# DeepRoute on Chameleon: Experimenting with Large-scale Reinforcement Learning and SDN on Chameleon Testbed

Bashir Mohammed[1], Mariam Kiran[2] and Nandini Krishnaswamy[1]

***Abstract*— As the numbers of internet users and connected devices continue to multiply, due to big data and Cloud applications, network traffic is growing at an exponential rate. WAN networks, in particular, are witnessing very large traffic spikes cause by large file transfers that last from a few minutes to hours on network links and there is a need to develop innovative ways in which flows can be managed in real-time.**

**In this work, we develop a reinforcement learning approach, in particular Upper-Confidence Algorithm, to learn optimal paths and reroute traffic to improve network utilization. We present throughput and flow diversions using Mininet and demo the technique using Chameleon's Testbed (Bring-Your-Own-Controller [BYOC] functionality). This work is initial implementation towards DeepRoute, which combines Deep reinforcement learning algorithms with SDN controllers to create and route traffic using deployed OpenFlow switches.**

## I. INTRODUCTION

Large scientific workflows involve analyzing, processing and moving large data files from experimental facilities, like the Large Hadron Collider (LHC) at CERN, to data centers across the world. To support these workflows, network engineers build and maintain a dedicated research wide area network (WAN) for scientific needs. These R&E networks see complex networking challenges and demands such as needs of fast, dedicated and secure data delivery, managing packet latency and ensuring performance reliability.

Continually upgrading network links can be an expensive solution. Network operators make active decisions to optimize available link bandwidth, using traffic engineering or network reconfiguration which are expensive engineering decisions [1]. Furthermore, failure to maintain links can lead to packet loss, which is a major issue for end-to-end performance. Current efforts are exploring machine learning approaches [2] and optimization techniques to improve current network utilization [3]. This practice of capping utilization is not scalable and prevents optimal use of available bandwidth across all links and resources - network link paths usually range from 10GB/s to 400GB/s in large WANs [4]. Network planning is always looking for efficient ways to manage these links.

In this paper we build reinforcement learning to optimize flow allocation over multiple paths. Working towards developing DeepRoute, a deep reinforcement learning (DRL), our objective is to develop a number of algorithms where an agent can learn and control incoming flows across multiple paths. DRL has demonstrated great success in managing computational resources and playing video games [5]. By directly interacting with its environment and getting feedback in the form of rewards, an agent can learn optimal strategies. Route optimization has been investigating in much efforts as a graph-based problem [6]. Our approach departs from graph-based problems and attempts to solve this per path selection at sources. Making decisions locally, our aim is to investigate if global optimization can be achieved. We demonstrate results in Mininet simulations.

[1]Scientific Data Management, Lawrence Berkeley National Lab, Berkely, CA bmohammed@lbl.gov, nkrishnaswamy@lbl.gov

[2] Energy Sciences Network (ESnet), Lawrence Berkeley National Lab, Berkeley, CA mkiran@lbl.gov

## II. METHODOLOGY

We are developing DeepRoute, an algorithm based on Deep Q-Networks (DQN), for network scenarios [7]. In this work, we develop an initial version of this using Upper Confidence Bound calculations for flow allocation. We demonstrate the results using Mininet.

**Mininet:** The mininet deployment allows us to develop, share, and experiment with OpenFlow and Software-Defined Networking (SDN). The performance of DeepRoute is tested by generating a collection of long and short-lived flows to mimic file transfers between a Source and Destination. We record the performance of the network links using Wireshark and mininet outputs.

## III. EXPERIMENT

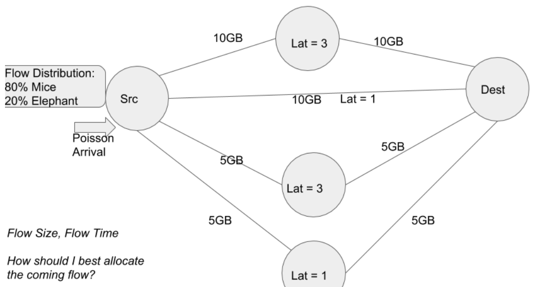
Figure 1a shows the diagram and mininet topology of the experiment built. We use iperf3 to start and measure the network throughput between the source (Src) and destination (Dest). Performance of the network link is measured using latency, jitter, packet loss

**DeepRoute Algorithm:** Inspired by the Multi-armed Bandit problem, the problem is to investigate adaptive routing to minimize delays in the network. Based on a number of distributions over paths $P = (R_1, R_2, R_n)$ and a set of rewards. The objective is to maximize the sum of collected rewards.

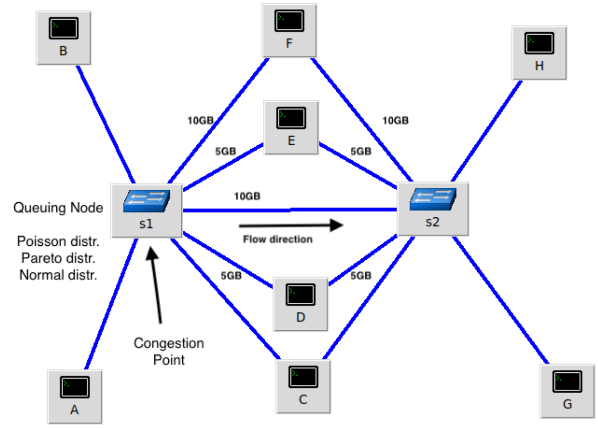
**Training and Test Data:** We ran 100 flows over mininet and collected the respective loss on all the 4 paths. This is then associated as a negative reward which the agent uses to make decisions in the future. Figure 2 shows the path selection results when making decisions randomly and RL approaches.

## IV. RESULTS

Figure 3 shows the throughput recorded across all paths. The RL approach (Figure 2) uses the packet loss as a measure of whether it was a good or bad transfer. By trying out all possibilities, it then learns that path 1 is the best route to

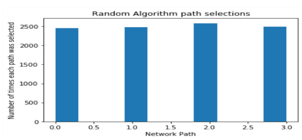



(a) Diagram Network Topology.

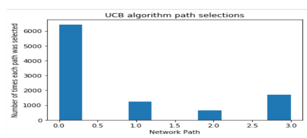



(b) Equivalent Mininet topology.

Fig. 1: Setting up network topologies.

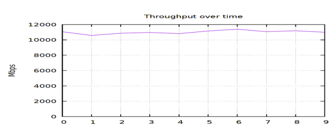



(a) Random selection.

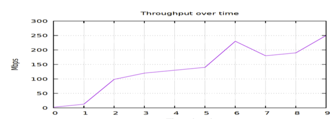



(b) UCB (RL approach) selection.

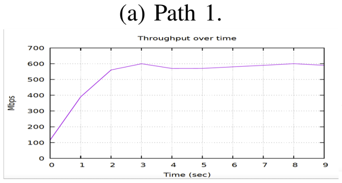
Fig. 2: Selection of paths made on Random and RL-based approaches.



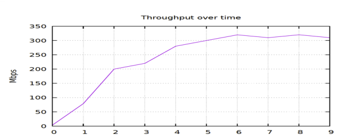

(a) Path 1.




(b) Path 2.




(c) Path 3.




(d) Path 4.

Fig. 3: Throughput across the paths during Testing.

take. Therefore, during the test phase, this is the path which is chosen most and the throughput increases. We still see some random traffic on other paths.

**Extending on Chameleon:**, we extend this by using the Bring-Your-Own-Controller (BYOC) functionality. This enables us to create and control our own exclusive, isolated OpenFlow network switches and attach DeepRoute decision-making.

## V. CONCLUSION

Since, this is still in test phases, we are developing the RL approaches using Mininet and network testbed, and not production networks yet. In the future, we can introduce delay and latency and see how it affects the traffic flows to alter their decisions. In the future, we will extend this to Chameleon Cloud testbed using dynamically created VLANs and external connections.

The need for efficient management of WAN topologies presents new research opportunities and our work in DRL has a number of implications in the field of network traffic research. In our demo, we will combine machine learning and controller decisions with the goals of improving bandwidth utilization and minimizing flow completion time over all possible paths.


## ACKNOWLEDGMENT

We would like to thank Paul Ruth of the Chameleon project for his technical support. This work is funded DOE ASCR Early Career Grant Deep Learning: FP00006145.


## REFERENCES


[1] A. Pathak, M. Zhang, Y. C. Hu, R. Mahajan, and D. Maltz, “Latency inflation with mpls-based traffic engineering,” in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC ’11. New York, NY, USA: ACM, 2011, pp. 463–472.

[2] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, “A comprehensive survey on machine learning for networking: evolution, applications and research opportunities,” *Internet Services and Applications*, vol. 9, no. 1, p. 16, Jun 2018.

[3] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, “B4: Experience with a globally-deployed software defined wan,” *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, Aug. 2013.

[4] K. Roberts, Q. Zhuge, I. Monga, S. Gareau, and C. Laperle, “Beyond 100 gb/s: capacity, flexibility, and network optimization,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 4, 2017.

[5] S. D. Holcomb, W. K. Porter, S. V. Ault, G. Mao, and J. Wang, “Overview on deepmind and its alphago zero ai,” in *ICBDE*. New York, NY, USA: ACM, 2018, pp. 67–71. [Online]. Available: http://doi.acm.org/10.1145/3206157.3206174

[6] F. Geyer and G. Carle, “Learning and generating distributed routing protocols using graph-based deep learning,” ser. Big-DAMA ’18. New York, NY, USA: ACM, 2018, pp. 40–45.

[7] T. Zahavy, N. B. Zrihem, and S. Mannor, “Graying the black box: Understanding dqns,” in *ICML*, 2016, pp. 1899–1908. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045390.3045591