

Machine Learning in the Control Plane of a Software Defined Network

Sergio Vidal-Beltrán¹, Brenda Ramírez-Gómez², José Luis López-Bonilla³, P. Siva Kota Reddy⁴
and A. Nagarjun⁵

¹⁻⁴Instituto Politécnico Nacional, Escuela Superior de Ingeniería Mecánica y Eléctrica, Zacatenco, Ciudad de México, Mexico

⁵JSS Science and Technology University, JSS TI Campus, Mysuru-570 006, India

Email: nagarjun@jssstuniv.in

Abstract—Transmission rate, packet delay, and packet loss can be improved by using machine learning in a software-defined network. The results obtained in this work show that machine learning implemented in the control plane improves the performance of a network compared to OSPF routing algorithms or static routes.

Index Terms— Machine learning, software defined networks, routing algorithms, OSPF, supervised learning.

I. INTRODUCTION

The use of machine learning and artificial intelligence has increased in many areas of science and engineering, including wireless communications [1-3], fiber optic communications [4-9], and fiber optic sensor applications. [9-13], Machine learning applied to network interconnectivity can be efficient because its capabilities include data prediction and classification. If sufficient data is provided to a machine learning network, it can learn from the state of the network using its characteristics and thus be able to predict its behavior by creating models and algorithms that modify its own codes according to the new parameters.

II. MOTIVATION

In this work, we hypothesize that it is possible to improve the performance of a data network by using machine learning in the control plane of a software-defined network. Routing decisions are not made through traditional routing algorithms such as shortest path (OSPF) but are made using a neural network that is trained with data from the behavior of the network, this training data is used by the neural network to later make predictions about which is the best route between two points in a data network. This work presents comparisons between the performance of a network that uses traditional routing algorithms such as static routes or OSPF against the performance using supervised machine learning.

III. PROBLEM FORMULATION AND SIMULATION SCENARIO

A software-defined network consists of a data plane, a control plane, and an application plane. The control plane is responsible for making routing decisions and subsequently communicating them to the data plane. The proposal of this work adds a neural network module in the SDN controller (control plane), which has the

following objectives:

- Network topology discovery
- Monitoring of performance parameters
- Collect a training data set
- Train the neural network with the training data
- Acquire learning skills (once training has concluded).
- Predict the best route (output) that avoids congestion.

Figure 1 shows the network scenario that was used to carry out this work.

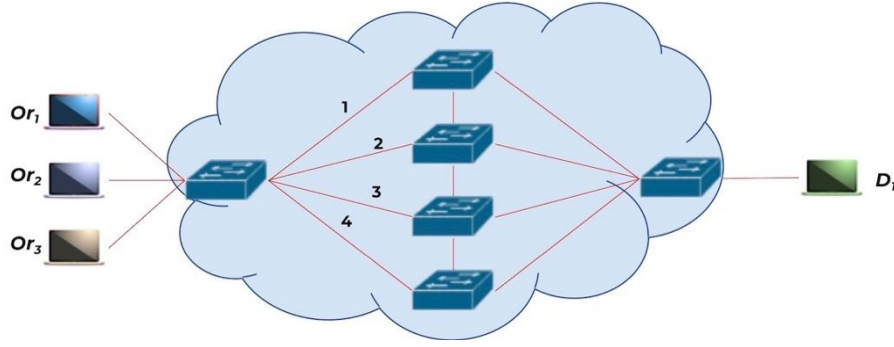


Figure 1. General Network Scenario

The figure 1 shows the network architecture of the simulation scenario, which consists of n source nodes indicated as $Or_1, Or_2, \dots, Or_n$, m destination nodes denoted as $D_1, D_2, \dots, D_m$, with which there are f data flows ($m \times n$) and there are R available connections between any source-destination pair; in addition to 6 Open Flow switches that provide connectivity to the network. The available routes (k) are defined by $k = R^n$, so we have a topology with the following characteristics: $n = 3, m = 1, R = 16, k = 4096$.

To evaluate the performance of the proposed network, the parameters shown in Table 1 were used.

TABLE I. SIMULATION PARAMETERS

Parameter	Value
Simulator	GNS3, Mininet
SDN protocol	Open Flow
Traffic generator	Iperf
Data traffic	UDP
Band with per link	250Mbps
Data rate	70Mbps → 150Mbps
Buffer size of each Open Flow switch	200 packets
Routing algorithm	Dijkstra (OSPF)
Simulation period	4 hrs
Monitoring period	3, 5, 10 segs
Origin nodes	3
Destination nodes	1
Available routes between each origin-destination pair	16

Between each origin-destination pair, UDP traffic is generated, varying the speed from 70 Mbps to 150 Mbps; To verify the performance of the network, simulations were carried out with a duration of 4 hours and

monitoring data were collected at different time periods (3, 5 and 10 seconds). Figure 2 shows the UDP flow between the origin-destination pair and the performance variables to be evaluated.

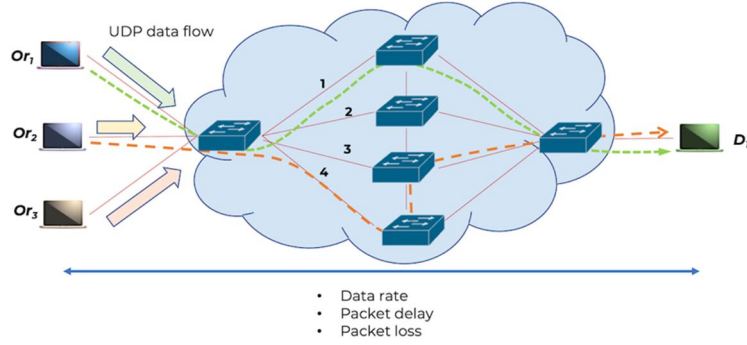


Figure 2. UDP traffic between origin-destination pair

IV. MONITORING AND TRAINING DATA

Once the network described in the previous section was configured, we proceeded to monitor and collect data that will serve as training data for the machine learning module. The following data was collected:

- Network congestion
- Transmission rates of all data streams, and
- Possible routes between origin-destination.

With the data collected, table 2 is built:

TABLE II. TRAINING DATA SET

Route k	Data rate (Mbps)			Route			Congestion
	$Or_1 \rightarrow D_1$	$Or_2 \rightarrow D_1$	$Or_3 \rightarrow D_1$	$Or_1 \rightarrow D_1$	$Or_2 \rightarrow D_1$	$Or_3 \rightarrow D_1$	
1	70	75	90	0	0	0	0.92
2	70	75	90	0	0	1	0.73
	70	75	90	0	0	2	0.64
...	...	...	...	...	...	...	...
4095	70	75	90	9	9	8	0.53
4096	70	75	90	9	9	9	0.84

The selected route is defined by the variable k then the transmission rate between each origin-destination pair $Or_n \rightarrow D_m$ is shown followed by which route selected, finally, the variable C represents the network congestion, which takes values of $\{0, 1\}$, $C = 1$ if there is one or more Open Flow switches with queuing greater than 80% in the selected path, $C = 0$ in any other case.

V. NEURAL NETWORK TRAINING

A total of 100,000 data were collected, which will serve as a training set, this set was divided into two:

- Training data
- Validation data

For the training data, 80% of the total data was selected, while the remaining 20% was used for the validation data. The first ones are responsible, as their name suggests, for the neural network to be trained (learn), and the validation data function as a control group, to validate the learning precision, which usually must be equal to or greater than 0.8. (80% accuracy) [14]

VI. NEURAL NETWORK CONFIGURATION

To select the best route, the SDN network routing module used machine learning based on neural networks. Figure 3 shows the neural network architecture used in this work, which is composed of 7 input nodes, three hidden layers and one output node. The number of neurons in the hidden layers varied according to the desired

precision of the neural network. Of the amount of data collected, 80% was used to train the model and the remaining 20% was used as validation data.

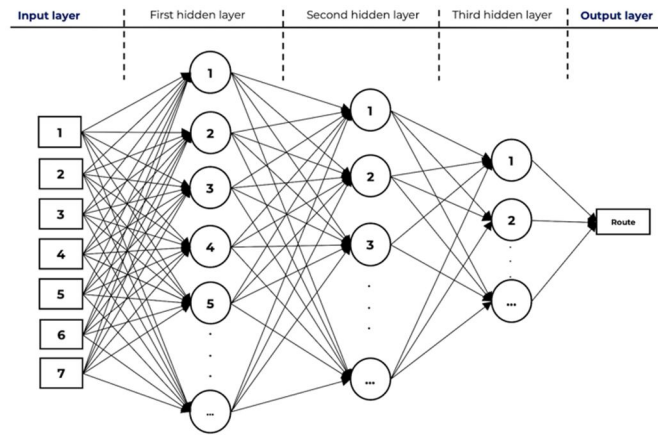


Figure 3. Neural network architecture

The technical specifications used by the neural network appear in Table III.

TABLE III. NEURAL NETWORK CONFIGURATION PARAMETERS

Parameter	Value
Hidden layers	3
Neurons per layer	512, 128, 64
Total data set	100,000
Training data set (80%)	80,000
Validation data set (20%)	20,000
Output nodes	1
Trained model accuracy	87 %
Optimization algorithm	Adam

Open Flow switches constantly monitor queuing on interfaces to determine if link congestion exists, and if so, activate route reconfiguration procedures. If congestion C is greater than or equal to 80% (0.8), the new route selection process begins. Considering Table 1, the congestion for route 1 is 0.92, with which the network selects a new route with less congestion and finds it on route 4095 which has a congestion of 0.53.

VII. RESULTS AND DISCUSSION

Once the testing and simulation scenario was configured, the data was collected and the neural network was trained, the performance tests were carried out, which were contrasted against static routing and dynamic routing (OSPF). To evaluate the performance of the network, the following parameters were selected.

- Average transmission rate.
- Packet delay.
- Packet loss.

The mentioned parameters were monitored every 3, 5 and 10 seconds.

VIII. AVERAGE TRANSMISSION RATE

The amount of data injected into the network was configured to start at 80 Mbps up to 150 Mbps.

Figure 4 shows the average transmission rate for static routing, dynamic routing and the proposed algorithm with Machine Learning. From Figure 4 it is observed that when the test starts with 80 Mbps the three algorithms present a similar performance, since the injected data is smaller compared to the capacity of the link, as the load

increases, different behaviors are observed. For static routing, predetermined routes are configured, which are maintained regardless of whether congestion occurs or not. This algorithm reaches its maximum performance around 100 Mbps, and subsequently declines as the amount of data injected into the network increases; This is due to the fact that after 100 Mbps the links begin to saturate and the static algorithm cannot react to this situation, which produces a progressive decrease in the transmission rate obtained.

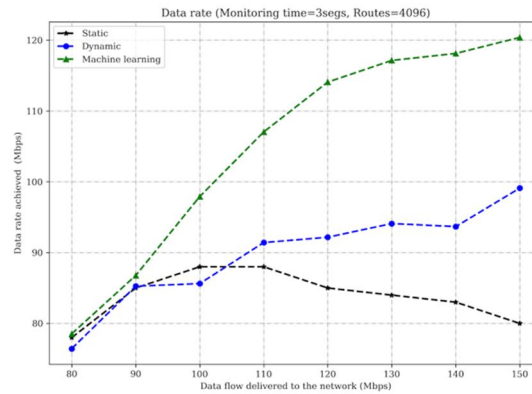


Figure 4. Average transmission rate of the three algorithms

Dynamic routing makes a selection of routes according to the conditions at that moment and monitors the performance of the network so that if the routes are congested, the route selection can be changed. Dynamic routing has better performance than static routing and reacts better to congestion since the algorithm monitors the link state constantly. Between 80 Mbps and 100 Mbps the algorithm has a performance that closely follows the injected traffic, however, after at 110 Mbps, even when performance increases, it does so slowly, as congestion causes performance to decrease. Finally, the performance graph using Machine Learning shows that it has better performance compared to its predecessors (static and dynamic), when 150 Mbps traffic is injected, routing with Machine Learning has a performance of 120 Mbps, while Dynamic routing shows approximately 100 Mbps and static routing drops to almost 80 Mbps.

In general terms, the results shown indicate that the network is offered more traffic than it can transmit, which causes the phenomenon of network congestion. Associated with the transmission rate, the packet delay is a parameter that is necessary to know to evaluate the performance of the network under analysis. In the following section, the results for the packet delay are presented.

IX. PACKET DELAY

As the amount of information injected into the network increases and the data is not processed and transmitted with the same speed with which it arrives, congestion occurs in the network, which causes a delay in the delivery time of the data to the receiver. The static routing algorithm, having predetermined routes, does not react to congestion so the delay will be greater than its dynamic routing and Machine Learning counterparts. Figure 5 shows the behavior of the three routing algorithms.

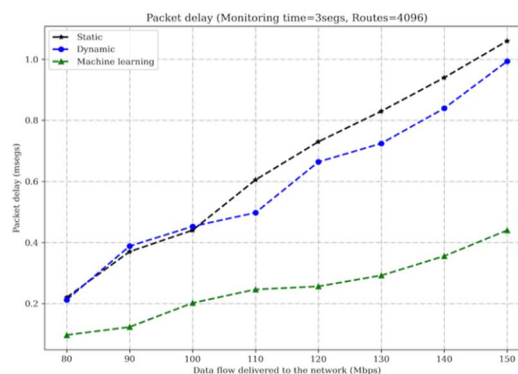


Figure 5. Packet delay

Figure 5 shows that static routing presents the greatest delay in packet delivery, maintaining an almost linear relationship as the data flow injected into the network increases. Dynamic routing has better performance since it can change the routes derived from the constant monitoring it performs; Routing based on Machine Learning has the shortest packet delay time, having approximately 0.44 m sec at 150 Mbps, while dynamic routing has a delay of almost 1 msec.

X. PACKET LOSS

As congestion increases, packet loss may occur in the network, since, since the incoming packets cannot be processed, the network elements discard them. From Figure 6, it can be seen that static and dynamic routing present packet loss from 80 Mbps, while packet loss for routing based on Machine Learning starts from 100 Mbps. Considering 150 Mbps, routing with machine learning had 12% packet loss, while static and dynamic routing had 53% and 36% respectively.

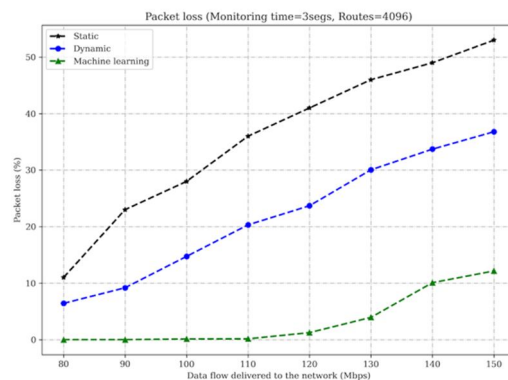


Figure 6. Packet Loss

XI. CONCLUSIONS

At the beginning of this work, it was hypothesized that it would be possible to improve network routing decisions through the use of machine learning in the control plane of a software-defined network. To test this hypothesis, three performance parameters were evaluated, which are: Transmission rate, packet delay and packet loss. Considering the case when monitoring every 3 seconds, the transmission rate with routing with Machine Learning improves by 18% compared to dynamic routing and by 50% compared to static routing. Regarding packet delay, there is an improvement of 52% and 58% in relation to dynamic and static routing. Finally, regarding packet loss, routing decisions based on Machine Learning improve by 60% compared to dynamic routing and by almost 77% considering static routing.

Due to the above, the results obtained in this paper show that routing based on Machine Learning improves routing decisions in the three performance parameters evaluated, so the working hypothesis posed at the beginning of the work is correct.

ACKNOWLEDGEMENTS

The authors wish to express their gratitude to Instituto Politécnico Nacional (IPN) for the support to conduct this study under project SIP 20230319 and Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) of México.

REFERENCES

- [1] Osamy, W., Khedr, A.M., Salim, A., Al Ali, A.I., El-Sawy, A.A.: Coverage, deployment and localization challenges in wireless sensor networks based on artificial intelligence techniques: A review. *IEEE Access* 10, 30232–30257 (2022)
- [2] Vidal-Beltran, S., Martínez-Piñón, F., López-Bonilla, J.L.: Supervised learning applied to the decoding of SCMA codewords. *IEEE Lat. Am. Trans.* 17(11), 1843–1848 (2019)
- [3] Vidal-Beltrán, S.; López-Bonilla, J.L. Improving Spectral Efficiency in the SCMA Uplink Channel. *Mathematics*, 9(6), 651(2021). <https://doi.org/10.3390/math9060651>

- [4] Zhang, L., Li, X., Tang, Y., Xin, J., Huang, S.: A survey on QoT prediction using machine learning in optical networks. *Opt. Fiber Technol.* 68, 102804 (2022)
- [5] Wang, D., Zhang, M.: Artificial intelligence in optical communications: from machine learning to deep learning. *Front. Commun. Netw.* 2, 656786 (2021)
- [6] Liang, M., Chu, L.: Research on channel compensation technology of DP-32QAM optical communication system based on DPC-KNN. In: *Asia Communications and Photonics Conference* (2021)
- [7] Lau, A.P.T., Khan, F.N.: *Machine Learning for Future Fiber-Optic Communication Systems*. Academic Press, New York (2022)
- [8] Wong, E., Mondal, S., Ruan, L.: Machine learning enhanced nextgeneration optical access networks—challenges and emerging solutions [Invited Tutorial]. *J. Opt. Commun. Netw.* 15(2), A49–A62 (2023)
- [9] Nevin, J.W., Nallaperuma, S., Shevchenko, N.A., Li, X., Faruk, M.S., Savory, S.J.: Machine learning for optical fiber communication systems: An introduction and overview. *APL Photonics* 6(12), 121101 (2021)
- [10] Ibáñez-Camarillo, D., Martínez-Piñón, F., Márquez-Olivera, M.V., Hernández-Herrera, V.G. and Vidal-Beltrán, S. Single-mode optical fibre digital decoder based on polarization using a K-nearest neighbour algorithm. *Electron. Lett.*, 59(18): e12921.(2023). <https://doi.org/10.1049/ell2.12921>
- [11] Venketeswaran, A., Lalam, N., Wuenschell, J., Ohodnicki Jr, P.R., Badar, M., Chen, K.P., Lu, P., Duan, Y., Chorpening, B., Buric, M.: Recent advances in machine learning for fiber optic sensor applications. *Adv. Intell. Syst.* 4(1), 2100067 (2022)
- [12] Keser, S., Hayber, S.E. Fiber optic tactile sensor for surface roughness recognition by machine learning algorithms. *Sens. Actuators A* 332, 113071 (2021)
- [13] Huang, H., Lin, J., Wu, L., Fang, B., Wen, Z., Sun, F.: Machine learningbased multi-modal information perception for soft robotic hands. *Tsinghua Sci. Technol.* 25(2), 255–269 (2019)
- [14] Vidal-Beltran, S., Martínez-Piñón, F., López-Bonilla, J.L., Yalja-Montiel, Jesús: Gradient Descent Optimization Algorithms for Decoding SCMA Signals. 20(1), 2150002 (2021).