A Machine Learning Approach for Traffic Flow Provisioning in Software Defined Networks

Subham Kumar, Gaurang Bansal, Virendra Singh Shekhawat

Birla Institute of Technology and Science, Pilani (BITS Pilani)

Abstract

With the recent surge of machine learning and artificial intelligence, many research groups are applying these techniques to control, manage, and operate networks. Software Defined Networks (SDN) transform the distributed and hardware-centric legacy network into an integrated and dynamic network that provides a comprehensive solution for managing the network efficiently and effectively. The network-wide knowledge provided by SDN can be leveraged for efficient traffic routing in the network. In this work, we explore and illustrate the applicability of machine learning algorithms for selecting the least congested route for routing traffic in a SDN enabled network. The proposed method of route selection provides a list of possible routes based on the network statistics provided by the SDN controller dynamically. The proposed method is implemented and tested in Mininet using Ryu controller.

Motivation

Machine Learning approaches to address network issues:
1) Feature Traffic Forecasting (FTF)
2) Time Traffic Forecasting (TTF)

FTF utilizes network characteristics while TTF utilizes network topology. Machine learning provides high accuracies on analysis of network traffic, they have been deployed extensively in Intrusion Detection Systems (IDS) and circuit switched networks. The major drawback of machine learning is that these techniques cannot be deployed in real time easily. The same goes for reinforcement learning as well.

Contributions

1. We propose an agile and efficient machine learning based path selection approach for traffic flow provisioning in software defined networks.

2. This paper provides a comprehensive comparison between ML models and the conventional routing techniques used today, in terms of time and accuracy.

3. Through extensive experiments we demonstrate substantial improvement of Machine learning methods adapted to address this issue.

Contact

Gaurang Bansal
Webpage: https://gaurang-bansal.github.io/
h20140128@pilani.bits-pilani.ac.in

Methodology

The setup contains two modules:
1. The training module learns from the paths provisioned in the recent past for the given state of the network.
2. In the deployment module, the controller queries the module at specified interval of time for the best possible path based on the current network state. It then provisions the new paths based on the information received.

K Means

K-Means is used for clustering similar states of the network based on Euclidean distance between them. The network state is represented through weights which are assigned to each of the links.

$$\rho(x_i, x_{i'}) = \frac{\sum_j (x_{ij} - \overline{x}_i)(x_{i'j} - \overline{x}_{i'})}{\sqrt{\sum_j (x_{ij} - \overline{x}_i)^2 \sum_j (x_{i'j} - \overline{x}_{i'})^2}}$$

In the training module, every state of the network along with its corresponding best path is read, and appropriately put into existing clusters, or the clusters are rearranged to form a new set of clusters.

Cosine Similarity

Cosine similarity finds the similarity between two points based on the angle between the two vectors which connect the origin to those points respectively.

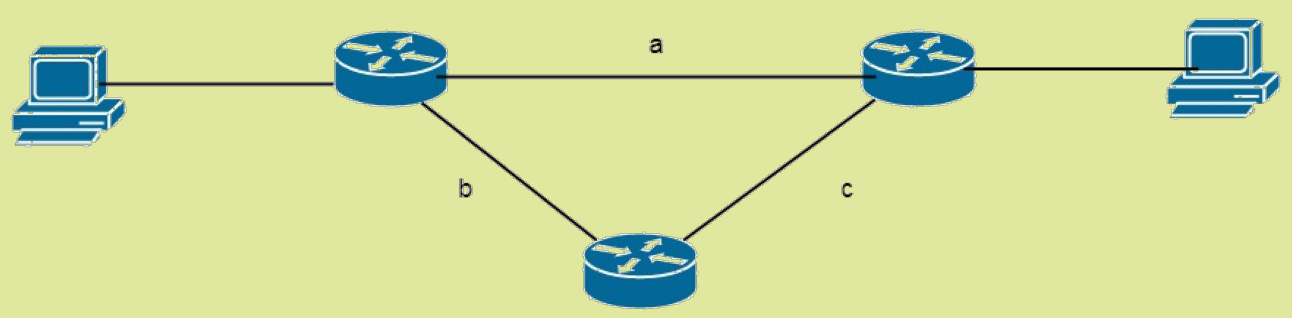
$$\cos(\mathbf{x_{ij}}, \mathbf{x_{i'j}}) = \frac{\sum_{i=1}^{n} \mathbf{x_{ij_i} x_{i'j_i}}}{\sqrt{\sum_{i=1}^{n} (\mathbf{x_{ij_i}})^2} \sqrt{\sum_{i=1}^{n} (\mathbf{x_{i'j_i}})^2}}$$

The network states are clustered in a similar fashion as in K-Means, but using cosine similarity as the clustering parameter instead of Euclidean distance

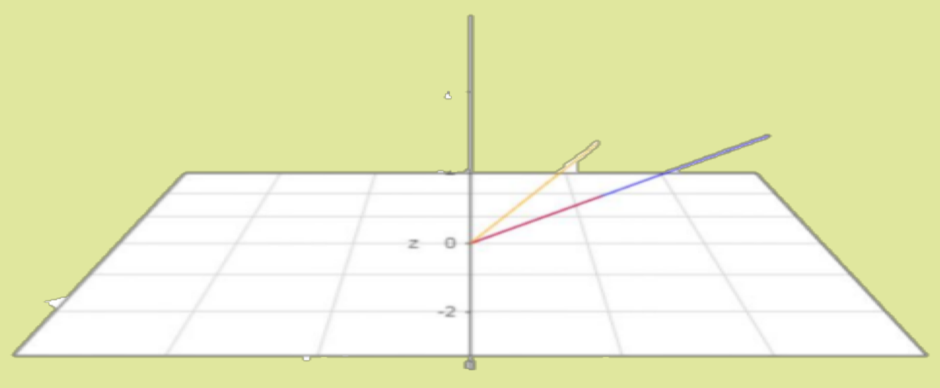
.

Comparative Analysis

Taking 2 clusters in the training module, with centers (4,4,4) and (3,1,1) respectively. The best path between the two host machines shown above in the first cluster is "a", while for the second cluster is "b → c".

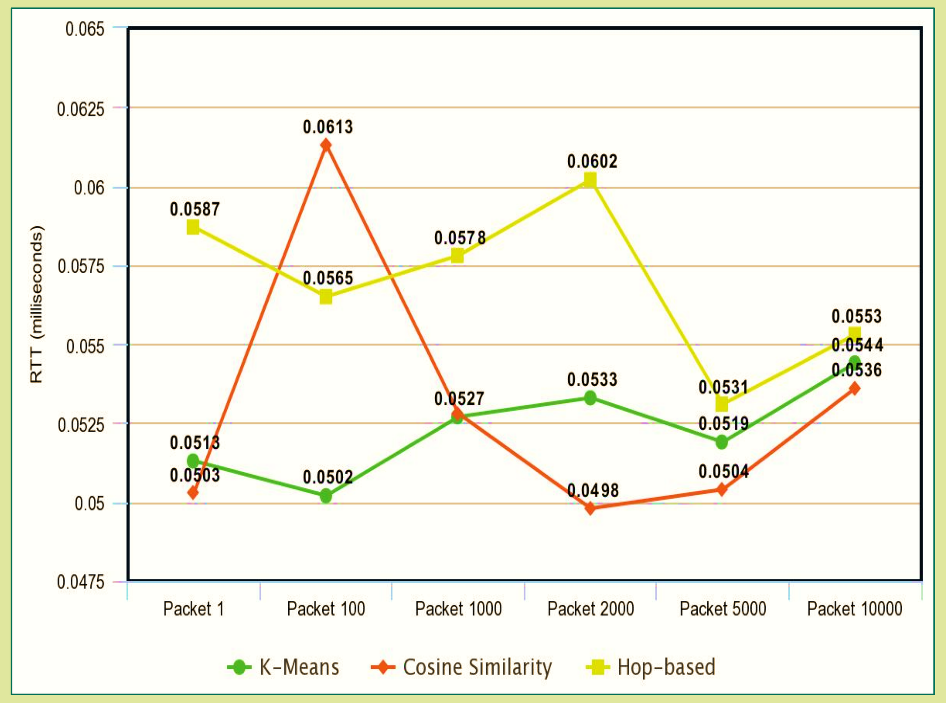


A test path, (2,2,2) has to be mapped to either of these them in the deployment module. (2,2,2) should be mapped to (4,4,4) because it is similar in nature to that state, giving similar best paths. However Euclidean Distance will return (3,1,1) as the closest, rather than (4,4,4). But on the other hand cosine similarity will give the highest value for (4,4,4) rather than (3,1,1) and will return (4,4,4) as the result.



Hence cosine similarity will perform better in such scenarios compared to K-Means.

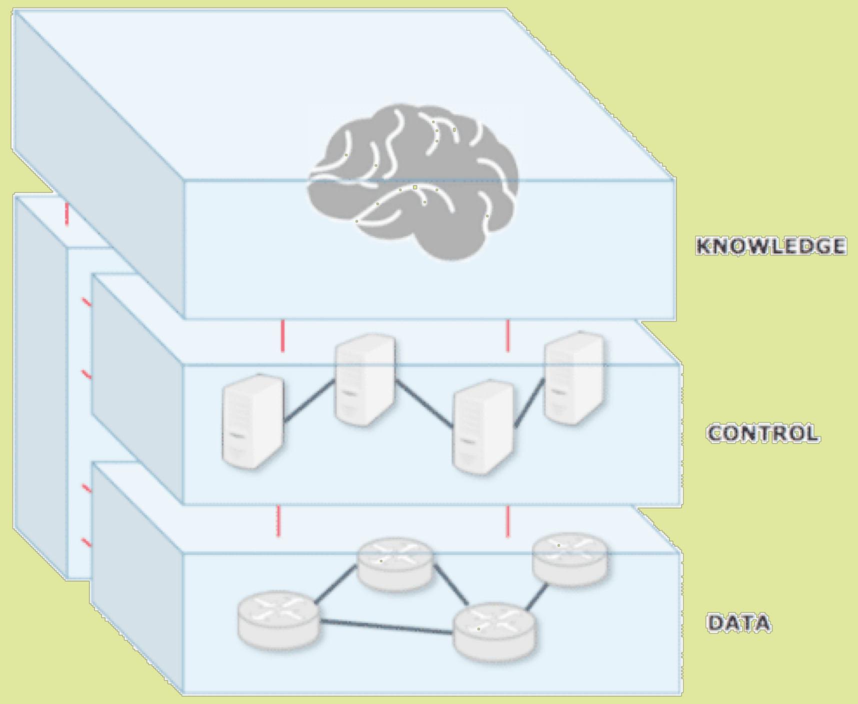
Results & Inferences



Paths selected by the K-means and Cosine similarity techniques provides lower RTT paths as compared to the conventional hop-based method. Cosine similarity gives higher RTT paths initially. But as time progresses and more number of packets get transmitted, Cosine similarity method selects better paths in terms of RTT as compared to K-means as well as hop based routing method.

Performance comparison of K-Means and Cosine Similarity to the best possible (optimal or ground truth) solution in terms of the average RTT, minimum RTT, maximum RTT and the mean deviation of the RTT. When we consider network as a whole for a considerable amount of time, then it can be seen that both methods select best possible paths or very close to that.

Conclusion & Future Scope



This paper presents and demonstrates the feasibility of machine learning to the field of networking, specifically traffic control. We have created a real time system, an algorithm which can be deployed in real time to find out the most optimal path and then push these changes to the switches. Further clustering algorithms can be tried and deployed, wit modifications to suit networking specifically

One of future scope of paper is to design a Knowledge plane or machine learning model that can be integrated with SDN model for flow prediction in real time with high accuracy.

References

1. Clark, D., et al. "A knowledge plane for the internet," Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. ACM, 2003.
2. Kreutz, D., et al. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE vol. 103.1, pp. 14-76, 2015.
3. Clemm, A., et al. "DNA: An SDN framework for distributed network analytics," Integrated Network Management (IM), IFIP/IEEE International Symposium on. IEEE, 2015.
4. Derbel, H., Agoulmine, N., and Salan, M. "ANEMA: Autonomic network management architecture to support self-configuration and self-optimization in IP networks," Computer Networks vol. 53.3, pp. 418-430, 2009
5. Oliner, A., et at. "Advances and challenges in log analysis." Communications of the ACM, vol. 55.2, pp. 55-61, 2012