Availability-Aware Network Slicing and Dynamic Function Placement in Virtualized RANs with Reinforcement Learning

A PROJECT REPORT

Submitted in partial fulfillment of the
requirements for the award of the degrees

of
BACHELOR OF TECHNOLOGY
in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:
Amber Deshbhratar (210001003)
Sushil Yadav (210001080)

Guided by:
Dr. Sidharth Sharma

INDIAN INSTITUTE OF TECHNOLOGY INDORE
November 2024

Table of Contents

CANDIDATE'S DECLARATION

We hereby declare that the project entitled **"Availability-Aware Network Slicing and Dynamic Function Placement in Virtualized RANs with Reinforcement Learning"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in **Computer Science and Engineering** completed under the supervision of **Dr. Sidharth Sharma,** IIT Indore is an authentic work.

Further, I/we declare that I/we have not submitted this work for the award of any other degree elsewhere.

29-11-2024 29-11-2024

Signature and name of the student(s) with date

CERTIFICATE by BTP Guide(s)

It is certified that the above statement made by the students is correct to the best of my/our knowledge.

Sidharth

29-11-24

Signature of BTP Guide(s) with dates and their designation

Dr. Sidharth Sharma, Assistant Professor, CSE, IIT Indore

Preface

This report presents the culmination of our efforts to develop a Reinforcement Learning (RL)-based framework for optimizing resource allocation in virtualized Radio Access Networks (vRAN) in 5G networks. The rapid evolution of 5G technology has opened up new possibilities for high-speed, low-latency communication while introducing complex challenges in managing diverse applications and their stringent requirements. With this project, we aimed to address the limitations of traditional optimization techniques by leveraging RL to enable dynamic and scalable resource management.

The project required a thorough understanding of cutting-edge concepts such as network slicing, functional splitting, lightpath provisioning, and reliability mechanisms in vRAN, combined with an exploration of machine learning techniques. Our work builds on prior research, particularly the SliAvailRAN framework, while extending it to incorporate dynamic, real-time decision-making using RL. The RL framework models the network state, dynamically selects configurations such as Virtual Network Configurations (VNCs), paths, and wavelengths, and evaluates them against constraints to ensure optimal resource utilization and reliability.

The following sections of the report are structured to provide a comprehensive understanding of the problem and the proposed solution. We begin by outlining the background and significance of vRAN functional splitting problem, emphasizing the challenges of incorporating availability, slice-awareness and lightpath provisioning into the mix. Next, we present a detailed review of related work, highlighting the gaps and opportunities in existing methodologies. Then we discuss the problem formulation, implementation details and conclude the report with the discussion of the results and future scope of our work.

This report is the result of the collaborative effort of our team under the guidance of our supervisor, Dr. SidharthSharma, who provided invaluable insights and direction throughout the project. It encapsulates our research, design, implementation, and evaluation of the RL-based framework, along with a critical analysis of its performance compared to traditional ILP approaches. We hope this work contributes to advancing the state-of-the-art in 5G network optimization and serves as a foundation for further exploration of adaptive techniques in future networks.

Amber Deshbhratar, Sushil Yadav
B.Tech. IV Year
Discipline of Computer Science and Engineering IIT Indore

Acknowledgements

We are deeply grateful to **Dr. Sidharth Sharma** for his invaluable guidance and support throughout the course of this project. His expertise, encouragement, and insightful feedback were instrumental in shaping the direction and execution of this work. Dr. Sharma’s unwavering commitment to fostering innovative thinking has been a source of inspiration and motivation at every stage of this research.

We would also like to express our gratitude to IIT Indore for providing the resources and environment necessary to carry out this project. The collaborative and intellectually stimulating atmosphere here has played a significant role in enriching my learning experience.

This project represents a collective effort, and we are deeply appreciative of everyone who contributed to its successful completion.

Amber Deshbhratar, Sushil Yadav
B.Tech. IV Year
Discipline of Computer Science and Engineering
IIT Indore

Abstract

The deployment of 5G networks requires virtualized Radio Access Networks (vRAN) to meet diverse application requirements, including Ultra-Reliable Low-Latency Communication (URLLC), Enhanced Mobile Broadband (eMBB), and Massive Machine Type Communication (mMTC). Traditional optimization techniques like Integer Linear Programming (ILP), while effective for static scenarios, are computationally expensive and unsuitable for dynamic deployments. In this work, we propose a Reinforcement Learning (RL) - based framework that models the network's dynamic state and enables an intelligent agent to optimize resource allocation by selecting Virtual Network Configurations (VNCs), primary and backup paths, and wavelengths. The agent maximizes a reward function defined as the difference between revenue and costs while adhering to critical constraints like wavelength continuity, latency, and resource capacity. Experimental results demonstrate that the RL framework achieves comparable performance to ILP in resource utilization and slice acceptance while significantly improving scalability and adaptability in dynamic network environments.

Introduction

1.1. Background

1.1.1 Virtualized RAN (vRAN)

The virtualization of Radio Access Networks (RAN) is a transformative approach in 5G networks, designed to enhance resource flexibility and efficiency. Traditional RANs consist of monolithic baseband units (BBUs), but vRAN disaggregates these into three components: Centralized Units (CUs), Distributed Units (DUs), and Radio Units (RUs). By leveraging Network Function Virtualization (NFV) and Software Defined Networking (SDN), vRAN enables the deployment of virtualized network functions (VNFs) across these components dynamically. This separation allows for centralized management, better resource pooling, and dynamic scaling, making it ideal for supporting the diverse requirements of 5G applications.

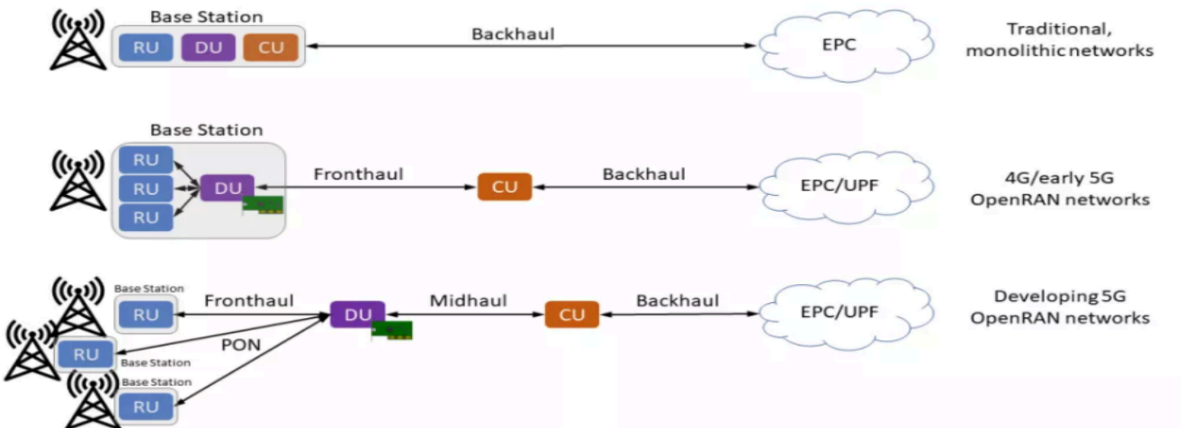

Fig.1. Evolution of Telecom Network Architecture and the concept of vRAN in 5G

1.1.2 Network Slicing

Network slicing is a key innovation in 5G that allows the physical network to be partitioned into multiple virtual networks or "slices." Each slice is tailored to a specific use case, such as Ultra-Reliable Low-Latency Communication (URLLC), Enhanced Mobile Broadband (eMBB), or Massive Machine Type Communication (mMTC). These slices share the same physical infrastructure but operate independently to meet distinct service-level agreements (SLAs). Network slicing enables efficient resource utilization while maintaining strict isolation between slices, ensuring reliability, performance, and security for diverse applications.

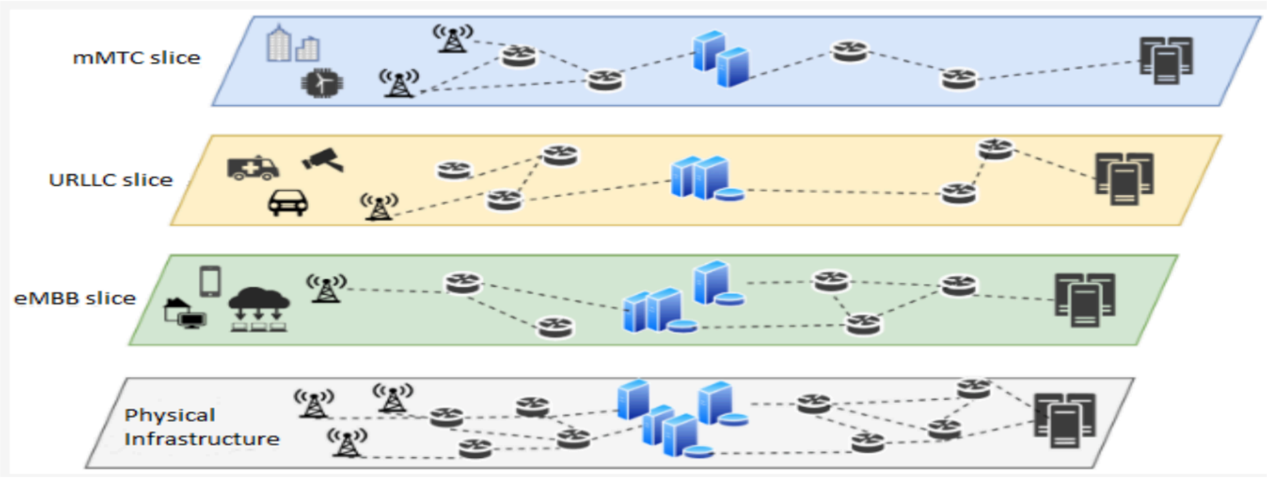

Fig 2. The Concept of Network Slicing

1.1.3 Functional Splitting

Functional splitting is a critical concept in vRAN architecture that allows the RAN protocol stack to be divided across different components (CU, DU, and RU) based on the level of centralization required. The 5G standards define multiple functional split options, ranging from highly centralized configurations, where most functions reside in the CU, to highly distributed setups, where functions are closer to the RU. Each split option balances trade-offs between latency, bandwidth, and computational demands. For example:

High Centralization: Functions like High-PHY and RLC are centralized in the CU, enabling resource pooling but requiring high-capacity backhaul links.

Distributed Setup: Functions are offloaded to DUs or RUs, reducing backhaul bandwidth requirements but increasing computational demands at edge nodes.

The choice of functional split impacts the network's performance, resource utilization, and ability to meet specific 5G use case requirements. Optimizing functional splitting is a key challenge in vRAN deployment, especially when combined with network slicing and reliability considerations.

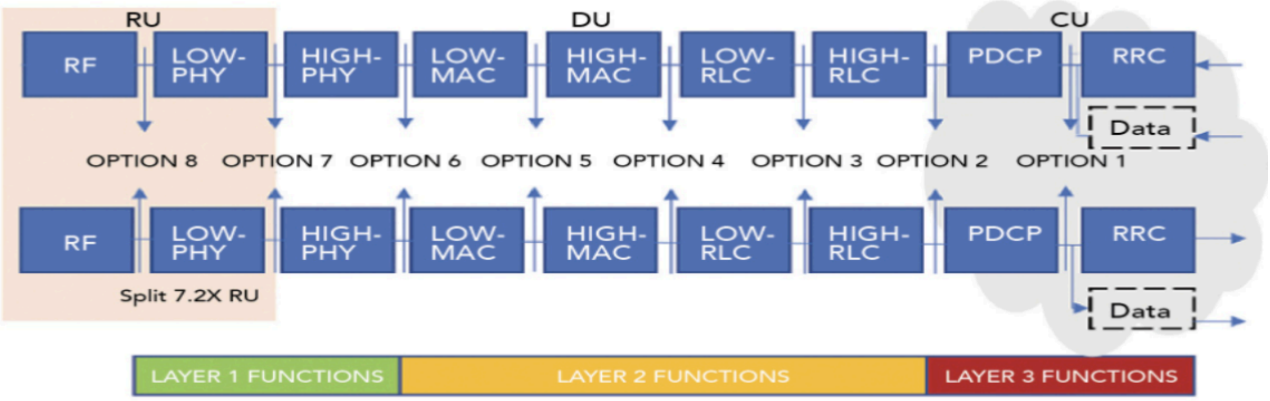

Fig 3. Eight Functional Split Options described by ITU, describing the placement of the functions from each protocol layer in the 5G New Radio (NR) protocol stack into CU, DU and RU

1.1.4. Wavelength Division Multiplexing in Optical Networks

Optical networks underpin the transport layer in 5G by providing high-capacity communication links. Wavelength Division Multiplexing (WDM) is a critical technology in these networks, enabling multiple data streams to coexist on a single optical fiber by assigning different wavelengths (or light frequencies) to each stream.

This capability necessitates careful lightpath provisioning to ensure **efficient routing and wavelength assignment (RWA)**. The RWA problem involves selecting a path and assigning a wavelength to each data stream, subject to constraints like wavelength continuity (using the same wavelength across a path) and avoiding wavelength conflicts on shared links. Optimizing RWA is essential for ensuring high bandwidth utilization and low latency.

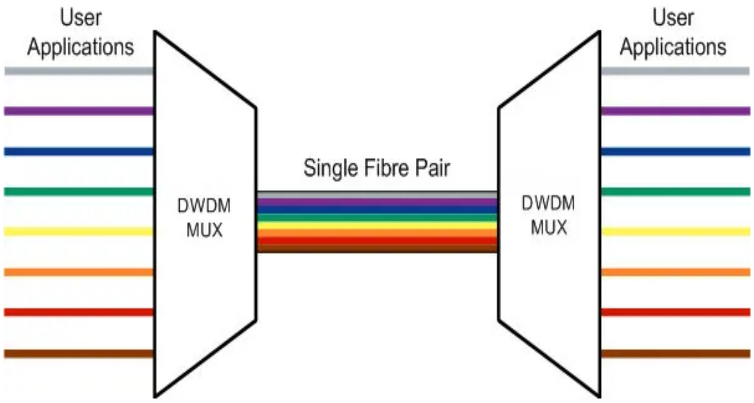

Fig 4.(a). Concept of WDM

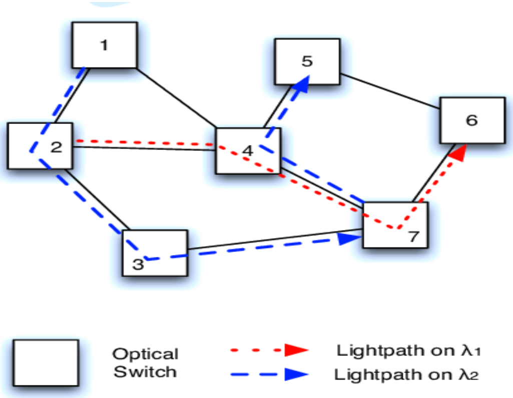

Fig 4(b). Lightpath Provisioning

1.1.5. Reliability in 5G Networks

Reliability is a cornerstone of 5G, especially for mission-critical applications like URLLC, which demand near-zero downtime. Reliability in vRAN is achieved by provisioning both primary and backup resources. A primary path handles the normal operation, while backup paths are reserved to maintain service continuity in case of failures. This redundancy is particularly critical in optical networks, where failures can disrupt multiple slices. Ensuring disjoint primary and backup paths, along with redundancy at the node level (CUs and DUs), enhances reliability and availability, meeting the stringent requirements of 5G services.

1.2. Motivation

The deployment of 5G networks introduces diverse and stringent requirements that challenge the traditional methods of network resource allocation and management. Virtualized Radio Access Networks (vRANs) enable dynamic slicing and functional placement, allowing service providers to cater to diverse use cases like URLLC, eMBB, and mMTC. However, the dynamic and resource-constrained nature of 5G networks requires a sophisticated approach to optimize resource allocation, functional placement, and wavelength provisioning.

Integer Linear Programming (ILP)-based frameworks, such as SliAvailRAN [7], provide optimal solutions but are inherently limited by their computational complexity, making them unsuitable for real-time and large-scale deployments. Additionally, ILP assumes a static network environment, which fails to account for the dynamic behaviour of real-world networks, where resource availability, traffic patterns, and slice demands change rapidly.

To address these limitations, Reinforcement Learning (RL) offers a promising alternative. RL enables an intelligent agent to learn and adapt to dynamic environments, making it ideal for scenarios like vRAN optimization. By modelling the network's current state and enabling real-time decision-making, RL can dynamically select optimal configurations, including Virtual Network Configurations (VNCs), paths, and wavelengths, while ensuring resource constraints are satisfied. This adaptability reduces computational overhead and provides scalability for large and complex networks.

The motivation for this project stems from the need to overcome the limitations of traditional optimization techniques and leverage the potential of RL to create a scalable, adaptive, and efficient framework for vRAN optimization. The proposed RL-based solution addresses key challenges, such as handling dynamic states, maintaining high resource utilization, and meeting stringent 5G slice requirements, making it a significant step forward in the evolution of network management techniques.

1.3 Objectives and Contributions

The primary objective of this work is to develop a scalable, adaptive, and efficient framework for optimizing resource allocation and network slicing in vRANs.

Key contributions include:

Designing an RL-Based Framework:

- Modeling the network state dynamically, incorporating parameters such as CPU capacity, link capacity, and wavelength availability.
- Developing an action space that allows the agent to select combinations of VNCs, paths (primary and backup), and wavelengths.

Reward Function:

- Defining a reward function that maximizes the difference between revenue and costs, analogous to the objective function of the ILP framework.

Validation and Performance Analysis:

- Implementing the RL framework and comparing its performance with the ILP approach in terms of slice acceptance rate, resource utilization, and computational efficiency.

Handling Constraints:

- Incorporating critical constraints like wavelength continuity, latency, bandwidth, and resource availability into the RL framework.

Related Work

2.1 Related Work

2.1.1. Functional Splitting and vRAN Centralization

One of the foundational aspects of vRAN optimization is functional splitting, where the protocol stack is divided across Centralized Units (CUs), Distributed Units (DUs), and Radio Units (RUs). The degree of centralization plays a pivotal role in determining resource efficiency and network performance.

Morais et al. in PlaceRAN [1] proposed an ILP-based framework for optimizing functional placement in vRANs, aiming to maximize centralization and minimize computational resources. However, while PlaceRAN effectively reduced resource costs, it did not account for dynamic network slicing or the reliability requirements of 5G use cases. Similarly, in [2] Garcia-Saavedra et al. investigated the trade-offs between centralization and bandwidth requirements, highlighting the importance of balancing computational loads and transport network capacities.

While these approaches provide valuable insights into centralization, they are limited to static network environments and fail to address the dynamic demands of 5G applications, such as Ultra-Reliable Low-Latency Communication (URLLC) and Enhanced Mobile Broadband (eMBB).

2.1.2. Network Slicing and vRAN Adaptation

Network slicing is a core innovation in 5G that enables the simultaneous support of diverse applications over a shared physical infrastructure. Recent works, such as Coelho et al.'s study [3], explored slice-aware function placement to meet the stringent requirements of individual slices, focusing on traffic isolation and functional splits. Multi-objective optimization techniques have been employed to balance bandwidth, latency, and resource utilization.

However, the scalability of these approaches is often limited by the static assumptions inherent in their optimization frameworks. Sen et al. in [4] proposed a heuristic-based function placement strategy that incorporates slicing but overlooks dynamic adaptation to changing network states, making it less effective for real-time deployments.

2.1.3. Lightpath Provisioning and Routing and Wavelength Assignment (RWA)

In optical transport networks, Wavelength Division Multiplexing (WDM) is critical for ensuring high-capacity communication links. The Routing and Wavelength Assignment (RWA) problem has been extensively studied, focusing on selecting paths and assigning wavelengths to maximize bandwidth utilization while avoiding conflicts.

Heuristic and ILP-based solutions for RWA have been proposed, addressing constraints like wavelength continuity and unique wavelength assignment. Alabbasi et al. [5] introduced an energy-efficient framework for RWA, emphasizing cost reduction. However, these methods primarily focus on static network states and do not integrate the dynamic requirements of vRAN optimization.

2.1.4. Reliability in vRAN

Reliability is paramount in 5G networks, particularly for applications like URLLC that require minimal downtime. Traditional approaches to reliability in vRAN rely on redundant resource allocation, including primary and backup paths. Ojaghi et al. [6] proposed methods to enhance fault tolerance through redundant function placement. However, these methods are computationally intensive and unsuitable for real-time deployment.

2.1.5. SliAvailRAN Framework

The SliAvailRAN framework [7] builds upon the limitations of previous works by incorporating network slicing, reliability, and functional splitting into an ILP-based model. SliAvailRAN introduces availability-aware slicing, ensuring that primary and backup paths are node-disjoint and meet slice-specific requirements.

However, it does not integrate lightpath provisioning, it is limited by its computational complexity, which hinders its scalability in large-scale and dynamic network environments. This limitation highlights the need for adaptive and scalable solutions, paving the way for Reinforcement Learning (RL)-based frameworks. RL offers real-time decision-making capabilities, overcoming the static and computationally expensive nature of ILP while retaining flexibility to meet the diverse requirements of 5G networks.

Problem Formulation

The problem of optimizing resource allocation in virtualized Radio Access Networks (vRANs) in 5G networks involves selecting Virtual Network Configurations (VNCs), primary and backup paths, and wavelengths while ensuring compliance with diverse constraints. The objective is to maximize the service provider's profit, defined as the revenue from accepted slice requests minus the costs of node activation, VNF instantiation, and wavelength activation.

3.1. System Model

1. **Network Model:**
   - The vRAN is represented as a graph G(N, L), where N is the set of nodes (CUs, DUs, and RUs) and L is the set of optical fiber links.
   - Each link supports multiple wavelengths, modelled using Wavelength Division Multiplexing (WDM).
2. **State Space:**
   - The current status of the network, including:
     - CPU availability at nodes.
     - Bandwidth availability on links.
     - Wavelength assignments and their availability.
     - Latency requirements for paths.
3. **Action Space:**
   - An action is defined as a tuple: (VNC, Primary Path, Backup Path, Wavelength), specifying:
     - The VNC selected for a slice.
     - The primary and backup paths assigned to meet reliability requirements.
     - The wavelength assigned to ensure lightpath continuity.
4. **Reward Function:**
   - The reward function is derived from the ILP objective:

     Reward=Revenue from accepted requests−(Node Activation Cost+VNF Instantiation Cost+Wavelength Activation Cost)

5. **Constraints:**
   - **Wavelength Continuity:** A single wavelength must be used along an entire path.
   - **Unique Wavelength Assignment:** No two lightpaths on the same link can share a wavelength.
   - **Primary and Backup path constraint:** Each RU will be associated with one slice instance of slice type URLLC, eMBB or MMTC, which will be assigned a Primary path and “tau” number of backup paths where the “tau” is calculated based on the QoS Availability requirement of the specific slice instance and its type.
   - **Primary and Backup path disjointness:** For each RU, the selected primary and backup path(s) should be node and link disjoint.
   - **Latency:** The selected paths must meet slice-specific latency requirements.
   - **Bandwidth:** The total bandwidth usage on a link cannot exceed its capacity.
   - **Node Processing:** The total CPU usage on a node must not exceed its capacity.

3.2. Reinforcement Learning Framework

3.2.1. State Representation

The state of the network at time t, S_t captures the current status of resources and ongoing slice requests.

It is represented as:

S_t=[CPU availability at nodes, Bandwidth availability on links, Wavelength availability, Slice-specific latency and bandwidth requirements]

The state dynamically updates as resources are allocated or released, reflecting the real-time status of the network.

3.2.2. Action Representation

An action A_t is a tuple representing the agent's decision at a given state:

A_t= (VNC, Primary Path, Backup Path, Wavelength)

- **VNC:** Specifies the functional split configuration for a slice.
- **Primary Path:** The primary route connecting the RU to the core network.
- **Backup Path:** A node-disjoint path for reliability.
- **Wavelength:** An assigned wavelength ensuring lightpath continuity.

The agent selects actions that adhere to the network constraints, ensuring valid and optimal resource allocation.

3.2.3. Reward Mechanism

The reward function incentivizes the agent to maximize profit while adhering to constraints:

R_t=Revenue from accepted slice−(Node Activation Cost+VNF Instantiation Cost+Wavelength Activation Cost)

Invalid actions, such as those violating wavelength continuity or exceeding resource limits, incur a negative penalty to discourage the agent from repeating such choices.

3.2.4. Policy Optimization with PPO

The Proximal Policy Optimization (PPO) algorithm is used to train the RL agent. PPO is particularly suited for this problem due to its efficiency in handling large state and action spaces and its stability during training. Key features include:

- **Clipped Surrogate Objective**: Ensures the policy does not deviate too far from previous policies, stabilizing training.
- **Exploration-Exploitation Balance:** Achieved using stochastic policies, enabling the agent to explore a wide range of configurations.

The Proximal Policy Optimization (PPO) algorithm is a reinforcement learning method that optimizes the policy using a clipped surrogate objective to ensure stability during updates. It falls within the family of policy gradient methods and is particularly designed to balance learning efficiency and stability.

In reinforcement learning, the goal is to maximize the expected cumulative reward:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \gamma^t r_t \right]$$

where:
$\pi_\theta(a|s)$: Policy parameterized by θ, mapping states s to action probabilities a.
τ: Trajectory generated by the policy.
r_t: Reward at time t.
γ: Discount factor.
The policy gradient theorem provides the gradient of $J(\theta)$:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\nabla_\theta \log \pi_\theta(a_t|s_t) \hat{A}_t \right]$$

where $\hat{A}_t$ is the advantage function:

$$\hat{A}_t = Q^\pi(s_t, a_t) - V^\pi(s_t)$$

with $Q^\pi(s_t, a_t)$ being the action-value function and $V^\pi(s_t)$ being the state-value function.

Surrogate Objective Function: To avoid large policy updates, PPO optimizes a surrogate objective function. The probability ratio between the new policy π_θ and the old policy $\pi_{\theta_{\text{old}}}$ is defined as:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$$

The surrogate objective is:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right]$$

where ϵ is a hyperparameter controlling the range of permissible updates (e.g., $\epsilon = 0.2$).

- The clipping function ensures that the policy update remains within a trust region, stabilizing training by preventing excessively large policy changes.
- If $r_t(\theta)$ deviates significantly from 1, the clipped term restricts its contribution to the objective.

Value Function Loss: PPO includes a value function loss to update the state-value estimator:

$$L^{\mathrm{VF}}(\theta) = \mathbb{E}_t\left[\left(V_\theta(s_t) - \hat{R}_t\right)^2\right]$$

where:

$\hat{R}_t$ is the target return (e.g., calculated via $\hat{R}_t = r_t + \gamma V(s_{t+1})$).

Entropy Regularization: To encourage exploration, PPO adds an entropy term to the objective function:

$$L^{\mathrm{ENTROPY}}(\theta) = \mathbb{E}_t\left[-\beta \cdot \sum_a \pi_\theta(a|s_t) \log \pi_\theta(a|s_t)\right]$$

where β is the entropy coefficient.

Combined Loss Function

The overall PPO loss combines the clipped surrogate objective, value function loss, and entropy regularization:

$$L^{\mathrm{PPO}}(\theta) = \mathbb{E}_t\left[L^{\mathrm{CLIP}}(\theta) - c_1 L^{\mathrm{VF}}(\theta) + c_2 L^{\mathrm{ENTROPY}}(\theta)\right]$$

where c_1 and c_2 are coefficients that balance the importance of the value function loss and entropy term.

Optimization Process:

1. **Sample Generation:** Collect trajectories τ\tau using the current policy π_θ by interacting with the environment.

2. **Advantage Estimation:** Compute $\hat{A}_t$ using methods like Generalized Advantage Estimation (GAE): $\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t}\delta_{T-1}$ where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$ is the temporal-difference error.

3. **Policy Update:** Optimize $L^{\mathrm{PPO}}(\theta)$ using gradient descent or Adam optimizer.

4. **Repeat:** Iterate until convergence or for a fixed number of training steps.

3.2.5. Training Process

1. **Environment Initialization**:
   - The network state is initialized with available resources and slice demands.
2. **Action Selection**:
   - The PPO agent selects an action A_t based on the policy.
3. **Validation and Execution**:
   - The action is checked for validity against constraints (e.g., wavelength continuity).

Invalid actions are penalized, and the agent retries.

4. **State Transition**:
   - The network state transitions to S_{t+1} based on the executed action, updating resource availability and slice allocations.
5. **Reward Assignment**:
   - The reward R_t is calculated based on the outcome of the action.
6. **Policy Update**:
   - The agent's policy is updated using PPO's objective, balancing exploration and exploitation.
7. **Repeat**:
   - This process is repeated over multiple episodes, enabling the agent to learn optimal configurations.

3.2.6. Evaluation Metrics

The performance of the RL framework is evaluated using the following metrics:

1. **Reward Convergence**:
   - Tracks the agent's ability to stabilize rewards over training episodes, indicating learning effectiveness.
   - The reward graph reflects the agent's increasing ability to select profitable and valid actions.
2. **VNC Distribution for RUs**:
   - Measures how many RUs are allocated to more centralized VNCs (e.g., VNC7, VNC8, VNC9).
   - A higher percentage of centralized VNCs indicates better resource pooling and higher centralization benefits.
3. **Computational Efficiency**:
   - Compares the time taken for training and decision-making with traditional ILP methods.

Implementation and Results

This section provides details of the implementation of the Reinforcement Learning (RL)-based framework using the **Proximal Policy Optimization (PPO)** algorithm for optimizing vRAN resource allocation. The implementation leverages the **Stable Baselines3** library in Python and is tested on multiple network topologies, ranging from small-scale to large-scale scenarios.

4.1. Software Tools and Libraries

1. **Programming Language:** Python
2. **Reinforcement Learning Library:**
   - **Stable Baselines3**: Provides pre-built PPO implementation with customizable hyperparameters.
3. **Simulation Tools**:
   - Custom-built simulation environment to emulate the vRAN network, including node and link properties.
4. **Optimization and Analysis Libraries**:
   - **NumPy** and **Pandas**: For data manipulation and reward computation.
   - **Matplotlib**: For visualizing reward convergence, VNC distribution, and other metrics.

4.2. Code Structure

1. **Environment**:

   - Implements the OpenAI Gym interface to integrate with Stable Baselines3.
   - Tracks the current network state (CPU, bandwidth, wavelengths, etc.).
   - Provides mechanisms for action validation and state transitions.
2. **Agent**:

   - The PPO agent interacts with the environment to select actions (VNC, paths, wavelengths).
   - Utilizes the Stable Baselines3 PPO implementation, enabling efficient training and exploration.
3. **Reward Calculator**:

   - Computes the reward based on the revenue minus the costs (node activation, VNF instantiation, wavelength activation).
   - Penalizes invalid actions to guide the agent toward feasible and profitable solutions.
4. **Simulator**:

   - Models network topologies, including optical links with Wavelength Division Multiplexing (WDM).
   - Implements constraints such as wavelength continuity, latency limits, and resource availability.

4.3. Workflow

1. **Environment Initialization**:

   - ○ Network topologies (8, 16, 32, 64, and 128 nodes) are initialized based on configurations from the SliAvailRAN paper.
   - ○ Each topology includes:
     - ■ Nodes representing CUs, DUs, and RUs with specified CPU capacities.
     - ■ Optical links with bandwidth capacities and wavelength sets.

2. **Agent Training**:

   - ○ The PPO agent is trained for each topology, with hyperparameters tuned for stability and performance.
   - ○ Training involves:
     - ■ Iterative action selection.
     - ■ Validation against constraints (e.g., wavelength continuity, resource limits).
     - ■ State transitions and reward assignment.

3. **Evaluation**:

   - ○ Post-training, the agent is tested on unseen slice requests to evaluate its performance in terms of reward convergence, VNC distribution, and slice acceptance rates.
   - ○ Results are compared across topologies to assess scalability.

4.3.1. Hyperparameter Settings

Key PPO hyperparameters used during training include:

- **Learning Rate**: 3×10^{-4}
- **Batch Size**: 64
- **Clip Range**: 0.2
- **Discount Factor (γ)**: 0.99
- **Policy Network Architecture**: Two fully connected layers with 64 units each and ReLU activation.

These hyperparameters were selected based on experimentation to balance exploration and exploitation while maintaining training stability.

4.4. Testing and Results Compilation

The RL framework was tested on the following topologies:

1. **8-node Topology**:
   - ○ Small-scale scenario to validate initial implementation.
   - ○ Provides insights into the agent's ability to allocate resources efficiently.

2. **16, 32, and 64-node Topologies**:

    - Medium-scale scenarios to test the agent's scalability.
    - Analyze VNC distribution to determine centralization patterns across RUs.

3. **128-node Topology**:

    - Large-scale scenario to evaluate the agent's performance under high resource demands.
    - Test the agent's ability to adapt to complex network states and maintain feasible solutions.

4.5. Results and Discussion

Fig. 5. (a)-(e) below depicts the reward convergence plots for different network topologies (8, 16, 32, 64, and 128 nodes), demonstrating the learning progression of the RL agent.

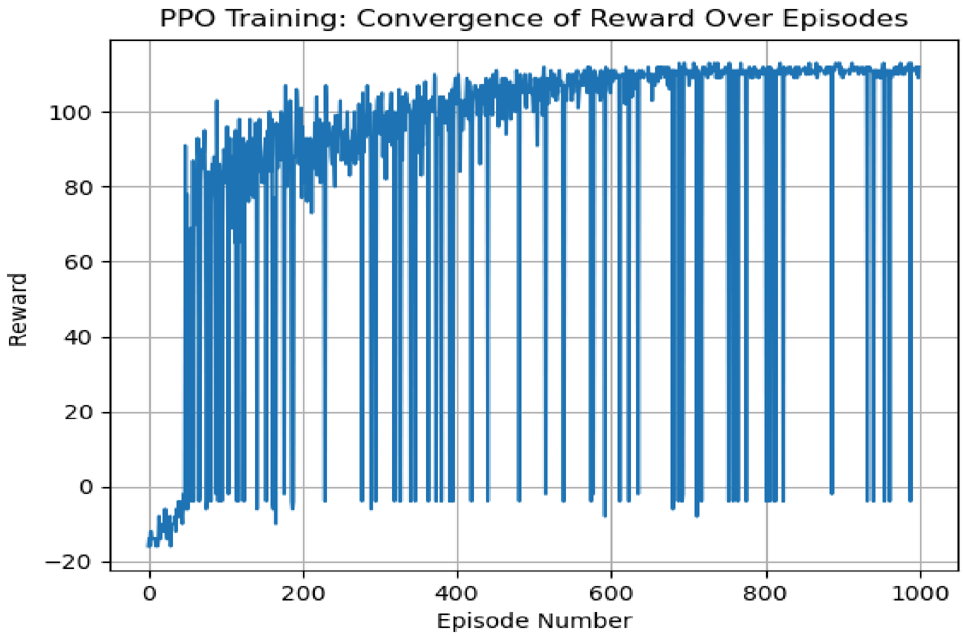

Fig 5 (a). Plot showing the reward convergence over 1000 episodes for 8-node topology.

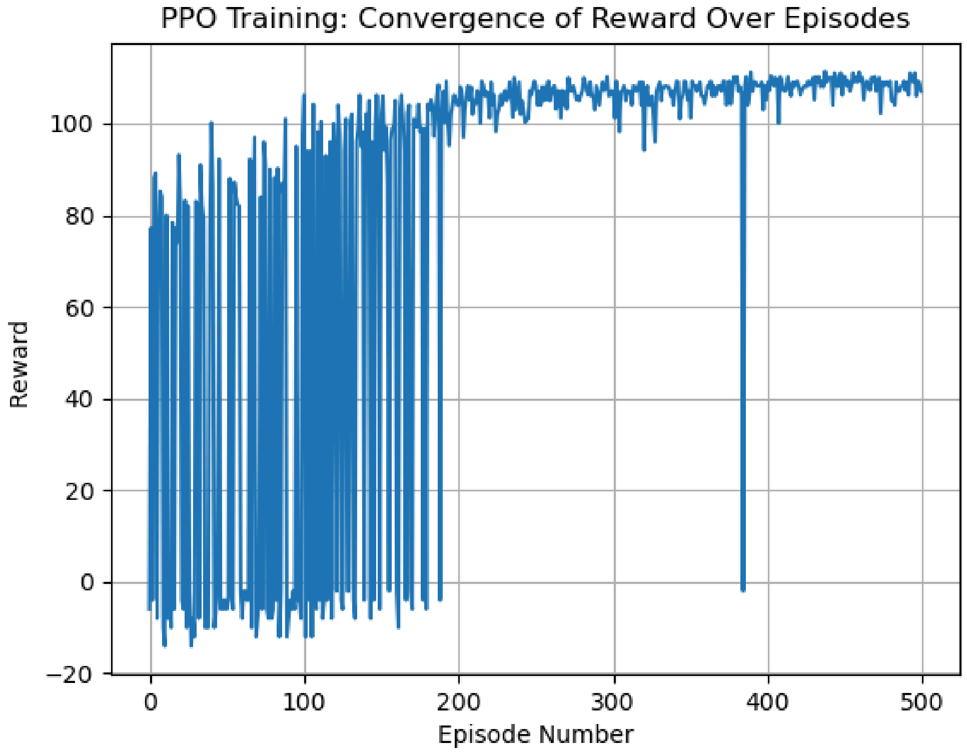

Fig 5 (b). Plot showing the reward convergence over 500 episodes for 16-node topology.

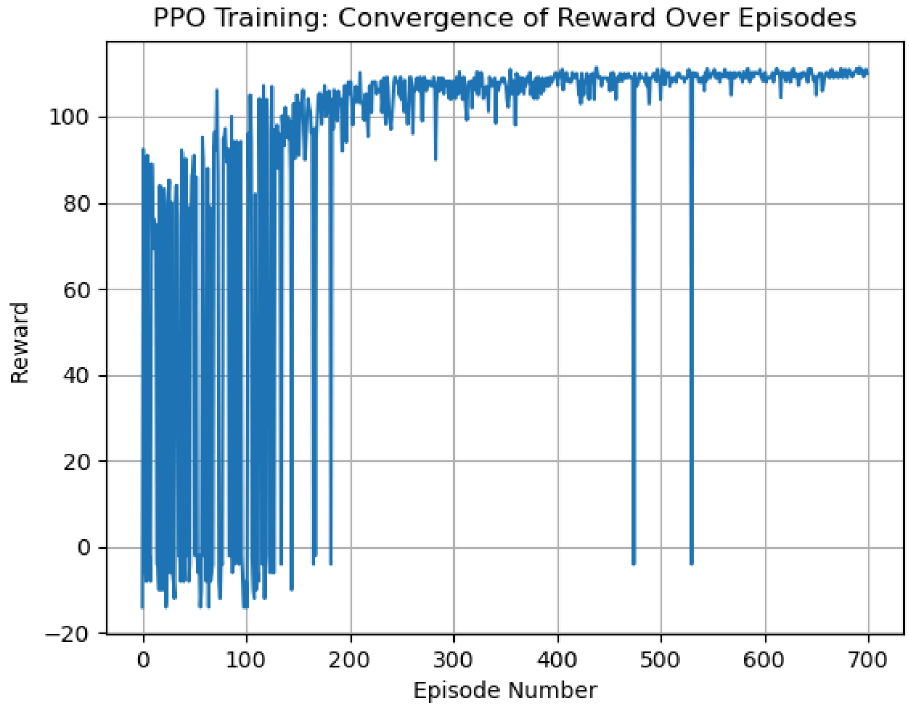

Fig 5 (c) Plot showing the reward convergence over 700 episodes for 32-node topology.

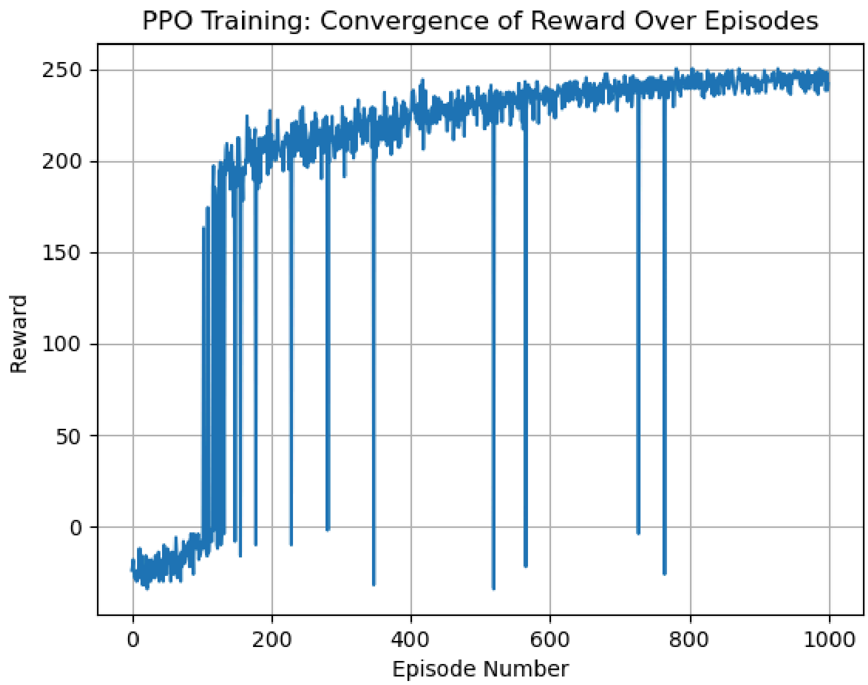

Fig 5 (d). Plot showing the reward convergence over 1000 episodes for 64-node topology.

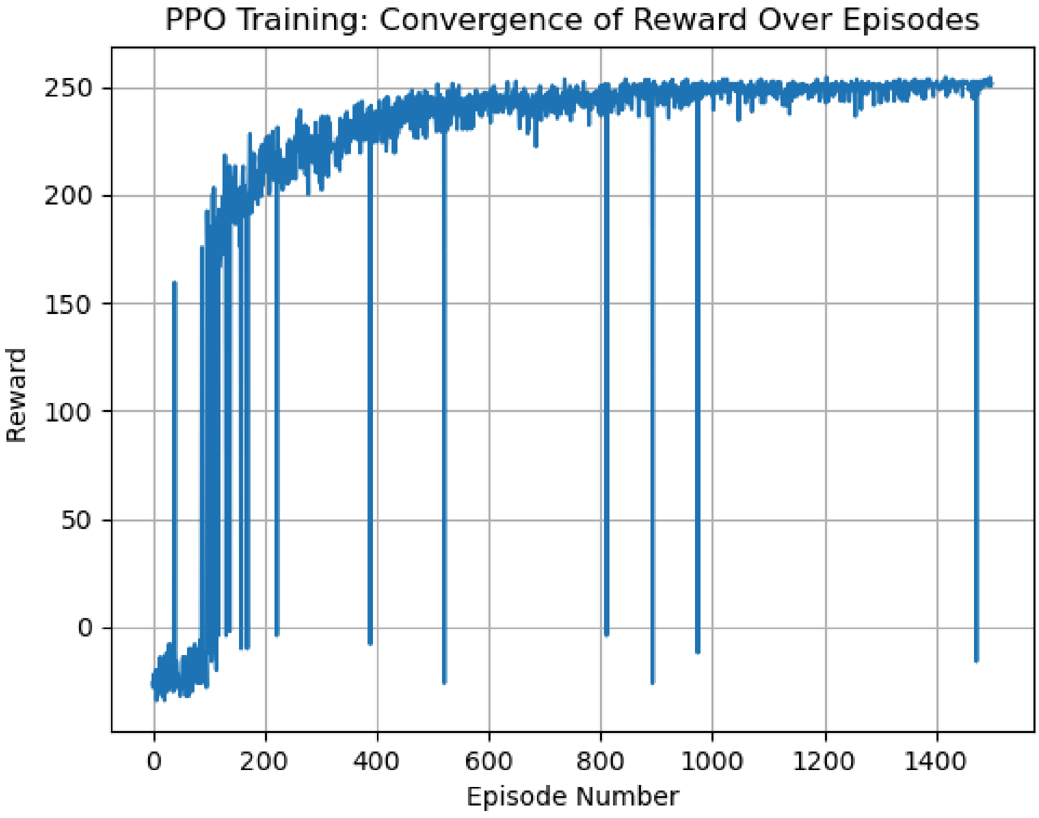

Fig 5 (e). Plot showing the reward convergence over 1400 episodes for 128-node topology.

In Fig 5 (a). we can see that the agent converges quickly, achieving a stable reward after approximately 1000 episodes. We can attribute this result to the fact that the smaller topologies tend to provide fewer action choices and simpler constraints, leading to faster convergence. We observe in Fig. 5 (b) that the convergence is achieved after around 500 episodes, slightly earlier than the 8-node scenario. This behaviour indicates better optimization due to a balanced action space and fewer resource bottlenecks compared to larger topologies (with a larger number of nodes and paths). In Fig. 5 (c), the convergence occurs after 700 episodes, showcasing the agent's ability to handle medium-scale networks effectively. This topology introduces a moderate increase in complexity, requiring the agent to explore more extensively before stabilizing. Reward convergence in Fig 5 (d), is achieved at approximately 1000 episodes, with occasional fluctuations indicating periodic exploration. The larger topology increases the complexity of resource allocation, particularly in ensuring disjoint primary and backup paths. In Fig 5 (e), Convergence requires around 1400 episodes, reflecting the significantly larger state and action space. This behaviour is expected due to the higher number of nodes, links, and constraints, which require more exploration for the agent to optimize effectively.

Note that here, in all the plots, the sudden “dips” in the curve, down to the negative values of rewards, are the cases where the agent is “exploring” some action which does not fulfil one or more of the constraints imposed on the actions, making the overall reward calculation down to a negative value.

Overall, the convergence trends across topologies validate the RL framework's scalability. Smaller networks converge faster due to simpler state-action spaces, while larger topologies demand extensive exploration but still achieve stability, demonstrating the robustness of the PPO algorithm.

Fig 6 below displays the allocation of VNCs across RUs, emphasizing the proportion of centralized VNCs (e.g., VNC7, VNC8, VNC9). This bar chart compares the distribution of Virtual Network Configurations (VNCs) allocated to RUs between the RL-based framework and the modified SliAvailRAN model with lightpath provisioning.

We can observe that, for smaller topologies, RL significantly allocates less proportionately the centralized VNCs as compared to modified SliAvailRAN formulation with lightpath provisioning however, as the topology size increases, the RL almost catches up to roughly similar proportions of VNC distributions as compared to its ILP counterpart. RL is expected to give slightly suboptimal results as compared to the exact ILP solution because there is a tradeoff here between the optimality and the computational complexity of the solution. In that regard, our RL solution is almost comparable to the exact ILP solution, especially in larger topologies, despite being faster in time complexity and a dynamic solution.

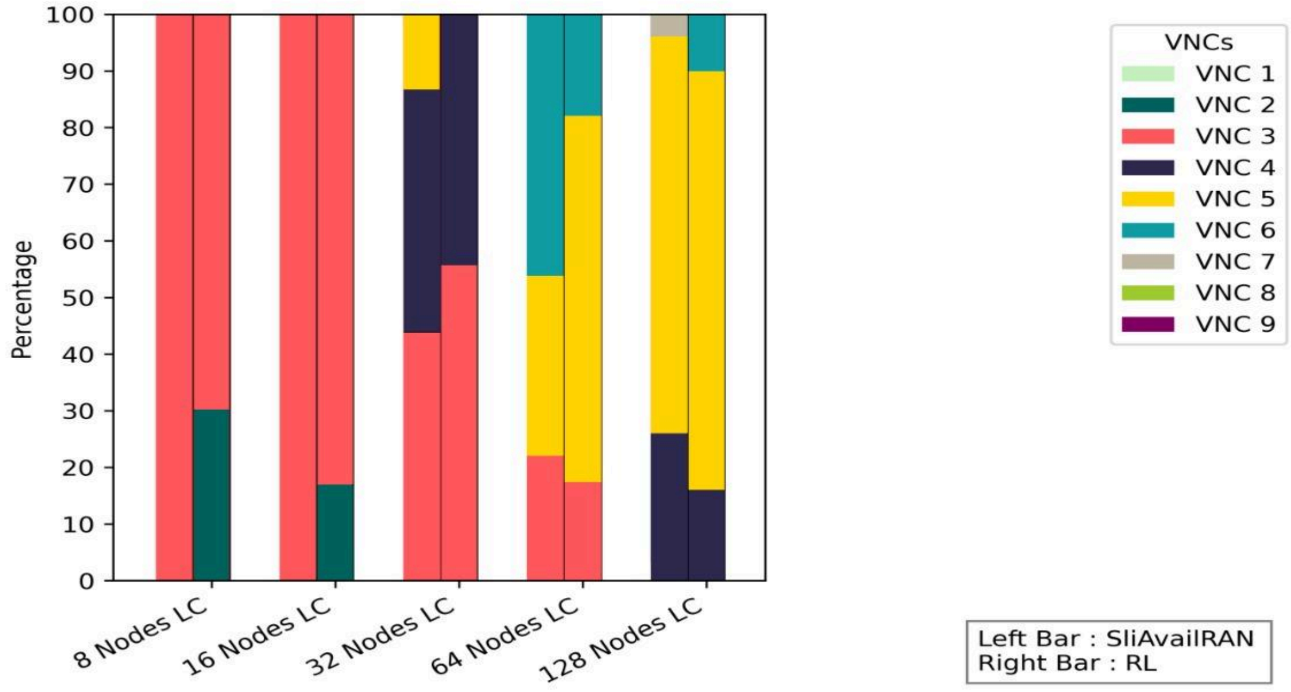

Fig. 6. Bar chart showing side-by-side comparison of the VNC allocation distribution (VNCs allocated to different RUs) for our RL-based Solution v/s modified SliAvailRAN [7] with Lightpath provisioning.

Conclusion and Future Scope

5.1. Conclusion

This project presents a Reinforcement Learning (RL)-based framework utilizing the Proximal Policy Optimization (PPO) algorithm for optimizing resource allocation in virtualized Radio Access Networks (vRANs). The proposed framework dynamically models the network state and enables an intelligent agent to select optimal configurations of Virtual Network Configurations (VNCs), primary and backup paths, and wavelengths. By integrating critical constraints such as wavelength continuity, latency, and resource capacity, the framework ensures efficient and reliable allocation of resources.

The framework was tested on various network topologies, ranging from small-scale (8 nodes) to large-scale (128 nodes), demonstrating its scalability and adaptability. Key evaluation metrics, including reward convergence, slice acceptance rate, and VNC distribution, validated the framework's performance. The RL-based approach achieved high resource utilization and reliability, addressing the limitations of traditional ILP methods by offering real-time, scalable solutions for 5G networks.

The implementation leveraging Stable Baselines3 in Python streamlined the development process, while the simulation of diverse topologies provided a robust testing environment. Results highlight the potential of RL for real-time vRAN optimization, paving the way for dynamic and flexible management of 5G networks.

5.2. Future Work

While this project achieves significant advancements in vRAN optimization, several opportunities for future work remain:

1. **Multi-Agent RL**:
    - Extend the framework to a multi-agent system where each RU, DU, or CU acts as an independent agent.
2. **Enhanced Reward Function**:
    - Refine the reward function to incorporate additional factors like energy efficiency, fairness in resource allocation, and QoS metrics.
3. **Integration with Real-World Data**:
    - Test the framework with real-world network data to validate its performance under practical conditions.
    - Incorporate real-time traffic patterns and dynamic topology changes for improved adaptability.
4. **Dynamic Topology Adaptation**:
    - Incorporate mechanisms to handle dynamic network changes, such as node failures, link congestion, and varying slice demands, for enhanced robustness.

References

[1] F. Z. Morais, G. M. F. De Almeida, L. L. Pinto, K. Cardoso, L. M. Contreras, R. da Rosa Righi, and C. B. Both, “PlaceRAN: Optimal Placement of Virtualized Network Functions in Beyond 5G Radio Access Networks,” IEEE Transactions on Mobile Computing, 2022.

[2] A. Garcia-Saavedra, J. X. Salvat, X. Li, and X. Costa-Perez, “WizHaul: On the Centralization Degree of Cloud RAN Next Generation Fronthaul,” IEEE Transactions on Mobile Computing, vol. 17, no. 10, pp. 2452–2466, 2018.

[3] W. da Silva Coelho, A. Benhamiche, N. Perrot, and S. Secci, “Function Splitting, Isolation, and Placement Trade-Offs in Network Slicing,” IEEE Transactions on Network and Service Management, vol. 19, no. 2, pp. 1920-1936, 2022.

[4] N. Sen and A. A. Franklin, “Slice-Aware Baseband Function Placement in 5G RAN Using Functional and Traffic Split,” IEEE Access, vol. 11, pp. 4521–4532, 2023.

[5] A. Alabbasi, X. Wang, and C. Cavdar, “Optimal Processing Allocation to Minimize Energy and Bandwidth Consumption in Hybrid CRAN,” IEEE Transactions on Green Communications and Networking, vol. 2, no. 2, pp. 545–555, 2018.

[6] B. Ojaghi, F. Adelantado, E. Kartsakli, A. Antonopoulos, and C. Verikoukis, “Sliced-RAN: Joint Slicing and Functional Split in Future 5G Radio Access Networks,” IEEE International Conference on Communications (ICC), 2019.

[7] S. Ahmed, M. Ramnani, and S. Sharma, “SliAvailRAN: Availability-Aware Slicing and Adaptive Function Placement in Virtualized RANs,” IEEE 25th International Conference on High Performance Switching and Routing (HPSR), 2024.