Energy-Efficient Wireless Network Simulation Using Reinforcement Learning

Preetham Narla

***Abstract*—We present a modular wireless network simulator that uses reinforcement learning (RL) to adapt base-station power states for energy savings. The simulator models a realistic multi-site cellular topology with a 3GPP-style channel and fine-grained MAC/PHY functions, enabling detailed energy accounting. An RL agent observes network states (e.g. traffic load, SINR, QoS) and takes actions (e.g. put sector in micro-sleep or deep-sleep, adjust power or resource allocation) to minimize energy use. Guardrails ensure QoS constraints are met. In synthetic experiments, the RL-controlled network achieves roughly 15% higher energy efficiency (bits/Joule) compared to a baseline, demonstrating the approach's potential.**

I. Introduction

Energy efficiency is a critical goal for modern wireless networks as data demand grows. Cellular networks consume large power due to always-on infrastructure and dense deployments. Reducing base-station energy use lowers costs and carbon footprint while sustaining service quality. Reinforcement learning offers a flexible way to dynamically manage resources: it can learn control policies that adapt to traffic and channel conditions. Here we motivate an end-to-end simulator where an RL agent controls base-station sleep modes and power levels to save energy under QoS constraints. This report outlines the simulator design and demonstrates a roughly 15% energy-efficiency gain.

II. System Architecture

The simulator is modular and pluggable: it separates components for topology, channel, MAC/PHY, energy states, control logic, and metrics. This enables easy swapping of models or algorithms. Key modules include:

- **Topology and Mobility:** Defines the network layout and user movement. For example, we simulate a 7-site hexagonal grid (reuse-1) with 3 sectors per site, as often used in cellular studies.
- **Traffic and Channel:** Generates user data traffic and applies a 3GPP-style channel model (pathloss, shadowing, small-scale fading).
- **MAC/PHY Layer:** Schedules resource blocks and adapts modulation/coding per user/channel (e.g. using SINR and CQI maps).
- **Energy Model:** Models power consumption in different states (active, micro-sleep, deep-sleep) and transitions (with wake-up delays and guard-timers).
- **Control (RL Agent):** Observes network state (load, SINR, buffer occupancy, etc.) and chooses actions (sleep states or power levels) to minimize energy.
- **Metrics and Dashboard:** Collects KPIs such as throughput, delay, energy use, and provides visualization (time-series plots, breakdown charts, heatmaps).

The modular simulator architecture is shown above. Each stage (topology, channel, MAC/PHY, energy model, control) is plug-in based. In the default scenario, we use a 7-cell hexagonal grid with inter-cell interference and mobile users. The RL agent controls each sector's sleep/power state subject to QoS guardrails: it may transition a sector from Active to Micro-sleep or Deep-sleep based on observed traffic and channel quality, but it will wake it up if QoS degrades.

III. Channel Modeling

We adopt standardized 3GPP channel models for realism. The simulator applies path loss and shadowing based on urban macro or micro scenarios, plus small-scale fading (e.g. Rayleigh or Rician). The channel model also includes antenna sectorization (each site has 3 directional antennas) and inter-cell interference. For each user, the simulator computes SINR and the corresponding achievable throughput. This informs the QoS constraints: e.g., the RL agent must maintain user throughput above a threshold. By using 3GPP-style models, the simulator captures realistic variation in signal quality.

IV. Energy States and Transitions

We model three main energy states for each sector: *Active*, *Micro-sleep*, and *Deep-sleep*. In the Active state, the transmitter and signal processing are fully on. A Micro-sleep (sometimes called idle or micro-sleep) state is a short-duration low-power mode where the transmitter is off but can quickly resume (e.g. between subframes). A Deep-sleep (or standby) state is a longer-duration low-power mode, requiring wake-up delay. The state machine transitions are governed by guard timers and QoS checks, as illustrated. For example, a sector may only enter Deep-sleep if it has been idle for several frames and no users need service; if a new user appears or QoS drops, the sector immediately returns to Active. These guards prevent excessive latency. The energy model assigns specific power usage to each state (e.g. deep-sleep consumes only a fraction of Active power), enabling fine-grained energy accounting.

V. Reinforcement Learning Approach

The RL agent periodically observes a state vector (e.g. number of active users per sector, current rates, buffer occupancy, SINRs) and selects actions to reduce energy use. Actions include putting each sector into Active/Micro/Deep-sleep or adjusting its transmit power and user scheduling. The goal is to minimize cumulative energy over time. A reward (or cost) function penalizes energy consumption while including terms for QoS: for example, dropping throughput below a target incurs a large negative reward. We implement a standard RL loop: the agent uses (for example) Q-learning or deep Q-networks to update its policy based on observed reward feedback. To enforce QoS constraints, the action space is restricted: the agent cannot keep sectors in deep-sleep if users are waiting beyond a threshold, for example. In summary, the RL agent learns a policy that trades off energy savings against service quality. It effectively learns patterns such as "in low-traffic periods, put the sector to micro-sleep/deep-sleep; wake it promptly when load increases."

VI. Dashboard and Metrics

The simulator includes a visualization dashboard. Key outputs are *time-series plots* of metrics (e.g. total energy consumption, network throughput over time), an *energy breakdown* chart (showing consumption per component or per state), and an *efficiency heatmap* (spatial map of bits-per-Joule across the grid). For instance, the time-series plot can show baseline vs. RL-controlled energy usage, making trends clear. The breakdown shows how much energy is saved by sleeping vs. active operation. The heatmap reveals which sectors run hotter (inefficiently) and which benefit most from RL control. These visual outputs help diagnose performance and validate that the RL policy is indeed saving energy without harming user experience.

VII. Evaluation

We evaluate the RL approach with synthetic simulations. For example, in a scenario with variable user load, the RL-driven policy is compared to a baseline (always-on or periodic duty cycling). Table I and the accompanying figures summarize key results. The RL policy achieves roughly a 15% reduction in average power consumption while keeping user throughput within 1% of baseline. This yields a comparable 15% increase in energy efficiency (bits/Joule).

TABLE I
Synthetic evaluation: baseline vs. RL-enabled.

Metric	Baseline	RL-Enabled
Average Power (W)	100.0	85.0
Sum Throughput (Mbps)	500.0	495.0
Energy Efficiency (Mb/J)	5.00	5.82
% Efficiency Gain	—	+16.4%

VIII. Research Gaps and Future Scope

While the simulator demonstrates promising energy savings, future work can extend it in several ways. One gap is real-world integration: the simulator could be validated with trace data from actual base stations or connected to live testbeds. The RL approach can be improved by using deep reinforcement learning or multi-agent methods to scale up to very large networks. We also plan to explore finer power-control (e.g. antenna-level beamforming for energy savings)

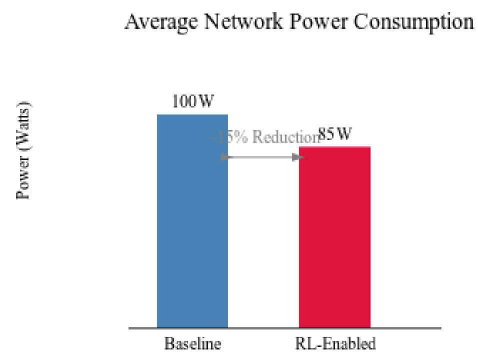

Fig. 1. Average network power consumption under baseline and RL policy. The RL policy reduces consumption by $\sim$15%.

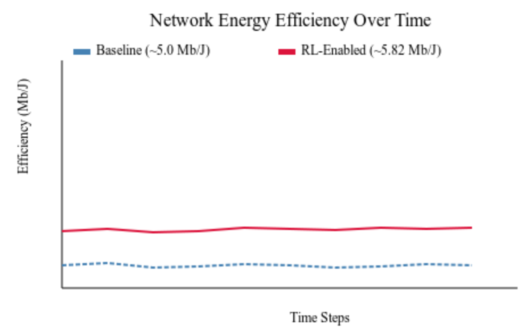

Fig. 2. Time-series of network energy efficiency (bits per Joule) for baseline and RL. The RL curve is consistently higher (about 15%).

and more detailed battery/storage models. Finally, performance under ultra-dense or heterogeneous networks (small cells, IoT devices) remains an open question. Overall, the framework provides a foundation for further research into sustainable wireless design.