# Experience-driven Networking: A Deep Reinforcement Learning based Approach

Kunal Sisodia

Summer Term 2020

*Abstract* - **The current era we are living in is marked with the evolution of high tech breakthroughs with technology like fifth-generation wireless technologies (5G) which have provided firm grip to the communication networks by increasing the potential and efficiency in terms of connectivity and transmission speed. For all of its advantages, the upcoming technologies are a double-edged sword which are followed by some drawbacks. Apparently, One prime problem is of Traffic Engineering (TE). To counter this a novel approach DRL-TE was developed which was inspired by Deep Neural Networks (DNNs). TE-aware exploration and actor-critic-based prioritized experience replay techniques were proposed as the part of the framework to boost the performance of Deep Reinforcement Learning (DRL) framework. ns-3 was deployed for implementation and extensive testing of this technique which later overshadowed all currently used techniques by offering improved utility, reducing delays, better throughput and thus better efficiency.**

## 1 Introduction

The main objective of the authors was to research and develop a self-reliance framework for the highly advance and dynamic modern communication network. The framework should be capable of controlling the network using its own experience rather instead of some pre-written algorithms. A challenging problem which arises in this domain is to forward the packet from source to destination in the network by minimizing the network congestion and increasing the overall network performance is termed as Traffic Engineering Problem. To counter this several techniques like OSPF, VLB, queuing theory and NUM were deployed but neither of them proves to be the optimal and best solution for this problem. Furthermore with the recent AI breakthrough of AlphaGo by Deepmind which managed to beat a human champion. Consequently it became a motivation to explore the field of DRL and its capabilities to solve some advance complex network problem using deep reinforcement learning. The DRL [2] Approach gave some significant advantages. Firstly this approach was model-free as it was able to solve the task of reinforcement learning by using samples without directly estimating the reward and transition dynamics. Secondly DRL technique was able to deal with highly dynamic time-variant environments. Moreover DRL was capable of handling a sophisticated state space, which is more advantageous over traditional Reinforcement Learning (RL). There were also some drawbacks associated with DRL Approach - Direct application of basic DRL technique such as DQN [2] based DRL did not worked well for TE prob as DQN was only capable to handle control problems with a limited action space. After that Deep Deterministic Policy Gradient (DDPG) [3] was deployed which is an off policy algorithm that means it can learn the value of optimal policy without the need of the agent's action. To learn the policy DDPG initially learns the Q-function which uses an off-policy data and the Bellman Equation. Furthermore, DDPG didn't not work well with Traffic Engineering (TE) problem as it is a continuous problem which means the agent have to find the best possible action for each time interval. After extensive research and testing authors pur-

posed two new techniques for optimizing DDPG approach. TE-aware exploration and actor-critic based prioritized experience replay to optimize the general DRL framework.

# 2 Problem Statement

A challenging problem which arises in this domain is the optimization of the network which is referred to as Traffic Engineering (TE) problem [1]. The overall goal is to avoid congestion in the network and to boost the total utility function [1] for all the communication sessions given by $\sum_{k=1}^{K} U_a(x)$
where utility of a session - $U_a(x) = \left(\frac{x^{1-a}}{1-a}\right)$ from the $\alpha$ fairness model [1]
x is the throughput
K - no of communication session
$\alpha$ - tradeoff between fairness and efficiency
if the $\alpha = 0$ it means no value to fairness and simply measures total throughput. and if $\alpha = 1$ is known as proportional fairness

The $\alpha$ fairness model is generally used for the approach Network Utility Function (NUM).

## 2.1 Network Utility Function

The Approach NUM [4] aims to maximize the overall utility function by allocating resources in the network and thereby providing a solution to the optimization problem. There were certain drawbacks linked with the approach which decreased its efficiency. Firstly the calculation of a complex utility function at the sender with lots of parameters are required like all the resource in the network, All possible routes, set of all possible source-sinks, link usages and the calculation can be done as referenced in [4]. Secondly, a mathematically model was required instead of a model free approach to correctly calculate the end to end delay which limited this approach to non dynamic networks only.

# 3 Deep Reinforcement Learning (DRL)

DRL approach of many fundamental terms like action, state, rewards, Environment policy and value. The state is the current situation in which the agent/actor finds itself. The place or the world in which the agent performs a task is called the environment. The environment consists of the set of rules which processes the actions of the agent and provides the result. Action - actions are all the possibles task performed by the agent in the environment Reward - The reward measures the success or the failure of the task performed by the agent. The feedback is provided by the environment for the actions of an agent for any given state. The results are the new state created and rewards if any

The technological developments have greatly increased the growth in the field of Deep Q Networks which jointly makes use Q learning and deep learning. DQN uses neural network instead of Q learning table to approximate the Q-value function.

The advance DQN [2] also had many pitfalls like - its performance was not optimal with the increasing sets of action. Plus the exploration strategy used by DQN was not that effective. To counter these policy gradients came into existence which tries to learn more in a robust way by evaluating the action to be taken rather than figuring out the value of each action. Furthermore to even increase its efficiency actor and critic model came into lights in which two sub agents learns together. One learns the policy for action called actor and other studies the Q-value for each action and state called a critic.

# 4 Proposed DRL Based Framework

DDPG [1] was chosen as a starting point by the author as it was capable to counter the problem of continuous control which was working fine with a few continuous control tasks but after the experiment done by the author, the results did not offer a decent performance against the TE problems because of two reasons. Firstly, the method given for physical control problems had some issues. Secondly, ignorance of transition samples as it was capable to deal with the uniform sampling method only. Thus to overcome these shortfalls and to give an optimal solution for TE problem the author planned to include an actor and critic based prioritized experience replay. The author describes the prerequisites as [1] which needs to be designed with great care in order to work with the DRL model. At first the

state space which is formed by two important factors namely throughput and delay for each communication session. Secondly, An action space which is a collection of split ratios for the different communication session and can be termed as a solution to the TE problem. Lastly, The reward which is basically the utility function of communication sessions which needs to be maximized to counter the TE problem. This model runs to provide the best action at a time frame by capturing the network state, transition samples and the action to the network.

### 4.1 Algorithm DRL-TE

To overcome these shortfalls of DDPG [3] method and to give an optimal solution for TE problem the author planned to include an actor and critic based prioritized experience replay [1]. To determine an experience driven approach which can fully control a dynamic network the author makes use of prioritized experience replay which samples the data on the basis of the priority assigned for each sample in a time frame. The author combined DQL-based DRL method with actor and critic model along with priority experience replay. The algorithm DRL-TE [1] uses a dual layer loop free connected network for its implementation for actor and critic network. The First layer consists of 64 neurons, leaky rectifier for activation. The Second layer consist of an activation function which guarantees that the sum of output values should be one. critic network model also uses the same configuration network as described above. Here is the brief summary of the DRL-TE algorithm [1].

- The algorithm in its first step assigns random weight to the actor network $\pi(\cdot)$ and the critic network $Q(\cdot)$ with weights $\theta^Q$ and $\theta^\pi$ respectively.
- Target networks $Q'(\cdot)$ and $\pi'(\cdot)$ with same weight as that of the original network were employed to improved the learning stability.
- Calculation of key factors for all the samples.
- Temporal-Difference (TD) error is calculated for training the critic network.
  $\delta_i := y_i - Q\left(\mathbf{s}_i, \mathbf{a}_i\right)$;
  TD error is the difference between y which is the target value for training the critic network and the function Q($s_i$,$a_i$) is the expected return for taking action $a_i$ while in state $s_i$.
- For training of the actor network it is essential to calculate the Q gradient $\nabla_{\theta^\pi} J_i := \nabla_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a})\big|_{\mathbf{s}=\mathbf{s}_i, \mathbf{a}=\pi}\left(\mathbf{s}_i\right) \cdot \nabla_{\theta^\pi} \pi(\mathbf{s})\big|_{\mathbf{s}=\mathbf{s}_i}$
  Actor function is given by $\pi(\mathbf{s})$ and the critic function - $Q(\mathbf{s}, \mathbf{a})$. To find the Q gradient a chain rule is applied to the expected reward J by the similar research done in [4].
- TD error and Q gradient both are combined to calculate the priority of the samples. $p_i$
  $p_i := \varphi \cdot \left(\left|\delta_i\right| + \xi\right) + (1 - \varphi) \cdot \overline{\left|\nabla_{\mathrm{a}} Q\right|}$
  The formula makes uses of factor $\varphi$ which denotes the relative importance of TD Error vs Q Gradients. While $\overline{\left|\nabla_{\mathrm{a}} Q\right|}$ if the average of Q gradients absolute values and $\xi$ used to avoid samples which have negligible samples once.
- weight changes are accumulated for actor network and critic network $Q(\cdot)$
- The accumulated weight change then used to update the actor and critic network. The final updation also happens in the target network and the rate of updation is defined by a factor $\tau$.

## 5 Implementation

For the implementation of Algorithm DRL-TE [1] pseudo code is present but in general there was no public implementation found.

## 6 Performance Evaluation

In the experimental approach ns-3 simulation environment along with tensor flow for actor and critic networks was used to evaluate the algorithm DRL-TE. System configuration [1] was an Intel Quad-core 2.6GHz CPU with 8GB ram. NSFNET, ARPANET and a random typologies were used to evaluate the performance of the algorithm [1]. Traffic demand followed a Poisson process and the scale was set increasing in each run. This experimental approach was then compared with techniques like shortest

path , load balance, network utility maximization and DDPG while keeping all the other factor identical, Some of the conclusions were drawn after evaluation.

- DRL-TE [1] Outperforms in terms of reducing end-to-end delay, maximizing total utility function, pretty good end to end throughput on various topologies when compared with other models like SP, LB, NUM and DDPG.

- Due to better performance result of DRL-TE [1] this technique is considered to be robust against the changes in network parameters like traffic load and topology.

- DRL-TE [1] proved to be the best by beating the DDPG approach as it was quickly able to reach a state with its own experience which provide a solution with higher reward. On the other hand DDPG was seen stuck with low rewarded solution and thus it was not able to counter DRL-TE.

# 7 Discussion

In this paper we saw how DRL based algorithm using an actor and critic network were able to counter a fundamental problem of traffic engineering. In comparison various approach were deployed to help solve the issues of traffic in an advance, modern and a highly dynamic networks. All the given approaches had there predefined rules and structure and mostly were model-based which hampered the performance. The DRL-TE algorithm was a model free approach which managed to train itself using transition samples and could figure how to control the network. This showcased the immense potential of DRL framework. In the current scenario where the scalability of internet connected systems has increased to multiple folds and the risk of cyber attacks are enormous. It would be interesting to see DRL based algorithms taking up challenges in the field of cyber security. As we saw DRL is capable of solving complex, dynamic, and especially high-dimensional problems.

# 8 Conclusion

In the end , authors provided an experience-driven DRL-TE [1] approach with actor and critic networks to counter TE problem. The performance of this approach when evaluated using NSFNET, APRANET topologies and a random topologies. This approach outperforms all the current baselines methods. As a result of the simulation the DRL-TE approach reduces end to end delay, was robust to network changes, offered better throughput and thus maximizing the total utility function.

# References

[1] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. Liu, and D. Yang. "Experience-driven Networking: A Deep Reinforcement Learning based Approach". In: 2018-April (Oct. 2018), pp. 1871–1879.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (Feb. 2015), pp. 529–533. URL: http://dx.doi.org/10.1038/nature14236.

[3] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. "Deterministic Policy Gradient Algorithms". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by E. P. Xing and T. Jebara. Vol. 32. PMLR, 22–24 Jun 2014, pp. 387–395.

[4] S. H. Low and D. E. Lapsley. *Optimization flow control. I. Basic algorithm and convergence*. Vol. 518. 6. 1999, pp. 861–874.