traffic
traffic rate	200 Mbps
Distance of background STAs	5 m
Distance of target STA	[5, 15, 20, 25, 30, 35, 40] m
Simulation time	50 / 150 seconds
Error rate model	TableBasedErrorRateModel
Path loss model	LogDistancePropagationLossModel
Propagation delay model	ConstantSpeedPropagationDelayModel
Mobility model	ConstantPositionMobilityModel RandomWalk2dMobilityModel
Target STA velocity	Uniform random variable [2-5] m/s

The hyper-parameters of DDQN algorithms are shown in Table 2. The neural network architecture simply contains a hidden layer composed of two fully connected layers.

TABLE II. Hyper-Parameters of JFRA Scheme

Parameter	Value
Hidden layer dimension	64 / 128
Interaction period	20 ms
Learning rate	5×10^{-3}
Soft update factor τ	1×10^{-3}
Reward discount factor γ	0.3
Batch size	64
Experience buffer B_e size	5000
Priority factor α	0.5
Priority epsilon ε	1×10^{-2}

We evaluate the performance of JFRA in terms of MAC throughput and data packet delay of the target STA. Two cases are considered:

- A static scenario for different distances between AP and the target STA. The number of BG STAs is set to 10;
- A dynamic scenario in which the target STA moves back and forth within a range of 2 to 40 meters. The velocity is determined by a uniform random variable within a range of 2 to 5 meters per second. The number of BG STAs is set to zero.

We compare the performance of the proposed JFRA scheme to two baseline algorithms: Minstrel HT and Thompson Sampling (TS), which are the default link adaption mechanism in Linux system, and a multi-armed bandit based reinforcement learning algorithm respectively. Minstrel HT adaptively select the most suitable transmission rate for each STA by tracking the probability of successfully sending a frame with each available rate. While Thompson Sampling maintains the number of successful and failed transmissions as the shape parameters of beta distribution for each rate. To select rate for transmission, TS samples a frame success rate from the beta distribution for each rate and then selects the one with the highest expected throughput. The frame length is set to the maximum size in both baseline algorithms. Every simulation is run 10 times and average results are utilized to eliminate the effect of randomness.

A. Static scenario

Considering that each algorithm requires a data statistic process (Minstrel HT, TS) or a training process (JFRA) to reach the best performance, the simulation is divided into two parts: the first 10 seconds are used for a warm-up for each algorithm and evaluation is performed during the last 10 seconds. The throughput and delay comparison in static scenario are shown in Fig. 2 and Fig. 3 respectively. Powered by the DDQN model, JFRA can learn the best frame length and rate in different channel condition. Minstrel HT suffers from the exhausted random sample for other rates, whereas JFRA always chooses the best data rate after sufficient exploration. Comparing to Minstrel HT, JFRA achieves a throughput gain from 12.6% to 21.3%, and reduces the delay by 4.2% to 18.7%. The performance of TS severely deteriorates since it fails to handle the transmission errors caused by frame collisions. This will impact the actual success rate distribution of each rate, leading to inaccurate rate selection. However, JFRA can correctly learn the contending environment and select the proper rate and frame length for transmission.

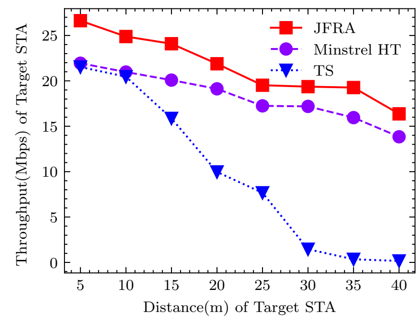


Fig. 2. Throughput Comparison in Static Scenario

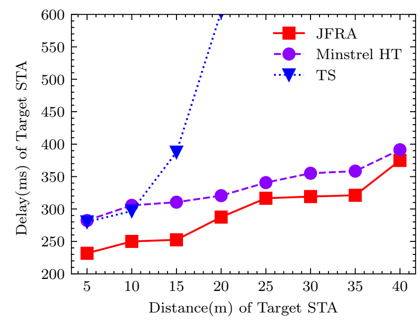


Fig. 3. Delay Comparison in Static Scenario

B. Dynamic scenario

The mobility of target STA will cause the variation of channel quality. Similarly, all the algorithms have been through a training process with same duration. The evaluation results of throughput and delay are shown in Fig. 4 and Fig. 5, respectively. It can be seen that JFRA increases the throughput by up to 60.4% comparing to TS, and up to 68.9% comparing to Minstrel HT. Meanwhile, JFRA reduces the delay by up to 38.2% comparing to TS, and up to 63.3% comparing to Minstrel HT. This is because by learning the experience, JFRA can timely select the best frame length and rate to adapt to the varying channel, also avoid unnecessary rate exploration. Through appropriate combination of frame length and data rate, JFRA achieves a tradeoff between improving throughput and reducing packet loss rate.

Furthermore, we analysis the algorithm convergence in dynamic scenario by observing the throughput and the accumulated reward. As is shown in Fig. 6(a), the throughput fluctuates drastically at the beginning due to the considerable action noise used in the initialization of the algorithm. The overall level of throughput is relatively low since most of the actions selected are used for exploration. The continually

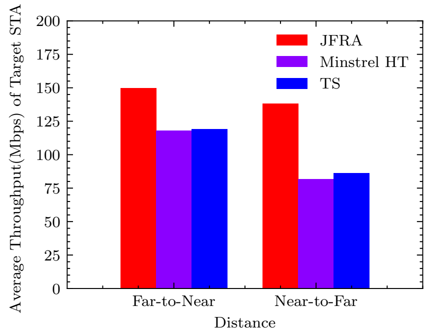


Fig. 4. Average Throughput in Dynamic Scenario

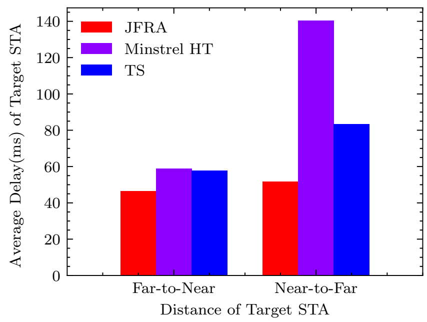
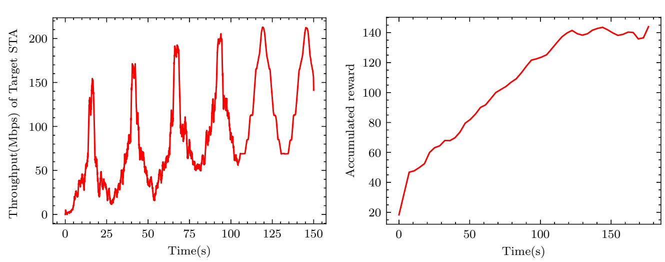


Fig. 5. Average Delay in Dynamic Scenario

increasing reward in Fig. 6(b) indicates that JFRA agent is performing better and better in frame length and rate selection as the number of training iterations increases. Through reward and TD-error based prioritized training, JFRA agent rapidly learns effective actions for different channel quality. After 110 seconds, JFRA agent completes the training phase and starts operating phase. Now JFRA agent consistently chooses the best action, so both throughput and reward reach to a high and stable state. Note that in static scenario, it only takes up to 10s for JFRA scheme to converge due to the simplicity of the stationary STAs and the fixed channel quality.



(a) Throughput Vs. Time (b) Accumulated reward Vs. Time

Fig. 6. Convergence of JFRA in dynamic scenario

V. Conclusion

In this paper, we propose a joint frame length and rate adaption scheme for 802.11 network based on the DDQN. The proposed JFRA scheme can dynamically select the frame length and data rate to adapt to the environment by learning the experience, thus address the tradeoffs in frame aggregation and several rate features. It balances the overhead of exploration and exploitation through a decaying action noise. The reward and TD-error based prioritized training helps JFRA rapidly converge to the best frame length and rate in a large number of available sets. We evaluate the JFRA in ns3-ai framework and compare to Minstrel HT and Thompson Sampling algorithm in both static scenario and dynamic scenario. The results show that our proposed scheme outperforms the other two algorithms, especially in dynamic scenario. The gain can be up to 68.9% in terms of throughput and 63.3% in terms of delay. This is attributed to its learning and dynamically adaptive capabilities. The possible further studies include implementing JFRA scheme on commodity hardware, and evaluating the overall system performance with multiple distributed JFRA agents operating within a same Wi-Fi network.

References

[1] "IEEE Draft Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment: Enhancements for Extremely High Throughput (EHT)," *IEEE P802.11be/D3.0, January 2023*, pp. 1–999, 2023.

[2] L. Galati Giordano, G. Geraci, M. Carrascosa, and B. Bellalta, "What will Wi-Fi 8 be? A primer on IEEE 802.11 bn ultra high reliability," *arXiv e-prints*, pp. arXiv–2303, 2023.

[3] G. Bhanage, "Amsdu vs ampdu: A brief tutorial on wifi aggregation support," *arXiv preprint arXiv:1704.07015*, 2017.

[4] E. Coronado, A. Thomas, and R. Riggio, "Adaptive ML-based frame length optimisation in enterprise SD-WLANs," *Journal of Network and Systems Management*, vol. 28, no. 4, pp. 850–881, 2020.

[5] E. Coronado, A. Thomas, S. Bayhan, and R. Riggio, "aios: An intelligence layer for sd-wlans," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020, pp. 1–9.

[6] S.-C. Chen, C.-Y. Li, and C.-H. Chiu, "An experience driven design for IEEE 802.11 ac rate adaptation based on reinforcement learning," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[7] R. Karmakar, S. Chattopadhyay, and S. Chakraborty, "Dynamic link adaptation in IEEE 802.11 ac: A distributed learning based approach," in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. IEEE, 2016, pp. 87–94.

[8] C. Shin, H. Park, and H. M. Kwon, "PHY-supported frame aggregation for wireless local area networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2369–2381, 2014.

[9] R. Karmakar, S. Chattopadhyay, and S. Chakraborty, "IEEE 802.11 ac link adaptation under mobility," in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*. IEEE, 2017, pp. 392–400.

[10] R. Karmakar, S. Chattopadhyay, and S. Chakraborty, "An online learning approach for auto link-Configuration in IEEE 802.11 ac wireless networks," *Computer Networks*, vol. 181, p. 107426, 2020.

[11] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[12] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[13] F. Fietkau and D. Smithies, "minstrel ht: New rate control module for 802.11 n," *Articles/376765*, 2010.

[14] A. Krotov, A. Kiryanov, and E. Khorov, "Rate control with spatial reuse for Wi-Fi 6 dense deployments," *IEEE Access*, vol. 8, pp. 168 898–168 909, 2020.

[15] R. Porat, M. Fischer, S. Merlin, S. Vermani, E. Au, D. Yangxun, J. Zhang, J. Luo, J. Pang, R. Stacy *et al.*, "11ax evaluation methodology," *doc. IEEE*, pp. 802–11, 2016.

[16] H. Yin, P. Liu, K. Liu, L. Cao, L. Zhang, Y. Gao, and X. Hei, "ns3-ai: Fostering artificial intelligence algorithms for networking research," in *Proceedings of the 2020 Workshop on ns-3*, 2020, pp. 57–64.