AI-Enabled Traffic-Aware Closed-Loop Framework for MAC-Layer Control in Multi-Hop IoT Networks

Ahmad Mahmod
ICube, University of Strasbourg
Pole API, 67412 Illkirch, France
mahmod@unistra.fr

Julien Montavont
ICube, University of Strasbourg
Pole API, 67412 Illkirch, France
montavont@unistra.fr

Thomas Noel
ICube, University of Strasbourg
Pole API, 67412 Illkirch, France
noel@unistra.fr

***Abstract*—Low-power and lossy wireless networks must balance communication reliability and energy efficiency despite constrained resources, dynamic multi-hop topologies, and fluctuating traffic conditions. This challenge is exacerbated when traffic alternates between sporadic and bursty event-driven phases, as a single MAC operating mode cannot efficiently accommodate all traffic regimes. This paper presents a traffic-aware closed-loop framework that combines runtime monitoring, short-horizon forecasting, recommendation, and online reconfiguration. Using per-node telemetry and learning-based models, the framework anticipates near-future performance degradation and triggers suitable MAC adaptations, combining contention-based duty-cycle timing reconfiguration under moderate traffic with TDMA-oriented protection under bursty traffic. Experiments on the SLICES-FR testbed show that the framework generalizes to unseen deployments, improves the timing of runtime actions compared with reactive control, and restores reliability during bursty phases without the permanent cost of always-on TDMA.**

***Index Terms*—Low-power and Lossy Wireless Networks, Closed-loop Adaptation, Machine Learning, Performance Forecasting, Traffic-aware Control.**

I. Introduction

Low-power and lossy wireless networks (LLWNs) are a core communication substrate for the Internet of Things (IoT), especially in environmental monitoring, industrial sensing, smart infrastructure, and critical supervision. Their operation is shaped by constrained nodes, low-rate radios, unreliable links, and dynamically evolving multi-hop topologies [1]. As a result, reliability and energy efficiency cannot be optimized independently: increasing radio activity usually improves delivery, but also increases duty cycle and shortens service life, whereas aggressive sleeping reduces energy consumption, but can severely degrade performance when the offered load or forwarding burden increases [2].

This trade-off becomes harder to manage because LLWN conditions are not stationary. Traffic can alternate between sporadic sensing and bursty event-driven phases. Forest monitoring is a representative example: during normal operation, nodes report low-rate measurements, whereas fire detection can trigger bursty and critical traffic. Under sporadic low-rate traffic, fixed duty-cycled contention-based settings may remain sufficient [3]. Under bursty traffic, however, the same configuration may become inadequate because simultaneous arrivals create queue buildup, repeated contention, retransmissions, and packet drops. In such cases, a transition to TDMA-like service can provide stronger protection by reducing contention and draining queued packets more deterministically [4]. Thus, the most suitable MAC configuration depends on the traffic regime, the node position in the routing topology, and the forwarding burden imposed by multi-hop communication.

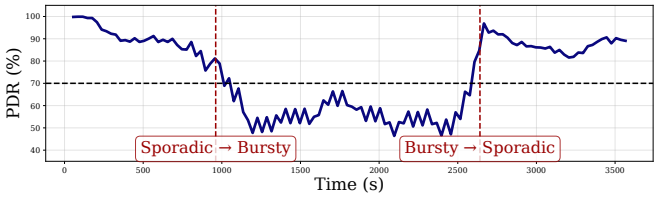


Fig. 1: PDR under sporadic and bursty traffic regimes.

Fig. 1 compares the packet delivery ratio (PDR) achieved by a leaf node in a tree topology under alternating sporadic and bursty traffic regimes, using contention-based duty cycling only, with a target PDR of 70%. The figure shows that the leaf node under contention-based duty cycling satisfies the target PDR during sporadic traffic. However, when the bursty phase begins, the node fails to maintain the target PDR, highlighting the need to explore more deterministic MAC configurations, including TDMA-like approaches. At the same time, such deterministic MAC configurations require higher synchronization overhead and longer radio-on periods due to synchronization and scheduled listening, which increase energy consumption. These observations motivate runtime adaptation to better balance energy consumption and QoS requirements under current network conditions.

In previous work [5], we proposed PENSIL, a flexible architecture based on a modular network stack in which network functions are implemented as virtual functions, enabling fast and fine-grained reconfiguration of the communication stack during operation, including parameter tuning, function updates, and full protocol replacement. The architecture includes a central orchestrator that exposes reconfiguration capabilities and hosts part of the control logic responsible for executing stack modifications. The present work builds upon this foundation by focusing on the design of the control logic that drives this orchestrator, i.e., determining when and what should be reconfigured at runtime, particularly at the MAC layer, where timing constraints make decision-making more challenging.

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

In this context, a key challenge lies in the fact that purely reactive decision-making remains insufficient in many LLWN scenarios: if reconfiguration is triggered only after performance degradation becomes observable, significant performance loss may already have occurred, especially at structurally critical forwarding nodes [6]. This motivates a shift toward a closed-loop control approach in which MAC management is continuously driven by network state. In this paradigm, the network is monitored, short-horizon performance evolution is anticipated, and reconfiguration decisions are proactively selected based on both current and predicted conditions.

To address these challenges, this work develops a traffic-aware closed-loop control framework for multi-hop LLWNs that unifies monitoring, short-horizon prediction, recommendation, and runtime configuration. The framework exploits per-node telemetry to capture communication quality, topology-dependent forwarding conditions, and traffic burstiness, and leverages learning-based models to anticipate near-future PDR and duty-cycle evolution. It then triggers suitable reconfiguration actions, enabling node-specific duty-cycle adaptation under contention-based operation and TDMA-oriented protection under bursty traffic within a single control loop. Unlike approaches that treat predictive modeling, adaptive MAC protocols, and closed-loop control as separate problems, the proposed framework combines them into one practical architecture deployed over realistic testbed experiments. More specifically, the work makes the following contributions:

1) A formulation of LLWN MAC adaptation as a proactive closed-loop control problem under heterogeneous node roles and mixed traffic regimes.
2) A real-world dataset of windowed telemetry and derived metrics for multi-hop LLWNs under diverse timing configurations and traffic profiles.
3) An experimental learning pipeline for short-horizon prediction and runtime recommendation based on real deployments.
4) A real-world evaluation of the proposed framework on unseen multi-hop deployments to assess forecasting accuracy, recommendation effectiveness, adaptability to traffic-profile changes, and scalability in larger heterogeneous topologies.

To promote reproducibility and support further developments, we provide open-source access to our dataset, implementation, and evaluation results[1].

II. Background and Literature Review

A. Traffic dynamics and adaptive MAC behavior in LLWNs

Many LLWN applications rely on periodic sensing, where nodes transmit at regular intervals and relatively stable rates. This traffic model has long guided the design of duty-cycled MAC protocols, since periodic low-rate reporting can be supported efficiently through repeated wake-up schedules and lightweight contention-based access. However, IoT traffic is not always smooth: event-driven sensing, alarms, and reactive applications often generate bursty traffic, in which multiple packets are produced within short intervals and followed by longer silent periods [3].

This distinction is important because the traffic regime directly affects MAC performance. Under sporadic or low-rate periodic load, a fixed duty-cycle configuration may remain efficient, whereas under bursty load it can create queue buildup, repeated contention, and degraded delivery [7]. Kim et al. propose a dynamic duty-cycle MAC protocol for IoT environments and wireless sensor networks in which the receiver adjusts its duty-cycle ratio and the sender adapts its contention-window size according to congestion conditions [8]. Other studies show that reservation-oriented or TDMA-based mechanisms can become preferable when traffic is highly concentrated in time [9]. More broadly, work on burst-aware clustering, mobility support, and bursty delivery modeling confirms that traffic irregularity strongly affects buffering, congestion exposure, and service predictability [10].

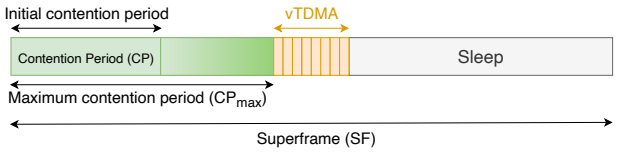


Fig. 2: GoMacH frame structure.

Protocols such as GoMacH [11], which we adopt in our evaluation, combine contention-based access with TDMA-like deterministic operation. GoMacH integrates duty cycling, dynamic slot allocation, and multi-channel communication to achieve high PDR with low radio duty cycle on commodity IEEE 802.15.4 hardware. As shown in Fig. 2, each superframe begins with a nominal contention period, CP, which may be extended up to $CP_{\max}$ when demand increases. Nodes contend during this interval and sleep during the remaining part of the superframe to reduce radio activity. When contention-based access becomes insufficient, GoMacH enables a contention-free virtual TDMA (vTDMA) phase to improve reliability under heavy traffic.

Together, these studies show that LLWN adaptation must be traffic-aware. A single contention-based configuration is unlikely to remain efficient under both sporadic and bursty regimes. Adapting contention-based duty-cycle timing may suffice under moderate traffic, whereas burstier conditions may require a transition toward TDMA-like service. However, TDMA-like operation requires continuous synchronization, which may increase energy consumption and control overhead. In low-rate sporadic traffic scenarios, this overhead can become comparable to the actual data traffic load. In this context, adapting the contention-based duty cycle or temporarily enabling TDMA-like operation, rather than maintaining it permanently, is often sufficient to meet the target PDR while reducing unnecessary overhead.

[1]https://github.com/ahmahmod/mswim2026

B. Machine learning for performance forecasting and control

Machine learning has become increasingly important in wireless networking and IoT resource management because it can capture complex interactions among traffic demand, topology, interference, protocol parameters, and heterogeneous node conditions that are difficult to model analytically. Recent surveys report a growing use of learning-based methods for performance prediction, resource allocation, MAC adaptation, and performance optimization [12].

The need to anticipate degradation before it becomes severe has motivated extensive research on prediction in wireless systems. A first group of studies focuses on predicting communication performance metrics such as delay, throughput, reliability, energy consumption, SNR, and PDR. For instance, a multi-parameter prediction study applies deep neural networks to estimate signal-to-noise ratio (SNR) and PDR in IoT/WSN environments [13]. Throughput estimation has also been investigated in [14], which applies regression models to estimate throughput in IEEE 802.15.4 networks under contention.

Another line of work applies ML to resource-management problems such as bandwidth allocation [15] and IoT resource optimization [16]. Other studies target MAC-level decisions more directly. Reinforcement learning and related approaches have been used for real-time access optimization, multiple-access selection, contention adaptation, TDMA scheduling, and broader channel-access control in IoT systems [17]–[19]. Together, these studies show that MAC behavior can be addressed as a learnable control problem rather than as a fixed handcrafted design.

At the same time, closed-loop control provides a natural mechanism for converting observations into actions. In such systems, an orchestrator repeatedly observes the network state, evaluates current performance, selects an action, and applies it at runtime. This feedback-driven perspective has been explored in industrial IoT [20]. However, its applicability to LLWN runtime control remains limited.

In conclusion, the literature establishes three main points. First, LLWN behavior is strongly influenced by the distinction between non-bursty traffic, such as sporadic and low-rate periodic traffic, and bursty traffic. Second, predictive and feedback-driven adaptation are promising for proactive network management, but existing studies are largely developed outside the specific context of constrained multi-hop LLWNs. Third, machine learning can support complex networking decisions when explicit modeling is difficult, yet most such efforts target broader wireless networks rather than LLWNs. What remains missing is an integrated framework for real-world multi-hop LLWNs that combines short-horizon prediction, traffic-aware control, node-specific adaptation, and runtime switching between contention-based and TDMA-oriented operation.

III. Methodology and Framework

The proposed framework consists of two stages, as illustrated in Fig. 3. The offline stage collects measurements under diverse configurations and traffic profiles, builds structured datasets, and trains models for prediction and recommendation. The selected models are deployed online. The online stage continuously reconstructs the current per-node state from live telemetry, forecasts near-future performance, and applies either contention-parameter reconfiguration or TDMA-oriented protection when needed on GoMacH. The individual components of the framework are described in greater detail in the following subsections.

TABLE I: Dataset collection parameters.

Parameter	Values
Testbed (site)	SLICES-FR (Lille)
Number of nodes	25
Measurement window (Δ)	15 s
Operating system	RIOT
Network stack	UDP/RPL/6LoWPAN/GoMacH
Contention period (CP)	$[10, 15, 20]$ ms
Max. contention period ($CP_{\max}$)	$[3, 7, 11] \times CP$
Superframe (SF)	$[10, 15, 20, 25, 30] \times CP$
Periodic data rate	2 packets/s
Max. allocated slots (max_slot)	$[10, 30, 50, 70, 90, 110, 130, 150]$
Burst packets ($burst_pkts$)	$[5, 10, 20, 30, 40]$

A. Dataset Collection

We consider a multi-hop LLWN composed of a set of nodes $\mathcal{N}$, organized in a tree topology rooted at a sink node. In time window t, node i is characterized by the GoMacH configuration

$$\mathbf{c}_{i,t} = (CP_{i,t}, CP_{i,t}^{\max}, SF_{i,t}),$$

where $CP_{i,t}, CP_{i,t}^{\max}$, and $SF_{i,t}$ represent the nominal contention period, the extended contention period, and the superframe duration illustrated in Fig. 2, together with the network-wide TDMA state $u_t^{\text{tdma}} \in \{0, 1\}$ and the common slot budget $\texttt{max_slot}_t$, which represents the maximum number of slots that a receiver node can schedule in TDMA.

The dataset is built from 85 experiments at the SLICES-FR[2] Lille site, using the parameters defined in Table I. A first full-factorial sweep explores general GoMacH timing through 45 configurations, each run for a 1-hour experiment, obtained from 3 values of CP, 3 values of $CP_{\max}$, and 5 values of SF. A second sweep isolates burst-sensitive TDMA behavior through 40 additional configurations, each run for a 30-min experiment, obtained from 8 values of `max_slot` and 5 burst sizes. This separation is important: the first dataset captures general duty-cycle timing adaptation, while the second isolates the interaction between burstiness and TDMA slot allocation. Each experiment involves 25 M3 nodes. Each node runs the RIOT operating system [21], on top of which we implemented software modules to periodically collect measurements and export them to CSV files. Each node runs a protocol stack composed of UDP/RPL/6LoWPAN/GoMacH.

In every experiment, one node acts as the root, that is, the data sink. The remaining nodes establish routes toward the root through RPL control exchanges. After an initial 3-minute warm-up period allowing routing convergence, each

[2]https://slices-fr.eu/

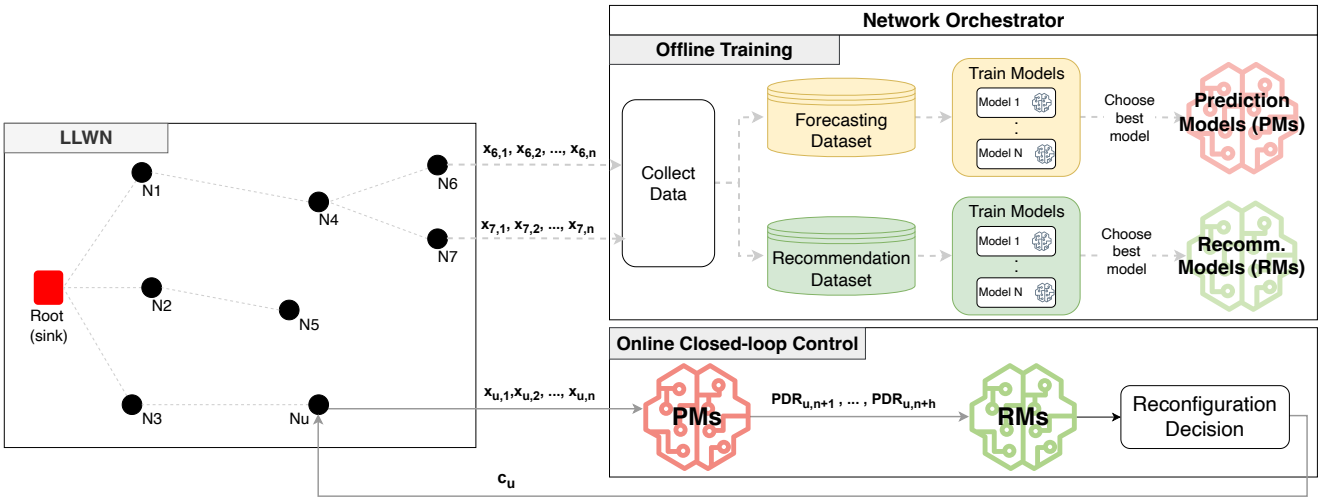


Fig. 3: Proposed framework consisting of two stages: offline training and online closed-loop control.

node generates a 1-byte data packet in bursts or periodically depending on the traffic scenario. During each run, every node exports MAC- and routing-level counters once per measurement window t of duration Δ seconds. These raw measurements are subsequently collected by the orchestrator, which transforms them into supervised datasets for training and evaluating the prediction and recommendation models.

B. Dataset Construction

For each node and time window t, we compute the following metrics. Let ΔS_t, ΔF_t, ΔB_t, and ΔQ_t denote the numbers of successful transmissions, failed transmissions, busy-channel events, and queue drops, respectively. In addition, ΔW_t and ΔT_t denote the radio wake time and the total elapsed time of the window. We define the total number of transmission-related events observed in the window $D_t = \Delta S_t + \Delta F_t + \Delta B_t + \Delta Q_t$.

From these quantities, we then define the main MAC-level performance indicators as:

$$\begin{aligned} \mathrm{PDR}_t &= \frac{\Delta S_t}{D_t + \varepsilon}, & \mathrm{Duty}_t &= \frac{\Delta W_t}{\Delta T_t + \varepsilon}, \\ \mathrm{BusyRatio}_t &= \frac{\Delta B_t}{D_t + \varepsilon}, & \mathrm{FailRatio}_t &= \frac{\Delta F_t}{D_t + \varepsilon}, \\ \mathrm{QueueLoss}_t &= \frac{\Delta Q_t}{D_t + \varepsilon}, \end{aligned} \tag{1}$$

Here, ε is a small constant to avoid division by zero. These quantities are complemented with topology and traffic descriptors, yielding the input vector:

$$\begin{aligned} \mathbf{z}_{i,t} = \Big(&\mathrm{PDR}_{i,t}, \mathrm{Duty}_{i,t}, \mathrm{BusyRatio}_{i,t}, \mathrm{FailRatio}_{i,t}, \\ &\mathrm{QueueLoss}_{i,t}, h_{i,t}, d_{i,t}, rssi_{i,t}, b_{i,t}, \mathrm{EWMAs}\Big), \end{aligned} \tag{2}$$

where $h_{i,t}$ denotes the hop distance to the root, $d_{i,t}$ the number of descendants, $rssi_{i,t}$ the received signal strength, and $b_{i,t} \in [0, 1]$ a normalized burstiness indicator.

The inclusion of these parameters is important because node performance in a multi-hop LLWN depends not only on local contention, but also on topological position, wireless channel conditions, and temporal structure of the offered traffic. The pipeline also computes exponentially weighted moving average (EWMA)-smoothed versions of key variables to capture short-term history with low online overhead. These features reduce short-term noise while remaining responsive to changes in LLWN conditions, thereby improving prediction and runtime control.

C. Offline Training and Model Design

1) Prediction Models: The prediction stage estimates the next-window response from the current observed state. For each node i, two regressors are learned:

$$\widehat{\mathrm{PDR}}_{i,t+1} = f_{\mathrm{pdr}}(\mathbf{z}_{i,t}), \qquad \widehat{\mathrm{Duty}}_{i,t+1} = f_{\mathrm{duty}}(\mathbf{z}_{i,t}), \tag{3}$$

Two bundles are trained offline, one for non-bursty (sporadic/periodic) conditions and one for bursty conditions, and the online orchestrator selects the appropriate bundle through a burstiness gate ($b_{i,t}$). For proactive control, the predictor is rolled over the next H windows and summarized as

$$\widehat{\mathrm{PDR}}_{i,t}^{(H)} = \frac{1}{H} \sum_{h=1}^{H} \widehat{\mathrm{PDR}}_{i,t+h}. \tag{4}$$

2) Recommendation Models: Whereas the predictor estimates what is likely to happen next, the recommender estimates which configuration should be applied. The framework includes two recommendation branches: a *CP-oriented recommender* for duty-cycle timing adaptation and a *TDMA-oriented recommender* for burst-driven slot allocation.

The *CP-oriented recommender* summarizes the last N windows of each metric into a vector $\mathbf{s}_{i,t}^{(N)}$ and predicts the run-level node performance expected under candidate GoMacH settings (c_i):

$$\widehat{\mathrm{PDR}}_i = g_{\mathrm{pdr}}(\mathbf{s}_{i,t}^{(N)}, c_i), \qquad \widehat{\mathrm{Duty}}_i = g_{\mathrm{duty}}(\mathbf{s}_{i,t}^{(N)}, c_i).$$

Affected nodes are those whose rolled predicted PDR over the next H windows falls below the target:

$$\mathcal{A}_t = \left\{ i \in \mathcal{N} \;\middle|\; \widehat{\mathrm{PDR}}_{i,t}^{(H)} < \mathrm{PDR}_{\mathrm{target}} \right\}.$$

A candidate c_i is feasible if $\widehat{\mathrm{PDR}}_i(c_i) \geq \mathrm{PDR}_{\mathrm{target}}$. The orchestrator then selects, among feasible candidates, the one minimizing predicted duty cycle. Otherwise, it resorts to a best-effort selection that prioritizes reliability improvement under a bounded duty-cycle cost.

The *TDMA-oriented recommender* selects a common slot-allocation budget for all nodes under bursty traffic:

$$\widehat{\mathrm{PDR}}_i = h_{\mathrm{pdr}}(\mathbf{s}_{i,t}^{(N)}, \texttt{max_slot}, b_{i,t}),$$

$$\widehat{\mathrm{Duty}}_i = h_{\mathrm{duty}}(\mathbf{s}_{i,t}^{(N)}, \texttt{max_slot}, b_{i,t}),$$

and chooses the configuration with the lowest predicted duty cycle among those satisfying the PDR target for the relevant nodes. Otherwise, it resorts to a best-effort selection.

In summary, offline learning is the foundation of the proposed control framework by combining prediction and recommendation models trained from experimental data. The prediction models provide a short-term view of future network behavior by forecasting next-window reliability and duty cycle, which supports proactive detection of degradation. The recommendation models extend this capability by selecting suitable GoMacH configurations, either through Contention Period (CP) adaptation in general conditions or TDMA slot allocation under bursty traffic. Together, these two model families transform historical measurements into actionable intelligence, enabling informed and adaptive runtime control.

D. Model Families and Performance Comparison

The contribution of this work does not lie in the design of a new machine-learning model, but in the integration of prediction and recommendation within our framework. Accordingly, we evaluate a set of established regression families and retain, for each task–target pair, the model that offers the best empirical performance. Since the mapping from observed network state to performance is nonlinear, interactive, and variable across nodes and operating conditions, the candidate set spans complementary inductive biases, from simple parametric models to flexible nonlinear learners:

- **Linear models: *Ridge* and *Lasso*:** Efficient and interpretable baselines, but limited in capturing strong nonlinear interactions.
- **Kernel and instance-based models: *SVR* and *KNN*:** Able to model nonlinear behavior, though often less robust in heterogeneous feature spaces.
- **Neural model: *MLP*:** Included as a generic nonlinear approximator for tabular telemetry data.
- **Tree-based ensembles and boosting: *Extra Trees*, *GBRT*, *HGBT*, and *XGBoost*:** Well suited to structured tabular data with strong interactions and threshold effects.

The final model is selected independently for each task–target pair by minimizing the Mean Absolute Error (MAE) under group-wise cross-validation:

$$\mathrm{MAE} = \frac{1}{N} \sum_{i=1}^{N} |y_i - \widehat{y}_i| \,, \tag{5}$$

where N is the number of evaluated samples, y_i the measured target, and $\widehat{y}_i$ its prediction.

Cross-validation is required because samples from the same experiment share the same MAC configuration, routing conditions, traffic setting, and environment. Let G_{train} and G_{test} denote the experiment identifiers assigned to the training and test folds. The split satisfies $G_{\mathrm{train}} \cap G_{\mathrm{test}} = \emptyset$, ensuring that each test fold contains only unseen runs and that the evaluation reflects cross-run generalization rather than memorization of run-specific trajectories.

TABLE II: Cross-validation MAE for prediction and recommendation models.

Model	Prediction				Recommendation			
	Non-bursty		Bursty		CP-oriented		TDMA-oriented	
	PDR (%)	DC (%)	PDR (%)	DC (%)	PDR (%)	DC (%)	PDR (%)	DC (%)
XGB	8.46	1.93	**8.19**	**1.81**	6.11	1.60	**4.67**	1.43
HGBT	8.42	1.93	8.21	1.84	6.15	**1.58**	4.71	1.63
GBRT	8.50	**1.92**	9.01	1.84	6.33	1.65	5.07	1.46
Lasso	8.74	2.08	10.45	2.17	6.53	1.78	5.45	1.71
ExtraTrees	**8.41**	1.95	8.82	1.82	**6.10**	1.58	4.70	**1.37**
Ridge	8.68	1.93	9.49	2.03	6.31	1.61	5.14	1.52
SVR	9.58	3.03	8.75	2.66	7.15	2.75	5.86	2.50
KNN	9.67	3.80	10.46	3.23	6.96	3.07	5.69	4.06
MLP	9.05	1.96	9.00	2.13	6.93	1.80	5.69	2.08

Table II shows that tree-based boosting and ensemble methods deliver the strongest overall performance. Extra Trees is selected for non-bursty PDR prediction, CP recommendation PDR and for TDMA-slot recommendation duty cycle, GBRT for non-bursty duty-cycle prediction, XGBoost for both bursty prediction targets and TDMA-slot recommendation PDR, and HGBT for CP recommendation duty cycle. Overall, these results indicate that tree-based ensembles constitute the most reliable model family for deployment, although simpler regularized models can remain competitive for specific targets.

E. Online Closed-Loop Control

The online orchestrator continuously reconstructs per-node state from runtime telemetry and uses the prediction and recommendation bundles to trigger actions. For each node i, a proactive trigger is generated when $\widehat{\mathrm{PDR}}_{i,t}^{(H)} < \mathrm{PDR}_{\mathrm{target}}$ for p $(patience)$ consecutive checks. Once activated, the corresponding recommender selects a new configuration and applies it to the affected nodes. To avoid oscillations, each reconfigured node enters a c $(cooldown)$ period during which further triggers are blocked. In parallel, a burst-protection mode monitors node burstiness. TDMA is enabled when $b_{i,t}$ exceeds a threshold B_{th} for consecutive P_{on} windows on at least a fraction R_{on} of the monitored nodes; it is disabled when the burstiness remains below B_{th} for consecutive P_{off} windows on at least a fraction R_{off}. When TDMA is activated,

TABLE III: Traffic profiles used in the evaluation.

Profile	Initial Config	*Phase-1*	*Phase-2*	*Phase-3*
Profile-1	$CP = 10$ ms $CP_{\max} = 30$ ms $SF = 400$ ms	**Periodic** (interval: 500 ms)		
Profile-2		**Periodic** (interval: 500 ms)	**Bursty** (30 pkts every 20 s)	**Periodic** (interval: 500 ms)
Profile-3		**Periodic** (interval: 1000 ms)	**Bursty** (15 pkts every 30 s)	**Periodic** (interval: 1000 ms)

the TDMA recommender selects the common `max_slot` that preserves reliability while limiting the cost of duty cycle.

In general, the proposed control framework combines node-specific timing reconfiguration under contention-based operation with network-wide TDMA activation and a shared slot-allocation decision under bursty traffic. This design enables the network to react proactively to predicted degradations while adapting its MAC operating mode and slot allocation to evolving traffic conditions.

IV. Performance Evaluation

The framework is evaluated in unseen experiments at the SLICES-FR Grenoble site, whose node placement and radio environment differ from those used for training, allowing us to assess generalization to new topologies and conditions.

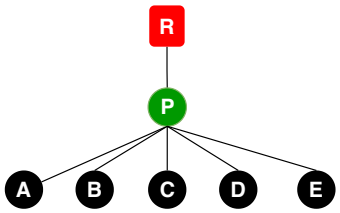


Fig. 4: Controlled topology used to evaluate the framework.

We first study the controlled topology of Fig. 4, composed of a root (R), an intermediate parent (P), and five leaves (A–E). This topology is used to enable interpretable comparisons across scenarios while preserving fixed structural roles in the routing hierarchy. Detailed analysis focuses on R, P, and one representative leaf (A), which together capture the behavior of nodes subject to different forwarding responsibilities. We then extend the evaluation to a larger heterogeneous deployment.

For the evaluation, we define three traffic profiles used across the experiments, as described in Table III. After a 3-minute warm-up period to allow routing convergence, each profile consists of three phases that may exhibit different traffic patterns: periodic traffic, in which 1-byte UDP packets are generated at a given interval, and bursty traffic, in which a group of 1-byte UDP packets is generated at a given period. A 3-minute no-traffic interval is inserted between consecutive phases. In all experiments, the initial duty-cycle configuration is fixed to $CP = 10$ ms, $CP_{\max} = 30$ ms, and $SF = 400$ ms.

The evaluation focuses on the closed-loop adaptation logic itself. Communication between the nodes and the network orchestrator is assumed to be reliable and is therefore considered outside the scope of this study. In practice, the network orchestrator collects telemetry through the SLICES-FR serial aggregation service and issues runtime reconfiguration commands accordingly. While this work does not directly rely on the full PENSIL implementation [5] for practical and experimental convenience, it builds on its core concept of a reconfigurable protocol stack, where network functions can be dynamically updated at runtime.

A. Forecasting Model Performance

We first evaluate the forecasting model over the controlled topology *with TDMA disabled*. We use Profile-2 (Table III) in a 3-hour experiment, with 30 minutes for Phase-1, 120 minutes for Phase-2, and the remaining time for Phase-3. Telemetry is collected every $\Delta = 15$ s. During rollout, the predictor is calibrated every k windows using real measurements; between calibrations, it relies on its own previously predicted state.

Fig. 5 reports the case $k = 1$, i.e., calibration at every window. The predicted curves reproduce both the average level and the short-term variations. The bursty phase, particularly for the leaf A, occasionally produces deep and sharp PDR drops that are more difficult to predict. Nevertheless, the model still captures the overall degradation pattern and the changes in averages. This is sufficient for control because the recommender relies on the average forecast over the next H windows rather than on an isolated point estimate.

Duty-cycle forecasts offer complementary information. Here, duty cycle is used as a proxy for power consumption, since SLICES-FR does not provide a radio-specific power measurement tool. The model reproduces the phase-dependent change in radio activity and also captures the lower duty-cycle baseline observed in the bursty phase. This lower duty cycle does not reflect higher efficiency; rather, the configured contention period is insufficient to absorb burst arrivals when TDMA is disabled, so many packets are dropped before transmission and radio activity decreases. Capturing this joint PDR/duty behavior is important because it helps distinguish true efficiency gains from degraded service.

In practice, frequent telemetry collection may be costly in LLWNs. For this reason, we study how the accuracy evolves as the calibration interval increases. Fig. 6 quantifies the effect of increasing the calibration interval from $k = 1$ window (15 s) to $k = 20$ windows (300 s). As expected, the error grows with k because the predictor relies longer on self-generated states before receiving a measurement-based correction. Nevertheless, the increase remains moderate. For PDR, the MAE rises from about 10% at short intervals to roughly 12.5% at $k = 20$. Duty cycle is easier to predict because it reflects aggregated radio activity over the window and is strongly constrained by the MAC timing configuration, whereas PDR depends on more stochastic packet-level events such as collisions, retransmission failures, queue drops, and burst-induced contention. As a result, duty-cycle trajectories are smoother and less sensitive to abrupt transient losses, leading to a median MAE that increases only from about 2%

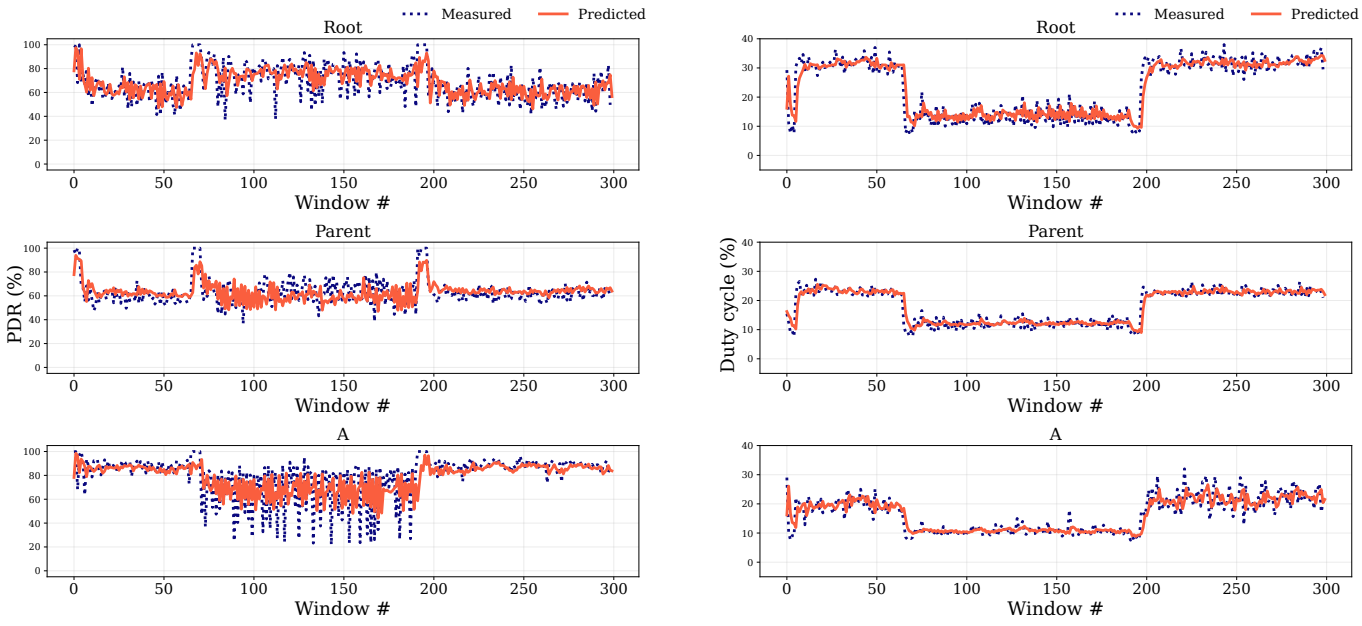


Fig. 5: Measured versus predicted trajectories for representative nodes: (a) PDR and (b) duty cycle.

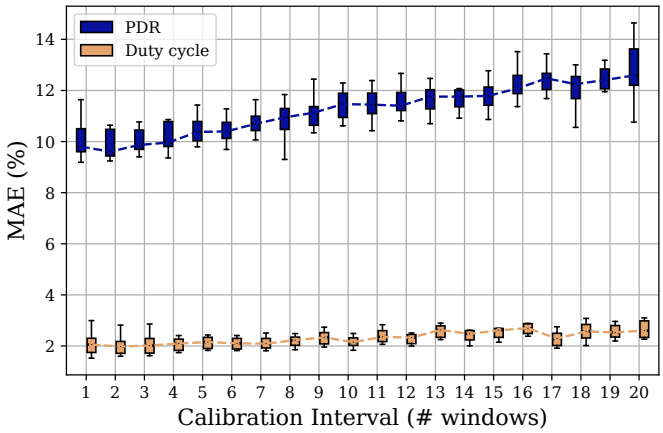


Fig. 6: Impact of the calibration interval k on forecasting error.

to 2.7%. The results indicate that the predictor remains useful even with relatively sparse calibration, which is important in LLWNs where traffic monitoring must remain limited.

B. Forecasting-based Recommendation Performance

To isolate the respective effects of recommendation and forecasting, and to highlight the importance of integrating both for proactive response, we run a 45-minute experiment with Profile-1 (Table III) running periodic data for the entire duration. The PDR target is set to 70%. In Fig. 7, we compare: *(i)* no recommendation or forecasting; *(ii)* recommendation without forecasting, where the orchestrator reacts to measured degradation; and *(iii)* recommendation with forecasting, where the orchestrator reacts to predicted degradation. The observation horizon is set to $N = 5$ and patience $p = 3$.

Fig. 7(a) shows that, under the static initial configuration with a superframe duration of 400 ms, leaf (A) remains relatively reliable and can sustain one packet every 500 ms. In contrast, the parent and the root remain well below the target because their forwarding burden, which includes both their own traffic and that of their descendants, is much higher. In addition to receiving upstream traffic, the root must continuously exchange routing information to maintain the topology, which further increases its load.

When recommendation is enabled without forecasting (Fig. 7(b)), the orchestrator eventually reconfigures the affected parent and root by reducing the duty-cycle period to 100 ms, thereby increasing the node wake-up rate and allowing both nodes to recover to a regime above the target. However, before reconfiguration is triggered, the root and the parent remain below the target for a substantial period, namely 252.1 s and 249.0 s, respectively, because the decision remains purely reactive.

The benefit of forecasting is evident in Fig. 7(c). In this case, the orchestrator anticipates degradation and updates the configuration before the moving-average PDR falls significantly below the target. The root remains below the target for only 62.4 s, while the parent avoids any sub-target period entirely, corresponding to reductions in degradation time of 75.2% and 100%, respectively. The few transient windows that still remain below the target are mainly due to the patience mechanism, since the orchestrator must still wait for $p = 3$ consecutive checks before applying the update. As a result, the parent and the root avoid the prolonged sub-target regime

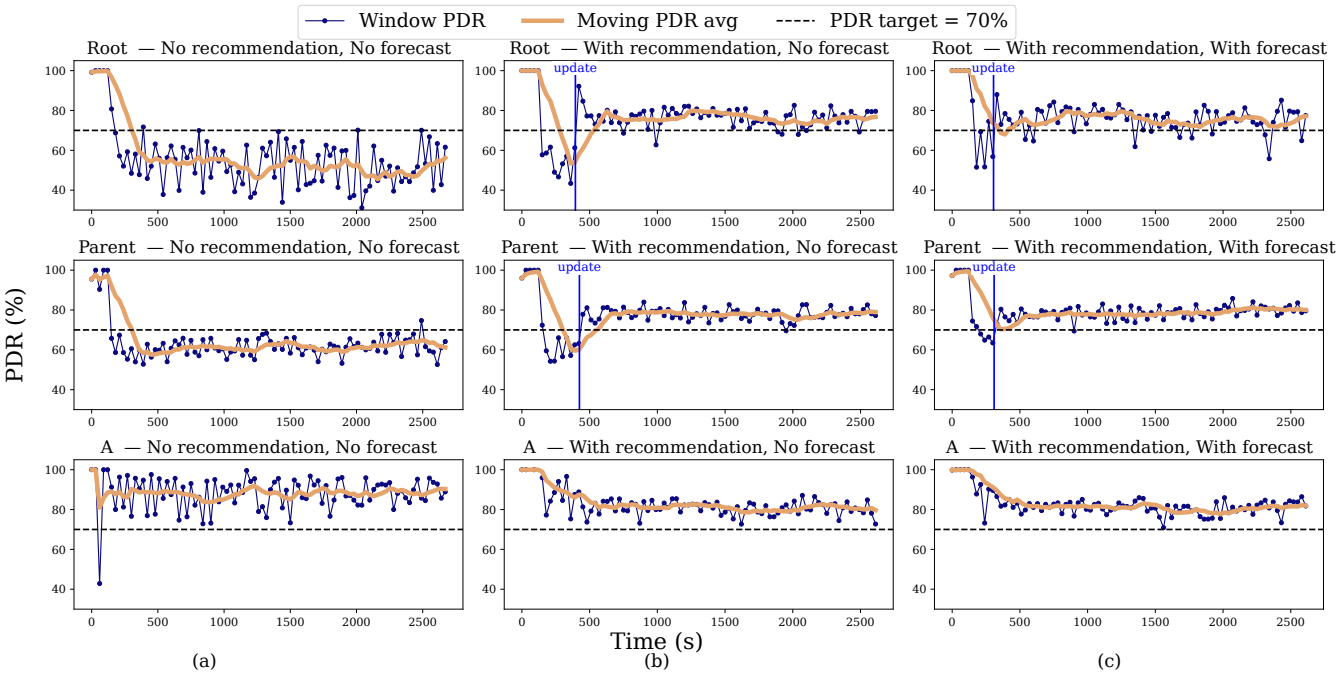


Fig. 7: PDR comparison under three scenarios: (a) no recommendation and no forecasting, (b) recommendation without forecasting, and (c) recommendation with forecasting.

observed in the purely reactive case. These results show that prediction improves the timing of control actions, thereby reducing both the duration and severity of performance degradation. Overall, they demonstrate that dynamic reconfiguration can achieve the target PDR in low-rate scenarios by adapting duty-cycle timing, without requiring a permanently active and costly TDMA service.

C. Adaptability to Traffic Profiles

In this section, we evaluate adaptability under one-hour experiments with a mixed traffic pattern defined by Profile-2 (Table III) in the controlled topology. Phase-1 lasts 10 minutes, Phase-2 lasts 30 minutes, and Phase-3 occupies the remaining time. The PDR target is set to 70%. We compare two cases: *(i)* TDMA unavailable, such that the orchestrator can only tune the duty-cycle timing; and *(ii)* TDMA available and activated when burstiness becomes high, namely when at least 20% of nodes satisfy $b_{i,t} \geq 0.5$. In the figures, solid vertical markers indicate duty-cycle timing reconfigurations, whereas dashed markers indicate TDMA-enabled updates.

Without TDMA (Fig. 8(a)), the orchestrator applies duty-cycle timing adaptation, and the configuration ($CP = 10$ ms, $CP_{\max} = 30$ ms, and $SF = 100$ ms) is sufficient during the periodic phases. However, the bursty phase generates sustained queue buildup, repeated channel conflicts, retransmissions, and queue drops at a rate that exceeds the ability of duty-cycled contention-based medium access to absorb them. As a result, all nodes fall well below the 70% target, despite the orchestrator continuing to apply best-effort contention-parameter updates.

TABLE IV: Comparison of performance with TDMA disabled, always on, and dynamically activated.

Metric	No TDMA	Always-on TDMA	Dynamic TDMA
PDR (%)	**66.4**	**90.9**	**85.3**
Queue loss (%)	21.8	5.3	5.5
Failure (%)	11.7	3.7	9.2
Duty-cycle (%)	**18.7**	**36.5**	**25.5**

When dynamic TDMA is available (Fig. 8(b)), the orchestrator applies duty-cycle timing adaptation during the periodic phase, but it switches to TDMA during the bursty phase and selects a suitable slot-allocation budget, `max_slot=30`. Reliability then recovers rapidly to a regime close to or above the target because scheduled access reduces contention and serves queued burst packets more deterministically. Once traffic returns to periodic operation, TDMA is disabled and the previous configuration is restored. Later reconfigurations may be applied either with the same parameters to include newly affected nodes or with a completely new configuration.

To highlight the importance of dynamically reconfiguring the contention-based duty cycle timing and enabling TDMA only when contention-based access becomes insufficient, we compare the PDR and duty cycle of three scenarios: TDMA permanently disabled, TDMA permanently enabled, and our framework, in which TDMA is activated dynamically only when needed, as described in Table IV. Without TDMA, the network cannot achieve the 70% target because the bursty phase generates a large queue loss that cannot be handled by reconfiguring only contention-based duty-cycle access.

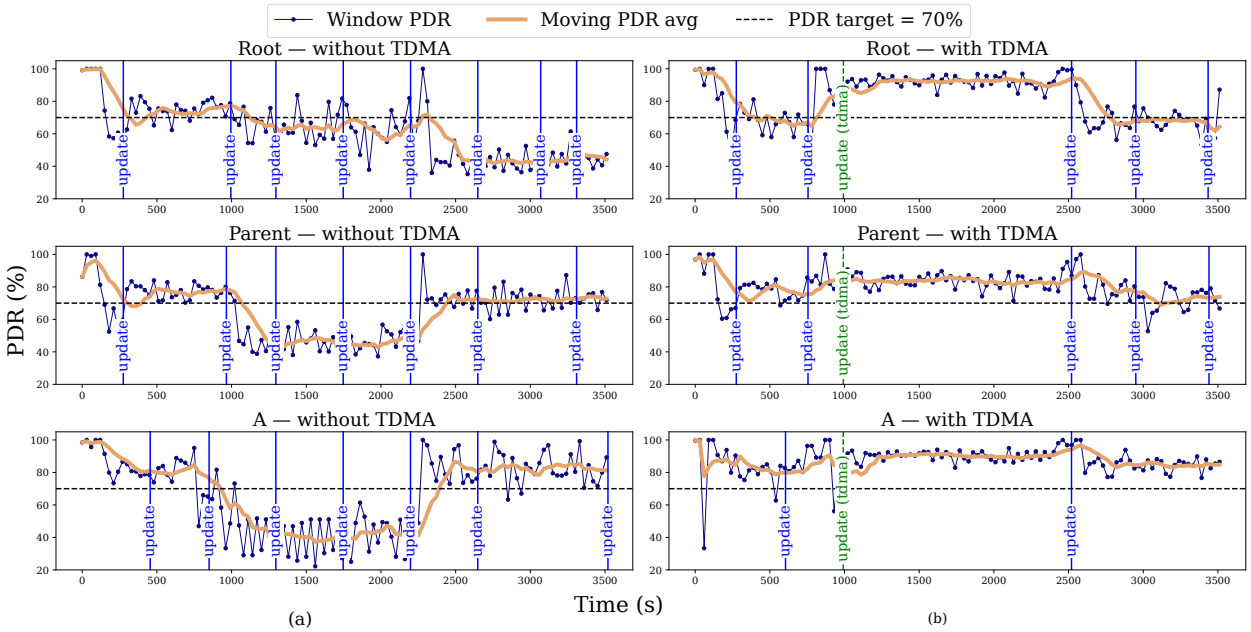


Fig. 8: PDR comparison under mixed traffic profile: (a) without TDMA and (b) with dynamic TDMA.

Always-on TDMA yields the highest PDR and the lowest queue loss, but at a substantially higher duty-cycle cost. In contrast, our dynamic-TDMA framework achieves the target by enabling TDMA only when necessary: it reaches queue-loss levels close to those of the always-on TDMA case while reducing the duty-cycle cost by 30.13% relative to permanent TDMA. These results show that TDMA should not remain permanently disabled or permanently enabled; rather, it should be activated selectively during bursty periods and released when contention-based operation becomes sufficient again. This behavior is justified by TDMA synchronization overhead, which can become comparable to the actual data traffic under low-rate operation and consequently increase radio activity and energy consumption.

D. Scalability over a Large Heterogeneous Topology

In this subsection, we evaluate the scalability of the proposed framework on a larger multi-hop deployment with stronger topology heterogeneity and forwarding asymmetry. We use the largest connected 50-node subset at the Grenoble site, where nodes form routes to the root over 1 to 4 hops. The network runs for one hour under the mixed traffic Profile-3 (Table III). Phase-1 lasts 10 minutes, Phase-2 lasts 30 minutes, and Phase-3 covers the remaining time. The PDR target is set to 80%. We compare two scenarios: *(i)* dynamic configuration using the proposed framework and *(ii)* a static configuration without runtime reconfiguration. We report only PDR results in this experiment, as duty-cycle values exhibit strong spatial and temporal heterogeneity across nodes in this irregular multi-hop topology, which makes aggregate statistics such as mean values and confidence intervals non-informative.

Fig. 9 reports the global average PDR with its standard deviation, together with the runtime reconfiguration actions applied during dynamic operation. At the beginning of the initial periodic phase, several nodes trigger duty-cycle timing updates to accommodate the generated traffic. After this initial adjustment, reconfiguration becomes more selective and mainly targets forwarding nodes, allowing the network to maintain the target PDR with lower variability than in the static scenario. When the bursty phase begins, the orchestrator issues a global TDMA-enabled update with a common slot-allocation budget `max_slot=100`, which keeps the average PDR above the target. By contrast, the static configuration suffers a severe degradation during the bursty phase. After the bursty phase, TDMA is disabled again, and only occasional duty-cycle timing updates are applied. This behavior illustrates the two complementary adaptation mechanisms at scale: contention-parameter tuning under moderate load to limit energy cost, and TDMA switching under burst-induced stress to preserve reliability. Quantitatively, the static configuration spends 31.7% of the experiment below the target PDR, whereas the proposed framework reduces this duration to 14.6%.

Taken together, these results show that the proposed framework remains effective in a larger and more heterogeneous multi-hop setting and adapts its decisions to both evolving traffic conditions and topology-dependent forwarding load.

V. Conclusion and Future Work

This paper presented a traffic-aware closed-loop control framework for the LLWN MAC layer that combines runtime monitoring, short-horizon forecasting, recommendation, and online reconfiguration. The framework exploits per-node telemetry to anticipate near-future degradation and select suitable MAC-layer configurations, enabling contention-based duty-cycle timing adaptation under moderate traffic and TDMA-oriented protection under bursty traffic.

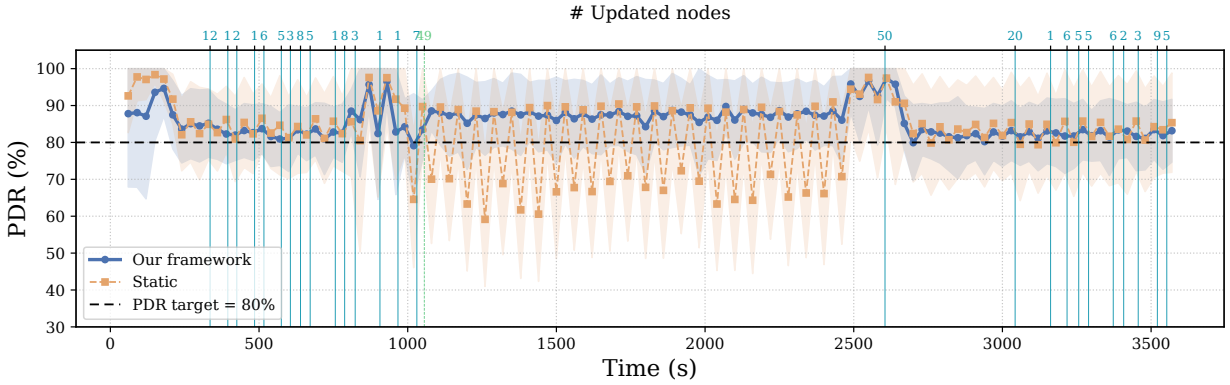


Fig. 9: Global average PDR in the 50-node heterogeneous topology under closed-loop control and static configuration.

Experimental results on the SLICES-FR testbed showed that the proposed framework generalizes to unseen deployments and remains effective under different topological and radio conditions. The forecasting models captured the main evolution of PDR and duty cycle while maintaining acceptable average prediction errors, with MAE values of up to 12.5% for PDR and 2.7% for duty cycle even under long calibration intervals. The integration of forecasting with recommendation improved the timing of runtime actions compared with reactive control, reducing degradation time by 75% to 100%. Under mixed traffic, the framework adapted its MAC operating mode to the traffic regime, restoring reliability during bursty phases while reducing the duty-cycle cost by more than 30% relative to always-on TDMA. Evaluation on a larger heterogeneous topology further showed that the framework scales beyond the controlled scenario and can maintain the global average PDR close to or above the target.

Future work will investigate online learning models that dynamically adapt prediction and recommendation, and will extend the validation of the proposed learning-based control toward finer-grained runtime adaptation across multiple layers using our programmable protocol stack proposed in [5].

Acknowledgment

This research has been funded by the French National Research Agency (ANR) under project ANR-23-CE25-0008. With a view to open access publication, the author has applied for an open access CC-BY licence for any manuscript accepted for publication (AAM) as a result of this submission.

References

[1] Y.-K. Chen *et al.*, "Challenges and opportunities of internet of things," in *17th Asia and South Pacific Design Automation Conference*, 2012, pp. 383–388.

[2] A. Oliveira *et al.*, "Low-power and lossy networks under mobility: A survey," *Computer Networks*, vol. 107, pp. 339–352, 2016.

[3] M. Mazo *et al.*, "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2456–2461, 2011.

[4] L. Wu *et al.*, "pQueue-MAC: an energy efficient hybrid mac protocol for event-driven sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 2, p. 160167, 2015.

[5] A. Mahmod, J. Montavont, and T. Noel, "PENSIL: Programmable network stack for low-power lossy IoT networks using lightweight-virtualization," *Internet of Things*, vol. 35, p. 101829, 2026.

[6] H. Lamaazi *et al.*, "Performance assessment of the routing protocol for low power and lossy networks," in *International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2015, pp. 1–8.

[7] B. Finley *et al.*, "How does enterprise IoT traffic evolve? Real-world evidence from a Finnish operator," *Internet of Things*, vol. 12, p. 100294, 2020.

[8] G. Kim *et al.*, "Dynamic duty-cycle MAC protocol for IoT environments and wireless sensor networks," *Energies*, vol. 12, no. 21, p. 4069, 2019.

[9] S. Ray *et al.*, "Supporting bursty traffic in wireless sensor networks through a distributed advertisement-based TDMA protocol (ATMA)," *Ad Hoc Networks*, vol. 11, no. 3, pp. 959–974, 2013.

[10] M. A. Y. Siam *et al.*, "An efficient burst traffic-aware clustering and mobility management model for mobile-sink based heterogeneous wireless sensor networks," in *Proceedings of the 11th International Conference on Networking, Systems, and Security*, 2024, pp. 96–103.

[11] S. Zhuo *et al.*, "GoMacH: A traffic adaptive multi-channel MAC protocol for IoT," in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*. IEEE, 2017, pp. 489–497.

[12] Y. Sun *et al.*, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3072–3108, 2019.

[13] M. Ateeq, M. K. Afzal, M. Naeem, M. Shafiq, and J.-G. Choi, "Deep learning-based multiparametric predictions for IoT," *Sustainability*, vol. 12, no. 18, p. 7752, 2020.

[14] A. Hameed *et al.*, "A machine learning regression approach for throughput estimation in an IoT environment," in *2021 IEEE International Conferences on Internet of Things (iThings)*. IEEE, 2021, pp. 29–36.

[15] R. Ala'a *et al.*, "Optimizing resource allocation in industrial IoT with federated machine learning and edge computing integration," *Results in Engineering*, p. 106387, 2025.

[16] S. O. Olatinwo *et al.*, "Deep learning for resource management in Internet of Things networks: A bibliometric analysis and comprehensive review," *IEEE Access*, vol. 10, pp. 94 691–94 717, 2022.

[17] N. Jiang *et al.*, "Reinforcement learning for real-time optimization in NB-IoT networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1424–1440, 2019.

[18] J. Xin *et al.*, "A deep-learning-based MAC for integrating channel access, rate adaptation, and channel switch," *Digital Communications and Networks*, 2024.

[19] C.-M. Wu *et al.*, "A Q-Learning-Based Adaptive MAC Protocol for Internet of Things Networks," *IEEE Access*, vol. 9, pp. 905–918, 2021.

[20] A. Aijaz *et al.*, "Closing the loop: A high-performance connectivity solution for realizing wireless closed-loop control in industrial IoT applications," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 11 860–11 876, 2021.

[21] E. Baccelli *et al.*, "RIOT: An open source operating system for low-end embedded devices in the IoT," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4428–4440, 2018.