A Reinforcement Learning Approach to Packet Routing on a Dynamic Network

Huanxi Liu (University of California, San Diego), Anthony Meza (University of California, Irvine), Christina Tran (Harvard University), Hwai-Liang Tung (Brown University)

Abstract

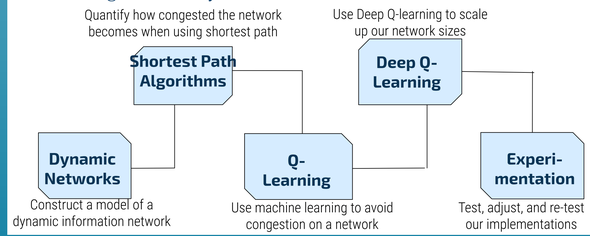
Finding optimal packet routing strategies for real world dynamic networks is a ubiquitous but largely open problem. In this work, we answer two main questions: (1) Can the reinforcement learning techniques Q-learning and Deep Q-learning be applied to the packet routing problem in order to reduce congestion and packet delivery times in comparison to the traditional shortest path algorithm? (2) Does this advantage hold for many different types of networks? To answer these questions, we develop a model for simulating packet routing on a dynamic network, in which a network's adjacencies and edge weights change in discrete. We then use this model to quantify each routing method's performance across a wide variety of connectivity regimes, where network size, edge distribution, and network dynamics are systematically varied. Overall, Q-learning and Deep Q-learning consistently lower congestion and decrease packet delivery times compared to the shortest paths algorithm on all tested networks. From this, we offer general guidelines for optimizing routing across various network topologies, and aggregate method-specific results for each topological regime. This work was part of UCLA RIPS 2020.

Introduction

MOTIVATION: Inefficiencies in satellite routing

Satellite networks are used for tasks such as GPS and communications. To perform these efficiently and reliably, suitable algorithms must be developed to decide how a packet of information should travel from satellite to satellite to reach its destination. Traditional algorithms direct packets on the calculated shortest path, but these cannot adapt to changes in satellite connections as they leave and re-enter service range. Furthermore, in a high-traffic network, they cause congestion in a few center satellites while underusing periphery satellites, slowing delivery time of these packets. We hypothesize that using machine learning to teach a satellite network how to route packets will result in faster delivery and less congestion.

OUR GOAL: Develop a reinforcement learning based program for routing data in a dynamic network.

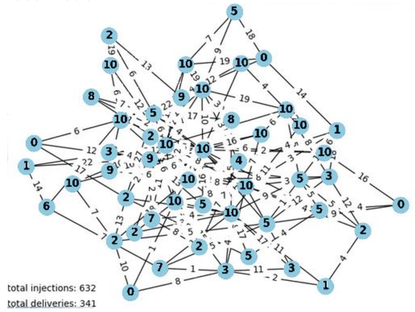


Methodology

DYNAMIC NETWORK MODEL

We developed a dynamic network model in Python with the following characteristics:

- At each discrete time step:
  - Edges are randomly deleted/restored
  - New packets are injected into the network
  - Edge weights change in a sinusoidal manner
  - Each node can send up to a fixed number of packets, Y.
- Each edge weight represents transmission delay for a packet traveling along that path
- Each node can only hold X packets at any given time step
- Packets are initialized randomly throughout the network with a random destination



SHORTEST PATH ROUTING

At each time step, recalculate each packet's shortest path using Dijkstra's algorithm.

Q-ROUTING

The machine learning algorithm first randomly sends packets and is given a reward based on how much closer the packets get to their destination and how much congestion it causes. Based on these rewards, it estimates a quality value, or Q-value, for each possible routing decision. After it has sufficiently learned about the satellite network, it can choose actions based on the highest Q-value.

DEEP Q-ROUTING

Like Q-routing, the algorithm estimates Q-values in order to make decisions. However, instead of simply storing them in a table, a neural network is used and repeatedly optimized in order to efficiently estimate these Q-values. We use a simple neural network with 2 linear layers with ReLu activations.

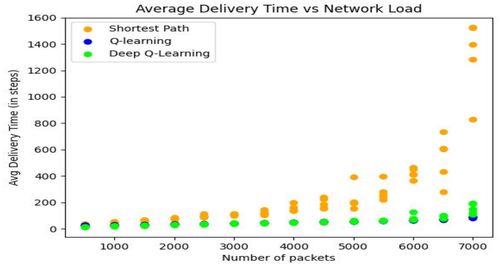
TESTING PERFORMANCE

Average Packet Delivery Time
(in the number of time steps)

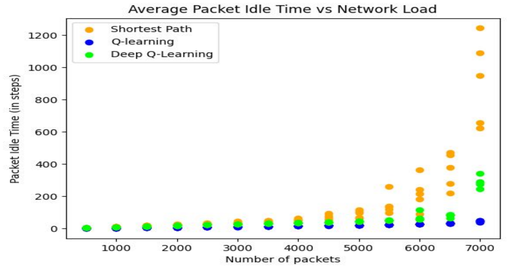
Level of Network Congestion
(in the average amount of time a packet spends idle)

Results

In terms of **delivery time**, both reinforcement learning methods strongly outperform shortest path. This pattern is resilient to changes in network topology, for example the size of the network, the capacity of each node, the average degree, and the type of random network. Q-routing and Deep Q-routing perform comparably.

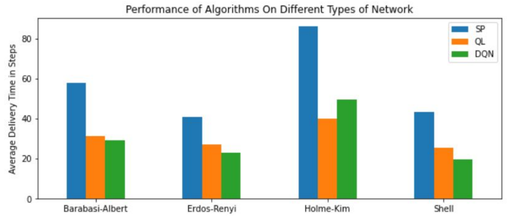


In terms of **network congestion**, this pattern is again repeated.



COMPARISON ON VARIED GRAPH TOPOLOGIES

Q-routing and Deep Q-routing outperform shortest path when used on four different graph topologies. The random graph generation models used were Barabasi-Albert, Erdos-Renyi, Holme-Kim, and Shell. We have observed similar results for the three algorithms' performance when varying other traits such as network size.



Conclusion

- Q-learning is able to deliver packets faster and reduce congestion compared to shortest path
  - Where shortest path cannot complete packet delivery, Q-learning always finishes, demonstrating its reliability under high-stress networks
- Deep Q-learning outperforms Q-learning at low network loads
  - The current simplistic implementation of deep learning delivers packets more slowly than regular Q-learning
  - Training time for a Deep Q-routing algorithm is significantly longer
- Reinforcement learning outperforms shortest path regardless of network topology
  - Deep Q-routing and Q-routing have nearly identical delivery times across network topologies
  - Training times increase with the complexity of the graph structure

Recommendations

- Deep Q-routing, performed with more powerful hardware and with some additional tuning of its learning parameters, has the potential to outperform Q-routing on all network sizes and types
- Potential improvements to the Deep Q-routing model include:
  - Switching to Double Deep Q-learning for better estimation of Q-values
  - Feeding additional inputs into the neural network
    - The current neural network accepts only the current position and destination; more information could lead to more complex routing decisions that take into account entire network dynamics (e.g. queue lengths of neighbor nodes, growth speed of queues)
- Additional lines of investigation include:
  - Does reinforcement learning for packet routing also outperform shortest path in very small networks?
  - Can reinforcement learning be resilient to large fluctuations in the number of packets?

Acknowledgements

This project was supported by the Institute for Pure and Applied Mathematics as well as the Aerospace Corporation, under the mentorship of Tom Merkh (University of California, Los Angeles).