CSCI 4179 Group 2 – Research Proposal

Evaluating a Q-learning based approach to wireless network load balancing

Eduard Kakosyan, B00887569

Ethan Rozee, B00863306

Jack Whitmer, B00769341

I Introduction

I-A Problem Statement

Q-learning, a policy-based reinforcement learning algorithm, enables optimal decision-making in complex environments through iterative state-action value updates without requiring a complete environmental model. In network environments, the dynamic nature of resource utilization, bandwidth demands, and signal quality presents an ideal application for Q-learning based optimization. The implementation of Q-learning in network load balancing has the potential to improve overall network performance, particularly in high-traffic scenarios. Therefore, we seek to evaluate the performance of a Q-learning based load balancing framework in a simulated wireless network environment.

I-B Importance

The evaluation of Q-learning's performance in network load balancing will contribute to the growing body of research on reinforcement learning applications in network management. While Q-learning has demonstrated success in various domains, its effectiveness in dynamic network environments, particularly regarding routing optimization and load distribution, remains understudied. Through simulation-based evaluation, we can better understand the potential benefits and limitations of Q-learning for optimizing resource allocation under various traffic patterns.

I-C Objectives

In our project, we aim to:

1. Implement a policy-based Q-learning framework for load balancing in a simulated wireless network environment with multiple access points of varying capacities
2. Evaluate the framework's performance in across different traffic patterns using key metrics including throughput, delay, and packet loss rate
3. Analyze our Q-learning algorithm's ability to optimize bandwidth allocation

The completion of these objectives will provide understanding as to how Q-learning adapts to varying network loads and usage patterns while optimizing performance.

II Related Work

Reinforcement learning (RL) involves training agents to make sequences of decisions by interacting with an environment to maximize cumulative rewards. The agent learns a policy that maps states to actions, optimizing for long-term returns rather than immediate gains. A key aspect of RL is balancing exploration (trying new actions to discover their effects) and exploitation (choosing known actions that yield high rewards). Common challenges in RL include designing a reward function that appropriately penalizes undesirable behavior while adequately rewarding positive actions, ensuring balanced learning. Additionally, determining the frequency and timing of rewards is crucial, as it influences the agent's learning efficiency and convergence.

II-A Literature Review

In the first paper by M. Boushaba et al. from 2013 (Boushaba, Hafid, Belbekkouche, & Gendreau, 2013), they discuss a new routing algorithm they designed which is called Reinforcement Learning-based Best Path Routing (RLBPR). They claim it is suitable for mesh networks to find the best path to a gateway given several variables about the state of the network. They also added analytical guarantees to the algorithms so that mesh networks with cycles do not send packets in a cycle. They developed two learning algorithms, RLBDR-PQ and RLBDR-ZPQ. RLBDR-PQ prioritizes high-quality links near gateways by penalizing poor-quality links as the path approaches the gateway. RLBDR-ZPQ gives extra weight to nodes in a critical zone (N-hops from the gateway) to reduce congestion and interference around gateways. Both algorithms use Q-learning with Q-tables to calculate the most optimal path for routing. The values in the Q tables are updated according to the network state which is determined by Gateway Load and Link Quality Metric (LQM). They tested their two algorithms against some common routing algorithms at the time of writing their paper: MIC, ETX, Nearest-G, BP2BG, and Load-G. The results of their testing were that their RLBPR algorithms both had better network performance based on throughput, delay, and packet loss rate. Furthermore, the RLBDR-PQ algorithm was better between their two proposed algorithms.

Another paper by Z. Mammeri from 2019 (Mammeri, 2019) is a review of RL techniques applied to routing in networks. They discuss the use cases in various other fields and explore the intricacies of how it works. Then they highlight the superior performance of Q-learning for routing which they conveniently name Q-routing. Furthermore, the crucial part of any RL solution is tuning the trade-off between exploration vs exploitation. Another important point is that RL is typically the option for situations where the goal is to optimize the performance of the network with numerous network state parameters with high variability. Examples of these include energy constraints, topology changes, and more. This paper also covers a plethora of various RL routing protocols of which the best can vary based on the network. They mention two key points for future research: to test performance with deep learning and to improve security concerns.

The last paper we read, by D. Godfrey et al. from 2021 (Godfrey, et al., 2021), they discuss another Q-learning based approach for software defined networks (SDNs). The key difference between a software defined network and something like the mesh network in the first layer is that the routing is performed on a centralized server which can control each of the routers individually. The protocol they developed periodically updated Q-values based on the network's current state (queue length, retransmission ratio, hop count). They tuned their reward function to optimize the performance based on those parameters. The name of their algorithm was Q-Learning Based Congestion-Aware Routing (QCAR) and the key difference between it and previous solutions is that it (as the name indicates) treats the congestion as a parameter to optimize. The result was that QCAR showed a higher packet delivery ratio, and a shorter end-to-end delay compared to traditional schemes.

II-B Connection to Our Work

These papers demonstrate that Q-learning is the best option to use for optimal routing performance in many real-world environments. They cover most of the required theory behind the implementations that we will use. They also discuss the techniques that do and don’t work in various situations. All of this will ultimately help us maintain a more informed implementation in hopes to replicate and improve upon the results found in the papers.

III Methodology

In this section, we present the proposed Reinforcement Learning-Based Load Balancing framework for Network applications. The methodology is divided into three main components: the environment/system architecture, the reinforcement learning model, and the experimental setup. Each of these is described below.

III-A Environment/System Architecture

The proposed load balancing framework is designed to dynamically allocate bandwidth across a set of access points to optimize network performance under different traffic patterns. Since live systems are not accessible, we will use a network simulator that supports reinforcement learning (RL) integration. For this study, we have selected GNS3 as the simulation environment. Within this setup, the following components are included

- **Network Scheduler**: This component generates network traffic with different characteristics, including smooth, periodic, and burst patterns. It continuously interacts with the RL agent to determine how to distribute bandwidth across access points.
- **Access Point Pool**: This is a collection of simulated access points that will carry out the tasks assigned by the network scheduler.
- **Reinforcement Learning Agent:** The RL agent monitors real-time metrics and dynamically decides which access point should handle each task. It learns from past actions and continuously improves its load balancing performance.

III-B Reinforcement Learning Model

Our framework will use a policy-based Q-learning algorithm for load balancing. This method was chosen because it can learn optimal policies without needing a full model of the environment and is relatively simple to implement.

The main components of the Q-learning algorithm are:

- State (S): Represents the current condition of the system, including each access point's resource utilization, the number of active tasks, and the overall system load.
- Action (A): An action corresponds to assigning a task to one of the available APs. The RL agent selects an action based on its learned policy to balance the workload efficiently.

- Reward (R): The reward is a measure of the system's performance after an action is taken. In our framework, the reward is based on response time, resource utilization, and the number of completed tasks. The goal is to minimize response time and prevent any AP from being overloaded, or underworked, which results in higher rewards.
- Q-Value (Q): The Q-value represents the expected future reward of taking a particular action in each state. The RL agent updates its Q-values through interactions with the environment, gradually converging to an optimal load balancing policy. A Q-table is used to store these values.

The Q-learning algorithm updates the Q-value inside our table using the following formula:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a}\big(Q(s',a') - Q(s,a)\big)\right]$$

Where:

- $\mathrm{Q}(s,\mathrm{a})$ is the Q-value for state s and action a.
- α is the learning rate
- r is the reward received after taking an action in each state.
- γ is the discount factor, which balances immediate and future rewards.
- $max_a Q(s',a')$ is the maximum expected future reward for the next state.

The Q-learning agent iteratively improves its policy by updating the Q-values until it learns the optimal strategy for distributing tasks across APs.

III-C Experimental Setup

To evaluate the performance of the proposed RL-based framework, we set up a simulated environment using GNS3. The experimental setup consists of the following components:

- Network Infrastructure: A simulated network environment with a pool of 3 AP's. Each access point will have different capacities to mimic real-world conditions.
- Traffic Patterns: The network scheduler generates three distinct traffic patterns:
  - *Smooth*: Consistent and steady data flow.
  - *Periodic*: Alternating high and low traffic phases
  - *Burst*: Sudden spikes in network demand.
- Performance Metrics:

- *Throughput*: The total amount of data successfully transmitted over the network.
- *Delay*: The average time taken for data packets to reach their destination.
- *Packet Loss Rate*: The percentage of packets that fail to reach their destination.

The RL agent will be trained over multiple iterations, and its performance will be analyzed across different traffic patterns. The results are evaluated based on how effectively the RL agent can improve throughput, while reducing delay and packet loss.

III-D Implementation

The Q-learning algorithm will be implemented in Python, using the GNS3 network simulation environment and OpenAI's Gymnasium library. The system will be designed to dynamically adjust bandwidth allocations based on real-time network conditions and learned policies.

IV Timeline

Milestone	Week of	Goals
I	**14 Feb.**	Research proposal submission
II	**28 Feb.**	Simulation setup with network topology complete
III	**7 Mar.**	Q-learning agent and comparison algorithms implemented
IV	**14 Mar.**	Performance testing and algorithm optimization
V	**21 Mar.**	Final testing and results compiled with visualizations
VI	**26 Mar.**	Group presentation
VII	**11 April**	Final report submission

V Works Cited

Boushaba, M., Hafid, A., Belbekkouche, A., & Gendreau, M. (2013). Reinforcement learning based routing in wireless mesh networks. *Wireless Networks, 19*, 2079–2091.

Godfrey, D., Kim, B. S., Miao, H., Shah, B., Hayat, B., Khan, I., . . . Kim, K. I. (2021). Q-learning based routing protocol for congestion avoidance. *All Works, 68*(3), 3671-3692.

Mammeri, Z. (2019). Reinforcement Learning Based Routing in Networks: Review and Classification of Approaches. *IEEE Access, 7*, 55916-55950.