Reinforcement Learning Network Optimization For 6G And Beyond

1st Abhishek Vaid
Department of Computer Science
San Jose State University
San Jose, USA
abhishek.vaid@sjsu.edu

2nd Parth Patel
Department of Computer Science
San Jose State University
San Jose, USA
parthamrutbhai.patel@sjsu.edu

I. Introduction

Congestion in network lines cause "LAG" or "Ping", where in you do not get your promised amount of Internet speed. Therefore, if you are consuming a time critical data like stock market updates or live streaming a high-definition video you face buffering problems. In this paper we discuss our work related to using reinforcement learning techniques to optimize the data packet transfer in networks leading to less delay and congestion. We have used Full Echo Q-Learning [1] for adaptive change of the routing path of data packets and compared it's performance with conventional Shortest Path algorithm. Simulation results show that Full Echo Q-Learning performs better than Shortest Path especially under high load levels. Conventionally Shortest Path algorithm was used to determine the packet routing path but with increase in customer pool the network congestion increased, this led to the algorithm becoming incapable to handle such scenarios. As Q-Learning has been there for a while, it has been known to solve network issues. With Q-Learning [2, 3] on successive iteration the agent learns the best possible action to route a data packet over a busy network.

Currently in 5G networks reinforcement learning is being used to solve network congestion issues. But as 6G in under development and promises to give TB/s [4] speed, we would need more usage of reinforcement learning technologies to primarily drive such network.

The primary reason of us choosing Full Echo Q-Learning is because, on our research of previous works [1] on the same topic we found out that Full Echo Q-Learning is considerably more stable to handle network congestion in comparison to regular Q-Learning. The purpose of choosing the network optimization problem was primarily because reinforcement learning works ideally for solving such problems. Going forward in this paper we will establish the correctness of previous statement.

II. Formal Problem And Evaluation Criteria

A. Formal Problem

In this paper our reinforcement learning agent will learn to choose the best possible path in a matrix environment representing the network of nodes. The agent will observe accurate statistics on each node position to learn which routing decisions lead to minimum delivery times. The agent at its current node will check the status of its neighbouring nodes before taking a decision for the best possible action, as in this communication it will receive the amount of load its neighbouring nodes have. This is an optimization problem where we minimize the data packet transfer time by choosing the best possible route.

Formally the Markov Decision Process (S,A,P,R,γ) for our problem is defined as the following - State S is the distance from current node to destination node. A is the action of choosing from one of the link paths, R reward is function of time spent in a node queue plus the time taken to transit from a node to another node. P is probability of choosing a particular path for routing packet through it and γ is the discount factor, γ ranges between 0 to 1.

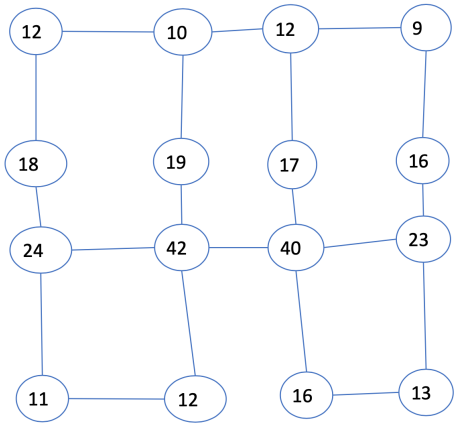

Fig. 1. Example of a matrix environment.

B. Evaluation criteria

The success of our project is achieved by making the agent learn to choose the best possible policy for network route selection and avoid congestion in network. We also compare the results of the Full Echo Q-Learning approach with the results of Shortest Path approach under similar simulation settings. The basis for comparison is the average packet

routing time for increasing load level on successive iterations between Full Echo Q-Learning and Shortest Path.

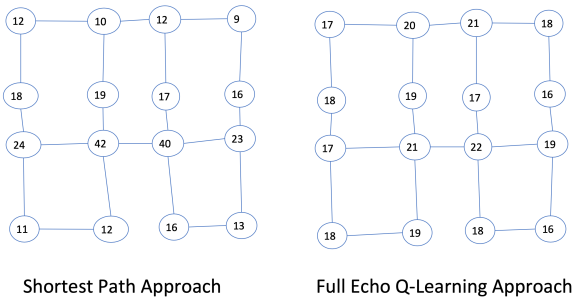

Fig. 2. Output contrast we want to achieve at the end [1]

As shown in figure 2, on success of our experiment the distribution of data packets across the network shall be evenly distributed with Full Echo Q-Learning approach. While it would be unevenly distributed and show high data traffic paths with Shortest Path algorithm approach.

III. Methodology

The shortest path algorithm only provides the shortest route from source to destination node, but it does not consider the load on high traffic networks, which ultimately leads to delayed data packet transfer. But when we use Full Echo Q-Learning the agent discovers and learns every possible outcome before taking a decision and developing a policy.

In Full Echo Q-Learning the agent will learn the outcome of every action from the current state to the next state. Then it will take the action which takes the lowest possible amount of time of transition of data packets between the states, thereby minimizing the reward. This approach helps in the exploration process which leads to finding of route with less delay time during high load level.

$$Q(S,A) \leftarrow Q(S,A) + \alpha \left(R + \gamma \max_{a'} Q\left(S', a'\right) - Q(S,A) \right)$$

Fig. 3. Q-Learning Equation [3]

In our methodology, S is the distance from current node to destination node, where distance is the time taken to route packet from one node to another set to 1 ms. Total distance from current to destination node is the summation of all inter node transitions the data packet travelled through. A is the action of choosing from one of the link paths. S' is the distance of the next node from the final destination node, R reward is time spent in node queue plus the time taken to transit from a node to another node. Where α is the learning rate which ranges between 0 to 1. γ is a discount factor, γ ranges between 0 to 1.

Our agent tries to correctly estimate the time delivery time of packet from one node x to its immediate node y and minimise this time, which eventually will minimise the delivery time from original source node to destination node.

$$t = \min_{z \in \text{neighbors of } y} Q_y(d,z)$$

$$\Delta Q_x(d,y) = \eta \, (\overbrace{q+s+t}^{\text{new estimate}} - \overbrace{Q_x(d,y)}^{\text{old estimate}})$$

Fig. 4. Learning Equations [1]

The equations used to calculate the reward which is the function of time and also update in Q-values are shown in figure 4.

IV. Evaluation Settings

For performing this experiment we have created our own grid matrix environment using the Gym library. We have created two networks, one is a 36x36 network and other is a 116x116 bigger network representing a more real world scenario. Our Q-Function is a 3D Array of Number-of-Nodes x Number-of-Nodes x Number-of-Actions, which stores the distance of each current node to destination node for a particular action. For load on network, we have made a priority queue which stores the current event to be performed, containing the delay for a packet route time of the node, current node and destination node information. The delay load increase per learning cycle. We perform 10000 iterations on each load level for our agent to learn and then we make it take the best possible action which then we evaluate against Shortest Path algorithm's chosen action. For mapping distances we have made a 36x36 or 116x116 2D array which stores the distance of each node to every other node in the network. We select our source and destination nodes randomly to generate new packet for each iteration and insert it into the priority queue. We have set our learning rate α to 0.7 and discount factor to 1. Also the agent only learns to route data packet for one step from the current node towards the destination node and does not travel the entire distance from source to destination. But as we are performing 10000 iterations for each load level in which we randomly choose the current and destination node therefore the agent will ultimately learn the Q-values for all possible actions throughout the network at different load settings.

V. Evaluation Result

On simulating experiments on both types of networks we observed that as load of network increases the shortest path algorithm starts taking more time than Full Echo Q-learning. These results are similar to what we expected while doing our research on our reference paper [1]. To understand this in more detail as the load increases on successive iteration it gets multiplied over a randomly selected time for load level for that iteration thereby increasing the load over successive iteration. The shortest path algorithm does not understand the difference in load thereby only providing a static shortest path action from source to destination node. So as the load increases it starts taking more time to route the data packet. On the other hand Full Echo Q-Learning learns on every successive iteration and updates the Q-values thereby taking the best possible action from source to destination and rather decreases

the time of data packet routing as it learns even with increasing load levels as shown in figure 5.

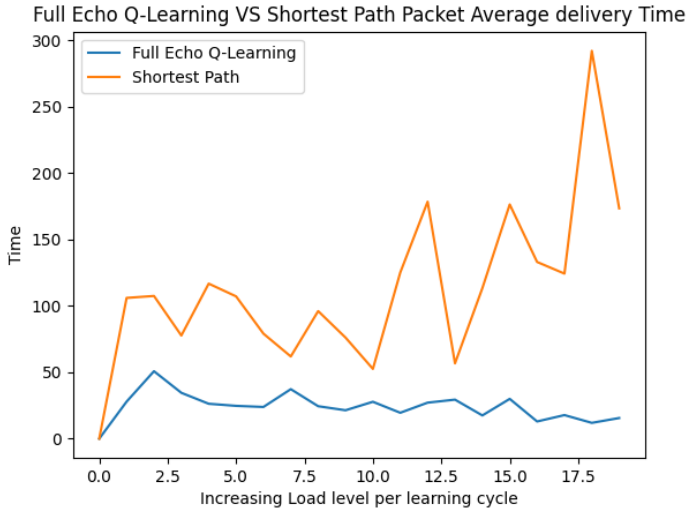

Fig. 5. Full Echo Q-Learning vs Shortest Path Packet Average Delivery Time

We also performed more experiments by changing the value of discount factor from 1 to 0.5.

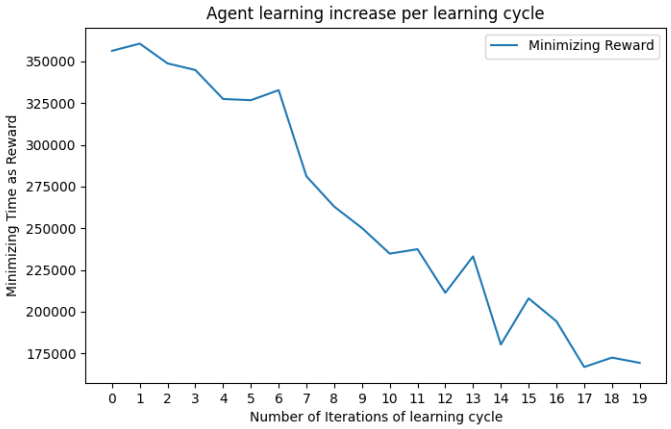

Fig. 6. Learning curve for discount factor 1.0

As shown in figure 6 with successive iterations the reward time minimises very clearly, which represents that the agent is leaning and the policy keeps on improving. Also the discount factor over here is 1 which is best suited for our scenario because the agent focuses on future reward, constantly trying to minimize it and thereby decreasing the routing time. After this we did another experiment in which we decreased the discount factor to 0.5. On doing so as shown in figure 7 the agent took a significantly more iterations before it started learning. This happened primarily because the agent's focus turned towards current state reward and it took time to start learning to minimise the reward time.

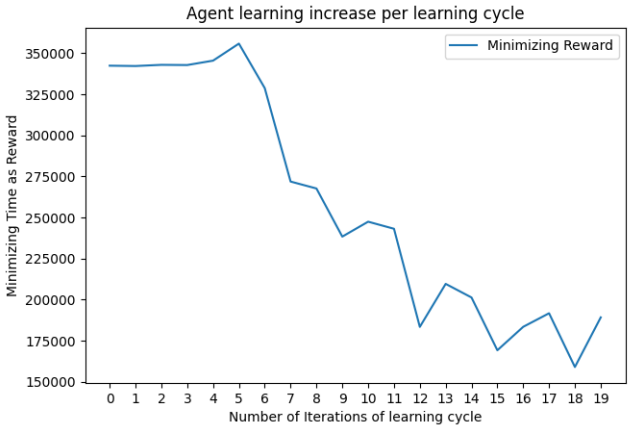

Fig. 7. Learning curve for discount factor 0.5

VI. Discussion

From the results we achieved we could clearly understand how reinforcement learning could be used in 6G networks for high speed data transfer. We believe that in coming years 6G will provide us with speeds so high that it would seem as though the data was already stored on system hard drive. The results clearly implied that in the coming future we would have self learning networks which would optimize themselves to the area in which they are installed. As shown throughout our experiments the conventional shortest path algorithm could not compete with reinforcement learning Full Echo Q-Learning algorithm. We also learnt that the higher is the discount factor the more future reward gets minimized leading to faster and stable learning rates. With upcoming machine learning and artificial intelligence age it would be a significant advantage in creating networks that can handle heavy data traffic.

VII. Summarizing Related Works

The inspiration behind this project can be found in the work Packet Routing In Dynamically Changing Networks by Justin A. Boyan and Michael L. Littman [1]. The authors talk about how reinforcement learning can be used to optimize networks and how they are better than conventional routing algorithms. The primary motivation to work apply reinforcement learning to 6G networks can be found in the paper Vision, Requirements, and Technology Trend of 6G: How to Tackle the Challenges of System Coverage, Capacity, User Data-Rate and Movement Speed by Shanzhi Chen and Ying-Chang Liang [4], this paper talks about how 6G networks will have TB/s speed by heavily making use of the machine learning age technologies. In the paper Full Echo Q-Routing with Adaptive Learning Rates: a Reinforcement Learning Approach to Network [5] by Yuliya Shilova, Maksim Kavalerov and Igor Bezukladnikov, the authors introduce a new routing technique making use of reinforcement learning and built on top the technique introduced in [1]. Their approach is extension of Full Echo Q-Learning technique where they use two learning rates to update the Q-values of nodes for better exploration.

REFERENCES

[1] J. Boyan and M. Littman, “Packet Routing In Dynamically Changing Networks: A Reinforcement Learning Approach,” Advances In Neural Information Processing Systems 6, 1994.

[2] Mammeri, Zoubir. (2019). Reinforcement Learning Based Routing in Networks: Review and Classification of Approaches. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2913776.

[3] Ishigaki, Genya CS 271 Model Free Control, 2021

[4] Chen, Shanzhi Liang, Ying-Chang Sun, Shaohui Kang, Shaoli Cheng, Wenchi Peng, Mugen. (2020). Vision, Requirements, and Technology Trend of 6G: How to Tackle the Challenges of System Coverage, Capacity, User Data-Rate and Movement Speed. IEEE Wireless Communications. PP. 1-11. 10.1109/MWC.001.1900333.

[5] Y. Shilova, M. Kavalerov and I. Bezukladnikov, ”Full Echo Q-routing with adaptive learning rates: A reinforcement learning approach to network routing,” 2016 IEEE NW Russia Young Researchers in Electrical and Electronic Engineering Conference (EIConRusNW), 2016, pp. 341-344, doi: 10.1109/EIConRusNW.2016.7448188.