Literature Review

Kyle Rumpza

rumpz020@umn.edu

1 Introduction

As the internet of things (IOT) grows rapidly our networks are becoming more complex and expansive, requiring more intelligent solutions to meet future networking demands. Software defined networks (SDN) completely changed the traditional understanding of network architecture by separating the control and data plane. This allows SDNs to have a logically centralized architecture able to dynamically implement different algorithms and get network state information (NSI). We also have been seeing a rapid rise in machine learning being implemented within SDNs to make systems more intelligent. In this survey are a collection of 8 papers published in Springer, and IEEE between 1989 and 2024. The survey conducted here will analyze papers and give a brief overview of their content. The research papers are organized based on their machine learning techniques used, routing algorithms explored and evaluated, the way in which testing was performed, performance improvements seen from testing, and the applications of traffic engineering using ML-techniques in current and future networking applications. This paper will provide a comprehensive review of the literature on this topic. This review is useful for those working on ML-techniques in SDNs that want to improve different evaluation metrics associated with communication networks.

2 Related Work

[1] Amin et. al have discussed one of the ways to optimize routing in SDN, so they addressed ML techniques divided into three categories supervised, unsupervised, and reinforcement learning. The research papers published between 2005 and 2021 are considered in the survey.

[2] Boutaba et. al provided brief introduction to ML techniques, engineering techniques, approaches and methods for data gathering in network traffic, overview of ML techniques in routing, traffic classification, QoS, anomaly detection, and intrusion detection. They also focus on the secure learning support, online learning and architectural design of systems so ML can be easily used. Survey covers 500 studies.

[3] Akin et. al Analyzes the use of existing routing algorithms in three different categories, ones that calculate static link costs, one which calculates dynamic link costs, and one that tries to evenly distribute load balance and minimize interference. It also talks about the issue of extracting network state information in terms of overhead computational cost, and inconsistencies created due to delay.

[4] Faezi et. al analyzes a wide range of surveys on the subject and extracts the performance evaluation, experiment

environment, and solution category from these surveys. Offers very useful metrics of how ML-techniques improve SDNs and how.

[5] Mammeri et. al Comprehensively analyze reinforcement learning approaches for routing for all types of networks. Provides a very good overview of the of the evolution of these specific ML-techniques and its application in communication networks

[6] Mendiola et. al survey approaches for TE, indirectly mentions applications in routing of SDNs

[7]Nunes et. al highly cited, diverse alternatives discussed, future SDN application trends

[8] Xie et. al Presents a comprehensive detail of the ML techniques, architecture, and working of SDNs. Different types of ML algorithms are explained and described in terms of QoS, security, resource management, and traffic classification.

3 SDN Background

In conventional networks there was a tight bond between the control and data plane but in software defined networks these planes are separated. The reason for this is that SDNs can overcome issues of flexibility and can update routing functions by exchanging control logic. SDNs are logically centralized usually with a remote controller that obtains a global view of the network by using topology discovery protocols to get accurate link-state information from switches. SDNs abstract the network and allows applications to interact with it. This centralized architecture provides faster overview of the network with smoother programmability and updates but requires much more carefully managed overhead control. It also means that SDNs make routing decisions per flow rather than per-hop. The advantages of SDNs make it very useful for future network programming and our network topologies become more dynamic.

Routing can now be parameterized based on types of optimal routing, cost functions, or resource usage to allow computer science algorithms to be directly translated instead of distributed though switches. Two prominent SDN architectures exist for forwarding packets, OpenFlow and ForCES. Because SDNs and OpenFlow use flow-based forwarding, traditional hardware switches (like ASIC switches) which aren't set up for this may see long delays. Nunes et. al brings up that solutions like installing general purpose CPUs can reduce packet delay for ASIC switches by 20% [7]. Some challenges that SDNs face also include memory limitations for forwarding devices as many switches only support a limited number of forwarding rules. Nunes et. al lists a number of algorithms like Devoflow which can reduce table size, or partitions rules based on hierarchy and policy [7]. Fortunately many commercial switches now have support for OpenFlow API. However the memory and computational overhead from the centralized controller are a major factor to consider when implementing these networks. Amin et. al places a lot of emphasis on "deciding how much control should be delegated to the controller to avoid bottlenecks" [1]. SDNs are seeing deployment for various services including enterprise networks, data centers, WANs, optical networks, vehicular networks, and mobile adhoc networks.

4 ML Techniques Classifications

To be put simply machine learning is the branch of AI that enables the system to learn automatically. This means that ML systems can male decisions and identify different patterns on their own. Boutaba et. al presents a comprehensive survey which looks all the way back to the 80s to analyze the history of ML algorithm development. In the 1980s Bayesian networks arose as a directed acyclic graph representation was used to show a link between conditions and probability. Then the end of the 80s saw the creation of the Q-learning algorithm, which is a model free learning technique that converges optimum action-values with probability. The 1990s was the last time so much attention was given to neural networks and many bio-inspired optimization algorithms like genetic algorithms and particle swarm optimization received increased attention [2]. These surveys divided their routing algorithms into several types. Both Amin et. al, Mameri et.al, and Xie et al. surveyed ML techniques for route optimization based on three categories; supervised learning, unsupervised learning, and reinforcement learning [1][5][8]. For supervised learning data is fed into the model and cross validated to ensure correctness. This is usually a form of informed search. Unsupervised learning seeks patterns among an unlabelled dataset and is usually a form of uninformed search. The last category reinforcement learning teaches an agent to make local decisions and take actions based on memory. The most common algorithm for this is Q-learning, and deep Q-learning. These surveys focus on how these different ML algorithms work with varying levels of information about the network work for the route optimization problem.

On the other hand Akin and Korkmaz in their survey of routing algorithms separated algorithms based on whether they are able to adapt to changing network conditions, in this case static vs. dynamic link-costs [3]. Their survey was focused much more on how the SDN controller obtains accurate network state information to get optimal paths from these algorithms. There were 3 distinct categories RA-SLCL (routing with static link costs), RA-DLC (routing with dynamic link costs), RA-DLCMI (routing with dynamic link costs and minimum interference). These surveys differed in their approaches to analyzing routing algorithms. The former classifications seek to analyze routing algorithms based on how they perform with different levels of information provided. This is important as the more topology information we need to discover from the controller, or the amount of data that needs to be in the memory ("knowledge base") of the RL-algorithm can drastically increase the overhead computational cost on the network. And also poses challenges to how we disseminate information to the other nodes. The latter approach used by Akin and Korkmaz et al. focuses on these issues and makes a case for obtaining accurate NSI to allow these algorithms to perform correctly. "We plan to investigate how to effectively collect the NSI while minimizing the load on the controller and the message overhead throughout the network" [3]. It is important to also note that while cloud computing has allowed for seemingly infinite storage and computational data for ML-algorithms it isn't set up for many traditional networks. In these papers these ML-algorithms are also tested with traditional routing algorithms and already implemented dynamic/ML-based routing algorithms to show the improvement of these algorithms in the routing problem over traditional solutions.

5 Applications of ML in Networking

Traditional algorithms are not suitable for SDN because their convergence and responses are slow to changing network conditions. We need new solutions for higher bandwidth and coverage demands. “The Internet has become extremely difficult to evolve both in terms of its physical infrastructure as well as its protocols and performance” [7]. In traditional networks trying to configure these new high-level policies combined with integrated “black-box” devices makes network management error-prone. This has opened up so much interest into the usage of SDNs and ML techniques as the future of networking. Amin et. al calls this inclusion of artificial intelligence in SDN environments the “Knowledge Plane” [1]. Most of these surveys and the focus of my own project is on how machine learning can improve traffic engineering in networks. Generally traffic engineering includes routing optimization, load balancing, congestion control, and quality of service (QoS). Some of these surveys such as Boutaba et. al gives a much more comprehensive view of the service improvements made by ML-based algorithms [2]. Some other services it can perform are predicting faults, resource management, security (intrusion detection), and traffic classification. Mendiola et. al specifically talks about Bandwidth on Demand (BoD) a service provided by ISPs and will require much more dynamic solutions as we require larger and larger datasets for training AI models, and for more distributed networks [6].

6 Results

[4] Had organized results spanning a large variety of surveys. In the study experiments run using deep RL and traditional RL algorithms showed a 14% increase in forwarding table hits, which means higher throughput. And out of the evaluation metrics used performance optimization was the highest improvement with 29%, and attack detection was the second with a 23% increase.

[6] Found improvements with various ML-based routing and forwarding protocols with different communication APIs operating on different networking layers. Concluded the best improvements and most complete solutions will involve D-CPI, A-CPI, and MI protocols which communicate information across all 3 layers.

[2] Found improvements for traffic prediction using time series forecasting (TSF), which constructs a regression model drawing correlation between future traffic demand and previously observed values. Also found applications for congestion control. He found that ML-based packet loss classifiers out performed TCP-vegas by using strategies such as active queue management (AQM), and more dynamic congestion window updates. ML AQM schemes mitigate the limitations of the “drop-tail” used to manage switch buffers and increase queue stability. And dynamic approaches to congestion window control is important for use on different networks such as satellite communication where congestion windows are updated differently.

[3] Showed the difference of algorithm performance with accurate NSI vs inaccurate NSI. And tested periodically getting NSI information for 3, 5, 10 seconds. The results showed that periodically getting NSI information produces inconsistencies due to delays between nodes and the controller. But found overall improvements of accurate NSI in performance. They also look into algorithms that route paths dynamically based on future network demands (RA-DLC) but this problem was found to be NP-Complete.

[5] Collected a lot of data from various routing protocols and analyzes their performances. There are too many to analyze but the algorithms most like some of the algorithms learned in class include RLGA-MAN, Q2-R, and FQLAODV which showed to have good throughput but a high overhead. The PFQ-AODV algorithm also was shows to have good throughput, minimal hop count, but a high overhead which is seen with many of these algorithms.

7 Conclusion

The studies observed conclude that ML-based techniques improve route optimization and have proven and tested applications for traffic engineering. And have potential to help with other networking problems such as security, fault management, and traffic prediction. The problem of trying to update traditional network infrastructure to handle newer challenges is new and recently has seen renewed interest from the wider networking community. The application of these new systems can improve our modern network and build upon it. Many studies proposes future research in the fields of hardware with new controller and switch designs. But also for use in cloud services, wireless networks, and in advanced network security.

References

[1] R. Amin et al. "A survey on machine learning techniques for routing optimization in SDN". In: *IEEE Access* 9.104582–104611 (2021).

[2] R. Boutaba et al. "A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities". In: *Springer* (2018).

[3] Akin E. "Comparison of routing algorithms with static and dynamic link cost in software defined networking (SDN)". In: *IEEE Access* 7.148629–148644 (2019).

[4] A. Faezi S. Shirmarz. "A comprehensive survey on machine learning using in Software Defined Networks (SDN) - human-centric Intelligent Systems". In: *Springer* (2023).

[5] Z. Mammeri. "Reinforcement learning based routing in networks: Review and classification of approaches". In: *IEEE Access* 7.55916–55950 (2019).

[6] A. Mendiola et al. "A survey on the contributions of software-defined networking to traffic engineering". In: *IEEE Communications Surveys* 1617–1634 (2017).

[7] B. Nunes et al. "A survey of software-defined networking: Past, present, and future of Programmable Networks". In: *IEEE Communications Surveys* 1617–1634 (2014).

[8] J. Xie et al. "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges". In: *IEEE Communications Surveys* 393–430 (2019).

[3] [1] [2] [4] [5] [6] [7] [8]