Reinforcement Learning Training Strategies for 5G Networks Latency Optimization

Andrea Pazienza
NTT DATA Italia
Via G. Amendola, 146 - 70126 Bari, Italy
andrea.pazienza@nttdata.com

Massimiliano Rossi
NTT DATA Italia
Via E. Calindri, 4 - 20143 Milan, Italy
massimiliano.rossi@nttdata.com

***Abstract*—This study investigates the application of reinforcement learning (RL) algorithms to minimize latency in 5G networks, focusing on edge selection in urban areas. Emphasizing the crucial role of latency reduction in enhancing network efficiency, the research employs real-time data processing and optimization strategies. Several RL algorithms are evaluated to determine the most effective approach for latency minimization, focusing on optimization strategies related to avoiding model overfitting. The insights contribute to advancing 5G network development, particularly in latency reduction strategies within Edge-Cloud Systems (ECS).**

***Index Terms*—Reinforcement Learning, Artificial Intelligence, 5G Networks, Network Optimization**

I. Introduction

In recent advancements, the Edge-Cloud continuum has emerged as a vital framework, enabling efficient allocation of computational resources between edge and cloud environments to improve performance and responsiveness. Supporting this technology stack, 5G networks offer the necessary infrastructure to facilitate real-time data exchange, critical for reducing latency and ensuring robust interactions between 5G base stations (gNodeBs) and data centers. The User Plane Function (UPF), a central component of 5G architecture, is key in low-latency edge computing. The N3 interface connects the gNodeB Radio Access Network (RAN) to the UPF, enabling user data transmission. For optimal operation, UPF positioning must be close to the network edge to meet latency and performance requirements, taking into account other network parameters that impact the end-user experience.

This paper introduces strategies and methodologies for optimizing network performance with a focus on RL, especially in avoiding model overfitting. Specifically, it explores three RL algorithms designed to reduce latency while dynamically adjusting to network conditions.

II. RL Optimization Strategies

Our approach centers on applying RL to reduce latency in 5G networks, particularly between gNodeB and target data centers. In this context, RL agents learn to make decisions in a simulated environment to maximize a reward signal, which is determined by network performance metrics such as latency, packet loss, and resource usage. We choose DQN, PPO, and A2C architectures for testing and implementation:

- Deep Q-Network (DQN) [1]: Employs a replay buffer, target network, and gradient clipping, which help the system learn from past actions, enhancing decision-making in diverse network conditions.
- Proximal Policy Optimization (PPO) [2]: Known for stability and efficiency, PPO iteratively refines policies to ensure adaptability in real-time, especially useful for high-latency and dynamic network conditions.
- Advantage Actor-Critic (A2C) [3]: Balances exploration and exploitation, allowing the network to make effective decisions for optimal performance.

The RL environment was implemented using the Gymnasium [4] Python library (a fork of OpenAI Gym), enabling realistic 5G network simulations. In this framework, actions are defined as selecting an edge node (in Milan, Rome, or Cosenza), observations as real-time data from these nodes, and rewards as functions of latency, CPU usage, and other key metrics. To facilitate RL optimization, a policy for target data center selection was implemented. This policy assigns weights to features within the reward calculation, guiding the RL agent to prioritize latency reduction while balancing other resource and network parameters. By ending episodes when the agent's selected node aligns with the target, the model effectively meets its latency reduction goals.

To ensure robust performance across different network conditions, we developed specific training strategies to prevent overfitting. Specifically, we employed empirical measurements from telco field operator experience to introduce best-practice knowledge within the optimization algorithm. Firstly, UPF selection constraints were introduced, limiting CPU usage to below 90% to avoid resource overload and maintaining balanced resource distribution across data centers. Additionally, the selected UPF must show at least a 20% reduction in packet loss over the previous UPF to ensure high-quality connections. We also implemented a refined reward system that issues positive rewards when latency and packet loss fall below dataset mean values, with penalties for higher values, encouraging the model to select high-performance UPFs. An additional accuracy bonus is applied when the selected UPF matches the optimal target, reinforcing the selection of the highest-performing data centers. The Italian data centers in Milan, Rome, and Cosenza represent various network conditions and

TABLE I
FEATURES WEIGHTS.

Feature	Weight	Unit of Measurement
cpu_usage_percent	0.6	Percent
memory_usage_percent	0.5	Percent
disk_usage_percent	0.5	Percent
net_in_percent	0.7	Percent
net_out_percent	0.7	Percent
latency_avg	1	ms
latency_mdev	0.2	ms
lost_percent	0.9	Percent

form the basis for testing our RL optimization strategies. Milan hosts the centralized core network with control plane functions, while each city has a UPF to manage the user plane. Each data center has a bandwidth cap to simulate real-world limitations. To build resilience against overfitting, data traffic was generated using a script that models different times of the day (Night, Busy Hour, Daytime) and traffic profiles (e.g., CPU load, bandwidth consumption). Each simulation cycle varies values such as throughput and session duration to reflect realistic usage patterns.

The goal of the RL model is to identify and select the optimal data center to minimize latency between the gNodeB and edge node. This RL environment emphasizes latency as the primary objective but incorporates additional Key Performance Indicators (KPIs) to ensure comprehensive performance optimization. For example, while low-latency data centers are preferred, high CPU utilization may affect overall network efficiency, so balancing multiple metrics is essential.

To facilitate optimal data center selection, we embedded a policy with weighted features within the RL environment, prioritizing latency and packet loss. This weighted approach allows the RL agent to consider multiple aspects of data center performance, ensuring decisions align with broader network goals and with empirical measurements. Feature weights, shown in Table I, were determined based on network dynamics, emphasizing latency and packet loss over other metrics like CPU and memory utilization.

In the Gymnasium environment, actions are defined as selecting an edge node (Milan, Rome, or Cosenza), observations as a triplet of data representing the status of each feature in real-time across the data centers, and rewards calculated as the weighted sum of features $\sum_{i=1}^{n} \frac{W(i)}{1+D(i)}$, where n is the number of features, $W(i)$ is the weight associated to the feature i and $D(i)$ is the best value respecting KPI for each feature. This reward system helps the model meet latency reduction goals while balancing other critical metrics. The episode ends when the selected edge node matches the optimal target, signaling that the model has met its performance objectives.

Using the Stable-Baselines3 library [5], we performed extensive hyperparameter tuning to achieve optimal performance. We explored various combinations of learning rate, batch size, and discount factor to identify the best configurations. Ultimately, a learning rate of 0.001, a discount factor of 0.45, and a batch size of 256 were chosen for DQN, showing optimal results for the task at hand.

III. EVALUATION

The three RL algorithms were evaluated on their ability to optimize latency and network performance. DQN outperformed both PPO and A2C, achieving a maximum reward of 338, indicating effective latency reduction and reward stability. PPO and A2C faced challenges with convergence and stability, suggesting they are less suitable for latency optimization in this context. KPIs included maximum reward achieved, reflecting the model's efficiency in meeting latency goals, reward convergence for consistent performance, and loss function analysis to identify configurations with minimal loss and thus higher efficiency.

To validate the RL agent's real-time predictions, we implemented a comprehensive evaluation strategy. The agent is evaluated continuously using real-time telemetry data from each data center, ensuring predictions closely align with current network conditions. Network experts labeled this data to establish the optimal data center selection according to set parameters, providing accurate comparison metrics. Performance metrics showed 77% accuracy, 77% precision, and 76% recall in UPF selection, with an improvement in latency of 30% on average.

IV. CONCLUSION

This study highlights the effectiveness of RL in enhancing 5G network latency optimization, with DQN emerging as the most successful model. DQN's configuration was fine-tuned to yield high stability and convergence, making it suitable for real-time 5G applications in Edge-Cloud Systems. By prioritizing latency and incorporating overfitting mitigation strategies, the model ensures robust performance across variable network conditions. The importance of selecting the right RL algorithm and hyperparameters was underscored, ultimately showing that RL techniques, particularly DQN, are highly effective for achieving low-latency and high-efficiency industrial networks.

Future research directions will explore different architectures, evaluation approaches, or any other dimensions for the observation or action space, such as sustainability parameters.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Horizon Europe Programme under the MLSysOps Project, grant agreement 101092912.

REFERENCES

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv:1707.06347*, 2017.

[3] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.

[4] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. d. Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. T. J. Shen, and O. G. Younis, "Gymnasium," Mar. 2023.

[5] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *JMLR*, vol. 22, no. 268, pp. 1–8, 2021.