INTER IIT
TECH MEET 14.0

Client Centric WiFi RRM

INTER IIT TECH MEET 14.0

End Term Submission Report

Team 33

ARISTA

Index

1 Introduction 1
1.1 Problem Statement 1
1.2 Work Summary 1
1.3 Remaining Challenges 1

2 System Architecure 1

3 Fast Loop (Reactive Management) 2

4 Event Loop 2
4.1 Response Strategy & Classification 2
4.2 Quiet Hours Management 3

5 Slow Loop 3

6 RRM-Plus AI Methods 3
6.1 Graph Neural Networks (GNN) 3
6.2 Graph Colouring for Channel Assignment 5
6.3 Causal Inference for Validation 5
6.4 Reinforcement Learning 6
6.5 Three-Phase Learning Pipeline 7

7 Stability V/S Churn Control 8

8 Service Level Objectives (SLOs) 8
8.1 Implementation Strategy 9
8.2 Pipeline Integration Summary 9

9 Advanced Client View 9
9.1 802.11k/mc/r Standards Integration 9
9.2 Transport-Layer QoE Validation 9
9.3 Crowd Sourced App Probes 10
9.4 Passive Inference 10

10 KPIs & Acceptance Criteria (End-Term) 10

11 Conclusion 10

References 11
Appendix 11

1 Introduction

1.1 Problem Statement

The challenge in Enterprise WiFi is moving beyond the traditional AP-centric model, which provides delayed and incomplete RF visibility. RRM-Plus aims to be an AI-driven, client-focused RRM system that uses a dedicated sensing radio and standards-based client telemetry to remove blind spots. Its goal is a closed-loop optimizer that adjusts channel, power, and bandwidth dynamically while maintaining strict stability controls.

1.2 Work Summary

In Mid-Term, we successfully established the foundational architecture for the RRM-Plus system that include

- **Sensing Orchestrator:** Implemented edge-intelligence pipeline using additional radio to perform continuous spectrum analysis without interrupting client traffic. This includes a lightweight CNN for classifying non-WiFi interference (e.g. Microwave, BLE), a Multi-Armed Bandit (MAB) algorithm for adaptive dwell time scheduling.
- **Client-View Integration:** We operationalized the IEEE 802.11k/v protocols to actively acquire client-side RSSI and SNR metrics, enabling the system to see the network from the device's perspective.
- **Optimization Engine:** We deployed a Bayesian Optimizer (BO) to tune RRM parameters (Channel Width, Transmit Power, OBSS-PD) to maximize our SLO subject to stability constraints.
- **Safety Mechanisms:** We designed and validated a Safe-Change Planner that enforces site change budgets and supports automatic rollbacks to prevent network instability.

1.3 Remaining Challenges

Current implementation successfully optimizes a single AP, but many critical challenges are to meet the End-Term:

- **Global Coordination:** The current system lacks a graph-based global planner. We need to implement Graph Neural Networks (GNN) to model complex inter-AP interference relationships and topology.
- **Multi-Timescale Control:** Decouple the control logic into distinct loops–Fast (seconds), Event (trigger-based), and Slow (hours)–to handle both transient interference spikes and long-term capacity planning effectively.
- **Causal Validation:** Moving beyond correlation, we need to integrate Causal Inference to mathematically validate that specific configuration changes are the true cause of QoE improvements.
- **Advanced Telemetry:** Client view -deepened by adding transport-layer metrics (TCP/QUIC RTT), 802.11mc RTT for location-aware decision-making, ensuring we solve application-layer issues, not just signal strength.

2 System Architecture

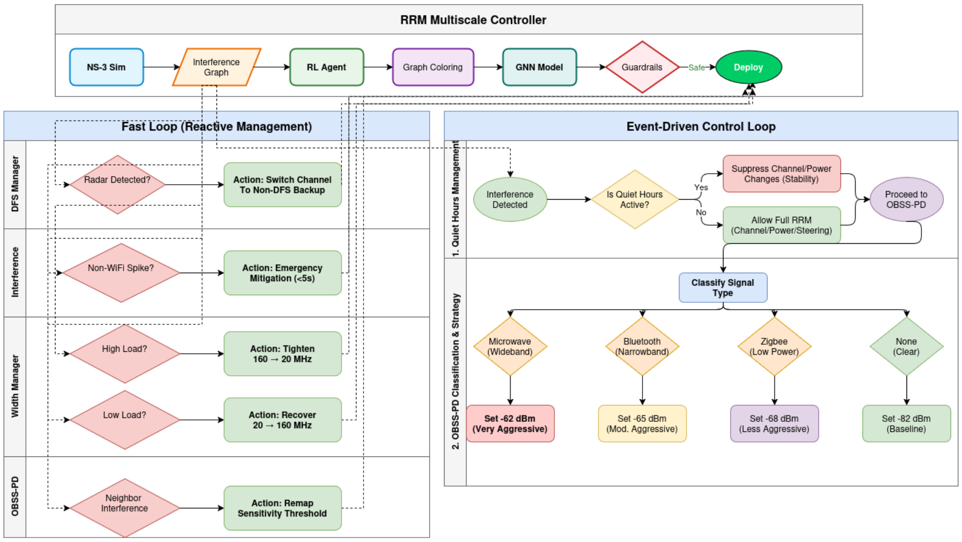

RRM Multiscale Controller
NS-3 Sim
Interference Graph
RL Agent
Graph Coloring
GNN Model
Guardrails
Safe
Deploy
Fast Loop (Reactive Management)
DFS Manager
Radar Detected?
Action: Switch Channel To Non-DFS Backup
Interference
Non-WiFi Spike?
Action: Emergency Mitigation (<5s)
Width Manager
High Load?
Action: Tighten 160 → 20 MHz
Low Load?
Action: Recover 20 → 160 MHz
OBSS-PD
Neighbor Interference
Action: Remap Sensitivity Threshold
Event-Driven Control Loop
1. Quiet Hours Management
Interference Detected
Is Quiet Hours Active?
Yes
No
Suppress Channel/Power Changes (Stability)
Allow Full RRM (Channel/Power/Steering)
Proceed to OBSS-PD
2. OBSS-PD Classification & Strategy
Classify Signal Type
Microwave (Wideband)
Bluetooth (Narrowband)
Zigbee (Low Power)
None (Clear)
Set -62 dBm (Very Aggressive)
Set -65 dBm (Mod. Aggressive)
Set -68 dBm (Less Aggressive)
Set -82 dBm (Baseline)

Figure 1: System Architecture

RRM-Plus functions as a closed-loop controller built on real-time telemetry that forms an AP interference graph which is optimized using an RL agent, a DSATUR-based channel planner and a GNN oracle to evaluate proposed configurations. Production guardrails regulate change budgets, hysteresis, rollout safety and rollback. The system continually updates the GNN through online learning based on real-world outcomes.

3 Fast Loop (Reactive Management)

The Fast Loop handles reactive optimization (DFS, bandwidth, OBSS-PD) on a seconds-to-minutes timescale.

Module	**Detection / Trigger Logic**	**Response / Strategy**
DFS Manager	**Radar Detect:** Channels 52–140. **Check:** Enforces 60s Channel Availability Check (CAC).	**Switch:** Move to Non-DFS Backup. • Low Band (36–48) **OR** • High Band (149–165).
Interference	**Method:** 10-sample window (3s vs baseline). **Trigger:** Jump > 15% **AND** Util > 75%.	**Action:** Emergency mitigation in < 5s. **Scope:** Ignores standard WiFi congestion; targets non-WiFi signatures only.
Width (Tighten)	**Trigger:** Util > 75% **OR** OBSS-PD > 30%. **Emergency:** Non-WiFi > 30%.	**Action:** Reduce 160 → 20 MHz. (Emergency triggers Channel Switch w/ 20s hold).
Width (Recover)	**Trigger:** Util < 50% **AND** OBSS-PD < 15%. **Stability:** Must hold 30s (5% hysteresis).	**Action:** Expand 20 → 160 MHz. Prevents "ping-pong" oscillations.
OBSS-PD	**Range:** -82 to -62 dBm. **Logic:** Neighbor interference ≥ 25%.	**Action:** Remap sensitivity threshold. **Network:** Dynamic AP graph; Co-channel (0.8) / Adj-channel (0.4) coeffs.

4 Event Loop

The Event Loop provides rapid interference response through spectrum analysis and adaptive OBSS-PD adjustments on a 100 ms cycle. It detects and classifies non-Wi-Fi sources and applies targeted mitigation for `30 seconds`. During this period, the system enters a **high-priority EVENT OVERRIDE MODE** that preempts the Fast Loop and ML Optimizations. Interference is confirmed only after 5 consecutive detections (500 ms) to minimize false positives.

4.1 Response Strategy & Classification

The system applies targeted OBSS-PD thresholds based on interference classification. Each type requires distinct mitigation due to different spectral characteristics.

Table 1: Event Loop Response Actions

Type	**OBSS-PD**	**Classification & Strategy**
Microwave	-62 dBm	**Wideband (>40 MHz) near 2.45 GHz.** Very aggressive threshold ignores high-power bursts.
Bluetooth	-65 dBm	**Narrowband (<2 MHz), 2.40–2.48 GHz.** Moderately aggressive for frequency-hopping transients.
Zigbee	-68 dBm	**Narrowband (1–5 MHz) near 2.405 GHz.** Less aggressive for continuous low-power sensors.
None (Clear)	-82 dBm	**No interference detected.** Restores baseline sensitiviRty (IEEE 802.11ax default).

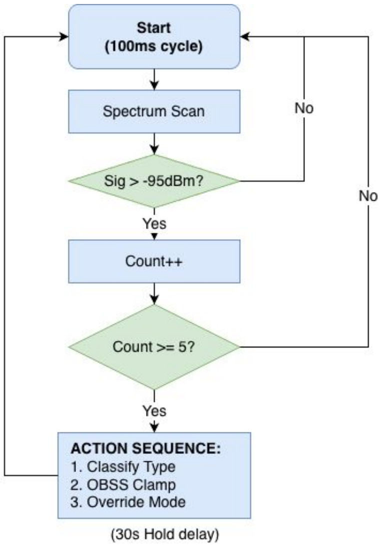


Figure 2: Event-Driven Control Logic

4.2 Quiet Hours Management

During exam periods, Quiet Hours temporarily pause most RRM actions—like channel changes, power adjustments, and client steering—while allowing OBSS-PD threshold tweaks to mitigate interference. This preserves stability in sensitive spaces (e.g., exam halls) without ignoring urgent interferences. APs are tagged with zone types, and scheduled windows activate the policy so the Event Loop keeps detecting interference but suppresses disruptive RRM changes.

5 Slow Loop

The slow loop performs global optimization every **4 hours** using network-wide data. It builds an interference graph of AP interactions, encodes it with a GNN, and feeds the embeddings to an RL agent that proposes parameter updates. A graph-coloring algorithm then computes channel assignments, and GNN message passing predicts next-state QoE. It works on a two-phase operation which are; **1. Passive Training:** RL learns from GNN-predicted QoE changes. **2. Online Deployment:** Every 4 hours, if expected QoE shows improvement, proposed changes are executed.

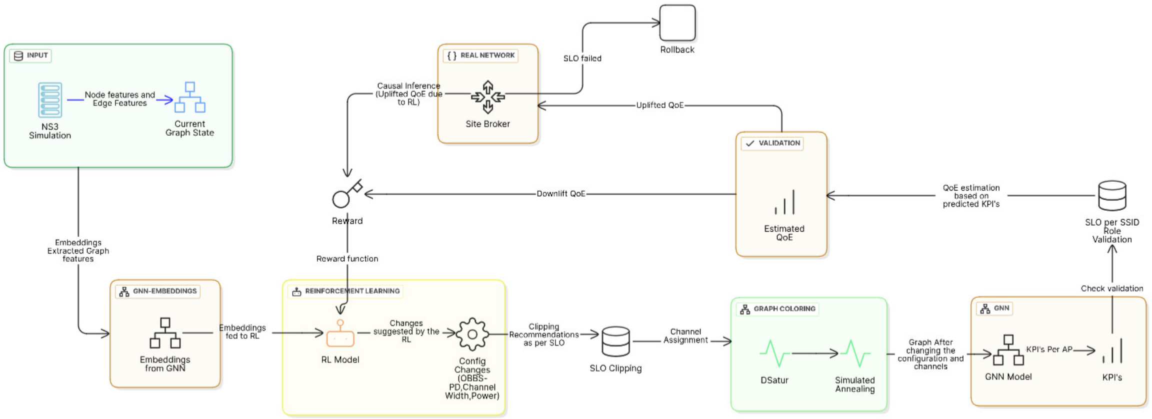


Figure 3: Slow Loop Process Flow

6 RRM-Plus AI Methods

Interference directed graph is modeled as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Nodes are APs and edges (i, j) exist if $\textbf{RSSI}_{ij} > -95$ **dBm**.

6.1 Graph Neural Networks (GNN)

The GNN acts as the **predictive engine**, modeling non-linear dynamics like hidden nodes and ripple effects.

Key Design Principles

- **Inductive Generalization:** No AP IDs; allows zero-shot transfer to new sites without retraining.
- **Physics-Based:** Learns from RF fundamentals (power, load, interference) rather than memorizing identities.
- **Client-Centric Focus:** Edge clients (RSSI $\in [-70, -65]$ dBm) weighted 3× in loss function.
- **Node Vector:** Encodes AP config, load state, and client distribution metrics.
- **Edge Vector:** Models **physical interference** (path loss, overlap) and **hidden node proxies** using feature enhancement.

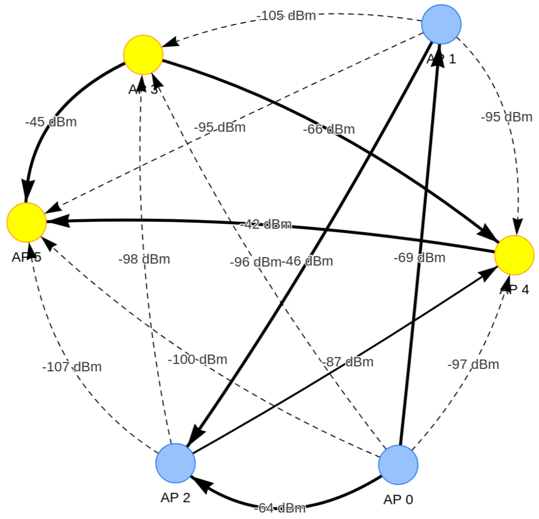


Figure 4: Topology of the Interference Graph. Edges denote contention links exceeding -95 dBm.

6.1.1 GNN Architecture & Hierarchical Objective

The model employs a **three-stage Graph Neural Network (GNN)** architecture optimized for inference speed (< 1ms for 50 APs) and inductive capability, allowing deployment across diverse network sizes without retraining.

6.1.2 Input Embedding & Fusion

Node features are processed via **parallel pathways** and fused using an additive connection with **layer normalization:**

$$\boxed{h_i^{(0)} = \text{LayerNorm}\left(\text{ReLU}\left(\underbrace{[E^W[w_i] \,\|\, E^{ax}[\nVdash_i^{ax}]]}_{\text{Categorical Embedding}} + \underbrace{(W_{num}\tilde{x}_i^{num} + b_{num})}_{\text{Numerical Projection}}\right)\right)} \tag{1}$$

6.1.3 Edge-Gated Message Passing ($L = 3$)

We employ edge-gated **multi-head attention**. For layer :

$$\boxed{\begin{aligned} m_{j\to i} &= \sigma(W_{gate}h_{ij}^{edge}) \odot (W_{msg}h_j^{(\ell)} + W_{bias}h_{ij}^{edge}) && (2)\\ a_i &= \sum_{k\in\{\text{max, sum, mean, std}\}} \alpha_{i,k} \cdot \text{Agg}_k(\{m_{j\to i}\}_{j\in\mathcal{N}(i)}) && (3)\\ h_i^{(\ell+1)} &= \text{LayerNorm}\left(\text{ReLU}\left(W_{up}[h_i^{(\ell)}\|a_i] + h_i^{(\ell)}\right)\right) && (4) \end{aligned}}$$

Here, $\alpha_{i,k}$ are **learnable weights** that allow each AP to dynamically prioritize different interference patterns (*max –> dominant*, *sum –> cumulative*, *mean –> desnity*, *std –> heterogeneity* interference).

6.1.4 Output Configuration & Loss Function

The final representation h_i^{final} aggregates all layers via **Jumping Knowledge attention**. This embedding feeds into specialized heads (Table 2) trained on a **hierarchical loss function**:

$$\boxed{\mathcal{L}_{total} = \frac{w_1\mathcal{L}_{Safe} + w_2\mathcal{L}_{Rel} + w_3\mathcal{L}_{Perf} + w_4\mathcal{L}_{Cal}}{\sum_i w_i}} \tag{5}$$

- **Safety ($\mathcal{L}_1$):** Weighted BCE on SLO violations, penalizing missed faults 10× more.
- **Reliability ($\mathcal{L}_2$):** Weighted MSE focusing on Edge Clients and Critical Regimes ($T < 5$ Mbps).
- **Performance ($\mathcal{L}_3$):** Log-space MSE ($\log(\hat{y}/y)^2$) to capture diminishing returns.

Table 2: Model Configuration and Prediction Heads

Model Config		**Output Heads**			
Component	**Value**	**Target**	**Description**	**Activation**	**Constraint**
Hidden Dim	64	$\hat{\mu}^{T,all}$	Mean Tput (All)	Softplus	> 0
Layers	3	$\hat{\mu}^{T,edge}$	Mean Tput (Edge)	$\alpha \cdot \hat{\mu}^{T,all}$	$\leq \hat{\mu}^{\mathbf{T,all}}$
Params	∼78k	$\hat{\sigma}^T$	Std Dev Tput	Softplus	> 0
Aggregators	4	$\hat{r}^{(95)}$	P95 Retry Rate	Sigmoid	$\in (0, 1)$
Inductive	Yes	$\hat{\ell}^{(95)}$	P95 Latency	Softplus	> 0

Inductive Generalization: The model can be trained on 50-AP networks and deployed on 1000, or any number of APs **without retraining**– a critical requirement for real-world deployment across diverse enterprise sites.

6.1.5 Explainability of GNN

Our framework provides transparency through axiomatic attribution methods, satisfying *Sensitivity*, *Invariance*, and *Completeness.*

GNN: Topology Impact

Integrated Gradients (Feature Importance): Attributes predictions to inputs (e.g.RSSI) by integrating gradients from a baseline x'.

$$\boxed{\text{IG}_i(x) = (x_i - x'_i)\int_0^1 \frac{\partial F}{\partial x_i} d\alpha} \tag{6}$$

Interference Flow Attribution (IFA): Quantifies interference $j \to i$ using attention weights γ and gradient sensitivity.

$$\boxed{\Phi_{j\to i} = \sum_{\ell} \gamma^{(\ell)} \left\| \frac{\partial \hat{y}_i}{\partial m_{j\to i}} \right\|} \tag{7}$$

Counterfactual Reasoning: Finds minimal config change x' for target $\hat{y}_{\text{tgt}}$.

$$\boxed{x^* = \arg\min \|x' - x\| + \lambda \text{Penalty}} \tag{8}$$

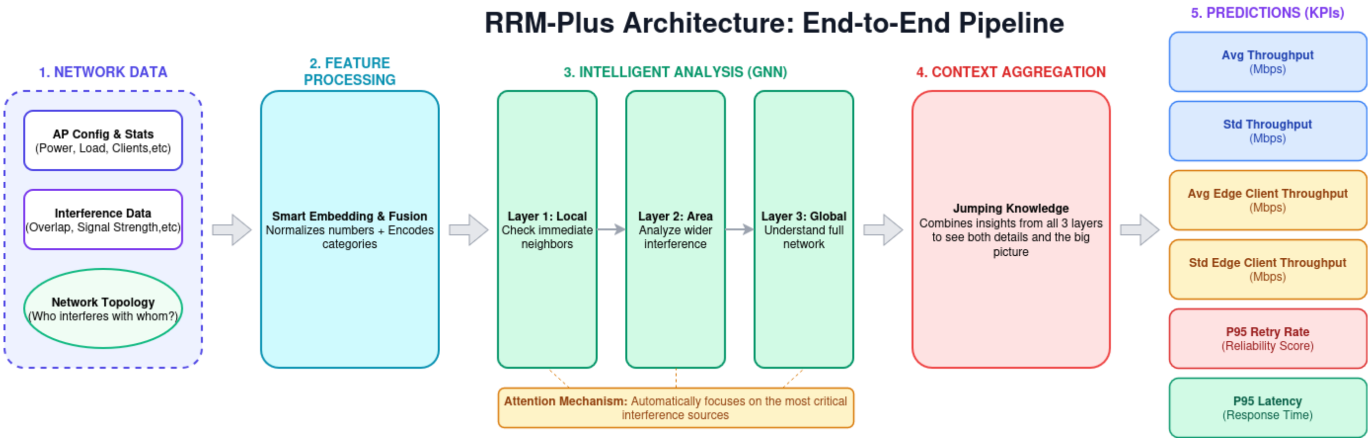

RRM-Plus Architecture: End-to-End Pipeline
1. NETWORK DATA
AP Config & Stats
(Power, Load, Clients,etc)
Interference Data
(Overlap, Signal Strength,etc)
Network Topology
(Who interferes with whom?)
2. FEATURE PROCESSING
Smart Embedding & Fusion
Normalizes numbers + Encodes categories
3. INTELLIGENT ANALYSIS (GNN)
Layer 1: Local
Check immediate neighbors
Layer 2: Area
Analyze wider interference
Layer 3: Global
Understand full network
Attention Mechanism: Automatically focuses on the most critical interference sources
4. CONTEXT AGGREGATION
Jumping Knowledge
Combines insights from all 3 layers to see both details and the big picture
5. PREDICTIONS (KPIs)
Avg Throughput
(Mbps)
Std Throughput
(Mbps)
Avg Edge Client Throughput
(Mbps)
Std Edge Client Throughput
(Mbps)
P95 Retry Rate
(Reliability Score)
P95 Latency
(Response Time)

Figure 5: GNN Architecture

6.2 Graph Coloring for Channel Assignment

We model the network as a weighted conflict graph $G = (V, E)$, integrating **Conservative Q-Learning (CQL)** outputs (bandwidth, power) as constraints. The channel assignment minimizes global interference energy using a two-phase approach.

Cost Function Architecture: Minimizes $\mathcal{E}_{\text{global}}$ by balancing interference severity, stability, and regulatory constraints. Edge cost $\mathcal{J}_{\text{edge}}$ quantifies spectral overlap:

$$\boxed{\mathcal{J}_{\text{edge}}(u, v) = W_{uv} \times \alpha(c_u, c_v) \times K_{uv}} \tag{9}$$

where $\boxed{\mathrm{W}_{uv} = 10^{(P_u - \text{PathLoss}uv)/10} \times (1 + \text{RoamFactor}uv)}$ is coupling strength (path loss + roaming), α is the overlap coefficient, and K_{uv} accounts for 802.11ax spatial reuse.

Algorithm 1: Robust DSATUR: Constructs an initial valid assignment. Prioritizes nodes based on saturation degree and load $(\text{SatDeg}(u), |N(u)|, \text{Clients}_u)$ to color highly constrained APs first.

Algorithm 2: Simulated Annealing (Refinement): Refines the DSATUR solution via **stochastic exploration.** Accepts worse solutions with probability $e^{-\Delta E/T}$ to escape local minima, using geometric cooling $(T_{k+1} = 0.95T_k)$.

6.3 Causal Inference for Validation

To **distinguish genuine improvements** from spurious correlations (e.g., natural traffic dips), we employ a robust causal framework to estimate the true treatment effect (τ). We rely on an **ensemble of four techniques** to strictly control for confounding variables.

Inference Ensemble Methods

- **Propensity Score Matching (PSM):** Matches treated APs with control units of similar configuration probability to balance covariate distribution.
- **Difference-in-Differences (DiD):** Isolates treatment impact by comparing pre-post evolution against a control group, removing common temporal trends.
- **Synthetic Control:** Constructs a custom counterfactual baseline for unique APs using a weighted combination of control units.
- **Double Machine Learning (DML):** Leverages ML to residualize high-dimensional confounders (e.g., complex interference patterns) and isolate causal effects.

Validation Criteria: We compute the median estimate across all four methods. Configuration gains are reported only if this aggregate effect is statistically significant *($p < 0.05$)*.

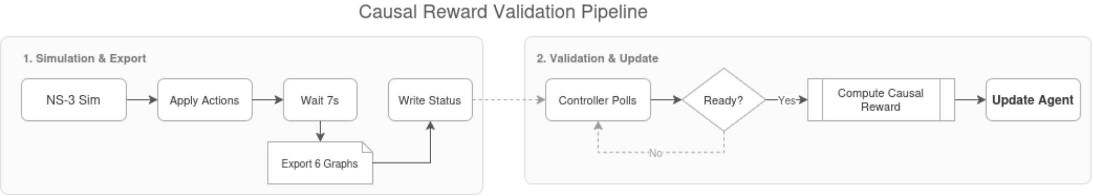


Figure 6: Causal Inference with RL

6.4 Reinforcement Learning (RL)

We use a **three-phase Safe RL pipeline** combining (1) **Offline Conservative Critics**, (2) **Policy Distillation**, and (3) **Online RCPO fine-tuning**. The system is modeled as a **Constrained MDP (CMDP)** $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, C, \gamma, \epsilon \rangle$, maximizing QoE while enforcing retry-rate SLOs.

$$\boxed{r = w_1 \Delta_{\text{QoE}} + w_2 \nVdash [\Delta_{\text{QoE}} < 0] \cdot 2\Delta_{\text{QoE}} + w_3 \text{QoE}_{\text{new}}}$$

$$\boxed{c = \mathbb{I}(\text{retry\_rate} > 8\%)}$$

$$\boxed{\max_{\pi} J_r(\pi) = \mathbb{E}[\sum_t \gamma^t r_t], \quad \text{s.t. } J_c(\pi) \leq \epsilon}$$

$$\boxed{\max_{\pi} \min_{\lambda \geq 0} J_r(\pi) - \lambda(J_c(\pi) - \epsilon)}$$

State: Graph with $N \in [10, 50]$ APs, encoded to 128-dim by GNN. **Actions:** 225 discrete $(P_{\text{tx}}, \text{OBSS-PD}, \text{Width})$ combinations. **Cost:** SLO violation; **Reward:** asymmetric QoE-drop penalty.

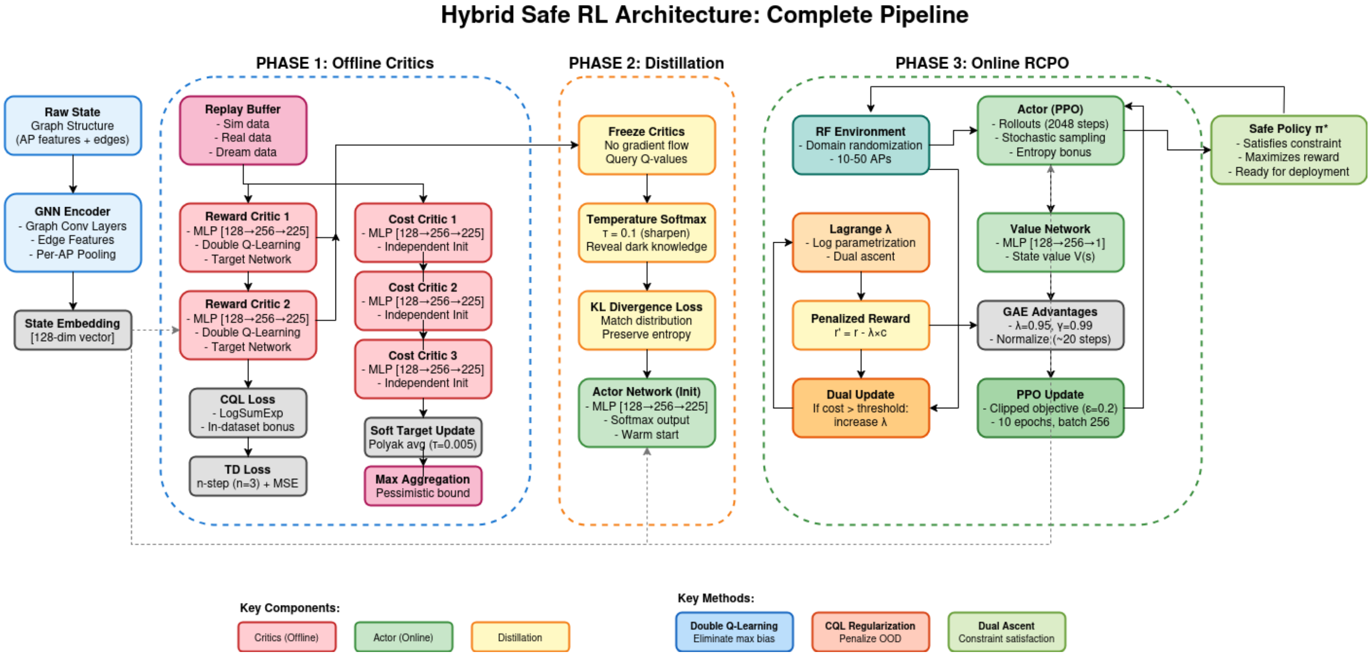


Figure 7: Hybrid RL Architecture

6.4.1 Explainability of Reinforcement Learning

The proposed framework creates explainable agent behavior by decoupling safety initialization (Phase I) from runtime risk management (Phase II).

- **Phase I: Safety via Pessimism (CQL)**
  - **Mechanism:** This phase utilizes Conservative Q-Learning (CQL) to enforce robust initialization.
  - **Logic:** The agent explicitly rejects Out-of-Distribution (OOD) actions—state-action pairs not seen in the offline dataset. It treats **low Q-values as "unfamiliar"** territory by applying an epistemic penalty proportional to the uncertainty ($\alpha \cdot$ Uncertainty).

$$\boxed{Q_{\text{CQL}} = Q_{\text{Bellman}} - \alpha \cdot \text{Unc.}} \tag{10}$$

- **Phase II: Risk-Reward Balancing (RCPO)**
  - **Mechanism:** This phase employs Reward Constrained Policy Optimization (RCPO) for online adaptation.
  - **Logic:** The agent **dynamically** balances the pursuit of Reward (R) against Safety constraints (C). A dynamic Lagrangian multiplier (λ_t) serves as a **"safety price."** If safety constraints are violated, λ_t increases, triggering a "recovery mode" where the agent prioritizes safety over reward.

$$\boxed{A_{\text{Lag}} = \hat{A}_R - \lambda_t \cdot \hat{A}_C} \tag{11}$$

- **Summary of Methodology**
  - **Offline Safety:** The system uses CQL for robust initialization, ensuring the agent begins with a pessimistic view of unknown states.
  - **Online Adaptation:** RCPO allows for adaptive tuning to handle real-world dynamics that may differ from the offline data.
  - **Sim-to-Real Gap:** The combination of **distillation** and the **λ-Regulator bridges** the gap between simulation and deployment.

6.5 Three-Phase Learning Pipeline

Methodology & Logic	**Mathematical Formulation**
1. Offline Conservative Critic (CQL) Combines **Double Q-learning** (bias removal) and **CQL regularization** (penalizes OOD actions). Uses ensemble critics ($Q_{c,i}$) with pessimistic aggregation to ensure safety during offline training.	$\mathcal{L}_{\text{CQL}} = \alpha \left[\log \sum_a e^{Q(s,a)} - \mathbb{E}_{\pi_\beta}[Q] \right] + \mathcal{L}_{\text{TD}}(\text{Double } Q)$
2. Policy Distillation (KL) The actor mimics the stable critic distribution using **KL divergence**. Critics use **temperature softmax** ($\tau = 0.1$) to generate targets, preserving relative action preferences and preventing premature collapse.	$p_\tau(a\|s) = \frac{\exp(Q(a)/\tau)}{\sum_{a'} \exp(Q(a')/\tau)}$ $\mathcal{L}_{\text{KL}} = \sum_a p_\tau(a\|s) \log \frac{p_\tau(a\|s)}{\pi_\theta(a\|s)}$
3. Online RCPO Fine-Tuning Deploys PPO + GAE with **domain randomization**. Enforces SLOs via **Dual Ascent**, which dynamically increases the penalty weight λ if the average cost $\bar{C}$ violates the safety budget ϵ.	$r'_t = r_t - \lambda c_t$ $\lambda_{k+1} = \max(0, \lambda_k + \eta(\bar{C} - \epsilon))$

$$\boxed{\begin{aligned} \log \lambda_{k+1} &= \log \lambda_k + \alpha_\lambda(\bar{c} - \epsilon) \\ \hat{A}_t &= \sum_l (\gamma \lambda_{\text{GAE}})^l \delta_{t+l} \\ \mathcal{L}_{\text{PPO}} &= \mathbb{E}[\min(r_t A_t, \text{clip}(r_t, 1 \pm \epsilon) A_t)] \end{aligned}}$$

6.5.1 RL Results

We proposed a three-phase Safe RL pipeline that successfully decouples safety estimation from policy improvement. By distilling offline conservative critics into an online adaptive actor, the system bridges the gap between static stability and dynamic fine-tuning.

Our method achieves a **+18.9% throughput gain** over offline baselines while strictly satisfying the retry rate SLO ($< 10\%$). As shown below, it offers the optimal trade-off compared to unconstrained or purely conservative baselines.

Method	Throughput	Violations	Status
Ours (Hybrid)	**2.78**	**8.7%**	**Optimal (Safe)**
Unconstrained PPO	2.89	14.3%	Unsafe
CPO	2.56	9.8%	Suboptimal
CQL-Only	2.34	$< 8.0\%$	Conservative

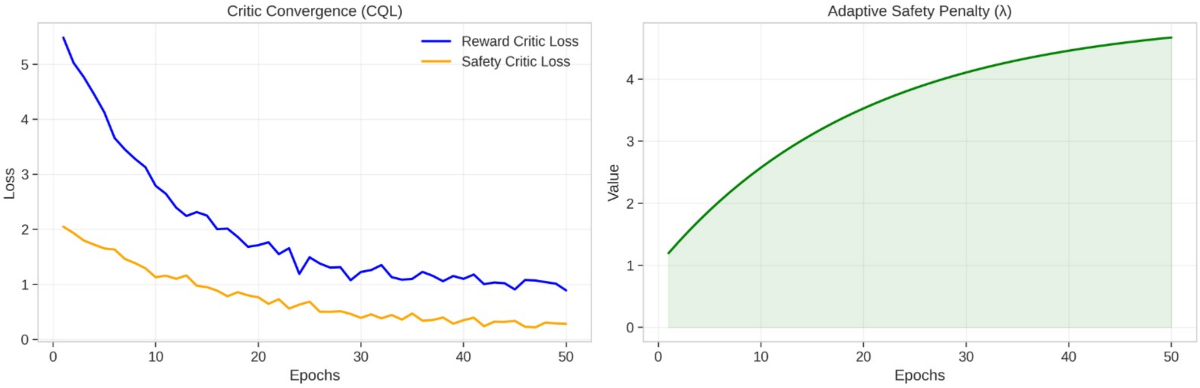


Figure 8: Critic Convergence (Left) and Adaptive Safety Penalty(Right)

Key Innovations

- **RL Integration:** Treats RL-decided parameters (width, power) as fixed constraints, separating continuous (RL) and discrete (Graph Coloring) optimization.
- **Efficiency:** Achieves interference-minimizing assignments for 50-AP networks in < 2 seconds.

7 Stability V/S Churn Control

To maintain production stability, the Safe Change Planner implements a phased rollout strategy with blast radius control, ensuring all network adjustments align with site occupancy and defined site change budgets. This architecture utilizes Dynamic Hysteresis Control with specific time-of-day windows to minimize churn, while simultaneously maintaining an audit trail and rapid rollback mechanisms for safety. This ensures churn control and enhances the operational readiness of our RRM system.

8 Service Level Objectives (SLOs)

Hard Constraints apply *before* GNN prediction **(clipping RL/GC outputs)** to ensure *safety* and regulatory compliance. **Performance SLOs** are soft constraints verified *against* GNN-predicted KPIs to **optimize** user experience.

Role	Hard Constraints	Perf. SLOs
Voice	• 20 MHz max • Tx: 12–18 dBm • 5 GHz pref	• P95 Lat $\leq$ 150ms • Retry $\leq 8\%$
Exam	• 20 MHz max • Freeze mode • OBSS-PD low	• Airtime $\leq 85\%$
IoT	• 2.4 GHz only • Tx $\leq$ 16 dBm	• RSSI ≥ -75 dBm
Guest	• 80 MHz max • Tx $\leq$ 20 dBm	• Airtime $\leq 70\%$

Table 3: Role-Based Hard Constraints and Performance SLOs (Split View)

8.1 Implementation Strategy

Aspect	Clipping Pass (Hard Constraints)	Freeze Pass (Time-Window Guardrails)	Rejection Pass (Performance SLOs)
Goal	Constrain RL output to valid physical limits.	Block RRM changes during restricted time windows.	Reject configurations predicted to violate performance SLOs.
When	Immediately after RL generates an action.	First step of Site Broker / Change Manager.	After GNN prediction.
Logic	1. Read RL-proposed parameters. 2. Identify AP role. 3. Clamp values to the role's hard limits.	1. Check system time. 2. Verify if the role has an active freeze window. 3. If active, return a **No-Op**.	. 1. Retrieve GNN predictions $\hat{Y}$. 2. Load role-specific SLO thresholds. 3. Reject if any predicted metric violates its SLO.

8.2 Pipeline Integration Summary

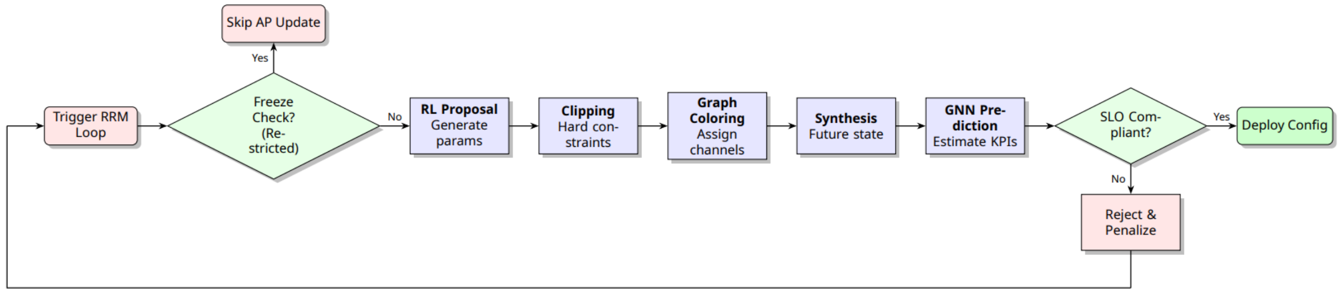


Figure 9: Pipeline Of SLO integration

9 Advanced Client View

RRM-Plus implements a **multi-layer client-view system** correlating MAC, transport, and physical-layer metrics to diagnose QoE degradation root causes.

9.1 802.11 mc/r/ Standards Integration

- **802.11mc FTM:** 5 frame FTM bursts measure RTT to estimate client-AP distances via $d = (RTT \cdot c)/2$. Measurements are spatially aggregated into 5-meter grid cells, identifying interference hotspots when per-cell retry rates exceed 15%. Privacy is maintained through spatial binning and 5-minute data retention limits.

- **802.11r Fast Transition:** Sub-50ms handovers are achieved via implementation in `ieee80211r-ft.h` with **PMK caching**. Target: P50 roam time < 100ms.
  - **ESS Requirement:** We assume all Access Points (APs) within the deployment area belong to the **same Extended Service Set (ESS)**, which is a prerequisite for standard 802.11r operation.
  - **PMK Caching:** The **Pairwise Master Key (PMK)** is securely cached across all APs belonging to the ESS using a common **PMK-R0 Key Holder (R0KH)** to eliminate the need for full re-authentication during a roam.
  - The function `ExecuteFastTransition()` implements the fast Basic Service Set (BSS) transition mechanism.

9.2 Transport-Layer QoE Validation

The system employs passive TCP/QUIC monitoring to detect client-side performance issues that remain invisible to MAC-layer metrics. By tracking end-to-end RTT variance and packet loss patterns, the framework identifies scenarios where clients experience poor quality despite clean wireless channel conditions (e.g., low retry rates, good RSSI).

High RTT variance ($\sigma > 50$ms) combined with negligible MAC retries ($< 5\%$) indicates client-side bottlenecks such as CPU throttling, application stalls, or receiver buffer limitations. This cross-layer correlation prevents unnecessary AP reconfigurations by accurately isolating transport-layer issues from wireless medium problems, enabling targeted remediation strategies.

9.3 Crowd Sourced App Probes

Idle-Window Sensing: Background ICMP probes (64-byte packets) are triggered during idle channel periods ($> 30\%$ idle time) at a maximum rate of 5 probes per second per client. The `CrowdsourcedProbeConfig` structure tracks per-client round-trip time (RTT), packet loss, and jitter statistics. Adaptive rate limiting adjusts probe frequency based on channel utilization: minimum 200 ms spacing during idle windows, maximum 2-second interval during congestion. Network overhead: approximately 2.56 Kbps per client at maximum rate. The collected latency metrics provide continuous visibility into transient channel behavior for network characterization.

9.4 Passive Inference

Legacy Device Support: For non-802.11k clients, passive inference uses UL/DL MCS divergence, ACK timing variance ($\sigma > 5$ms), and retry asymmetry to detect hidden nodes and interference. Enables graceful degradation without requiring client support for advanced standards.

10 KPIs & Acceptance Criteria (End-Term)

KPI Improvements for the End-Term Goal: A Comparison Between the Start and End of Our Observation Period.

Metric	**Start**	**End**	**Change**	**Notes**
Edge Client Throughput (kbps)	5605442	7679456	37%	>35% Achieved
P95 Retry Rate (%)	11.052	4.682	−57%	<-30% achieved
BSS-TM Success Rate (%)	80.2	93.1	12.9%	>90% achieved
FT Roam Time P50 (ms)	150	33	117	<100ms achieved
RU Utilization Gain (%)	4.17	17.63	+13.46%	>15% achieved

Table 4: New KPI index

Metric	**BO**	**RL**
Avg. QoE Score	-0.03	0.96
Edge Median Throughput	1159.98 kbps	767956 kbps
P95 Retry Rate	0.16%	0.09%
BSS-TM Acceptance	90.3%	93.1%
BSS-TM Requests	40584/44946	50455/55322

Table 5: System Performance Metrics

11 Conclusion

We have successfully simulated an AP–Client environment in NS-3, integrated with GNN and RL to enhance user experience. The system also includes SiteBroker, SLO enforcement, rollback mechanisms, 802.11k/v/r/mc, audit trails, causal inference with explainability for both GNN and RL, seamless client roaming, and improved airtime efficiency and stability—supported by fast, event, and slow control loops.

References

[1] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *International Conference on Learning Representations (ICLR)*, 2018. https://arxiv.org/abs/1710.10903

[2] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How Powerful are Graph Neural Networks?" *International Conference on Learning Representations (ICLR)*, 2019. https://arxiv.org/abs/1810.00826

[3] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic Attribution for Deep Networks," *International Conference on Machine Learning (ICML)*, 2017. https://arxiv.org/abs/1703.01365

[4] R. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "GNNExplainer: Generating Explanations for Graph Neural Networks," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019. https://arxiv.org/abs/1903.03894

[5] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-Learning for Offline Reinforcement Learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020. https://arxiv.org/abs/2006.04779

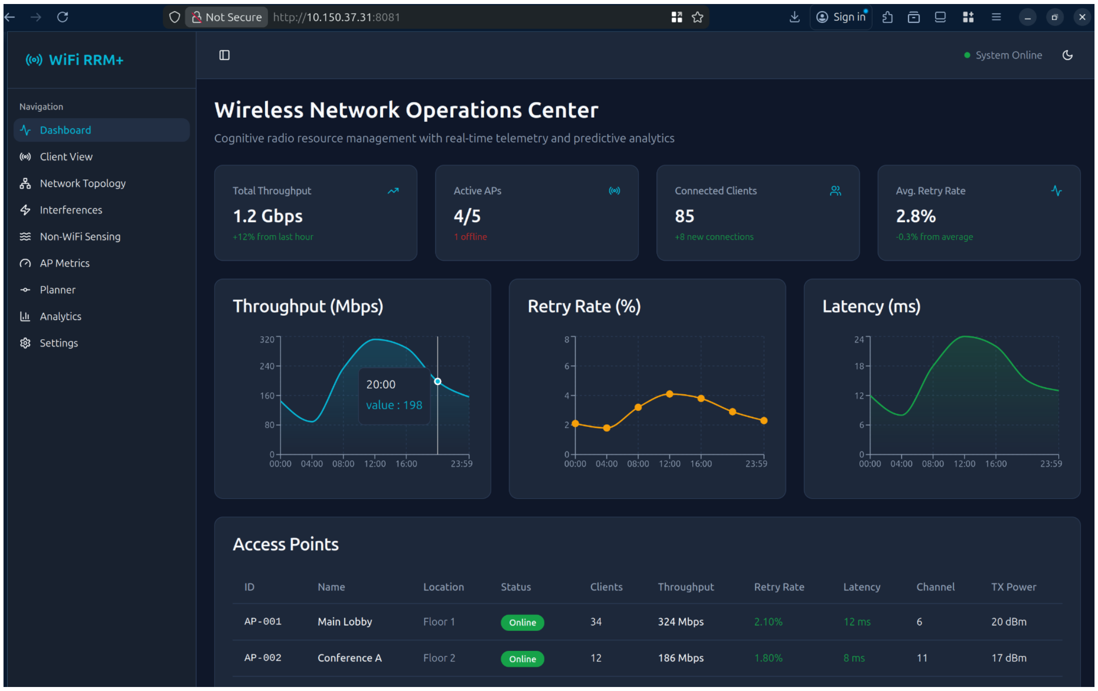
[6] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained Policy Optimization," *International Conference on Machine Learning (ICML)*, 2017. https://arxiv.org/abs/1705.10528

[7] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Hyperparameters," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 25, 2012. https://arxiv.org/abs/1206.2944

[8] D. Brélaz, "New Methods to Color the Vertices of a Graph," *Communications of the ACM*, vol. 22, no. 4, pp. 251–256, 1979. https://dl.acm.org/doi/10.1145/359094.359101

[9] P. Gutierrez and J. Y. Gérardy, "Causal Inference and Uplift Modelling: A Review of the Literature," *JMLR Workshop and Conference Proceedings*, vol. 67, 2017. http://proceedings.mlr.press/v67/gutierrez17a.html

Appendix



Wireless Network Operations Center

Cognitive radio resource management with real-time telemetry and predictive analytics

Total Throughput
1.2 Gbps
+12% from last hour

Active APs
4/5
1 offline

Connected Clients
85
+8 new connections

Avg. Retry Rate
2.8%
-0.3% from average

Access Points

ID	Name	Location	Status	Clients	Throughput	Retry Rate	Latency	Channel	TX Power
AP-001	Main Lobby	Floor 1	Online	34	324 Mbps	2.10%	12 ms	6	20 dBm
AP-002	Conference A	Floor 2	Online	12	186 Mbps	1.80%	8 ms	11	17 dBm

Figure 10: Website Dashboard