

Dual Mind World Model Inspired Network Digital Twin for Access Scheduling

Hrishikesh Dutta, *Member, IEEE*, Roberto Minerva, *Senior Member, IEEE*, and Noel Crespi, *Senior Member, IEEE*

Abstract—Emerging networked systems such as industrial IoT and real-time cyber-physical infrastructures demand intelligent scheduling strategies capable of adapting to dynamic traffic, deadlines, and interference constraints. In this work, we present a novel Digital Twin-enabled scheduling framework inspired by Dual Mind World Model (DMWM) architecture, for learning-informed and imagination-driven network control. Unlike conventional rule-based or purely data-driven policies, the proposed DMWM combines short-horizon predictive planning with symbolic model-based rollout, enabling the scheduler to anticipate future network states and adjust transmission decisions accordingly. We implement the framework in a configurable simulation testbed and benchmark its performance against traditional heuristics and reinforcement learning baselines under varied traffic conditions. Our results show that DMWM achieves superior performance in bursty, interference-limited, and deadline-sensitive environments, while maintaining interpretability and sample efficiency. The proposed design bridges the gap between network-level reasoning and low-overhead learning, marking a step toward scalable and adaptive NDT-based network optimization.

Index Terms—Network Digital Twin, Dual Mind World Model, IoT Scheduling, Reinforcement Learning, Deadline-Constrained Networks, Wireless Sensor Networks, Network Optimization.

I. INTRODUCTION

The next generation of wireless communication systems, such as smart city infrastructure, industrial IoT, and autonomous networks, demands unprecedented levels of adaptability, reliability, and intelligence. These systems must operate under bursty traffic, stringent Quality of Service (QoS) constraints, and interference-limited environments. As such, network scheduling - the problem of deciding which nodes can access the channel at any given time - becomes a critical, yet highly complex, decision-making challenge.

Traditional scheduling algorithms in wireless networks, such as Max-Weight Scheduling (MWS), focus on maximizing long-term throughput for packets without strict delay constraints. MWS is throughput-optimal, achieving any feasible throughput vector, while simpler policies like Longest-Queue-First (LQF) [1] or distributed CSMA ([2], [3]) reduce computational overhead but may only attain a fraction of the throughput region. However, emerging applications require deadline-aware scheduling, which introduces a discontinuity in packet utility, making scheduling more challenging than traditional frameworks. Existing approaches to deadline-aware scheduling often rely on restrictive assumptions. Frame-based

traffic models [4], where time is divided into frames, and arrivals/deadlines are known a priori. Here, the optimal solution per frame is a Max-Weight schedule with weights based on deficit counters (tracking unmet delivery ratios). However, this approach is non-causal unless traffic is periodic and synchronized. Partial generalizations in [5] lack performance guarantees. The greedy policies like Largest-Deficit-First (LDF) [6] extend LQF to real-time scheduling, achieving an efficiency ratio (fraction of the real-time throughput region guaranteed) of at least $\frac{1}{\beta+1}$, under i.i.d. arrivals and no fading, where β is the network's interference degree [7]. There have been works ([8], [9]) that introduce randomized algorithms (AMIX-ND, AMIX-MS) for improving efficiency ratios with relaxed traffic assumptions, but face implementation barriers in large-scale networks.

Furthermore, data-driven methods such as Reinforcement Learning (RL) ([10]–[12]) have shown promise, but often fail to guarantee feasibility under domain-specific constraints, such as interference or packet deadlines. Furthermore, purely model-free agents tend to lack interpretability and require significant training data, making them unsuitable for rapidly changing network conditions.

With the goal of ameliorating the above limitations and inspired by cognitive architectures in neuroscience and artificial intelligence, we propose a novel hybrid scheduling framework, built using the concept of *Dual Mind World Model (DMWM)* [13]. The proposed approach combines reactive heuristics with symbolic planning within a network digital twin platform [14]. The developed DMWM-inspired NDT maintains a *fast reactive module* (the “Fast Mind”) for heuristic-driven decision making, and a *deliberative symbolic module* (the “Slow Mind”) that uses internal simulation (imagination) to predict the consequences of actions over a finite horizon. Feasibility is enforced using a *Informed Constraint Navigation (ICN)* layer, which acts as a symbolic constraint checker for interference, deadlines, and queue availability.

DMWM operates in a dual-phase loop: *Fast Mind* triggers lightweight scheduling when constraints are simple or imagination fails; and *Slow Mind* evaluates all feasible scheduling candidates using forward rollouts within the digital twin environment, selecting the schedule that optimizes long-term throughput or delay. Unlike black-box neural planners, DMWM offers *interpretable planning*, *constraint compliance*, and *real-time responsiveness*. Its modular design allows for direct integration within real or emulated NDTs.

Specifically, this paper has the following key contributions:

- 1) We introduce a hybrid cognitive scheduling model that integrates a symbolic planner with a reactive fallback

The authors are with the Data Intelligence and Communication Engineering Lab, Telecom SudParis, Institut Polytechnique de Paris, France (email: hrishikesh.dutta@telecom-sudparis.eu; roberto.minerva@telecom-sudparis.eu; noel.crespi@telecom-sudparis.eu)

mechanism, enabling both adaptability and low-latency execution.

- 2) We design an efficient symbolic state-transition function that allows the agent to simulate future queue states, providing long-term foresight without neural dynamics modeling.
- 3) We formulate a logic-based constraint filtering mechanism that enforces per-node deadlines, interference avoidance, and queue-awareness before planning.
- 4) We construct a simulation-driven digital twin environment that models bursty traffic, heterogeneous nodes, and scheduling constraints, enabling comprehensive performance benchmarking.
- 5) We benchmark DMWM against classical and learning-based scheduling baselines under multiple realistic scenarios, demonstrating its superiority in delay management, throughput, and feasibility adherence.

II. RELATED WORK

The challenge of scheduling in wireless networks under deadline, interference, and delay constraints has been previously studied, yet many existing approaches fall short in developing solutions for dynamic, heterogeneous networks.

Deadline-Constrained Scheduling: Traditional scheduling policies such as Max-Weight Scheduling (MWS) offer throughput-optimal performance but are agnostic to real-time constraints [4]. Extensions like Largest-Deficit-First (LDF) [6] and its analytical generalization via the Real-Time Local Pooling Factor (R-LPF) [9] provide bounded efficiency ratios under specific traffic assumptions. Recent works have explored randomized scheduling [7], offering improved delivery ratios, particularly under stochastic arrivals and fading channels. However, these methods often assume frame-based traffic or rely on offline knowledge of traffic statistics, limiting their adaptability in dynamic settings.

Learning-Based Scheduling: Reinforcement learning (RL) has gained traction as a means to learn scheduling policies from interactions with the network environment. For instance, cooperative RL schemes [12] enable decentralized scheduling in energy-harvesting IoT networks, while others apply RL in full-duplex communication environments to bypass CSI estimation [11]. More recently, offline RL methods [8] have been proposed to train deadline-aware policies using behavior cloning and actor-critic learning. Despite their promise, most RL-based schedulers are model-free, requiring large training samples and offering limited interpretability, which poses challenges in real-time and safety-critical applications.

Hybrid and Graph-Based Approaches: Graph convolutional network (GCN)-based DRL models have been used for joint optimization of time-sensitive and best-effort traffic [10], improving generalization and convergence speed. Similarly, fast-mixing CSMA variants [4] and distributed CSMA protocols with centralized coordination [3] have shown improved scalability. However, such approaches still face limitations in integrating foresight and constraint reasoning within the scheduling loop.

Symbolic and Model-Based Methods: Foundational work by Deng et al. [5] characterized the delay constrained capac-

ity region for general traffic patterns via MDPs, laying the groundwork for model-based scheduling. While their solution achieves theoretical optimality, the complexity of infinite-horizon MDPs restricts practical deployment. Input-queued switch scheduling strategies like HRF and CHRF [1] showcase the potential of rank-based arbitration, but require specialized queue structures and are difficult to generalize for wireless networks.

Positioning of This Work: Compared to the aforementioned techniques, our proposed *Dual Mind World Model (DMWM)* framework takes a hybrid cognitive approach, combining symbolic planning with reactive heuristics in a simulation-driven *Network Digital Twin (NDT)* environment. Unlike purely model-free RL agents, DMWM performs forward rollouts over a logic-driven world model, enabling constraint-aware scheduling decisions with minimal training. Our design bridges the interpretability and efficiency of symbolic schedulers with the adaptability of learning agents, yielding superior performance under bursty, interference-limited, and deadline-sensitive conditions.

III. SYSTEM MODEL

We consider a slotted-time wireless network consisting of N sensor nodes, denoted by the set $\mathcal{N} = \{1, 2, \dots, N\}$. Each node maintains a local buffer (queue) and competes for channel access through centralized scheduling. The system operates under constraints typical of resource-constrained IoT networks, including interference, buffer overflows, and deadline-sensitive traffic.

A. Physical Network Model

At each time slot t , the queue length at node $i \in \mathcal{N}$ is denoted by $q_i(t) \in \mathbb{N}_0$. Packets arrive at node i according to a Poisson process with rate $\lambda_i(t)$, which may vary with time to model bursty or periodic behavior. Nodes transmit based on a schedule $\mathcal{S}(t) \subseteq \mathcal{N}$, where $|\mathcal{S}(t)| \leq K$ is the maximum number of simultaneous transmissions per slot, dictated by the number of time-frequency resources.

We define a set of *conflict pairs* $\mathcal{C} \subset \mathcal{N} \times \mathcal{N}$ to encode interference constraints. A valid schedule must satisfy:

$$\forall (i, j) \in \mathcal{C}, \quad i \in \mathcal{S}(t) \Rightarrow j \notin \mathcal{S}(t) \quad (1)$$

Packets experience delays, and some nodes have deadline constraints D_i . A packet arriving at time t_a must be transmitted by $t_a + D_i$, otherwise it incurs a deadline violation. The goal is to maximize long-term throughput while minimizing queue backlog, average delay, and deadline violations.

B. Network Digital Twin Environment

To enable planning and evaluation, we construct a *Network Digital Twin* that serves as a virtual mirror of the physical network. It models the dynamics of:

- **Traffic:** time-varying Poisson arrivals with configurable burstiness.
- **Queueing:** bounded FIFO queues with maximum buffer size B .

- **Delays and Deadlines:** per-packet age tracking for deadline evaluation.
- **Constraints:** interference (via \mathcal{C}) and transmission limits.

The digital twin supports *symbolic imagination*—a simulated forward pass from a given state under hypothetical actions—without interacting with the real network. This facilitates the implementation of foresight-driven decision-making under the DMWM framework.

C. Scheduling Actions and Rewards

At each slot t , the scheduler selects a feasible action $\mathcal{S}(t)$ that:

- 1) Satisfies all logic constraints (non-empty queues, conflict avoidance, deadlines).
- 2) Maximizes a reward signal based on queue state and prediction of future congestion.

Each successfully transmitted packet yields unit reward. The total system reward at time t is:

$$r(t) = \sum_{i \in \mathcal{S}(t)} \mathbb{I}[q_i(t) > 0] \quad (2)$$

To support planning, the digital twin is used to simulate H -step rollouts from the current state:

$$R(\mathcal{S}) = \sum_{\tau=0}^{H-1} \sum_{i \in \mathcal{N}} \mathbb{I}[q_i^{(\tau)} > 0] \quad (3)$$

where $q_i^{(\tau)}$ represents the simulated queue length of node i at future step τ .

D. Dual-Mind Decision Architecture

The scheduler operates under a *Dual Mind World Model (DMWM)*:

- **Slow Mind (Symbolic Planner):** Uses the digital twin to perform multi-step simulation over all valid scheduling candidates. Selects the schedule that maximizes predicted reward $R(\mathcal{S})$.
- **Fast Mind (Reactive Heuristic):** Falls back to simple rules (e.g., LQF) when imagination is inconclusive, infeasible, or too expensive to compute in time-constrained environments.

Constraint checking is implemented using an Informed Constrained Navigation (ICN) module that filters actions based on feasibility, ensuring compliance with protocol logic.

IV. DUAL MIND WORLD MODEL DRIVEN SCHEDULING

The proposed scheduling architecture is based on a cognitively inspired framework that mimics dual-process decision-making in intelligent agents. This dual-mind architecture, as shown in Figure 1 allows the scheduler to balance between *speed* and *foresight*, switching modes dynamically based on current feasibility, complexity, and urgency.

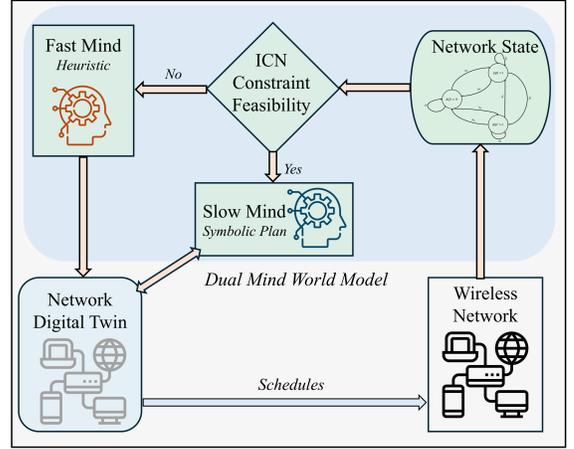


Fig. 1: System Model

A. Fast Mind: Reactive Scheduling

The fast mind selects a schedule $\mathcal{S}_{\text{fast}}(t)$ using a heuristic based on current queue lengths:

$$\mathcal{S}_{\text{fast}}(t) = \arg \max_{\mathcal{S} \subset \mathcal{N}} \sum_{i \in \mathcal{S}} q_i(t), \quad |\mathcal{S}| = K \quad (4)$$

This policy is computationally lightweight and ensures responsiveness under high-load or emergency conditions but lacks long-term optimality.

B. Slow Mind: Symbolic Rollout Planning

The slow mind considers a set of feasible candidate schedules:

$$\mathcal{F}(t) = \{\mathcal{S}_c \subset \mathcal{N} : |\mathcal{S}_c| = K, \text{ICN}(\mathcal{S}_c, q(t), t) = \text{True}\} \quad (5)$$

where ICN is a constraint-checking function that verifies:

- Non-empty queues: $q_i(t) > 0$ for all $i \in \mathcal{S}_c$
- Interference constraints: $(i, j) \notin \mathcal{C}$ for all $i, j \in \mathcal{S}_c$
- Deadline feasibility: if D_i exists, schedule packets within allowable time

For each feasible schedule $\mathcal{S}_c \in \mathcal{F}(t)$, the symbolic world model simulates a rollout of horizon H :

$$q_i^{(\tau+1)} = \begin{cases} \max(q_i^{(\tau)} - 1, 0), & \text{if } i \in \mathcal{S}_c \\ q_i^{(\tau)}, & \text{otherwise} \end{cases} \quad (6)$$

starting from $q_i^{(0)} = q_i(t)$. The cumulative symbolic reward over the horizon is defined as:

$$R(\mathcal{S}_c) = \sum_{\tau=0}^{H-1} \sum_{i \in \mathcal{N}} \mathbb{I}[q_i^{(\tau)} > 0] \quad (7)$$

The slow mind then selects the rollout-optimal schedule:

$$\mathcal{S}_{\text{slow}}(t) = \arg \max_{\mathcal{S}_c \in \mathcal{F}(t)} R(\mathcal{S}_c) \quad (8)$$

C. Combined Decision Rule

The final scheduling decision at time t is based on the availability of feasible plans:

$$\mathcal{S}(t) = \begin{cases} \mathcal{S}_{\text{slow}}(t), & \text{if } \mathcal{F}(t) \neq \emptyset \\ \mathcal{S}_{\text{fast}}(t), & \text{otherwise} \end{cases} \quad (9)$$

This rule ensures the best available action is chosen—either from predicted symbolic outcomes (slow mind) or from fast heuristics when symbolic planning fails or is too expensive.

D. Complexity and Feasibility

The complexity of the slow mind’s planning step is $O\left(\binom{N}{K} \cdot H \cdot N\right)$, stemming from simulating H -step queue evolution for each of the $\binom{N}{K}$ candidate schedules. In practice, the ICN filter significantly prunes the search space by rejecting infeasible or illogical candidates early.

E. Integration with the Digital Twin

All symbolic simulations are performed within the *Network Digital Twin*, which replicates the queuing, interference, and traffic dynamics of the physical system. By using the twin for internal rollouts, the slow mind achieves foresight without affecting real-time operation. This separation enables predictive evaluation, safety checking, and explainability within a unified decision framework. The detailed working of the proposed scheduling framework is given in Algorithm 1, with the ICN module in Algorithm 2.

TABLE I: Baseline Simulation Hyperparameters

Parameter	Value
Number of Nodes (N)	5
Max Scheduled Nodes (K)	3
Queue Capacity (B)	50
Time Steps per Run	200
Poisson Arrival Base Range (λ_i^{base})	[0.5, 1.0]
Imagination Horizon (H)	3
Q-Learning α (Learning Rate)	0.1
Q-Learning γ (Discount Factor)	0.95
Q-Learning ϵ (Exploration Rate)	0.2

V. EXPERIMENTAL SETUP

To evaluate the proposed Dual Mind World Model (DMWM) scheduler, we implement a simulation-driven digital twin of a time-slotted wireless network. The digital twin captures queue dynamics, stochastic arrivals, interference constraints, and per-node deadlines, in addition to forwarding the scheduling policies learnt by the DMWM scheduler. All experiments are carried out across multiple network scenarios with diverse constraints, compared against baseline policies.

A. Network Simulation Environment

We simulate a network of N nodes, each maintaining a bounded FIFO queue of size $B = 50$. Time is slotted, and up to K nodes can be scheduled in each slot. Traffic arrivals at each node follow a time-varying Poisson process with dynamic

Algorithm 1 DMWM-Enabled Scheduling

Require: Current queue state $q(t)$, deadline vector $D = \{D_1, \dots, D_N\}$, conflict graph \mathcal{C} , number of slots K , planning horizon H

Ensure: Selected schedule $S(t) \subseteq \{1, \dots, N\}$ with $|S(t)| = K$

- 1: $\mathcal{F} \leftarrow \emptyset$ {Feasible schedule set}
- 2: **for all** $S \subseteq \{1, \dots, N\}$ with $|S| = K$ **do**
- 3: **if** ICN_CHECK($S, q(t), D, \mathcal{C}, t$) **then**
- 4: $\mathcal{F} \leftarrow \mathcal{F} \cup \{S\}$
- 5: **end if**
- 6: **end for**
- 7: **if** $\mathcal{F} \neq \emptyset$ **then**
- 8: $R^* \leftarrow -\infty, S^* \leftarrow \emptyset$
- 9: **for all** $S \in \mathcal{F}$ **do**
- 10: $q^{(0)} \leftarrow q(t), R \leftarrow 0$
- 11: **for** $\tau = 0$ to $H - 1$ **do**
- 12: **for all** $i \in S$ **do**
- 13: $q_i^{(\tau+1)} \leftarrow \max(q_i^{(\tau)} - 1, 0)$
- 14: **end for**
- 15: $R \leftarrow R + \sum_{i=1}^N \mathbb{I}[q_i^{(\tau)} > 0]$
- 16: **end for**
- 17: **if** $R > R^*$ **then**
- 18: $R^* \leftarrow R, S^* \leftarrow S$
- 19: **end if**
- 20: **end for**
- 21: **return** S^*
- 22: **else**
- 23: **for** $i = 1$ to N **do**
- 24: $u_i \leftarrow q_i(t) \cdot (2 \text{ if } D_i < \infty \text{ else } 1)$
- 25: **end for**
- 26: $S_{\text{fallback}} \leftarrow \text{top-}K \text{ indices by } u_i$
- 27: **return** S_{fallback}
- 28: **end if**

arrival rates $\lambda_i(t)$. The traffic profile includes both smooth and bursty patterns to mimic real-world variability:

$$\lambda_i(t) = \lambda_i^{\text{base}} \cdot \left(1 + 0.75 \cdot \sin\left(\frac{2\pi t}{50}\right)\right) + \delta_i(t)$$

where $\delta_i(t)$ introduces occasional burst spikes on selected nodes.

Scheduling actions are constrained by a conflict graph \mathcal{C} (interference model) and optional per-node packet deadlines D_i .

B. Evaluation Scenarios

We evaluate the policies in four distinct scenarios:

- **Default:** Moderate interference (pairs of conflicting nodes) and mixed deadline constraints on alternate nodes.
- **Bursty Traffic:** High variance in arrival rates with no deadlines or conflicts.
- **Deadline-Sensitive:** Alternating nodes with short (5) and long (15) deadlines, plus pairwise interference.
- **Interference-Constrained:** Circular conflict graph without deadlines.

Algorithm 2 ICN_CHECK($S, q(t), D, \mathcal{C}, t$)

```

1: for all  $i \in S$  do
2:   if  $q_i(t) = 0$  then
3:     return False
4:   end if
5:   if  $D_i < \infty$  and oldest packet age  $> D_i$  then
6:     return False
7:   end if
8: end for
9: for all  $(i, j) \in S \times S$  do
10:  if  $(i, j) \in \mathcal{C}$  then
11:    return False
12:  end if
13: end for
14: return True

```

C. Baseline Scheduling Policies

The proposed DMWM scheduler is compared against the following baselines:

- **Random:** Selects K nodes uniformly at random.
- **Longest Queue First (LQF):** Prioritizes nodes with highest queue length.
- **Deadline-Priority:** Assigns urgency weights based on packet deadlines.
- **Fair Round Robin:** Ensures fairness over time using a memory-based rotation.
- **RL-Agent (Q-Learning):** Online learning-based scheduler with ϵ -greedy exploration, updated via Q-table.

The DMWM agent uses a symbolic forward model (imagination) with a planning horizon of $H = 3$ to evaluate candidate schedules filtered through the ICN constraint layer.

D. Hyperparameters and Performance Metrics

All policies are tested over 30 independent simulation runs, each lasting 200 time steps. The baseline parameters used in the simulation environment are summarized in Table I. We evaluate each policy using the following metrics: average throughput, queue length, delay, and deadline violations. Each metric is reported with its mean and standard deviation over multiple simulation runs.

VI. RESULTS AND ANALYSIS

To assess the effectiveness of the proposed DMWM-inspired scheduling framework, we conduct extensive simulations across diverse network conditions that mimic real-world challenges in IoT and wireless access systems. As explained earlier in section V, we compare the performance of DMWM against traditional baselines and a reinforcement learning (RL) agent under four distinct traffic and constraint scenarios: default, bursty, deadline-sensitive, and interference-limited. First, to gain insight into the internal dynamics of DMWM, we analyze two behavioral visualizations shown in Figure 2. The heatmap shown in Figure 2 (a) illustrates the node scheduling pattern selected by DMWM over time. The schedule adapts dynamically to queue states and constraint violations. The

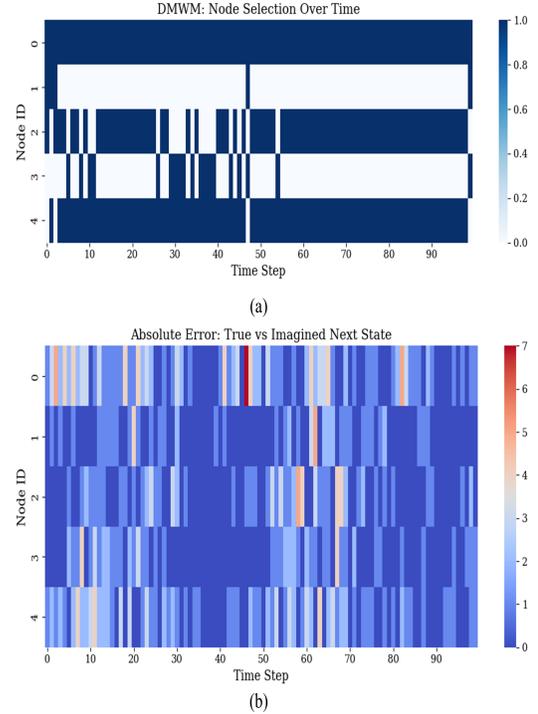


Fig. 2: DMWM Scheduler Behavior and Model Accuracy: This plot illustrates the node scheduling behavior of the DMWM policy, where each row corresponds to a node, and each column represents a time step. A blue cell (in (a)) indicates that the corresponding node was selected for transmission at that time. Subplot (b) visualizes the absolute difference between the actual queue state transitions and those predicted by the symbolic world model used in the Slow Mind’s internal simulations.

diversity and structure in activation patterns demonstrate the interplay between the reactive (Fast Mind) and symbolic (Slow Mind) modules. A comparison between imagined and actual next states (Figure 2 (a)) reveals low absolute prediction errors across nodes, confirming the reliability of the internal model used for forward simulations. This validates the ability of the digital twin to anticipate the dynamics of the queues over the planning horizon.

Figure 3 presents comparative results of all policies across four scenarios, averaged over multiple runs. The following key findings are observed. First, DMWM-enabled scheduling consistently achieves the highest or near-highest average throughput across all scenarios. Notably, it outperforms both LQF and the RL-agent under bursty and deadline-sensitive loads, due to its planning-based imagination module that anticipates congestion and avoids conflicts. Second, in terms of queue occupancy, DMWM maintains lower average queue sizes compared to reactive policies, indicating more effective congestion mitigation. Particularly in interference-heavy scenarios, its symbolic constraint module allows it to navigate contention zones more efficiently than LQF and Random. Finally, the most significant improvements from DMWM are observed in latency-sensitive metrics. Under the deadline-sensitive scenario, DMWM reduces the average delay and number of deadline violations substantially relative to the RL-agent and heuristics. This is attributed to its forward rollout mechanism, which explicitly evaluates the impact of delay

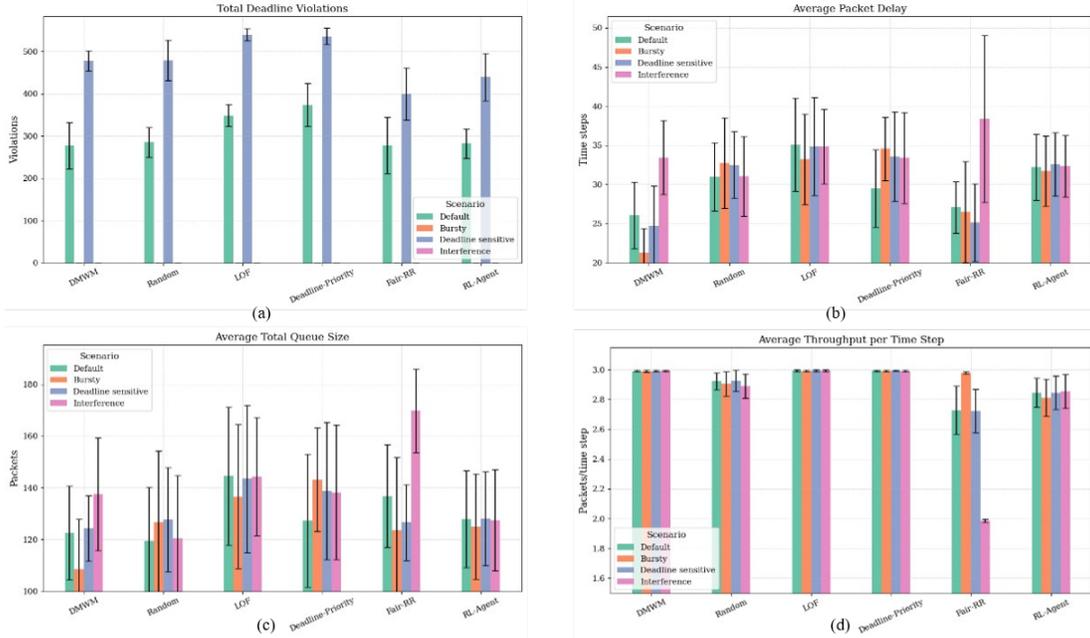


Fig. 3: This figure compares six scheduling policies—DMWM, Random, LQF, Deadline-Priority, Fair Round-Robin, and a Q-learning agent—across four scenarios: Default, Bursty, Deadline-Sensitive, and Interference-Constrained. Results show mean performance over 30 runs with standard deviation bars. DMWM attains the best or near-best throughput while maintaining low delay and queue occupancy, achieving efficient utilization without violating interference or deadline constraints.

during scheduling. To be noted that DMWM not only performs well on average but also shows lower variance across trials, indicating robustness under dynamic network conditions.

VII. CONCLUSION

In this paper, we proposed a novel Dual Mind World Model (DMWM) approach for network scheduling, designed to operate within a Network Digital Twin (NDT) simulation environment. The DMWM framework integrates symbolic planning and lightweight model-based prediction, offering a balance between adaptability and interpretability. By leveraging imagined state transitions over a short planning horizon, DMWM allows the scheduler to anticipate and mitigate network congestion, interference, and deadline violations without requiring deep function approximators or extensive retraining. We evaluated the proposed scheduler across multiple simulated network scenarios characterized by bursty traffic, deadline constraints, and interference. Comparative results against classic baselines demonstrated DMWM’s robustness, particularly in non-stationary or constrained environments. In addition to superior performance in key metrics like throughput, delay, and deadline adherence, DMWM offers a transparent decision-making process through interpretable rollout mechanisms. This makes it suited for critical or resource-constrained systems requiring accountability and insight.

REFERENCES

- [1] B. Hu, F. Fan, K. L. Yeung, and S. Jamin, “Highest rank first: A new class of single-iteration scheduling algorithms for input-queued switches,” *IEEE Access*, vol. 6, pp. 11 046–11 062, 2018.
- [2] Z. Zhang, S. Atapattu, Y. Wang, and M. Di Renzo, “Distributed mac for ris-assisted multiuser networks: Cdma/ca protocol design and statistical optimization,” *IEEE Transactions on Mobile Computing*, 2025.
- [3] C. Huang and X. Wang, “Distributed scheduling with centralized coordination for scalable wireless mesh networking,” *IEEE/ACM Transactions on Networking*, vol. 31, no. 1, pp. 436–451, 2022.
- [4] B. Li and A. Eryilmaz, “Optimal distributed scheduling under time-varying conditions: A fast-csma algorithm with applications,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 7, 2013.
- [5] L. Deng, C.-C. Wang, M. Chen, and S. Zhao, “Timely wireless flows with general traffic patterns: Capacity region and scheduling algorithms,” *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, 2017.
- [6] X. Kang, W. Wang, J. J. Jaramillo, and L. Ying, “On the performance of largest-deficit-first for scheduling real-time traffic in wireless networks,” in *ACM international symposium on Mobile ad hoc networking and computing*, 2013, pp. 99–108.
- [7] C. Tsanikidis and J. Ghaderi, “Randomized scheduling of real-time traffic in wireless networks over fading channels,” *IEEE/ACM Transactions on Networking*, vol. 31, no. 4, pp. 1688–1701, 2022.
- [8] J. Wan, S. Lin, Z. Zhang, J. Zhang, and T. Zhang, “Scheduling real-time wireless traffic: A network-aided offline reinforcement learning approach,” *IEEE internet of things journal*, vol. 10, no. 24, 2023.
- [9] C. Tsanikidis and J. Ghaderi, “On the power of randomization for scheduling real-time traffic in wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 29, no. 4, pp. 1703–1716, 2021.
- [10] L. Yang, Y. Wei, F. R. Yu, and Z. Han, “Joint routing and scheduling optimization in time-sensitive networks using graph-convolutional-network-based deep reinforcement learning,” *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23 981–23 994, 2022.
- [11] H. Fawaz, M. El Helou, S. Lahoud, and K. Khawam, “A reinforcement learning approach to queue-aware scheduling in full-duplex wireless networks,” *Computer Networks*, vol. 189, p. 107893, 2021.
- [12] H. Dutta, A. K. Bhuyan, and S. Biswas, “Cooperative reinforcement learning for energy management in multi-hop networks with energy harvesting,” *IEEE Transactions on Green Communications and Networking*, 2025.
- [13] L. Wang, R. Shelim, W. Saad, and N. Ramakrishnan, “Dmwm: Dual-mind world model with long-term imagination,” *arXiv preprint arXiv:2502.07591*, 2025.
- [14] S. M. Raza, R. Minerva, N. Crespi, M. Alvi, M. Herath, and H. Dutta, “Exploiting the efficient data modeling in network digital twin to empower edge-cloud continuum,” in *2024 20th International Conference on Network and Service Management (CNSM)*. IEEE, 2024, pp. 1–4.