Asynchronous Multi-Agent Reinforcement Learning for 5G Routing under Side Constraints

Extended Abstract

Sebastian Racedo
Concordia University
Montreal (Qc), Canada
sebastian.racedo@mail.concordia.ca

Brigitte Jaumard
Concordia University
Montreal (Qc), Canada
brigitte.jaumard@concordia.ca

Oscar Delgado
École de Technologie Supérieure (ETS)
Montreal (Qc), Canada
oscar.delgado@etsmtl.ca

Meysam Masoudi
Ericsson
Kista, Sweden
meysam.masoudi@ericsson.com

ABSTRACT

Modern 5G/O-RAN networks must route heterogeneous services with distinct QoS constraints (e.g., bandwidth, compute resources, and strict end-to-end latency) over shared resources. Centralized reinforcement learning (RL) controllers and synchronized multi-agent training can encounter serialization bottlenecks and delays due to stragglers. We propose Asynchronous Multi-Agent Reinforcement Learning (AMARL): one Proximal Policy Optimization (PPO) agent per service plans routes in parallel on local environment snapshots and commits resource deltas to a shared global state via a lock-guarded commit/abort mechanism that preserves feasibility. On an O-RAN-like simulator driven by 24-hour Montreal traffic traces, AMARL matches a strong single-agent PPO baseline in Grade of Service (GoS) and latency, while reducing training wall-clock by 29.9% and evaluation time by 15.0%.

KEYWORDS

Multi-Agent Reinforcement Learning; Asynchronous Learning; PPO; Routing; O-RAN; Network Optimization.

ACM Reference Format:
Sebastian Racedo, Brigitte Jaumard, Oscar Delgado, and Meysam Masoudi. 2026. Asynchronous Multi-Agent Reinforcement Learning for 5G Routing under Side Constraints: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/NMHJ4078

1 INTRODUCTION

Programmable 5G and beyond networks concurrently serve services with sharply different QoS envelopes (eMBB, uRLLC, mMTC), turning routing into a time-critical control problem under nonstationary demand and resource contention. Classical non-learning baselines (e.g., fixed shortest-path routing, heuristic re-optimization, or MILP-style formulations) face a trade-off between solution quality and recomputation cost, making near-real-time adaptation difficult at scale. RL is attractive because inference is fast once trained, enabling online adaptation to congestion and service mix shifts. A common starting point is a single centralized RL policy that routes all flows [1, 4, 9]. However, the state and action spaces and mixed QoS objectives grow quickly with network size and service heterogeneity, and decision-making becomes serialized [2, 6]. Synchronized MARL alleviates some scaling limits but still incurs barriers and straggler effects [5, 10]. Motivated by asynchronous RL successes in other domains, we ask: Can we preserve routing feasibility and QoS while training and executing per-service agents fully asynchronously?

Contribution. We present AMARL, an asynchronous, service-specialized MARL design for routing under bandwidth,compute, and latency constraints in an O-RAN-like setting. A full version is available on [7].

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/NMHJ4078

2 METHODOLOGY

System model and constraints: We model the network as a directed graph $G = (V, E)$ with link capacity C_e and propagation delay t_e for each $e \in E$, and compute nodes $V_c \subseteq V$ with usable compute budget κ_v (e.g., capped at 80% CPU). Each flow request p has bandwidth demand b_p, latency limit τ_p, and a service-specific SFC of length $n_{s(p)}$ whose placement $\bar{x}_{v,f}$ is fixed.

Flows are unsplittable: for each SFC segment k (i.e., the end-to-end path between two consecutive SFC waypoints, from stage k to stage k+1, including source and destination as boundary waypoints), the traffic of p is routed on a single path.

Let $y_{e,p,k} \in \{0, 1\}$ indicate whether link e is used by flow p on SFC segment k, and let $a_{v,k,p} \in \{0, 1\}$ indicate whether SFC stage k of request p executes at node v. The routing must satisfy:

$$\sum_{p \in P} \sum_{k} b_p \, y_{e,p,k} \leq C_e \qquad \forall e \in E, \tag{1}$$

$$\sum_{p \in P} \sum_{k} b_p \, c_{v,f_{s(p),k}} \, a_{v,k,p} \leq \kappa_v \forall v \in V_c, \tag{2}$$

where $c_{v,f}$ is the per-Mbps compute demand of function f at node v. These are the hard feasibility checks enforced at commit time. End-to-end latency feasibility is checked during the local episode by accumulating propagation delay and processing delay (a function

of the local snapshot utilization and the traversed functions), and requiring the realized delay to respect τ_p.

Asynchronous multi-agent design (AMARL): We run one Proximal Policy Optimization (PPO) [8] agent A_s per service s (see Figure 1). Agents never synchronize; coordination occurs through shared resource contention in a global environment $E^\star$. Each agent repeatedly: (i) fetches the next request of its service, (ii) instantiates a local environment E_e^s from a snapshot of state($E^\star$), (iii) routes the request step-by-step, and (iv) emits a resource delta $\Delta(p)$ that is applied to $E^\star$ via a lock-guarded `commit()`. If applying $\Delta(p)$ would violate the capacity constraints, the commit aborts and the episode is rejected.

Actions and masking: At each step, the agent selects the next hop among neighbors; the environment provides an `action_mask` that removes actions that would immediately violate hard constraints, improving sample efficiency.

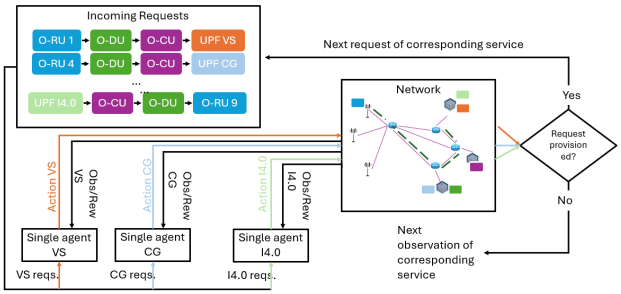

Figure 1: Asynchronous Multi-Agent AMARLDiagram

3 RESULTS

We evaluate AMARL on an O-RAN-like simulator with a fixed feasible placement and nearly real-time 24-hour traffic traces from Montreal. We compare against a centralized single PPO agent using Maskable PPO [3] (an extension of PPO [8] that handles dynamic action space that routes all services sequentially).

QoS parity with lower wall-clock. AMARL achieves comparable GoS to the single-agent baseline while substantially reducing training time and evaluation time (see Table 1 and Figure 2). These wall-clock gains follow from parallel rollouts and service-specialized learning that avoids a monolithic policy having to internalize heterogeneous service objectives simultaneously.

Failure modes under contention. Figure 3 summarizes dominant failure causes. Tight-latency/high-bandwidth services (e.g., AR) are frequently pruned by capacity and latency constraints, leading to `dead_end` termination when no feasible next hop remains. This aligns with the intended semantics of action masking under hard constraints: feasibility is guaranteed for accepted flows, while unrouteable flows fail fast without destabilizing other services.

4 CONCLUSION AND FUTURE WORK

AMARL enables fully asynchronous, per-service routing policies coordinated through shared state rather than synchronization barriers. On a realistic O-RAN-type routing network with 24/7 city-wide demand, AMARL preserves GoS/latency compared to a high-performance single-agent PPO reference database, while reducing

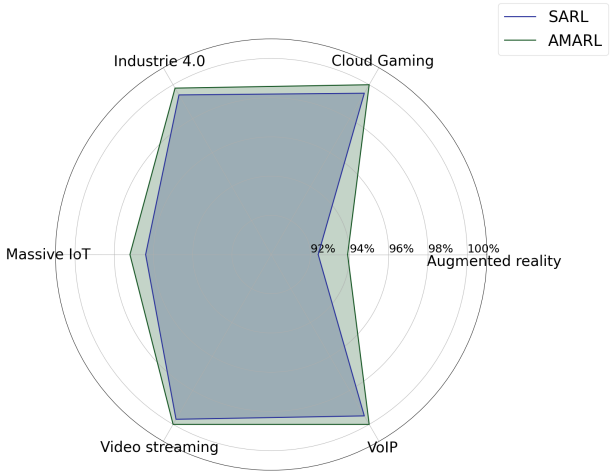

Figure 2: GoS comparison (SARL vs. AMARL).

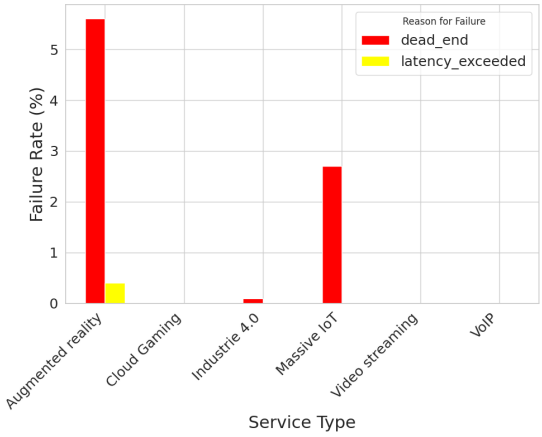

Figure 3: Failure analysis on test data.

Table 1: Time and GoS comparison (test set).

Method	Avg. GoS	Train	Test
SARL (1 agent)	97.81%	07:25:05	00:45:22
AMARL (6 agents)	98.48%	05:12:00	00:38:33

training and evaluation time. This points to asynchronous, modular service agents as a practical control architecture for near-real-time (and future 6G) network automation.

ACKNOWLEDGMENTS

This work was supported by NSERC (under project ALLRP 566589-21) and InnovÉÉ (INNOV-R program) through the partnership with Ericsson. We are grateful to Adel Larabi at GAIA, Ericsson Montréal for clarifying some concepts of the current 5G technology.

REFERENCES

[1] P. Almasan, J. Suárez-Varela, K. Rusek, P. Barlet-Ros, and A. Cabellos-Aparicio. 2022. Deep reinforcement learning meets graph neural networks: Exploring a routing optimization use case. Computer Communications 196 (2022), 184–194.

[2] Lucian Busoniu, Robert Babuska, and Bart De Schutter. 2008. A Comprehensive Survey of Multiagent Reinforcement Learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 38, 2 (2008), 156–172.

[3] Shengyi Huang and Santiago Ontañón. 2022. A Closer Look at Invalid Action Masking in Policy Gradient Algorithms. The International Florida AI Research Society Conference (FLAIRS) 35 (2022), 1 – 6.

[4] B. Jaumard, C. Boudreau, and E. Janulewicz. 2024. Dynamic Service Function Chaining Provisioning with Reinforcement Learning Graph Neural Networks. In 3rd GNNet Workshop on Graph Neural Networking Workshop. Association for Computing Machinery, Los Angeles, CA, USA, 53–58.

[5] V. Mnih, A.P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. In 33rd International Conference on Machine Learning. PMLR, New York, NY, USA, 1928–1937.

[6] Afshin Oroojlooy and Davood Hajinezhad. 2022. A review of cooperative multi-agent deep reinforcement learning. Applied Intelligence 53, 11 (2022), 13677–13722.

[7] Sebastian Racedo, Brigitte Jaumard, Oscar Delgado, and Meysam Masoudi. 2026. Asynchronous MultiAgent Reinforcement Learning for 5G Routing under Side Constraints. arXiv:2602.00035 [cs.NI] https://arxiv.org/abs/2602.00035

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. 2017. Proximal policy optimization algorithms. (2017). arXivpreprintarXiv:1707.06347

[9] Zhiyuan Xu, Jian Tang, Jingsong Meng, Weiyi Zhang, Yanzhi Wang, Chi Harold Liu, and Dejun Yang. 2018. Experience-driven Networking: A Deep Reinforcement Learning based Approach. In IEEE INFOCOM 2018 - IEEE Conference on Computer Communications. IEEE, Honolulu, HI, USA., 1871–1879.

[10] C. Yu et al. 2023. Asynchronous Multi-Agent Reinforcement Learning for Efficient Real-Time Multi-Robot Cooperative Exploration. In International Conference on Autonomous Agents and Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems, London, United Kingdom, 1107–1115.