Deep Reinforcement Learning-Based Scheduling for Wi-Fi Multi-Access Point Coordination

David Nunez[⋆], Francesc Wilhelmi[⋆], Maksymilian Wojnar[♯], Katarzyna Kosek-Szott[♯], Szymon Szott[♯] and Boris Bellalta[⋆],

[⋆]*Wireless Networking, Universitat Pompeu Fabra, Barcelona, Spain*
[♯]*AGH University of Krakow, Poland*

***Abstract*—Multi-access point coordination (MAPC) is a key feature of IEEE 802.11bn, with a potential impact on future Wi-Fi networks. MAPC enables joint scheduling decisions across multiple access points (APs) to improve throughput, latency, and reliability in dense Wi-Fi deployments. However, implementing efficient scheduling policies under diverse traffic and interference conditions in overlapping basic service sets (OBSSs) remains a complex task. This paper presents a method to minimize the network-wide worst-case latency by formulating MAPC scheduling as a sequential decision-making problem and proposing a deep reinforcement learning (DRL) mechanism to minimize worst-case delays in OBSS deployments. Specifically, we train a DRL agent using proximal policy optimization (PPO) within an 802.11bn-compatible Gymnasium environment. This environment provides observations of queue states, delay metrics, and channel conditions, enabling the agent to schedule multiple AP-station pairs to transmit simultaneously by leveraging spatial reuse (SR) groups. Simulations demonstrate that our proposed solution outperforms state-of-the-art heuristic strategies across a wide range of network loads and traffic patterns. The trained machine learning (ML) models consistently achieve lower 99th-percentile delays, showing up to a 30% improvement over the best baseline.**

***Index Terms*—coordinated spatial reuse, IEEE 802.11bn, machine learning, multi-access point coordination, reinforcement learning, scheduling, Wi-Fi 8.**

I. Introduction

The rapid growth of wireless traffic, the rise of latency-sensitive applications, and the increasing density of user devices are imposing unprecedented demands on Wi-Fi networks. Traditional contention-based channel access mechanisms do not scale effectively in dense deployments with diverse traffic patterns and high levels of co-channel interference [1]. These limitations lead to poor quality of service (QoS), large delays, and inefficient spectrum utilization. Future Wi-Fi technologies must embrace more deterministic channel access and latency-sensitive scheduling to meet the demands of emerging use cases such as augmented reality, industrial automation, and dense public venues [2].

In this context, multi-access point coordination (MAPC), introduced in IEEE 802.11bn, represents a fundamental architectural shift from uncoordinated to coordinated spectrum sharing. By enabling centralized scheduling decisions across multiple access points (APs), MAPC facilitates joint optimization of time, frequency, and spatial resources. Coordinated spatial reuse (Co-SR) stems as a specific MAPC technique to allow multiple AP–station (STA)[1] pairs to transmit concurrently by coordinating spatial reuse (SR) opportunities across the network [3]. Unlike time—or frequency—division coordination [4], Co-SR aims to maximize spectral efficiency by allowing overlapping transmissions whereby interference constraints are satisfied. Among the various MAPC strategies, Co-SR offers a compelling trade-off between complexity and performance, making it particularly well-suited for latency-sensitive applications in high-density scenarios while keeping implementation costs low. For this reason, our analysis focuses on MAPC networks employing Co-SR as the underlying coordination mechanism. However, the effectiveness of MAPC networks using Co-SR hinges not only on the feasibility and quality of parallel transmissions but also on the ability to make timely and informed scheduling decisions under diverse traffic and channel conditions. Designing efficient MAPC schedulers remains a challenging task. The scheduler should make decisions influenced by traffic, signal quality, and interference for various independent basic service sets (BSSs), resulting in complex relationships that are hard to capture by heuristic algorithms, such as the use of maximum number of packets (MNP), oldest packet (OP), and traffic-alignment tracker (TAT) scheduling strategies for Co-SR [5].

In this work, we address the MAPC scheduling problem by formulating it as a sequential decision-making task and proposing a model capable of learning scheduling policies from real-time network observations. The proposed solution consists of a deep reinforcement learning (DRL) agent that interacts with an environment through a standardized Gymnasium interface.[2] Our main aim lies in demonstrating that the learned policies consistently outperform heuristic-based schedulers, such as those previously proposed in [5], by effectively adapting to diverse topologies, loads, and traffic patterns. Our results show that the proposed model achieves up to a 30% reduction in worst-case delay compared to the best-performing non-ML baseline. The main contributions of this paper are summarized as follows:

[1]In the IEEE 802.11 standard, client devices are referred to as non-AP STAs, which we refer to as STAs in the remainder of this paper.

[2]https://gymnasium.farama.org/

- We formulate the 802.11bn MAPC scheduling problem as a sequential decision-making optimization problem.
- We propose a DRL solution to learn dynamic MAPC scheduling policies based on real-time metrics such as the queueing delays of the entire overlapping basic service set (OBSS).
- We develop a novel Gymnasium-compatible simulation environment that integrates AI-driven decision-making with a detailed 802.11bn-compliant simulator, enabling the training and evaluation of DRL-based MAPC schedulers under diverse traffic and channel conditions.
- We showcase the performance of our proposed DRL-based scheduling solution and compare it to heuristic baselines such as MNP, OP, and TAT, in a comprehensive set of simulation scenarios.

The rest of this paper is structured as follows. Section II delves into the state-of-the-art solutions on MAPC and Co-SR. Section III discusses the IEEE 802.11bn MAPC setup and our scheme for creating SR-compatible groups of devices from multiple BSSs. Then, Section IV describes the proposed machine learning (ML) setup and DRL-based scheduling solution. The evaluation setup and simulation details are provided in Section V, whereas the simulation results are presented in Section VI. Section VII concludes the paper.

II. Related Work

Since its introduction by the IEEE 802.11ax standard, SR has received a lot of attention, and many works have attempted to improve its performance by adopting ML techniques. Examples include [6]–[9], where multi-armed bandit (MAB) solutions drive the selection of SR parameters. MABs are a suitable framework for online decision-making due to their simplicity and effectiveness, thus being a good match for Wi-Fi's MAC optimization.

Learning-based approaches have also been explored to tackle the inherent complexity and the combinatorial action space of MAPC networks. In [10], the authors showcase the potential of reinforcement learning (RL)-based solutions, including decentralized and coordinated approaches, for driving the optimization of SR in dense Wi-Fi deployments. Later, the authors propose a multi-agent multi-armed bandit (MA-MAB) approach to jointly configure SR parameters—specifically packet detection thresholds and transmit power—across multiple coexisting APs, based on coordinated (shared) reward strategies [11]. Their results show significant gains in throughput and fairness compared to uncoordinated baselines. Similarly, [12] introduces a hierarchical MAB framework for Co-SR group selection in IEEE 802.11bn. Using RL to select compatible AP subsets, the study finds that upper confidence bound (UCB)-based bandits offer fast convergence and sustained performance across heterogeneous topologies. Finally, the work in [13] addresses the Co-SR scheduling problem in IEEE 802.11bn by proposing a practical solution based on MABs, including both flat and hierarchical variants. Simulations and testbed experiments show that hierarchical MABs can improve aggregate throughput by up to 80% over legacy IEEE 802.11, without compromising the number of assigned per-STA transmission opportunities.

Regarding Wi-Fi scheduling, the main focus of this paper, recent efforts have been put into ML as well. In [14], an RL-based scheduling framework is proposed and implemented to optimize the application-layer QoS of a Wi-Fi network with commercial devices. The method adjusts contention window sizes and throughput limits using a Q-network trained from historical QoS feedback, achieving significant improvements over enhanced distributed channel access (EDCA) in a real testbed. In addition, the authors in [15] use deep learning to schedule transmissions based solely on node location, bypassing channel estimation and handling dense interference scenarios. Linked closely to this paper, [16] proposes a DRL-based solution to optimize SR in 802.11bn networks. However, the approach considered in [16] is substantially different from ours, as DRL decisions are used to perform channel access and link adaptation. In this paper, instead, we focus on the selection of suitable SR groups for transmission, as part of a DRL-based MAPC scheduler. This is something that has already been tackled in [5], where non-artificial intelligence (AI) schedulers are evaluated. Our paper advances the state-of-the-art by demonstrating the potential of ML solutions, which are expected to effectively address the increasing complexity in multi-AP scenarios.

III. Multi-AP Coordination in Wi-Fi 8: Overview and System Model

A. Main Operation

Based on ongoing discussions about a unified MAPC framework for Wi-Fi 8 [17], Fig. 1 exemplifies a plausible mechanism for enabling Co-SR in a group of four APs that can potentially transmit to their corresponding STAs. We consider only downlink transmissions in accordance with the MAPC specification in IEEE 802.11bn. The figure illustrates the interaction among the coordinated APs over the course of two transmission opportunities (TXOPs). All the APs are within mutual communication range and share a common wireless channel, contending for access using the distributed coordination function (DCF), which involves selecting a random backoff value and decrementing it while the channel is sensed to be idle before transmitting on the channel.

To facilitate the coordination among APs, once the backoff counter reaches zero, the AP that wins the contention acquires the role of Sharing AP—AP_1 in TXOP #1 and AP_4 in TXOP #2 (Fig. 1). The Sharing AP is responsible for initiating channel reservation by transmitting a MAPC-initial control frame (MAPC-ICF). This control frame, among other operations (e.g., indicating the MAPC feature to be used), is proposed here to solicit buffer status information from neighboring APs—called Shared APs—within the MAPC coordination range. In response, these Shared APs transmit MAPC-initial control response (MAPC-ICR) messages, which contain their intention to participate and, for the sake of enabling the mechanism proposed here, also include the number of buffered packets ϱ_i and the timestamp of the head-of-line

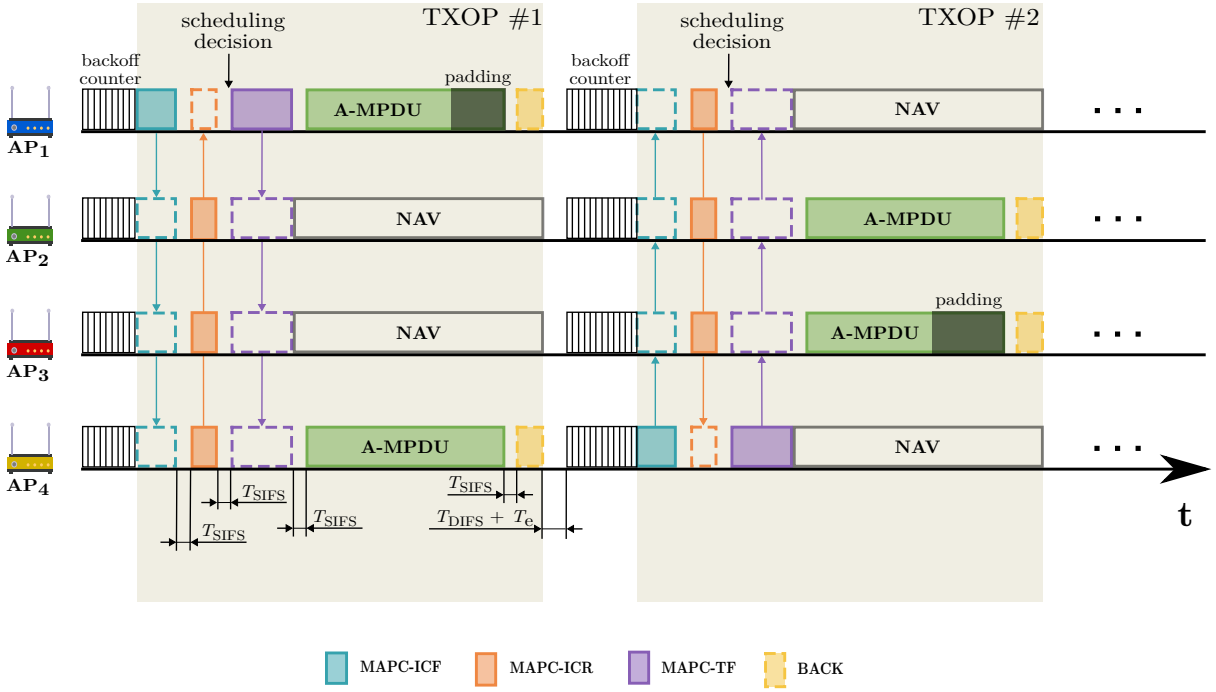


Figure 1: The proposed MAPC network.

(HoL) packet ε_i for each of their associated STAs. Orthogonal frequency-division multiple access (OFDMA) is used to enable the simultaneous transmission of MAPC-ICR frames.

While contention-based access enables flexible medium sharing, it can also result in collisions and delay fluctuations. A collision is assumed to occur when no MAPC-ICR responses are received within a timeout equal to $T_{\text{MAPC-ICF}} + T_{\text{SIFS}} + T_{\text{MAPC-ICR}} + T_{\text{DIFS}} + T_{\text{e}}$, where T_{SIFS} and T_{DIFS} denote the short interframe space (SIFS) and DCF interframe space (DIFS), respectively, and T_{e} represents a backoff slot time. In such cases—indicative of two or more devices in the OBSS having chosen the same backoff value—each affected device increases its contention window (CW) and draws a new random backoff value within the updated CW range. The contention process is then restarted using the new backoff value.

On the other hand, if the channel access attempt is successful, the Sharing AP proceeds to parse the received MAPC-ICR messages and selects the group of STAs to serve in the downlink during the current TXOP, based on the defined scheduling policy. This decision is made independently at each TXOP.

Once a group of potential SR transmissions is selected (including multiple APs and their corresponding STAs), the Sharing AP broadcasts a MAPC-trigger frame (MAPC-TF) that identifies the selected STAs (it may contain a single STA if that yields the best outcome according to the scheduler) and specifies their transmission parameters, including bandwidth, modulation and coding scheme (MCS), and the duration of the TXOP. Following this trigger, the chosen devices transmit their buffered data using an aggregate MAC protocol data unit (A-MPDU), while the remaining devices set their network allocation vectors (NAVs) to defer access until the channel is expected to become idle again. Note that in the proposed framework, channel access follows the legacy contention mechanism, i.e., DCF. However, the set of selected devices for transmission does not necessarily include the Sharing AP, as illustrated in TXOP #2 of Fig. 1. Finally, each STA sends back its corresponding block acknowledgment (BACK) upon successful reception. Any frame not successfully received is retained in the buffer and reattempted in the STA's next scheduled TXOP.

B. SR Group Creation

To enable Co-SR in multi-AP networks, we precompute a set of feasible SR groups as in [5].[3] Each group consists of a subset of AP–STA pairs that can be scheduled for concurrent transmissions, but guaranteeing certain quality in the transmissions. The feasibility of a group is evaluated by estimating the transmission rate used to serve each STA i in the group n, which yields

$$R_{i,n}^{\text{CoSR}} = \frac{N_{\text{bps}} R_{\text{c}} N_{\text{sc}} N_{\text{ss}}}{T_{\text{OFDM}} + T_{\text{GI}}}, \tag{1}$$

[3]Alternative approaches for spatial reuse group creation, such as those presented in [12], [13], are also viable. However, we adopt the method proposed in [5] to ensure an identical group selection process for both the ML-based models and heuristic algorithms.

where N_{sc}, N_{ss}, T_{OFDM}, and T_{GI} denote the number of data subcarriers, the number of spatial streams used, the duration of an orthogonal frequency-division multiplexing (OFDM) symbol, and the duration of guard intervals, respectively. The values N_{bps} and R_{c} correspond to the bits per symbol and the coding rate defined by the selected MCS for STA i, which is selected on a per-transmission basis according to the expected signal-to-interference-plus-noise ratio (SINR).

Let $\mathcal{M}_n$ be the subset containing the STAs in group n. To assess the feasibility of concurrent transmissions, the performance of each STA i in $\mathcal{M}_n$ is compared to its single transmission case, denoted R_i^{ST}, which matches with the case in which the STA is served alone (without being exposed to inter-BSS interference). A group is admitted if the following condition is satisfied:

$$|\mathcal{M}_n| \frac{R_{i,n}^{\mathrm{CoSR}}}{R_i^{\mathrm{ST}}} \geq 1, \quad \forall i \in \mathcal{M}_n. \tag{2}$$

Note that the equality in (2) defines the minimum acceptable transmission rate for an STA to be included in an SR group n. This condition ensures that, in the long run, under continuous high traffic loads, the per-STA mean throughput is at least as high as that achieved with single transmissions. The expression captures the trade-off between increased interference—due to concurrent transmissions within a group—and the efficiency gained by serving multiple STAs within the same TXOP. However, satisfying throughput alone does not guarantee that latency requirements are met. Therefore, smart scheduling mechanisms are necessary to effectively manage buffered traffic and minimize delay.

IV. Proposed DRL-based MAPC Scheduling Solution

This section provides an overview of the overall solution designed to integrate a DRL agent into the Wi-Fi's MAPC scheduling operation. As illustrated in Fig. 2, we define a system that comprises three core modules: *i)* the network environment, *ii)* the Gymnasium-compatible interface, and *iii)* the learning agent. Next, we define each block in detail.

A. Environment

The environment integrates an 802.11 simulator implemented in Python,[4] which characterizes particular 802.11 deployments and manages network operations such as traffic arrivals, channel conditions, and transmission events. For the integration of the 802.11 simulator and the DRL agent, we consider that, in each episode, a single deployment involving multiple APs and their associated STAs is simulated for a specific configuration (including a particular random traffic realization). The simulation proceeds in steps, each corresponding to a TXOP, as illustrated in Fig. 1, where multiple transmissions from different APs can be scheduled. Further details on the scenario configuration, channel and traffic models, and scheduling policies are provided in Section V.

[4]The simulator is available in https://github.com/dncuadrado/11bn-py-Simulator.

B. Gymnasium

To enable the training and evaluation of learning-based schedulers, we integrate the 802.11 simulator from Section IV-A into the Gymnasium application programming interface (API)—a standardized interface widely used for reinforcement learning environments [18]. This abstraction allows the interaction between the agent and the network environment to follow a consistent loop of observation, action, and reward, while decoupling the learning logic from the simulator's internal implementation.

At the beginning of each episode, the `reset()` function initializes the environment with a fresh deployment configuration. The initial observation returned to the agent reflects the current network state, including per-STA features such as queue size, delay metrics, and channel quality indicators.

As part of the interaction between the Gymnasium and the environment, the `step()` function is used by the Gymnasium to send the action selected by the agent (Section IV-C), which corresponds to the index of a valid SR group and is used by the simulator in the upcoming TXOP. The environment, upon executing the proposed action, updates the queue states and delay metrics, and returns the next observation, the reward signal, and the episode termination flags (`terminated` and `truncated`). This feedback enables the agent to learn which scheduling decisions are effective over time.

C. DRL-based MAPC scheduling agent

The problem of efficiently scheduling transmissions in an MAPC network can be formulated as a Markov decision process (MDP) [19], where an agent learns an optimal policy through interactions with the environment. As part of our solution, the agent is trained using the proximal policy optimization (PPO) algorithm, implemented in the actor-critic framework [20]. This approach maintains two distinct neural networks: an actor network, which maps the current observation s to a probability distribution over actions a, and a critic network, which estimates the expected return from observation s. Both networks are optimized simultaneously to improve policy quality while ensuring stable learning dynamics. The core components of this DRL framework are the observation space, the action space, and the reward function, which are elaborated below.

1) Observation Space: At each decision step with timestamp t, the agent receives an observation vector s containing normalized features from all N STAs in the network. The resulting observation vector, $s = [\bar{\boldsymbol{\delta}}, \bar{\boldsymbol{\varrho}}, \bar{\boldsymbol{h}}] \in [0,1]^{3N}$, is composed of the following information:

- **Delay vector** $\bar{\boldsymbol{\delta}} = [\delta_1, \ldots, \delta_N]/T_{\mathrm{sim}}$, where $\delta_i = (t - \varepsilon_i)$, $i = 1, ..., N$, is the per-STA elapsed time since the HoL packet arrived to their corresponding AP queue, normalized by episode duration, T_{sim}.
- **Queue size vector** $\bar{\boldsymbol{\varrho}} = [\varrho_1, \ldots, \varrho_N]/\varrho_{\max}$, which represents the current occupancy of the transmission queue for each STA, ϱ_i (in number of frames) normalized by the maximum buffer length, $\varrho_{\max}$.

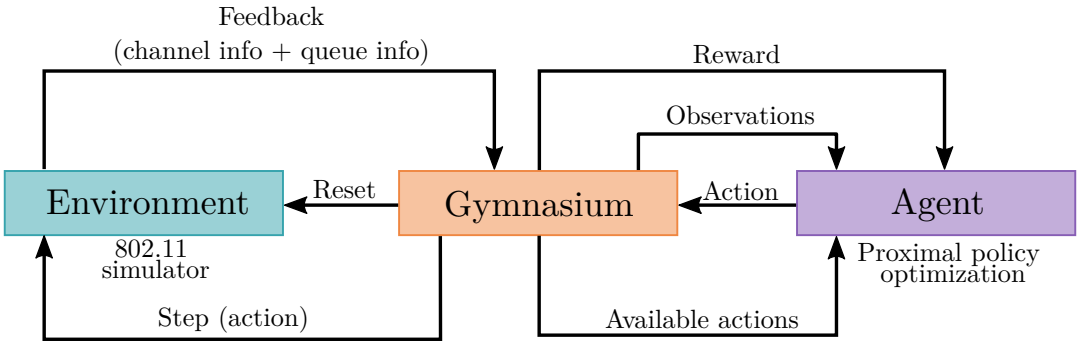


Figure 2: Proposed framework for the integration of ML into 802.11 networks.

- **Channel coefficient vector** $\bar{\boldsymbol{h}} = [h_1, \ldots, h_N]/h_{\max}$, which includes the per-STA link quality, i.e., the channel coefficients normalized by the path loss at a 1 meter distance in line of sight (LoS), $h_{\max}$.

The considered observations are passed through a shared feature extractor composed of a multi-layer perceptron (MLP) of two layers, where each hidden layer contains 64 units and employs the hyperbolic tangent (`Tanh`) activation function. The resulting feature vector is used as input to two output heads [21]:

- The **actor** computes action logits, which are transformed into a probability distribution over a discrete action space via a categorical distribution.
- The **critic** outputs a scalar value that estimates the expected cumulative reward from observation s, guiding the policy update by reducing variance in the gradient estimates.

2) Action Space and Masking: The discrete action space $\mathcal{A}$ is composed of $Z > 0$ actions, i.e., valid SR groups. For the sake of efficiency, at each step, a binary mask $m \in \{0,1\}^Z$ dynamically filters out infeasible or non-beneficial actions.[5] An action a_z is masked (i.e., $m_z = 0$) if any of the following conditions apply:

1) *Offline filtering:* The group is preemptively discarded for the entire deployment realization if it contains at least one receiver that does not meet the condition in (2).
2) *Online masking:* At each scheduling decision, the group is masked if none of the intended receivers have pending packets in their respective transmitter queues.

Masking ensures that the agent selects only from a subset of valid and effective scheduling actions at each step, which considerably reduces the exploration space and accelerates convergence by guiding the agent away from invalid or non-productive actions.

3) Reward Design: To guide the agent toward reducing delays and accelerating learning, we construct a reward function that balances long-term feedback with immediate signals through reward shaping as follows:

$$r = r_{\text{sh}} + r_{\text{lg}},$$

[5]Implemented via `MaskablePPO` from the `sb3_contrib` library: https://github.com/Stable-Baselines-Team/stable-baselines3-contrib.

where r_{sh} is a reward shaping component that delivers immediate feedback for selecting actions aligned with the current network state. In contrast, r_{lg} reflects the cumulative effect of previous decisions on the system's overall delay. While r_{lg} is provided at every step (i.e., it is not sparse), its magnitude diminishes sharply when the worst-case queueing delay increases. Both terms are discussed in the following.

Reward shaping: This component allows the agent to recognize effective actions even when the overall delay remains high. It contributes a positive reward when the head-of-line packet, observed across all STAs at the previous decision point, is successfully transmitted during the current TXOP, resulting in:

$$r_{\text{sh}} = \min\{\boldsymbol{\varepsilon}\} - \min\{\boldsymbol{\varepsilon}'\},$$

where $\boldsymbol{\varepsilon}' = [\varepsilon'_1, \ldots, \varepsilon'_N]$ and $\boldsymbol{\varepsilon} = [\varepsilon_1, \ldots, \varepsilon_N]$, denote the head-of-line delays before and after the current TXOP, respectively. If the same packet remains at the head of the queue, indicating it was not dispatched, then $r_{\text{sh}} = 0$.

Long-term reward: This term encourages the agent to maintain queueing delays low in the long run:

$$r_{\text{lg}} = \min\left(\frac{\beta}{t - \min\{\boldsymbol{\varepsilon}\} + \nu}, 1\right),$$

where $\beta > 0$ is a scaling parameter, and $\nu > 0$ prevents indetermination in division by zero. The denominator captures the maximum system-wide queueing delay, with the term decreasing as delays increase.

At early stages of training, the knowledge collected by the agent might still be insufficient to determine the best actions and, thereby, delay tends to accumulate. The term r_{sh} enables the agent to recognize specific actions (such as clearing the oldest buffered packet) leading to meaningful improvements. As the policy matures, r_{lg} becomes more informative, encouraging sustainable low-delay operation. The combined reward, therefore, guides the learning process from initial exploration toward effective long-term scheduling strategies.

V. Evaluation Setup

In this section, we describe the details of our IEEE 802.11bn simulations and the considered scheduling techniques. The simulation parameters are summarized in Table I.

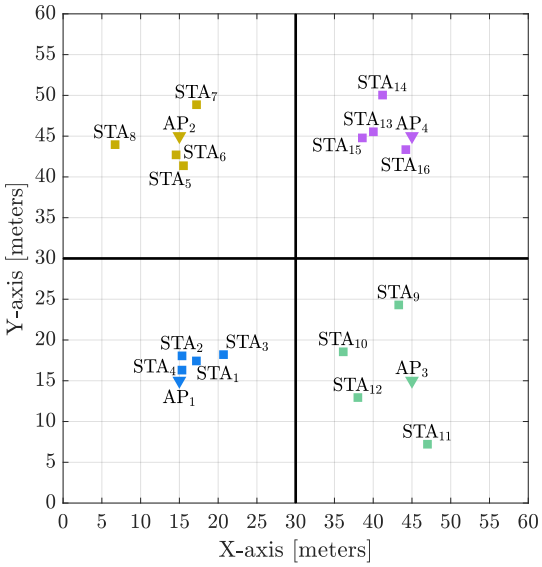


Figure 3: Sample deployment with 4 APs and 16 STAs.

Table I: Simulation parameters.

Parameter	Description	Value
d_{STA}	Distance between AP and STAs [meters]	[1, 10]
B_{p}	Break-point distance [meters]	10
B	Bandwidth [MHz]	80
f_{c}	Carrier frequency [GHz]	6
σ	Standard deviation of shadowing [dB]	5
N_{SC}	Number of data subcarriers	980
N_{SS}	Number of spatial streams	2
T_{OFDM}	OFDM symbol duration [μs]	12.8
T_{GI}	Guard interval duration [μs]	0.8
T_{max}	Max TXOP duration [ms]	5
$T_{\text{MAPC-ICF}}$	MAPC Initial Control Frame duration [μs]	74.4
$T_{\text{MAPC-ICR}}$	MAPC Initial Control Response duration [μs]	88
$T_{\text{MAPC-TF}}$	MAPC Trigger Frame duration [μs]	74.4
T_{BACK}	Block ACK [μs]	100
T_{SIFS}	Duration of a SIFS slot [μs]	16
T_{DIFS}	Duration of an DIFS slot [μs]:	34
T_{e}	Duration of an empty slot [μs]	9
CW_{min}	Min contention window	15
CW_{max}	Max contention window	1023
P_{max}	Max transmission power [mW]	200
ϱ_{max}	Max buffer size	10^4
h_{max}	Max channel gain	10^{-3}
W	Noise power [Watts]	3.2×10^{-13}
CCA	Clear channel assessment threshold [dBm]	-82
T_{ON}	Bursty traffic ON period mean duration [ms]	1
T_{OFF}	Bursty traffic OFF period mean duration [ms]	10
T_{sim}	Simulation duration [s]	5
L	Length of single data frame [bits]	12×10^3

A. Scenario

We consider an 802.11 enterprise scenario [22], where multiple APs coexist within a common area. As illustrated in Fig. 3, and for the evaluation of Co-SR, we focus only on a subset of $J = 4$ APs operating on the same frequency channel. They are deployed in distinct office rooms separated by walls, with an inter-AP distance of 30 meters. Each AP j has N_j associated STAs, placed randomly at distances $d_{\text{STA}} \in [1, 10]$ meters from their serving AP. For simplicity, we assume that all APs have the same number of associated STAs.

Assuming that all the transmissions utilize a fixed transmission power equal to P_{max}, we consider all the APs to be within the same communication range, so that they contend for the channel according to DCF. Once an AP wins access to the channel, it selects an SR group for transmission. The set of spatially compatible groups available in each deployment is precomputed using the strategy described in Section III-B.

We consider only downlink traffic, with heterogeneous traffic patterns across STAs. Specifically, the traffic profile of each STA is independently drawn from either a Poisson or a bursty traffic model (Section V-C), selected with equal probability. Furthermore, the offered load for STA i, ω_i, is independently and uniformly sampled from the range $[\omega_{\text{min}}, \omega_{\text{max}}]$. Additionally, only deployments in which at least one scheduling mechanism achieves a 99th-percentile delay below 100 ms are included in the results. The remaining cases are considered overloaded and are excluded from the analysis. Such deployments typically arise due to unfavorable STA placements or traffic patterns, making them unsuitable for meaningful latency evaluation. The proportion of discarded deployments is reported for each evaluated case.

B. Channel Model

The SINR perceived at STA i, denoted as ξ_i, depends on the subset $\mathcal{K}$ of simultaneously active interfering transmitters (if any), and is computed as:

$$\xi_i = \frac{P_j h_{i,j}}{W + \sum_{k \in \mathcal{K}} P_k h_{i,k}}, \tag{3}$$

where P_j is the power used by AP j to deliver the intended transmission to STA i, and W stands for the noise power. The terms P_k represent the power transmitted by interfering AP k, which contributes to the interference experienced at STA i. The channel gain $h_{i,j}$ between STA i and AP j is defined as $h_{i,j} = 10^{-\frac{P_{\text{L}}(d_{i,j})}{10}}$, a function of the transmitter-receiver separation $d_{i,j}$. Likewise, $h_{i,k}$ represents the channel gain from AP k to STA i, based on distance $d_{i,k}$. The term $P_{\text{L}}(d_{i,j})$ represents the path loss in decibels and is modeled using the TGax specification for enterprise environments [22]:

$$\begin{aligned} P_{\text{L}}(d_{i,j}) =& 40.05 + 20 \log_{10} \left(\frac{\min(d_{i,j}, B_{\text{p}}) f_{\text{c}}}{2.4} \right) \\ &+ \mathbb{1}_{d_{i,j} > B_{\text{p}}} P' + 7 W_{\text{n}} + \mathcal{X}, \end{aligned} \tag{4}$$

where $d_{i,j} \geq 1$ denotes the distance in meters, f_{c} is the carrier frequency in GHz, W_{n} is the number of intervening walls, and $P' = 35 \log_{10}(d_{i,j}/B_{\text{p}})$ accounts for extra loss beyond the break-point distance B_{p}. The term $\mathcal{X}$ models log-normal shadowing effects with $\ln(\mathcal{X}) \sim \mathcal{N}(0, \sigma)$.

We adopt the IEEE 802.11be MCS set (no major changes are expected in 802.11bn), and determine the appropriate MCS

index for each transmission using pre-generated packet error rate (PER) against signal plus noise ratio (SNR) curves gathered from MATLAB simulations [23]. These mappings reflect the simulation conditions used later in our study, including fixed packet size L, bandwidth B, channel model, and antenna settings. The selection criterion ensures that only MCS values satisfying $\mathrm{PER} < 10^{-2}$ for the estimated ξ_i are employed. As a result, STAs closer to their APs are assigned higher MCS values, while distant STAs operate with more robust (lower) MCS indexes.

The number of aggregated frames (A-MPDU size) sent to STA i per TXOP is denoted by U_i, which depends on the selected MCS and the TXOP duration. Interference from concurrent transmissions—i.e., $\sum_{k\in\mathcal{K}} P_k h_{i,k}$ in (3)—directly affects MCS selection and must be precisely accounted for when grouping STAs for SR compatibility.

The number of MAC protocol data units (MPDUs) successfully decoded by STA i is modeled as a binomial random variable, i.e., $\mu_i \sim B(U_i, q)$, where $q = 1 - \mathrm{PER}$ is the success probability and U_i the number of transmitted frames. Frames that STA i fails to decode are retained in its corresponding AP's buffer and are re-attempted[6] during the next TXOP scheduled for STA i.

C. Traffic Model

Downlink traffic is considered, and each AP_j maintains independent logical transmission queues for its associated N_j STAs, with incoming packets being buffered. We assume that each STA supports a single flow, and $\varrho_i(t)$ denotes the queue depth for STA i at time t. For each TXOP, the queue dynamics evolve according to

$$\varrho_i(t + T_\mathrm{s}) = \max\{\varrho_i(t) - \mu_i(T_\mathrm{s}), 0\} + \lambda_i(T_\mathrm{s}), \qquad (5)$$

where $\mu_i(T_\mathrm{s})$ is the number of successfully delivered frames during a TXOP of duration T_s (drawn from a random variable with expected value $\bar{\mu}$), and $\lambda_i(T_\mathrm{s})$ represents the number of arrivals during that same interval (drawn from a random variable with expected value $\bar{\lambda}$).

The following two traffic generation models are used to evaluate system performance, and in all cases, the load at STA i is defined as ω_i [Mb/s].

1) *Poisson*: Packet arrivals follow a Poisson process, resulting in exponentially distributed inter-arrival times for each STA.
2) *Bursty*: This model captures intermittent traffic where packet bursts occur in ON periods separated from OFF periods. Burst arrivals are modeled as a two-state (ON/OFF) Markovian process, with ON and OFF durations drawn from exponential distributions with means T_ON and T_OFF, respectively. During the ON periods, the traffic load aggregates all the arrivals from an equivalent Poisson process.

[6]We assume that retransmissions are not subject to any maximum attempt limit.

We assume equal priority for all packets within each STA queue and employ a first in first out (FIFO) scheduling policy for managing each queue.

D. MAPC Scheduling Mechanisms

We evaluate the performance of five different MAPC scheduling mechanisms, comprising three heuristics and two learning-based agents:

- *Maximum Number of Packets (MNP):* This scheduler selects the SR group with the largest number of packets in the current TXOP. For each feasible group, the number of schedulable packets is estimated based on the current queue lengths, regardless of packet age.
- *Oldest Packet (OP):* The OP policy prioritizes the SR group that includes the STA with the highest HoL packet delay. The goal is to reduce the worst-case delay by targeting the most delayed flow at each decision step. This mechanism implicitly promotes fairness but may underutilize TXOP capacity when the most delayed flows cannot be scheduled concurrently.
- *Traffic-Alignment Tracker (TAT):* Initially proposed in [5], TAT seeks to balance worst-case delay reduction and efficiency by selecting the SR group that minimizes a delay-alignment cost, i.e., the difference between the arrival times of the HoL packets among all the participants in a group. As a result, TAT tends to maintain stable delay distributions across heterogeneous deployments.
- *Machine Learning-General Agent (ML-G):* This DRL-based scheduler is trained to generalize across a wide range of traffic and deployment scenarios. At each decision step, the agent observes normalized queue lengths, HoL delays, and channel coefficients for all STAs, and selects a valid SR group using a policy trained via the PPO algorithm [20]. The agent's objective is to minimize the system-wide worst-case delay across all packets and STAs. Action masking is used to dynamically filter out infeasible or non-beneficial SR groups.
- *Machine Learning-Expert Agent (ML-E):* ML-E shares the same architecture, observation space, and masking procedure as ML-G, but it is trained under a specialized setting: a fixed deployment topology (corresponding to Fig. 3) with diverse traffic realizations. This focused training enables the agent to fine-tune its scheduling strategy to the spatial structure and interference patterns of a single environment, while still generalizing over different traffic conditions.

VI. Performance Evaluation

This section delves into the performance evaluation of the proposed DRL-based scheduling mechanism. We first show the process of training the ML schedulers and then examine their performance in new (unseen) deployments.

A. Training

The training of the DRL agent is conducted using the PPO algorithm, implemented via the `MaskablePPO` from the

`sb3_contrib` library [24]. The agent is trained within the custom Gymnasium environment described earlier, where each training episode includes the simulation of a Wi-Fi deployment, including STA placements with their corresponding channel conditions and the generation of new traffic patterns with per-STA load of ω_i [Mb/s], which remain fixed throughout the episode. Given that the channel model accounts solely for path loss (as a function of transmitter-receiver distance) and shadowing, we assume the channel conditions remain static throughout each episode. Moreover, the duration of each episode is the same as the duration of the simulation, that is, T_{sim}.

During the training of the model, we apply cosine annealing to gradually decay the learning rate over time, which is helpful to improve policy generalization and performance. The model is periodically evaluated, and training is halted when the agent converges to a stable policy (i.e., no improvements are observed after 20 successive evaluations) or upon reaching a predefined number of environment steps, n_{total}. Model checkpoints and performance metrics are logged using the Weights & Biases platform.[7] For better computational efficiency and faster convergence, we employ parallelized training using the `SubprocVecEnv` wrapper to instantiate multiple independent environments [24]. According to this, the agent interacts with a total of n_{env} parallel environments during each rollout phase. The values of the main hyperparameters used in training after tuning are reported in Table II.

Next, to assess the importance of the reward function, Fig. 4 shows the long-term reward evolution of two ML-G agents—one trained with reward shaping and one without—each evaluated across 10 independent training runs of 10^7 steps. As shown, the agent using reward shaping exhibits significantly faster initial learning, achieving higher rewards within the first 4 million steps. This indicates that the additional feedback guides early exploration toward more promising scheduling actions. While both agents eventually converge to similar performance levels around 6–10 million steps, the shaped reward continues to provide a slight advantage in terms of final training performance stability and peak reward. These results suggest that reward shaping not only accelerates convergence but also leads to better policy refinement in later training stages.

Analogously, Fig. 5 shows the actual worst-case delay performance (as the worst 99$^{\text{th}}$-percentile delay of each episode across all STAs) of the two agents when trained with and without reward shaping. The shaded regions represent the min–max bounds over the same 20 trainings previously analyzed (10 with and 10 without reward shaping), while the solid curves show the mean values. Both agents progressively reduce delay as training advances; however, the model with reward shaping converges significantly faster and reaches lower worst-case delay values throughout the entire training process. The agent without shaping exhibits higher variability and slower convergence, particularly during early stages. The

[7]https://wandb.ai/site

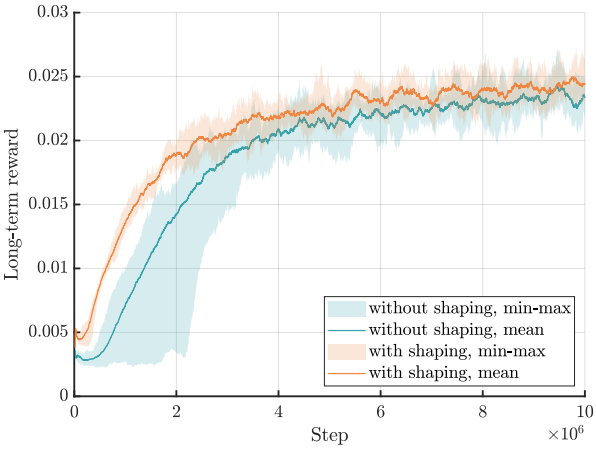


Figure 4: Long-term reward evolution during an ML-G agent training with/without reward shaping.

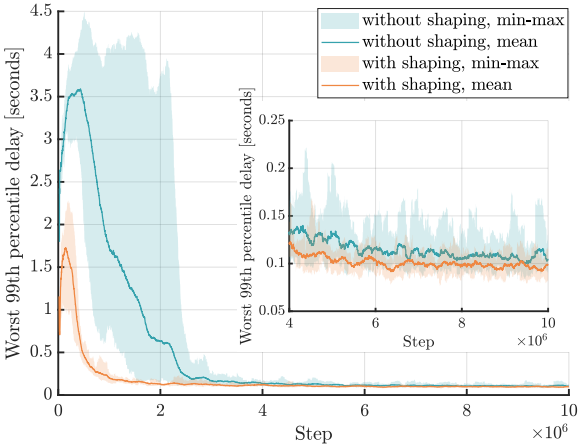


Figure 5: Evolution of the worst 99$^{\text{th}}$-percentile delay during an ML-G agent training with/without reward shaping.

inset plot on the bottom right provides a close-up of the stable region (from 4×10^6 to 10^7 steps), where the advantage of reward shaping remains consistent.

B. Inference Results

To validate our proposal, we first evaluate the performance of our trained DRL agents, as well as the heuristic algorithms, in the deployment shown in Fig. 3. Then, we extend the experiment to other randomly generated deployments to verify the adaptability and generalization of our proposal. Finally, we analyze how the ML-based agents scale as the number of users increases from $N = 8$ to $N = 20$.

1) Sample Deployment: Fig. 6 presents the 99$^{\text{th}}$-percentile and mean delays achieved by all the evaluated schedulers for 100 different traffic realizations in the deployment shown in Fig. 3, with 4 APs and 16 STAs. Every STA is independently assigned either a Bursty or Poisson traffic flow in each traffic realization, and its offered load ω_i is randomly drawn from the

Table II: Training parameters.

Parameter	Description	Value
`gamma`	Discount factor	0.99
`gae_lambda`	GAE (Generalized Advantage Estimation)	0.92
`n_steps`	Steps per environment per update	128
`batch_size`	Minibatch size	256
`clip_range`	Clipping parameter	0.2
α_{init}	Initial learning rate	6.5×10^{-4}
α_{schedule}	Learning rate schedule	Cosine decay
n_{env}	Number of parallel environments	10
n_{total}	Total number of steps	10^7
β	Scale factor for long-term reward	10^{-3}
ν	Factor to avoid indetermination	10^{-6}

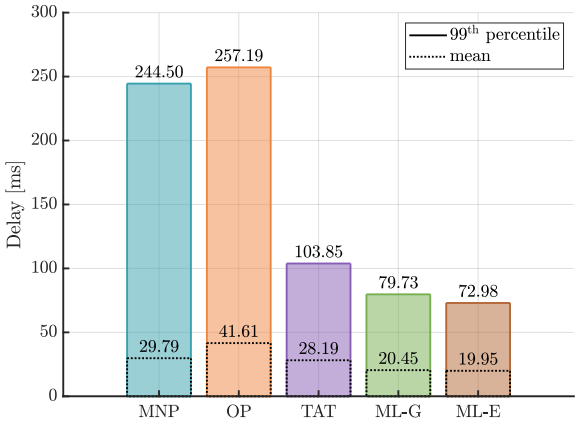


Figure 6: 99$^{\text{th}}$-percentile and mean delay for all the scheduling strategies in the sample deployment, after 100 traffic realizations, with $\omega_i \in [10, 90]$ and 10% of traffic realizations discarded.

range $[10, 90]$—which corresponds to a mean load of 50 Mb/s, comparable to the medium-high traffic regime discussed later in this section. Among the heuristic baselines, MNP and OP exhibit the highest worst-case delay, exceeding 240 ms. This confirms their poor adaptability in highly loaded or Bursty scenarios, where prioritizing either the aggregate queue length (MNP) or the head-of-line packet (OP) can lead to starvation and inefficient group selections. In contrast, TAT, which explicitly balances traffic alignment and efficient transmissions, significantly reduces the 99$^{\text{th}}$-percentile delay to 103.85 ms.

Both learning-based schedulers, ML-G and ML-E, outperform all heuristics by a wide margin. Notably, ML-E achieves the lowest worst-case delay of 72.98 ms, representing a reduction of 8% compared to ML-G, a 30% improvement over TAT, and over 70% compared to MNP and OP. These results suggest that the DRL agents effectively schedule groups under diverse traffic variations while balancing the worst-case delay and efficient transmissions. In both scenarios, the ML-based schedulers consistently achieve lower mean delay compared to all heuristic-based mechanisms.

Fig. 7 presents the normalized selection frequency per priority index for each evaluated scheduling strategy, measured in the deployment depicted in Fig. 3 with one traffic realization. The priority index ranges from low (L) to high (H), where L corresponds to selected actions that contain—at least—the STA with the lowest HoL delay, and H to those with the highest. Note that since buffer states continuously evolve as transmissions occur, the same action may correspond to different priority levels across successive TXOPs. The y-axis shows the fraction of TXOPs in which each priority level was selected. Note that MNP distributes its selections relatively evenly, reflecting its throughput-oriented design, while OP always concentrates on the highest priority, consistently serving the oldest packets, but at the cost of serving also STAs with low priority that also belong to the selected SR group. TAT also favors high priorities, though with a smoother distribution that reflects its delay-balancing mechanism. In contrast, ML-G exhibits a gradual increase across the priority spectrum, indicating a learned policy that balances delay reduction with broader scheduling diversity. Similarly, ML-E shows a high diversity but slightly sharper bias toward high-priority levels, due to its specialization in this deployment.

2) Random Deployments: Fig. 8 provides a detailed performance analysis by showing the distribution of worst-case delays across 100 random deployments. In each deployment, 4 APs are positioned as in Fig. 3, while $N_j = 4$ STAs per AP (totaling $N = 16$ stations) are randomly placed around their corresponding APs. We evaluate all scheduling strategies under three per-STA traffic load regimes, low: $\omega_i \in [10, 30]$, medium: $\omega_i \in [30, 50]$, and high: $\omega_i \in [50, 70]$. For low and medium traffic loads, all deployments are included in the analysis. Under high load conditions, however, 41% of the deployments are excluded due to network overload. Results for ML-E are omitted in this setting, as its performance degrades significantly when evaluated on deployments different from the one used during training.

Under low load, Fig. 8a, all schedulers maintain bounded delay, but MNP displays significantly higher variability, occasionally producing worst-case delays over 30 ms. The ML-based method exhibits tighter delay bounds overall, but does not outperform OP or TAT under low-load conditions, with OP emerging as the most effective scheduler in this regime, because the network is underutilized and the optimal solution is to serve the HoL packet.

As the load increases, Fig. 8b, non-ML schedulers begin to diverge. MNP and OP show higher median delay and a broader spread of outliers. TAT remains more stable, but ML-G consistently outperforms all baselines, with a narrower interquartile range and fewer outliers, because it balances addressing the worst-case differently, not always dispatching the oldest packet, as shown in Fig. 7.

Under high load, Fig. 8c, the performance gap between ML-G and baselines widens substantially. MNP and OP experience frequent worst-case delays above 100 ms. TAT degrades more gracefully but still trails the ML agent. ML-G maintains both a lower median and a tighter delay distribution, as it adopts a

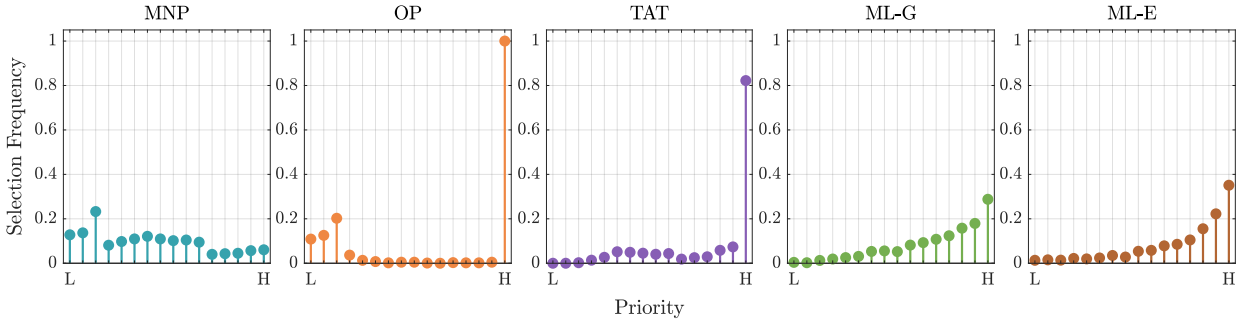


Figure 7: Normalized selection frequency per priority index for each evaluated scheduling strategy in the deployment of Fig. 3 and one traffic realization.

more flexible strategy for handling delay, rather than strictly serving the oldest packet at every decision step, confirming the learned policy's resilience under network stress.

Finally, Fig. 9 presents the 99th-percentile and mean delay achieved by the four scheduling strategies—MNP, OP, TAT, and ML-G—across 100 deployment realizations, where $\omega_i \in [10, 90]$—the same range used during training of the ML-G agent—and 15% of the deployments are excluded from the analysis due to overload. The results demonstrate that the proposed ML-based scheduler (ML-G) outperforms all heuristic baselines in terms of both mean and worst-case delay. Specifically, ML-G achieves the lowest 99th-percentile delay at 64.54 ms, representing a 15% improvement over TAT (75.92 ms), the best-performing heuristic policy. In terms of mean delay, ML-G also performs competitively, reaching 16.87 ms—slightly higher than MNP's 15.55 ms but significantly lower than OP (23.20 ms) and TAT (23.00 ms).

These results collectively demonstrate that the learning-based scheduler not only outperforms heuristic baselines in average and tail delay metrics but also exhibits superior stability across different load conditions. Its ability to capture cross-scenario scheduling patterns gives it a distinct advantage over fixed-priority policies.

3) Scalability: Fig. 10 shows the distribution of worst-case delays for the four scheduling strategies—MNP, OP, TAT, and ML-G—as the number of users increases, with $J = 4$ APs and $N \in \{8, 12, 16, 20\}$ users. Each boxplot aggregates the results from 100 random deployment realizations, with a portion of the deployments discarded due to overload (ranging from 4% for 8 STAs to 24% for 20 STAs). The total network load is kept constant (on average) across all scenarios and deployment realizations, with a mean network load of $\omega_{\text{net}} = 800$ Mb/s and a standard deviation of $\sigma_{\text{net}} = 92.4$ Mb/s. Note that this setup yields a per-user traffic distribution comparable to the previously analyzed case with 16 STAs, i.e., $\omega_i \in [10, 90]$ Mb/s. To match each configuration, four different ML-G agents were trained using the corresponding network load distributions.

Across different number of STAs, the ML-G agents consistently achieve a lower worst-case delay compared to the heuristic baselines. The performance gap becomes more pronounced as the number of STAs increases, reflecting ML-G's better adaptability to high-density scenarios. While TAT shows competitive behavior for $N \in \{8, 12, 16\}$, its performance deteriorates under scenarios with a higher number of users ($N = 20$). In contrast, ML-G maintains a tighter delay distribution under 100 ms (including outliers) in all evaluated scenarios, demonstrating robust queue management and scheduling decisions under varying levels of network density, and confirming that ML-based approaches scale well in terms of delay performance as user density increases.

VII. Conclusions

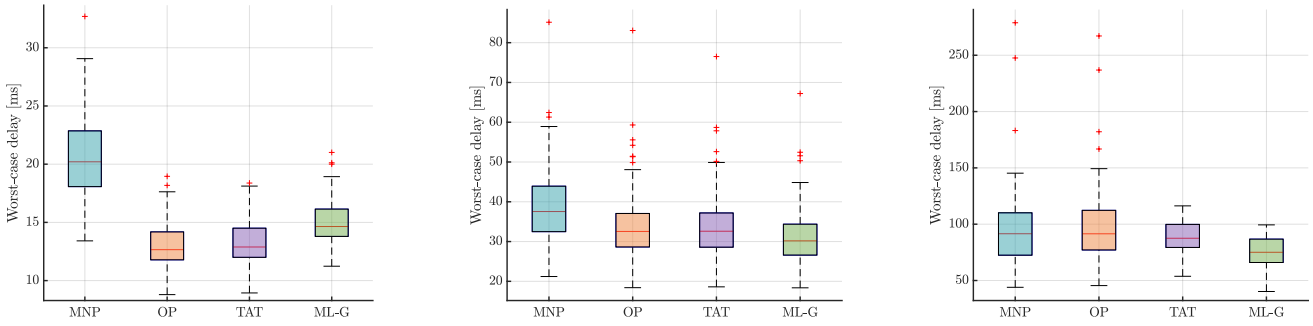
This work investigates the use of DRL to address the MAPC scheduling problem in IEEE 802.11bn Wi-Fi networks. We design and implement a DRL-based agent capable of observing per-station worst delay, queue size, and channel conditions, and making scheduling decisions aimed at minimizing worst-case delay. The agent is trained using PPO within a Gymnasium-compatible Wi-Fi environment that models Co-SR and heterogeneous traffic.

Our results show that the proposed model not only generalizes across diverse deployments and traffic patterns but also consistently outperforms three established MAPC scheduling heuristics (MNP, OP, and TAT). These findings highlight the potential of learning-based approaches to improve delay-sensitive performance in coordinated Wi-Fi networks, paving the way for more intelligent scheduling in next-generation wireless standards.

Advanced scheduling strategies—such as those involving group management and transmit power control—are left for future work, along with the integration of emerging features like multi-link operation (MLO). Additionally, developing a generalizable model capable of scaling to arbitrary network sizes, regardless of the number of APs and STAs, remains an open research direction.

VIII. Acknowledgments

This paper is supported by the CHIST-ERA Wireless AI 2022 call MLDR project (ANR-23-CHR4-0005), partially funded by AEI and NCN under projects PCI2023-145958-2 and 2023/05/Y/ST7/00004, respectively. The work of D.



(a) Low load (b) Medium load (c) High load

Figure 8: Worst-case delay distribution over 100 random deployments.

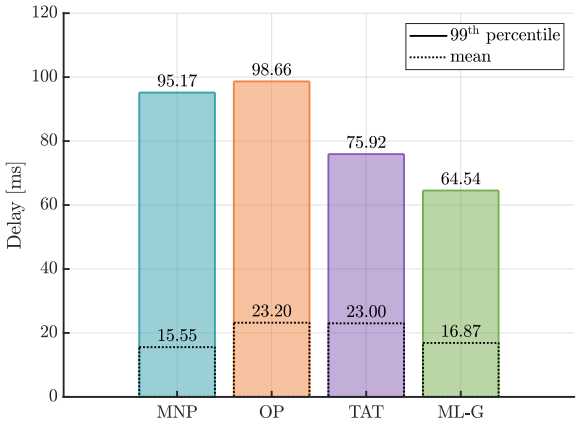


Figure 9: 99$^{\text{th}}$-percentile and mean delay for all the scheduling strategies after 100 deployments realizations, with $\omega_i \in [10, 90]$ and 15% of deployments discarded.

Nunez, F. Wilhelmi and B. Bellalta is also partially supported by Wi-XR PID2021-123995NB-I00 (MCIU/AEI/FEDER,UE), by MCIN/AEI under the Maria de Maeztu Units of Excellence Programme (CEX2021-001195-M), and ICREA Academia 00077.

References

[1] R. Costa, P. Portugal, F. Vasques, C. Montez, and R. Moraes, "Limitations of the IEEE 802.11 DCF, PCF, EDCA and HCCA to handle real-time traffic," in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*. IEEE, 2015, pp. 931–936.

[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," *IEEE Communications Magazine*, vol. 62, no. 8, pp. 126–132, 2024.

[3] D. Nunez, F. Wilhelmi, L. Galati-Giordano, G. Geraci, and B. Bellalta, "Spatial Reuse in IEEE 802.11bn Coordinated Multi-AP WLANs: A Throughput Analysis," 2024. [Online]. Available: https://arxiv.org/abs/2407.16390

[4] I. Val, D. López-Pérez, A. Kijanka, S. Schelstraete, L. Muñoz, D. Arlandis, and M. Martínez, "Wi-Fi 8 Unveiled: Key Features, Multi-AP Coordination, and the Role of C-TDMA," 2025.

[5] D. Nunez, P. Imputato, S. Avallone, M. Smith, and B. Bellalta, "Enabling Reliable Latency in Wi-Fi 8 Through Multi-AP Joint Scheduling," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 2090–2101, 2025.

[6] F. Wilhelmi, C. Cano, G. Neu, B. Bellalta, A. Jonsson, and S. Barrachina-Muñoz, "Collaborative spatial reuse in wireless networks via selfish multi-armed bandits," *Ad Hoc Networks*, vol. 88, pp. 129–141, 2019.

[7] F. Wilhelmi, S. Barrachina-Munoz, B. Bellalta, C. Cano, A. Jonsson, and G. Neu, "Potential and pitfalls of multi-armed bandits for decentralized spatial reuse in WLANs," *Journal of Network and Computer Applications*, vol. 127, pp. 26–42, 2019.

[8] A. Bardou, T. Begin, and A. Busson, "Improving the spatial reuse in IEEE 802.11 ax WLANs: A multi-armed bandit approach," in *Proceedings of the 24th International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 2021, pp. 135–144.

[9] Y. Huang, "Reinforcement Learning Approaches to Improve Spatial Reuse in Wireless Local Area Networks," Ph.D. dissertation, University of Wollongong, 2022.

[10] F. Wilhelmi, S. Szott, K. Kosek-Szott, and B. Bellalta, "Machine Learning and Wi-Fi: Unveiling the Path Toward AI/ML-Native IEEE 802.11 Networks," *IEEE Communications Magazine*, 2024.

[11] F. Wilhelmi, B. Bellalta, S. Szott, K. Kosek-Szott, and S. Barrachina-Muñoz, "Coordinated Multi-Armed Bandits for Improved Spatial Reuse in Wi-Fi," 2025. [Online]. Available: https://arxiv.org/abs/2412.03076

[12] M. Wojnar, W. Ciezobka, K. Kosek-Szott, K. Rusek, S. Szott, D. Nunez, and B. Bellalta, "IEEE 802.11bn Multi-AP Coordinated Spatial Reuse With Hierarchical Multi-Armed Bandits," *IEEE Communications Letters*, vol. 29, no. 3, pp. 428–432, 2025.

[13] M. Wojnar, W. Ciężobka, A. Tomaszewski, P. Chołda, K. Rusek, K. Kosek-Szott, J. Haxhibeqiri, J. Hoebeke, B. Bellalta, A. Zubow, F. Dressler, and S. Szott, "Coordinated Spatial Reuse Scheduling With Machine Learning in IEEE 802.11 MAPC Networks," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2025.

[14] Q. Li, B. Lv, Y. Hong, and R. Wang, "ReinWiFi: Application-Layer QoS Optimization of WiFi Networks with Reinforcement Learning," 2025. [Online]. Available: https://arxiv.org/abs/2405.03526

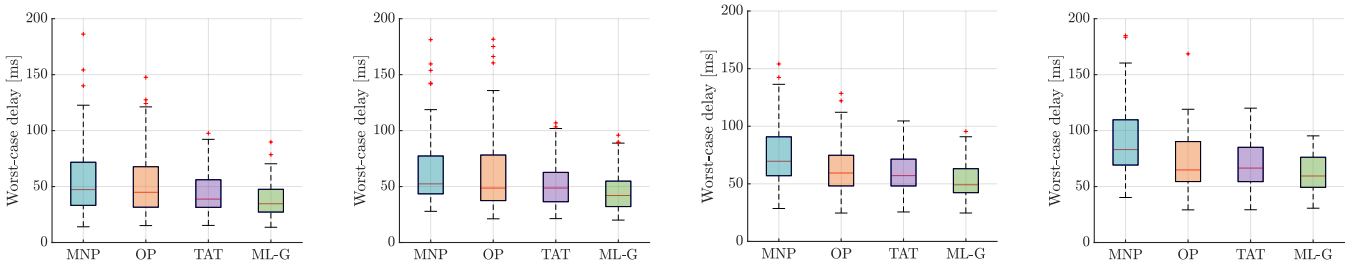
[15] Cui, Wei and Shen, Kaiming and Yu, Wei, "Spatial Deep Learning for Wireless Scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, p. 1248–1261, Jun. 2019. [Online]. Available: http://dx.doi.org/10.1109/JSAC.2019.2904352

[16] M. Du, R. Yan, P. Liu, Z. Guo, and X. Sun, "Deep Reinforcement Learning Based Spatial Reuse for IEEE 802.11 bn," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2025, pp. 1–6.

[17] "IEEE 802.11-25/0502r0: Details on the unified MAPC framework," 2025.

[18] M. Towers, A. Kwiatkowski, J. Terry, J. U. Balis, G. De Cola, T. Deleu, M. Goulão, A. Kallinteris, M. Krimmel, A. KG *et al.*, "Gymnasium: A standard interface for reinforcement learning environments," *arXiv preprint arXiv:2407.17032*, 2024.

[19] R. BELLMAN, "A Markovian Decision Process," *Journal of*



(a) $N = 8$ STAs (b) $N = 12$ STAs (c) $N = 16$ STAs (d) $N = 20$ STAs

Figure 10: Worst-case delay distribution for a different number of users $N \in \{8, 12, 16, 20\}$ across 100 random deployment realizations in each scenario.

Mathematics and Mechanics, vol. 6, no. 5, pp. 679–684, 1957. [Online]. Available: http://www.jstor.org/stable/24900506

[20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[21] V. Konda and J. Tsitsiklis, "Actor-Critic Algorithms," in *Advances in Neural Information Processing Systems*, S. Solla and T. Leen and K. Müller, Ed., vol. 12. MIT Press, 1999. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[22] S. Merlin *et al.*, "TGax Simulation Scenarios," Nov. 2015, doc.: IEEE 802.11-14/0980r16.

[23] The MathWorks Inc., "802.11be Packet Error Rate Simulation for an EHT MU Single-User Packet Format," Natick, Massachusetts, United States, 2024. [Online]. Available: https://www.mathworks.com/help/wlan/ug/802-11be-packet-error-rate-simulation-for-eht-mu-single-user-packet-format.html

[24] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html