Decentralized Intelligence for Centralized Control: Multi-Agent Reinforcement Learning for SD-WAN

Elshan Khanlari[1], Luca Borgianni[1], Davide Adami[2], Stefano Giordano[1]

[1]*University of Pisa, Dept. of Information Engineering, Via G. Caruso 16, 56122 Pisa, Italy*
[2]*CNIT - University of Pisa, Dept. of Information Engineering, Via G. Caruso 16, 56122 Pisa, Italy*

***Abstract*—Modern Software-Defined Wide Area Network (SD-WAN) deployments are required to manage traffic over heterogeneous underlay networks while meeting stringent Quality of Service (QoS) requirements. In scenarios where multiple branches share overlay resources, independent tunnel selection decisions often lead to congestion and degraded performance. Existing approaches lack coordination mechanisms to handle the dynamic interactions between agents competing for shared resources. This paper presents a Multi-Agent Reinforcement Learning (MARL) framework for distributed overlay selection in SD-WANs. Each branch is modeled as an autonomous agent that learns routing policies through interaction with the network environment. To account for the mutual impact of decisions across branches, we adopt the Centralized Training with Decentralized Execution (CTDE) paradigm, enabling agents to learn globally consistent behaviors while preserving scalability at inference. To encourage cooperative policies, we introduce a λ-weighted reward shaping mechanism that balances local QoS goals with global resource fairness. We evaluate our approach using both PPO and DQN algorithms in a simulated SD-WAN environment. The findings highlight the necessity of MARL in addressing resource contention and ensuring equitable shared overlay utilization.**

***Index Terms*—SD-WAN, Multi-agent reinforcement learning, network reliability**

I. Introduction

Software-Defined Wide Area Networking (SD-WAN) has emerged as a foundational technology for modern enterprise connectivity, providing centralized orchestration, intelligent traffic management, and enhanced cost-efficiency across geographically distributed networks [1].

Each site or branch in an SD-WAN topology may have unique performance requirements and access to a distinct set of underlay transport options. Variables such as bandwidth availability, latency sensitivity, link degradation, and cost further complicate decision-making. Moreover, when multiple branches simultaneously select the same overlay, contention can arise, resulting in congestion and degraded performance.

Traditional SD-WAN solutions often employ static policies or reactive rule-based heuristics (e.g., latency thresholds or SLA violations) to guide overlay selection. While straightforward, these approaches typically lack adaptability to evolving network conditions and do not account for the collective effect of distributed routing decisions made across multiple branches. This limitation underscores the need for machine learning techniques, particularly Reinforcement Learning (RL), which can autonomously learn and adapt policies through continuous interaction with the network environment.

However, RL-based approaches often suffer from scalability issues. In [2], they tackled this challenge by employing decentralized Multi-Agent Reinforcement Learning, which distributes decision-making across multiple agents to reduce system complexity. Nevertheless, the fully distributed nature of this approach results in agents acting independently, without accounting for their mutual impact on the network.

We designed and implemented two simulation environments, Centralized-MARL and Independent Learners, for SD-WAN that accurately reflect the challenges of heterogeneous branch connectivity over multiple shared overlays. These environments serve as testbeds in which branch agents contend for overlay resources whose capacity, latency, and cost characteristics vary over time. To facilitate effective coordination among agents without sacrificing deployment flexibility, this work will employ the Centralized Training with Decentralized Execution (CTDE) paradigm.

Another key contribution of our work is the development of a lambda-weighted reward function that encapsulates heterogeneous branch requirements while encouraging cooperative overlay usage. Each agent's reward will be scaled by a parameter λ reflecting its performance sensitivity, while the complement $(1 - \lambda)$ of the other agent's parameter will modulate the reward signal, thereby implicitly discouraging simultaneous selection of the same overlay.

The proposed approach is evaluated using a combination of queuing-theoretic analysis and empirical performance metrics. Queuing theory is applied to characterize traffic intensity under varying load conditions, while additional network metrics are assessed through simulation-based experimentation. To evaluate the generality and practical effectiveness of the MARL-based solution, experiments are conducted across diverse scenarios involving heterogeneous link capacities, latency requirements, and cost constraints.

II. Related work

The problem of tunnel selection in SD-WAN has been presented and evaluated in many works. In particular, the use of RL with a single agent has been well analyzed in [3]–[5].

However, single-agent formulations implicitly assume that a centralized entity has full observability and control over the decision process. In realistic SD-WAN scenarios, multiple edge nodes make concurrent and potentially conflicting deci-

978-3-903176-75-1 ©2025 IFIP

sions under partial information. This naturally motivates the adoption of MARL, where agents learn policies not only from their local environment but also by adapting to the behavior of other agents.

A possible approach is centralized MARL, a single controller or critic has access to all agents' observations and actions and computes a joint policy over the combined state–action space as introduced in [6]. However, this approach suffers from scalability issues, as the joint action space grows exponentially with the number of agents. In contrast, in decentralized MARL each agent learns and acts with only its own local information. No central controller aggregates information; each agent has its own policy based on local observations and receives rewards. Fully decentralized agents offer advantages in scalability and robustness, but without coordination, purely independent learning often fails to achieve globally optimal cooperation [7]. A third hybrid approach in [8], [9], MARL with Networked Agents (MARL-NA), has gained attention. A relevant advancement in this context is reward sharing, where agents distribute a portion of their rewards to neighbors. As shown in [10], this mechanism promotes local cooperation by encouraging agents to consider the impact of their actions on peers.

A recent study in [11] applies MARL-NA in dynamic overlay selection; it proposes a fully decentralized MARL solution in which each edge agent trains locally using its own observations and lightweight information exchanged with neighbors; cooperation emerges through a local reward-blending mechanism (a tunable λ) that mixes an agent's return with those of its neighbors [11].

In contrast, our work adopts a Centralized Training with Decentralized Execution (CTDE) paradigm: agents are trained with access to global state and joint returns, but execute independently at run time.

III. System Architecture and Methodology

To evaluate the effectiveness of coordinated learning in multi-agent SD-WAN environments, we implemented two architectural setups. The first is a centralized MARL approach, where joint policy optimization is performed with global observability and shared rewards. The second one is an independent learner setup (decentralized MARL), in which each agent operates using only local observations and individual reward. This contrast enables a quantitative comparison between coordinated and uncoordinated learning, highlighting the impact of inter-agent communication and reward sharing on performance. Agents are trained using both value-based and policy-based algorithms, Deep Q-Network (DQN) and Proximal Policy Optimization (PPO), across both architectural settings.

A. Reference SD-WAN Topology

To simulate realistic routing decisions in a distributed SD-WAN setting, our environment includes two branch agents (Branch A and Branch B) and three overlay links with distinct service characteristics. Each overlay is defined by its service rate, latency, packet loss behavior, and queuing capacity. Overlay 1 is shared between both branches, while Overlay 2 and Overlay 3 are exclusive to Branch A and Branch B, respectively.

Traffic arrivals at each branch are generated stochastically using a Poisson process, independently of the agent's actions. At each timestep, the agents observe the current network state and select an overlay through which to route their arriving traffic. This action determines how load is distributed across the overlays, directly affecting queuing behavior and congestion.

- Agent A chooses between Overlay 1 and Overlay 2.
- Agent B chooses between Overlay 1 and Overlay 3.

Because Overlay 1 is shared, agents must learn to coordinate their actions implicitly to avoid overloading the same path. The learning objective is to develop routing strategies that maximize individual performance while maintaining overall network efficiency. As shown in Figure 1, three overlay networks are established on top of heterogeneous underlay technologies, each offering distinct bandwidth and latency characteristics.

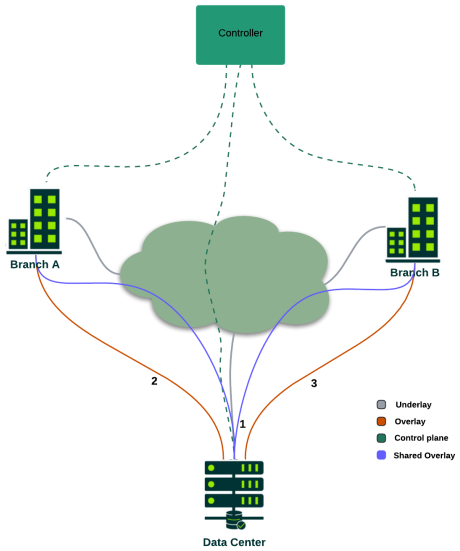

Fig. 1: SD-WAN topology with two branch agents (A and B) selecting among three overlays. Overlay 1 is shared between both branches, while Overlay 2 and Overlay 3 are exclusive to Branch A and Branch B, respectively.

B. Centralized Training with Decentralized Execution (CTDE)

To enable coordination among agents while preserving scalability and realistic deployment, our system adopts the CTDE paradigm. In this approach, agents are trained using global state information and shared rewards, but make decisions independently during runtime.

The CTDE strategy is implemented within the orchestration plane, which acts as a centralized decision-making layer (see Figure 2). A centralized RL controller operates here with full

visibility over the entire network. This controller is responsible for generating coordinated routing decisions based on the current state of the network.

Unlike decentralized approaches, where each agent only observes local information, this centralized agent collects and aggregates data from all nodes, links, and traffic flows. Specifically, it receives the state of every site, allowing the centralized agent to compute globally coordinated actions.

This setup contrasts with fully decentralized MARL approaches, where each agent must learn independently from its limited local perspective, making coordination more difficult and learning slower without communication.

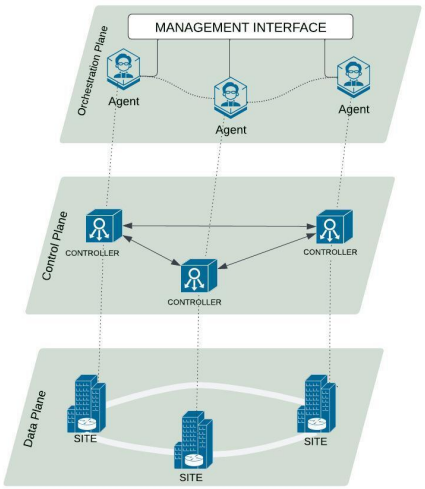

Fig. 2: Illustration of the three-layer SD-WAN reference architecture, emphasizing the interaction between network agents, controllers, and edge routers.

Once the centralized agent selects the optimal actions, these decisions are propagated downward through northbound APIs to the local SD-WAN controllers at each network site (see Figure 2).

C. MARL Environment System

Let us define the MARL Environment System with observation space, the action space, and the reward.

1) ***Observation Space****:* At each time step t, the environment emits a 12-dimensional **observation vector** $s_t \in \mathbb{R}^{12}$:

$$s_t = [bw_1,\ lat_1,\ Loss_1,\ bw_2,\ lat_2,\ Loss_2,\ bw_3,\ lat_3,\ Loss_3,\ |Q_1|,\ |Q_2|,\ |Q_3|]$$

where:

- bw_i: Available bandwidth on Overlay i
- lat_i: Latency on Overlay i
- $Loss_i$: Loss accumulated due to congestion
- $|Q_i|$: Queue length of ith overlay

2) ***Action Space****:* We define a joint discrete action space with a tuple of overlay choices for both agents:

$$\begin{aligned} a_t = 0 &\Rightarrow (\mathrm{A} \to O_1,\ \mathrm{B} \to O_1) \\ a_t = 1 &\Rightarrow (\mathrm{A} \to O_1,\ \mathrm{B} \to O_3) \\ a_t = 2 &\Rightarrow (\mathrm{A} \to O_2,\ \mathrm{B} \to O_1) \\ a_t = 3 &\Rightarrow (\mathrm{A} \to O_2,\ \mathrm{B} \to O_3) \end{aligned}$$

3) ***Per-branch Step Reward****:* Let the selected overlay for Branch A at timestep t have bandwidth B_A, packet loss L_A, and queuing delay D_A with initial capacity $B_{0,A}$, the individual reward for Branch A (and similar for Branch B) is:

$$\tilde{r}_A(t) = r_{\text{single}}(B_A, L_A, D_A),$$

(Where $r_{\text{single}}(\cdot)$ represents a performance-based utility function penalizing congestion, delay, and loss.)

4) ***Centralized MARL Reward****:* To promote cooperation, a mixed reward is computed using a scalar parameter $\lambda \in [0, 1]$:

$$R_{\text{joint}}(t) = \lambda \tilde{r}_A(t) + (1 - \lambda)\tilde{r}_B(t)$$

This joint reward is used during training to guide agents toward collaborative behavior. In our experiments, we use $\lambda = 0.8$, giving higher weight to Branch A, simulating stricter QoS requirements.

5) ***Independent Learners Reward****:* For comparison, in the decentralized setting, each agent learns independently with its own reward $\tilde{r}_A(t)$ or $\tilde{r}_B(t)$ without weighting:

$$R^A_{\text{IL}}(t) = \tilde{r}_A(t), \quad R^B_{\text{IL}}(t) = \tilde{r}_B(t)$$

6) ***Episodic Returns****:* Over an episode of T steps, the *cumulative* rewards are defined as:

$$G_A = \sum_{t=1}^{T} \tilde{r}_A(t), \quad G_B = \sum_{t=1}^{T} \tilde{r}_B(t), \quad G_{\text{joint}} = \sum_{t=1}^{T} R_{\text{joint}}(t)$$

The centralized policy is trained to maximize G_{joint}, while for independent learners, each G_A and G_B maximizes their own reward return.

IV. Evaluation and Results

To evaluate the behavior of the proposed SD-WAN environment and reward logic under various traffic conditions, we designed a series of simulation test cases. Each case aims to represent a specific network load scenario by varying the traffic intensity (ρ) across overlays. The test case scenarios are Underload ($\rho \ll 1$), High load ($\rho \approx 1$) and Overloaded ($\rho > 1$).

We evaluate our approach using the custom SD-WAN simulation described in Section III. Two branch agents (A and B) route traffic over three overlays: Overlay 1 (shared), Overlay 2 (exclusive to A), and Overlay 3 (exclusive to B).

Traffic arrivals at each branch follow independent Poisson processes with mean rates chosen to generate three operating regimes:

$$\rho = \frac{\lambda \cdot \bar{s}}{\mu},$$

where λ is the request rate, $\bar{s}$ the mean flow size, and μ the overlay service rate. We fix $\bar{s}$ for A and B, service rates μ_{O1}, μ_{O2}, μ_{O3}, and λ-weighting parameter $\lambda = 0.8$. Agents are trained for 500 episodes with both Proximal Policy Optimization (PPO) and Deep Q-Network (DQN) under centralized (CTDE) and independent learning MARL settings.

To reflect service differentiation, Branch A was assigned a higher lambda weight, prioritizing its traffic. We measure:

- **Convergence speed**: episodes count until episodic return plateaus.
- **Overload events**: fraction of time steps where any overlay's $\rho > 1$. The $\sum \rho > 1$ column in Tables I to III sums the overlays where $\rho > 1$, which allows for quickly identifying which action combinations lead to congestion and its severity. By analyzing this value, we can evaluate which decisions are more congestion-prone and identify the best-effort action, helping to assess the quality of our environment and the agent's expected behavior.

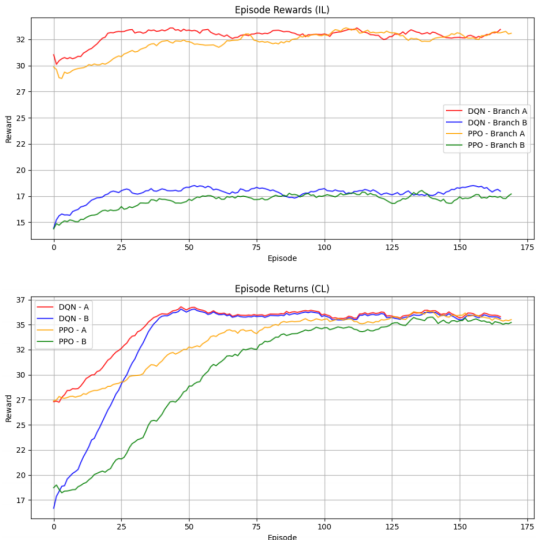

Fig. 3: Learning curves for independent learners (IL) and centralized learning (CL) under low-load conditions.

A. Test Case 1: Underload Regime ($\rho \ll 1$)

When arrival rates are low, all overlays operate below capacity. Table I reports the traffic intensity calculated through queuing analysis, confirming that CTDE does not degrade performance when resources are ample (see Figure 3). In this underloaded regime, the system remains stable and uncongested across all overlays. This setting provides an ideal baseline for assessing whether agents can identify and exploit the most advantageous routing paths without the influence of traffic bottlenecks. Both independent learners and centralized MARL agents consistently favored Overlay 1, the shared link, due to its superior service rate. This is desirable behavior, as routing through Overlay 1 minimizes latency and queue buildup. The learning curves in Figure 3 show a clear and steady improvement in performance across all agents, indicating successful policy convergence. Since all routes are available, the main differentiator becomes how efficiently each agent learns to optimize throughput and latency. However, the reward gap between Branch A and Branch B agents is noticeably larger for independent learners, whereas in centralized MARL this gap is significantly reduced. This suggests that independent learners exhibit stronger competition, while centralized MARL promotes cooperation. Consequently, even in an underloaded scenario—where cooperation is less critical—centralized MARL achieves higher total reward returns.

TABLE I: Overlay utilization (ρ) per action pair over underload regime.

Action	A→O	B→O	ρ_{O1}	ρ_{O2}	ρ_{O3}	$\sum \rho > 1$
0	1	1	0.144	–	–	0.144
1	1	3	0.096	–	0.04	0.04
2	2	1	0.04	0.004	–	0.04
3	2	3	–	0.04	0.04	0.04

B. Test Case 2: Near Saturation ($\rho \approx 1$)

In this test, our goal is to analyze the system's behavior near capacity limits. As shown in Table II, the expected behavior is that all overlays experience overload, significant losses are expected, therefore agents must learn to avoid congestion dynamically. The learning curves in Figure 4 demonstrate that centralized MARL (CL) consistently converges to the optimal cooperative routing policy corresponding to action (A $\rightarrow$ Overlay 2, B $\rightarrow$ Overlay 3), which leads to the lowest overall overlay traffic intensity (see Table II).

TABLE II: Overlay utilization (ρ) per action pair in near-saturation load.

Action	A→O	B→O	ρ_{O1}	ρ_{O2}	ρ_{O3}	$\sum \rho > 1$
0	1	1	2.10	–	–	2.10
1	1	3	0.97	–	0.96	0.97
2	2	1	1.20	0.92	–	1.20
3	2	3	–	0.92	0.96	0.96 (OK)

Notably, the numerical summary in Table IV shows that the CL policy converged significantly faster than Independent Learners (IL), the action frequency histograms in Figure 5 and Figure 6 further illustrate that the number of actions (count) taken for CL to converge to a final policy is considerably less than that of IL. This indicates that IL struggled to converge to a final policy, it failed to adapt to high-load conditions and to choose the best-effort action, resulting in a suboptimal performance and inefficient use of available resources.

Therefore, independent learners did not succeed in the near-saturation regime, particularly due to their inability to observe and adapt to the combined congestion impact of their

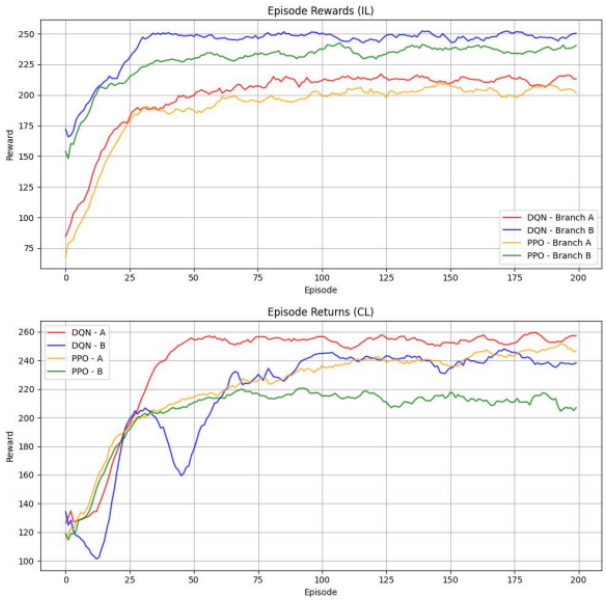

Fig. 4: Learning curves for independent learners (IL) and centralized learning (CL) under the near-saturation regime. DQN and PPO results for branch agents show that centralized MARL successfully prioritized Branch A's rewards over Branch B due to the higher assigned λ for Branch A, demonstrating improved fairness and SLA-aware decision-making.

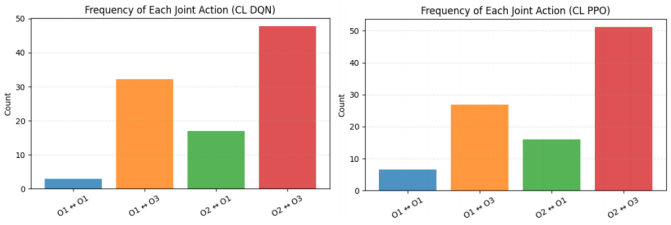

Fig. 5: Test 2's Action frequency distribution for centralized MARL agents. Coordinated policy favors $O_2 \rightarrow O_3$, to offload traffic from the shared overlay and balance network utilization.

decisions. Notably, both DQN and PPO agents frequently chose actions that sent Branch B traffic through Overlay 1, even though this link was already near or over capacity. This behavior led to poor routing efficiency, particularly under the stricter SLA needs of Branch A, since the shared overlay (O_1) was saturated by Branch B's misaligned decisions; Branch A's quality-of-service targets were violated, with increased delays and potential packet drops. This highlights the failure of decentralized agents to internalize global network conditions. In contrast, centralized MARL agents successfully coordinated to select the best-effort routing action corresponding to (O_2, O_3), which avoided shared congestion and more fairly distributed load across the available exclusive overlays. Their joint policy minimized congestion ($\rho < 1$) on each path (see Table II) while respecting per-branch needs. Figure 4 shows that centralized MARL successfully prioritized Branch

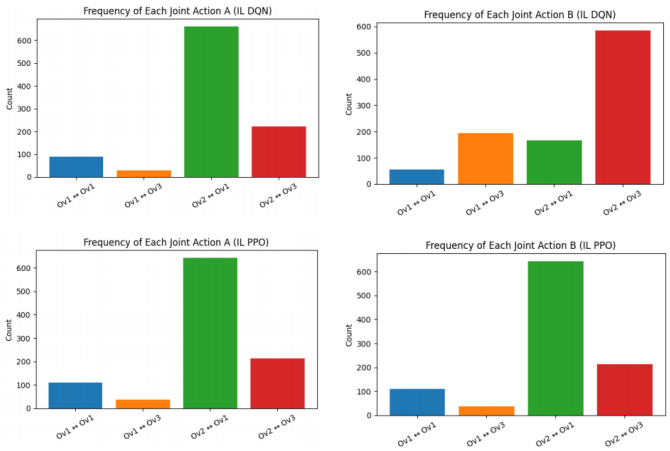

Fig. 6: Test 2's Action frequency distribution for Independent Learners. DQN and PPO policies often route traffic through Overlay 1 despite the risk of saturation, reflecting limited awareness of its policy.

A's rewards over Branch B due to the higher assigned λ for Branch A, demonstrating improved fairness and SLA-aware decision-making. This scenario showcased the growing necessity of cooperative behavior when the system is operating near its limits. Without shared context, independent agents degrade system-wide performance, and more critically, jeopardize SLA compliance. Centralized MARL, on the other hand, demonstrates its value by balancing load intelligently and respecting prioritization among agents. This case underscores that coordination is essential at the edge of capacity.

C. Test Case 3: Overload Regime ($\rho > 1$)

The overload scenario represents the most extreme condition in the test suite. In this setting, all overlays are subjected to traffic levels that exceed their service capacity, meaning that congestion is inevitable under all routing actions, where congestion rate is defined as the number of steps during which an overlay experienced zero bandwidth throughout an entire episode. Learning agents must now focus not on avoiding congestion, but on minimizing its occurrence and distributing it intelligently to maintain service quality, especially for SLA-critical branches such as Branch A. The per-episode congestion rate plot (see Figure 7) for this scenario captures how well each group of agents learns to mitigate unavoidable congestion over time. For independent learners, the congestion rate remains high and nearly constant across episodes (see Table IV). This indicates that decentralized decision-making leads to persistent, inefficient traffic assignments, particularly overloading Overlay 1. Often, both branches default to this shared link in pursuit of its higher service rate, without considering that its simultaneous use exacerbates congestion. As a result, neither agent adapts to the system's realities, and overall network performance suffers. Conversely, centralized MARL agents demonstrate a more intelligent response to overload. Their congestion rate begins high (see Figure 7), corresponding to actions involving O_2 for Branch A, as seen

in the previous test, unlike independent learners, CL chooses the best-effort action (see Table III) and as its result O_1's curve gradually decreases with training, hence over time agents converge to routing decisions that minimize congestion on the shared overlay(O_1). In particular, they tend to route Branch A (with stricter SLA requirements) through Overlay 1, and direct Branch B to its exclusive Overlay 3, which can handle the load just below its limit. This strategy does not eliminate congestion entirely but localizes it to less impactful links and protects high-priority traffic. Moreover, the influence of the λ-parameter in reward shaping ensures that Branch A's performance is prioritized without starving Branch B of resources.

Fig. 7: Test 3's per-episode congestion rate for centralized MARL in the overloaded regime.

TABLE III: Overlay utilization (ρ) per action pair under overload regime.

Action	A→O	B→O	ρ_{O1}	ρ_{O2}	ρ_{O3}	$\sum \rho > 1$
0	1	1	1.8	–	–	1.80
1	1	3	1.44	–	0.96	1.44 (best-effort)
2	2	1	0.48	7.20	–	7.20
3	2	3	–	7.20	0.96	7.20

V. Final Remarks

This work has demonstrated that MARL offers a clear advantage over the Independent Learners baseline for adaptive overlay selection in SD-WAN environments. Across varying network conditions, centralized coordination consistently led to more optimal path selection, mitigating the inefficiencies observed in IL, where agents exhibited competitive behavior over shared overlays, leading to congestion and suboptimal routing in the absence of cooperation.

The findings position our approach at the intersection of MARL and next-generation WAN management, offering a foundation for future work in scalable, adaptive, and fairness-driven overlay selection in networks with heterogeneous traffic demands.

Acknowledgment

This work was partially supported by the European Union - Next Generation EU under the Italian National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.3, CUPE83C22004640001 partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART") Project WATCHEDGE and by the Italian Ministry of Research (MUR) in the framework of the CrossLab and Forelab Projects (Departments of Excellence).

TABLE IV: Convergence speed and overload events across traffic regimes and algorithms.

Regime	Metric	C-PPO	I-PPO	C-DQN	I-DQN
Underload	Episodes to converge	120	130	140	150
	Overload events (%)	0%	0%	0%	0%
Near-Saturation	Episodes to converge	210	320	250	370
	Overload events (%)	5%	40%	10%	50%
Overload	Episodes to converge	300	450	350	500
	Overload events (%)	15%	70%	25%	80%

References

[1] S. Troia, L. Borgianni, G. Sguotti, S. Giordano, and G. Maier, "A comprehensive survey on software-defined wide area network," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2025.

[2] L. Busoniu, R. Babuška, and B. De Schutter, "A comprehensive survey of multi-agent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, no. 2, pp. 156–172, 2008.

[3] S. Troia, F. Sapienza, L. Varé, and G. Maier, "On deep reinforcement learning for traffic engineering in sd-wan," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2198–2212, 2020.

[4] M. A. Ouamri, G. Barb, D. Singh, and F. Alexa, "Load balancing optimization in software-defined wide area networking (sd-wan) using deep reinforcement learning," in *2022 International Symposium on Electronics and Telecommunications (ISETC)*. IEEE, 2022, pp. 1–6.

[5] L. Borgianni, S. Troia, D. Adami, G. Maier, and S. Giordano, "Assessing the efficacy of reinforcement learning in enhancing quality of service in sd-wans," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023, pp. 1765–1770.

[6] W. Mao, L. Yang, K. Zhang, and T. Başar, "On improving model-free algorithms for decentralized multi-agent reinforcement learning," in *Proceedings of the International Conference on Machine Learning*, ser. ICML '22. PMLR, 2022, pp. 15 007–15 049.

[7] A. Botta, R. Canonico, A. Navarro, G. Stanco, and G. Ventre, "Scalable reinforcement learning for dynamic overlay selection in sd-wans," in *Proceedings of the IFIP Networking Conference (IFIP Networking)*. IEEE, 2023, pp. 1–9.

[8] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*. PMLR, 2018, pp. 5872–5881.

[9] S. Sarkar, "An algorithm for adversary aware decentralized networked marl," https://arxiv.org/abs/2305.05573, 2023.

[10] Y. Yi, G. Li, Y. Wang, and Z. Lu, "Learning to share in networked multi-agent reinforcement learning," in *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, 2022.

[11] A. Botta, R. Canonico, A. Navarro, G. Stanco, and G. Ventre, "Adaptive overlay selection at the sd-wan edges: A reinforcement learning approach with networked agents," *Computer Networks*, vol. 243, p. 110310, 2024.