DOI: http://dx.doi.org/10.26483/ijarcs.v15i5.7133

Volume 15, No. 5, September-October 2024

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

AN EVALUATION AND EXAMINATION OF SOFTWARE-DEFINED NETWORKS AND ITS ROUTING ENHANCEMENTS

S. Kalaivani
Research Scholar Department of Computer Science
KPR College of Arts Science and research,
Coimbatore-641407, India

Dr. A. Sumathi
Asso. Professor and Head Department of Information Technology, KPR College of Arts Science and research,
Coimbatore-641407, India

Abstract: Software-Defined Networking (SDN) has enabled the creation of sophisticated, adaptable, and customizable network management solutions. It enables the centralized management and flexible adjustment of routing by utilizing its decoupled architecture. Therefore, efficient routing is essential in SDN to improve network performance, scalability, and efficiency. While conventional models primarily emphasize heuristic and metaheuristic methods, recent progress has incorporated Machine Learning (ML) techniques into some of these models providing adaptive and intelligent solutions to routing challenges. These ML-enhanced models specifically target problems related to delay, traffic congestion and efficient use of resources. This survey provides a comprehensive analysis of different routing strategies in SDN with a specific emphasis on the subset of approaches that incorporate ML techniques. We evaluate the influence of ML on network performance, emphasizing their benefits and constraints, and examine the difficulties and future prospects in using ML for SDN routing. The survey concludes with suggestions for enhancing routing efficiency and network performance by employing advanced techniques selectively.

Keywords: SDN, Optimal routing, Latency, ML, Network Performance, Scalability

INTRODUCTION

The data plane and the control plane are physically separated in Software-Defined Networking (SDN), which is a revolutionary method of designing and managing networks [1]. This decoupling allows for a more adaptable and programmable network architecture, which overcomes the shortcomings of conventional network models. SDN is underpinned by several core principles:

Separation of Control and Data Planes: Traditional network devices combine control logic with data forwarding capabilities, which can limit flexibility and complicate management. SDN separates these functions, centralizing control in a software-based SDN controller while network devices (switches and routers) focus solely on packet forwarding. This separation enables more efficient network management and dynamic configuration [2].

Centralized Control: As the network's central processing unit (CPU), the SDN controller keeps an overview of the entire network at all times. It takes broad policy directives and breaks them down into specific instructions that the data plane hardware can follow. With centralized control, managers may make changes once and have them applied to the whole network [3].

Network Abstraction: SDN hides the complexity of the underlying network architecture by offering a logical representation of the network. This level of abstraction makes it possible to build new network services and apps without having in-depth understanding of the underlying physical network, which in turn makes network programming easier. [4].

I. ARCHITECTURE OF SDN

There are three primary levels in the SDN design: the data plane, the application plane, and the control plane. Services and programs that run on networks and make use of their resources are part of the Application Layer. By means of northbound APIs, applications communicate with the SDN controller in order to solicit policies and services related to the network. Located in the control layer, the SDN controller is in charge of making high-level decisions regarding the network according to the policies specified by applications.

The controller maintains a global network view and communicates with data plane devices via southbound APIs to enforce the desired network behaviors. Physical network devices, such switches and routers, make up the data plane layer. They are responsible for forwarding data depending on instructions and rules received from the SDN controller. Data plane devices are simplified focusing primarily on packet processing rather than complex control logic [5]. Figure 1 depicts the fundamental architecture of SDN and Table 1 briefly explain components of SDN and their purposes.

A. *Optimal routing in SDN*

Data packets must be able to traverse a network from one node to another, and routing is the technique by which this is accomplished [6]. The data plane and control plane are physically separated in SDNs, which allows for real-time path modifications in response to changing circumstances and network policies, as well as more flexible and dynamic routing.

However, Routing in SDN presents several intricate challenges that affect network efficiency and performance. One significant issue is dynamic routing where the need to adapt to fluctuating traffic patterns and network conditions can strain the SDN controller potentially leading to performance bottlenecks and increased latency. Scalability further compounds this challenge as the controller must handle a growing volume of routing information and frequent updates across a large number of data plane devices which can result in substantial communication overhead [7].

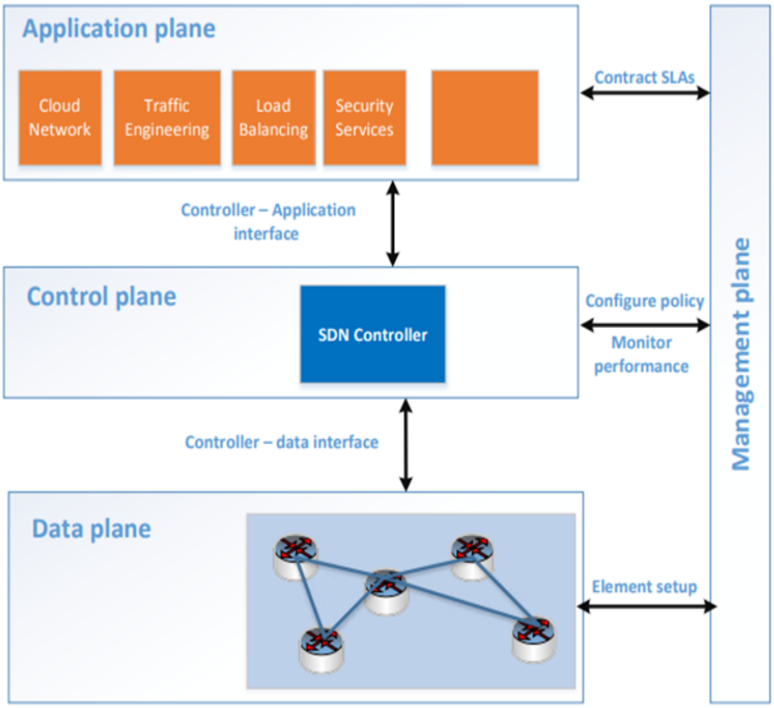

Figure 1. Architecture of UWSN

Table I. Transmission Components and Functions of SDN Architecture

Layers	Component	Function
Application plane	Network Service Applications	The SDN controller facilitates communication with network infrastructure, providing services with an abstracted global view, enabling them to achieve their objectives.
	Business Services	SDN controllers provide business functions and network interaction.
	Security Services	Interact with the SDN controller to secure the network.
	Other Services	Other applications that use the SDN controller's network capabilities.
Control plane	SDN Controller	Network center that directs data plane flows. Converts application commands to data plane communication protocol.
Data plane	Network Forwarding Elements (Switches)	Send incoming flows using flow table routes. Network traffic processing and forwarding in real time.
Management plane	Network Setup	Configures network parameters. Securely isolated from control, application, and data planes to prevent network attacks.
	Configuration Management	Maintains secure and efficient network configuration. Externally unprogrammable to prevent unauthorized access.
Interfaces	North-Bound Interface (NBI)	Accesses network resources from applications. Though undefined, it may include role-based application authorization and authentication.
	South-Bound Interface (SBI)	Links the SDN controller to network

		forwarders. Open Flow (OF) is widely used in SDN networks.
Conflict management	Orchestrator	Solves application-specific network behavior rules. Integrates into the controller, management plane, or as a separate application to manage complex network control tasks.

Additionally, ensuring routing performance and efficiency involves addressing latency concerns and optimizing resource utilization through effective load balancing. To fully utilize SDN and achieve optimal network performance, it is essential to address certain routing challenges.

To address the complexities of routing in SDN, heuristic and metaheuristic algorithms offer effective solutions for optimizing routing paths and improving network performance [8]. Heuristic algorithms use problem-specific knowledge to find good-enough solutions within a reasonable time frame. For instance, algorithms such as Dijkstra's [8, 9] and Bellman-Ford [10, 11] are often employed for their simplicity and efficiency in finding shortest paths or optimizing specific metrics like latency or cost. However, as networks become more complex, these traditional heuristics may struggle to find optimal solutions due to their computational limitations. This is where metaheuristic algorithms come into play.

Metaheuristics such as Genetic Algorithms (GAs), Simulated Annealing (SA) and Tabu search offer more sophisticated approaches to explore a larger solution space and avoid local optima. By using mechanisms inspired by natural processes or complex systems, metaheuristics can adaptively refine solutions by handling dynamic changes in network conditions and constraints effectively. These algorithms provide a flexible framework for addressing various optimization objectives in SDN including load balancing, traffic engineering, and fault tolerance thereby enhancing the overall routing performance and network efficiency.

Natural selection and evolution are the driving forces behind GA. In order to evolve a population of possible solutions over generations, they employ operations such as mutation, selection, and crossover [12]. The annealing process, used in metallurgy to stabilize a material by heating and cooling it gradually, is an inspiration for SA. It departs from local optima and investigates a larger solution space using probabilistic methods [13]. To prevent re-exploring solutions that have already been investigated, Tabu Search employs local search strategies that incorporate memory structures. It uses a tabu list to remember previous actions and avoid repeating them [14].

B. *Integration of Machine Learning with Metaheuristics*

Combining machine learning (ML) with metaheuristic algorithms is a powerful way to tackle complex optimization problems in SDNs. This blend brings together the best of both worlds, greatly improving the performance and adaptability of these algorithms. Here is how ML can enhance metaheuristic algorithms:

- ML models can dynamically adjust key parameters of metaheuristics, such as mutation rates in GAs or cooling schedules in Simulated Annealing SA based on real-time data. This adaptive tuning improves the efficiency of the search process and accelerates convergence towards optimal solutions [15].
- ML algorithms can predict future network states and traffic patterns. By incorporating these predictions metaheuristic algorithms can adjust routing paths and resource allocations, optimizing performance and reducing latency before issues arise [16].
- ML models can offer more nuanced evaluations of potential solutions. By integrating sophisticated metrics and analysis, these models provide a more comprehensive assessment of solution quality, improving the accuracy of optimization outcomes [17].
- ML algorithms can identify patterns and correlations within large datasets, guiding metaheuristic searches more effectively. This capability helps in recognizing promising solution areas and focusing the search process, thus improving optimization efficiency.

C. *A Comprehensive Approach to Machine Learning-Driven Routing in SDNs*

Step 1 (Collecting Data): First, gather detailed information about network traffic. Track how much traffic there is, what types are coming through, and when they peak—whether that's hourly, daily, weekly, or monthly. Attention to busy times, average traffic loads, and any changes due to special events or seasons. This will give you a solid understanding of past network behavior and help spot any trends. Also, keep an eye on current traffic in real-time to make quick routing decisions.

Step 2 (Data pre-processing): Next, tidy up the collected data. Remove any noise, outliers or incomplete records to ensure data is accurate. Normalize the data to make sure everything is on the same scale or format. Extract important features like peak traffic times and performance metrics. This will help create useful inputs for the ML models [18].

Step 3 (Training the Models): Use of historical data to label different outcomes, like the best routing paths or points of congestion. Train the supervised machine learning models like regression, decision trees, or NNs using this labeled data to predict network conditions and routing decisions. Also the use of clustering algorithms to uncover patterns or anomalies in data. For real-time decisions, apply Reinforcement Learning (RL) so models can learn from network feedback and improve over time [19].

Step 4 (Integrating Real-Time Data): Combine real-time data with trained models to keep predictions and decisions up-to-date. Regularly update models with the latest data to keep them sharp, which might mean retraining or tweaking them as new information comes in.

Step 5 (Predicting and Analyzing): Use of time-series analysis and predictive models to forecast future traffic and potential congestion points. This will help manage the network proactively and plan for capacity needs. Set up anomaly detection to spot unusual patterns or potential issues quickly like unexpected traffic spikes or equipment problems.

Step 6 (Making Routing Decisions): Create routing algorithms that use ML models' predictions to find the best paths for traffic. These algorithms should be able to adjust routes in real-time to avoid congestion and keep traffic flowing smoothly. Adaptive routing mechanisms will help the network respond quickly to changes and maintain top performance.

By following these steps, SDN environments can leverage the power of ML to enhance predictive routing, improve network performance, and ensure reliable QoS for various applications and traffic types.

The remaining section are organized as follows. Section II delves into the review on protocols for and the integration of ML to some of the protocols. Section III measures its performance evaluation and Section IV concludes the whole survey and paves the way for future researches.

II. Survey On The Usage Of Software-Defined Networks For Improving Routing Protocols In Various Network Models

In SDNs, the centralization of control functions can impact network efficiency and overall performance, potentially affecting data transmission and management. To address these challenges, routing protocols have been developed to optimize control node selection and data flow. These protocols enhance network reliability and performance by minimizing control overhead and ensuring efficient data routing through well-defined paths. This section reviews several routing protocols in SDNs.

Bano et al. [20] proposed a careful and progressive migration of Wireless Mesh Networks (WMNs) to SDNs, which would result in a strong routing architecture for both types of networks, called Soft-Mesh. The primary intention was to address various routing challenges in WMNs, such as scalability, reliability and network management issues by integrating SDN architecture. This integration facilitates network management and routing while ensuring full interoperability with existing IP devices to mitigate technical, operational and economic problems. Soft-Mesh enhanced the architecture of SDN nodes by enabling them to cohabit with legacy IP-based nodes. The architecture modified SDN nodes to react to network topology changes dynamically without frequent controller queries. The overarching goal of Soft-Mesh was to ease the migration of WMNs from conventional to SDN-enhanced routing by providing an affordable and frictionless solution for interoperability between SDN and legacy nodes.

Duong [21] developed the Load Balancing Routing under Constraints of Quality of Transmission (LBRCQT) algorithm to tackle the dual challenges of load balancing and maintaining Quality of Transmission (QoT) in Wireless Mesh Networks (WMNs). The algorithm was implemented at the SDN controller, uses a centralized approach to select the best routes based on real-time QoT and traffic load data. The network is organized into three layers: infrastructure for packet switching, control for routing and signaling, and application layer, with the open-flow protocol managing communication between these layers. LBRCQT improves WMN performance by efficiently balancing traffic, enhancing QoT and reducing end-to-end delays through dynamic route management.

Alidadi et al. [22] presented a Path Selection with Low Complexity (PSLC) algorithm to tackle the challenges of bandwidth-restricted routing in SDN-based Multi-Protocol Label Switching (SDN-MPLS) by balancing network load, route length and energy savings while maintaining low complexity. The SDN architecture enhanced MPLS-TE by incorporating SDN controller, Path Computing Element (PCE) to calculate and manage routing tables for SDN forwarding components, Path Computing Client (PCC). PCCs sent measuring traffic metrics to the PCE after performing packet forwarding. PCE then used this data to dynamically adjust routing tables and respond to changing traffic conditions. Finally, the PSLC algorithm utilized data from the PCE node and link weights to effectively manage constrained bandwidth routes.

Qian et al. [23] developed an advanced downlink routing control strategy for LoRaWAN that combines SDN with an improved Auto-Regressive Integrated Moving Average (ARIMA) model. Incorporating Savitzky-Golay (S-G) filtering to manage data volatility and a sliding window method for pre-processing, this strategy utilized SDN to efficiently monitor network traffic and regulate routing. The approach enhanced bandwidth usage and optimizes route selection, resulting in reduced packet loss and transmission delay, and improved reliability and performance of LoRaWAN networks.

The State-Action Reward-State-Action (SARSA) based Delay-aware Route Selection (SDRS) algorithm was studied by Shi et al. [24] for power distribution Internet of Things (PD-IoT) networks that enable wireless Power Line Communications (PLC) through SDN. Problems with route selection in dynamic networks, such as those caused by electromagnetic interference, imprecise global status information, and the combination of PLC and wireless communications, were the motivation for this approach. SDRS functioned admirably in the presence of both weak and heavy electromagnetic interference, greatly decreasing transmission delay and improving dependability.

Samadi et al. [25] introduced an Intelligent Energy-Aware Routing system for Mobile IoT Networks (IERMIoT) to improve network lifespan and minimize energy waste through centralized management of mobile nodes using SDN. This approach involved creating clusters of nodes within the network and employing an intelligent evolutionary algorithm to optimize the number and distribution of these clusters dynamically. By centralizing energy-intensive tasks to a central controller and implementing a loyalty mechanism to maintain cluster stability, IERMIoT effectively manages the dynamic topology changes due to node mobility, reduces energy consumption and improves overall network performance.

In this study, the notable evaluated performance metrics are End-to-End (ETE) Delay, Throughput, Energy Consumption, call blocking ratio (CBR), Packet Delivery Ratio (PDR), Packet Loss Rate (PLR), Bandwidth Occupancy rate (BOR), Routing overhead (RO) and Latency. These metrics are critical for assessing the efficiency and effectiveness of the network under various conditions and scenarios. Table 2 compares usage of SDNs for improving routing protocols in various network models in terms of their advantages, limitations and performance metrics.

III. SURVEY ON ROUTING IMPROVEMENTS OF SOFTWARE-DEFINED NETWORKS

Because SDNs separate the control plane and data plane, they enable dynamic configuration and centralized network management, which is a revolutionary step in networking. Traditional routing approaches in SDNs usually have problems with scalability, adaptation, and optimizing performance, even though they have certain benefits. This section explores various improvements in SDN routing, focusing on innovative approaches and methodologies designed to enhance network performance, optimize routing strategies and address the inherent limitations of conventional routing protocols in dynamic and complex network environments.

Li et al. [26] investigated a Fuzzy-based Rapid and Efficient Routing algorithm with guaranteed Latency-Throughput (FRLR) for managing traffic flows in SDN. FRLR seeks to take advantage of SDN by guaranteeing good QoS and energy efficiency, while also separating the control and data planes to improve network management. FRLR uses fuzzy logic to classify traffic flows based on their latency requirements and resource demands, which helps in efficiently managing and allocating network resources. It made routing adjustments based on current network conditions. It also identifies critical links to minimize interference and maintain performance. This algorithm ensured path latency by maximizing the utilization of link bandwidth.

Zhou et al. [27] developed the Asynchronous Advantage Actor-Critic (A3C) QoS-aware Routing Optimization Mechanism (AQROM) to enhance Quality of Service (QoS) and dynamic routing in SDNs. AQROM improved network QoS and cut down training time by adjusting routing strategies in real-time. It was inspired by the A3C method which allows for high-dimensional input and output sets and can handle both discrete and continuous states and actions. This algorithm improved network QoS by dynamically updating the reward function based on optimization objectives, independent of network topology and traffic patterns.

Santana et al. [28] developed a piecewise stationary Bayesian Multi-Armed Bandit (MAB) approach, Reactive Upper Confidence Bound (React-UCB) for optimal routing of information in SDNs. React-UCB was employed with various enhanced features to opt bidirectional route having lowest delay. React-UCB skilfully addressed sudden shifts in reward distributions through several methods. It embraced an optimistic approach in uncertain situations, prioritized recent feedback by diminishing the importance of older rewards, reset previously acquired data when changes in path delay distributions were detected, and utilized reward correlations to facilitate learning across various paths. These combined strategies effectively minimized the agent's accumulated regret and enhanced the efficiency of network resource operations.

Pathan et al. [29] proposed a Multi-Objective Integer Linear Programming (MILP) approach to optimize routing in SDN-enabled data centre networks. They developed two greedy algorithms. Priority-based Energy Minimization Algorithm (PEMA) was to enhance flow priority and reduce energy consumption and Priority-based Even Load Distribution Algorithm (PEDL) to balance data loads while maximizing flow priority. Their results demonstrated that this approach improved network performance for varying flow rates and dynamic flow needs.

Riveros-Rojas et al. [30] developed a solution based approach for addressing Routing and Device Assignment (RDA) issues in SDN using MILP and Genetic Algorithm (GA). The MILP model optimized SDN by minimizing blocked flows and energy usage. It worked out the best paths between nodes and assigned devices based on their usage to find the most efficient configurations. The GA then took these MILP solutions and fine-tuned them by exploring a broader range of SDN devices. Using non-binary permutation coding, GA iteratively improved these configurations through selection, crossover and mutation thereby enhancing energy saving and reducing blocked flows.

Zhang et al. [31] developed a holistic SDN deterministic network (HSDDN) to provide a real-time, reliable communication for industrial environments using combined SDN and TSN to achieve deterministic network behavior with both wired and wireless modes. Additionally, a SD-PHY was employed to provide a flexible and efficient backscatter communication system which supports various protocols through dynamic signal modulation and low-power operation. Finally, Tabu search for routing and scheduling with dual-stages (TSRS-DS) algorithm was introduced to solve co-design problem. During joint routing and scheduling optimization phase of TSRS-DS, Tabu search was utilized to mitigate combinatorial optimization issues.

Bouzidi et al. [32] introduced a Deep Q-Network and Traffic Prediction based routing optimization (DTPRO) approach through Neural Networks (NN) in SDN to dynamically predict traffic congestion and optimize routing. The DTPRO architecture consisted of four planes: Data, Control, Management, and Knowledge. It improved SDN by adding a Knowledge Plane (KP) that leveraged AI and telemetry data for smarter network management. The KP used ML to optimize routing and predict congestion while enabling proactive traffic control.

Kim et al. [34] devised a routing optimization method for SDNs using Deep Reinforcement Learning (DRL) with a Deep Deterministic Policy Gradient (DDPG) algorithm to understand the relationship between switch traffic loads and network performance. Also, to minimize end-to-end delay and packet loss, the Aggregated Traffic Volume Matrix (ATVM) was used to select the optimal set of link weights. The SDN controller determined the routing paths based on these link weights and configured the flow rules on SDN-enabled switches. The DRL learning process employed an M/M/1/K queue model to address network performance degradation issues.

Godfrey et al. [35] illustrated a Multi-Objective Routing Protocol using a RL algorithm with Dynamic Objective Selection (DOS-RL) to optimize energy consumption in IoT networks and adapt to sudden network changes in SDWSN-IoT. This approach utilized Q-learning to dynamically select and optimize objectives based on real-time system states and confidence in Q-values. Additionally, a Correlated Multi-Objective Markov Decision Process (CMOMDP) was applied to integrate multiple related objectives such as energy efficiency, load balancing and reliability into a dynamic reward function. The results indicated that this method improved network performance by balancing competing objectives and adaptively selecting the most pertinent optimization criteria.

Ke et al. [36] presented Q-learning Widest-Path Routing Algorithm (Q-WPRA) based on RL for SDN to determine the optimal path for transmission. This algorithm comprised of three stages. Path searching and loop phase executed the recognition of the complete transmission path and also prevented returning and looping. The purpose of the maximum bandwidth reward function was to dynamically update and optimize the transmission path based on the reward value. Finally, widest-path verification was responsible to ensure that the selected transmission path provides the maximum available bandwidth.

Gunavathie et al. [37] introduced a ML-driven Proactive Re-routing System (MLPRS) designed to improve Quality of Service (QoS) and dynamic load distribution in real-time SDN topologies. This approach involved continuous monitoring of traffic loads along network paths and triggering flow redirection when congestion was detected. Utilizing ML, the system categorized applications and assigned priorities to facilitate re-routing. Results indicated a substantial decrease in Round Trip Time (RTT) with the implementation of this method.

In this study, the notable evaluated performance metrics are Jitter, Bandwidth, End-to-End (ETE) Delay, Throughput, Energy Consumption, call blocking ratio (CBR), Packet Delivery Ratio (PDR), Packet Loss Rate (PLR), Link utilization (LU), Flow success rate (FSR), Bandwidth Occupancy rate (BOR), Routing overhead, Round Trip Time (RTT) and Latency. These metrics are critical for assessing the efficiency and effectiveness of the network under various conditions and scenarios.

Table 3 compares improved routing protocols in SDNs in terms of their advantages, limitations and performance metrics.

IV. PERFORMANCE EVALUATION

The performance evaluation of routing protocols for SDNs is discussed in this section, with a focus on two distinct studies. The protocols were tested under various simulation environments to determine their efficiency in real-world applications.

A. *Using SDN for Improving Routing Protocols in Various Network Models*

This evaluation was conducted using delay and throughput as the key metrics. Figure 2 shows the comparison of delay performance of various models that utilize SDN for improved routing protocols. It is clear that LBRCQT [21] outperforms other models with minimal delay. Lower delay indicates quicker responses and faster processing, which generally results in better performance for real-time applications and systems.

Table II. Comparative Study of Cluster-oriented Routing Protocols in UWSNs

Ref No.	Protocols	Advantages	Limitations	Environment	Major performance metrics
[20]	Soft-Mesh	Inexpensive Implementation		Mininet-WiFi	Throughput=10mbps, ETE delay = 7 ms, PDR = 10% RO = 70 Mbps
[21]	LBRCQT	Blocking probability of data packets are reduced		OMNeT++, INET 2.0	PDR = 95% ETE delay = 1.3ms Throughput = 247 mbps
[22]	SDN-MPLS, PCC, PCE, PSLC	Easy monitoring and maintenance of network.		PAYTON simulator	CBR=0.17 route length=4.03 flow=500 CPU time=5s
[23]	ARIMA, LLDP	Better data transmission reliability		Mininet	PLR= 15% Delay = 0.12s BOR= 70%
[24]	MPSRCS,SDRS	Real-time Interaction between the environments			Delay= 0.87s Retransmission Ratio =0.06
[25]	IERMIoT, GA, PSO, ICA, GWO, WOA	Increased steady-state duration		MATLAB	Energy Consumption =0.5J, Network stability = 1900 s, Network coverage = 125, PDR=85% Routing overhead = 0.38

Table III. Comparative Study of Cluster-oriented Routing Protocols in UWSNs

Ref No.	Protocols	Advantages	Limitations	Environment	Major performance metrics
[20]	FRLR	Usage of critical links saves resources for future	Energy consumption increases when traffic flow increases	MATLAB R2019a	Throughput 10.2 gbps Jitter=0.005 ms, Energy consumption =1000W, Latency (Delay) = 0.024 ms
[21]	AQROM	Reduced training time & memory resources	Adjustment of learning rates is still inconvenient	OMNeT ++	Latency= 3.456s PLR=57.701% Throughout= 5.882 kbps
[22]	React-UCB, MAB	Low computational complexity	It is a centralized structure	Mininet	For 400 arms, Execution time = 22.46ms
[23]	MILP, PEMA, PEDL	Prevent single node bottlenecks.	Struggles to manage in the case of link breakdown or path congestion	C++	Energy savings =35% Load distribution = 5.8 FSR= 60%
[24]	MILP, GA	Highly suitable for dynamic and high-traffic network environments	Implementation is computationally intensive	IBM ILOG CPLEX version 12.6	Energy consumption =2700 W, Blockage rate =10
[25]	HSDDN, TSRS-DS	Custom interface simplifies control and monitoring		OMNET++	Average Runtime=2 s
6]	DTPRO, DQN, NN, ML	Defines all paths in one action	Not fit for distributed SDN controllers	OpenvSwitch	Latency = 1.2 ms, Delay = 1.25 ms, PLR = 0.12%, LU = 6 mbps, Throughput = 8.1 mbps
7]	RL, Q-Learning STP	Network controlled remotely via web		Raspberry pi B+	PDR=0.99%
8]	DRL, DDPG, ATVM, M/M/1/K	Offline training avoids performance degradation		Python networkX	For 150 flows, Throughput=10 mbps Delay=2.4 s
9]	DOS-RL, Q-Learning CMOMDP	Incorporates multiple QoS metrics	Time complexity depends on state space	NS-3 AI module	For 60 nodes, PDR=0.92 Energy consumption for 1% received packets = 0.91 Delay = 2.4 ms
0]	Q-WPRA, RL	Prevents path loops effectively	No significant performance improvement	Mininet	Bandwidth= 3.82 mbps, Overhead time = 0.022 s
1]	MLPRS	Flows are updated every minute enabling dynamic load balancing		Mininet	For 250 packets, Throughput = 31.05 gbps Bandwidth= 33.33 gbps, RTT (Delay) = 0.486 ms

Figure 3 depicts the comparison of throughput performance of various models that utilize SDN for improved routing protocols. From the above analyses, it is obvious that LBRCQT [21] again performed well. It is because it transfers reduced number of data packets with minimum network traffic.

Therefore, it is evident that the LBRCQT [21] performed better in every possible way. This evaluation indicates that [21] is the efficient protocol making it more suitable for improving routing using SDN architecture.

B. ***Routing Improvements of SDN***

This evaluation was conducted using throughput and bandwidth as the key metrics. Figure 4 demonstrates the performance analysis of improved routing SDN models based on their throughput. It shows that MLPRS [37] outperformed other protocols with an exceptional increase in throughput ensuring higher data transfer rates. However, [26] also performed well achieving the second-highest throughput due to its optimized routing strategies. It is important to note that AQROM [27] is omitted in this evaluation due to its very low throughput value, which is literally in kbps.

Figure 5 illustrates the performance analysis of routing protocols based on their bandwidth. In this metric, MLPRS [37] again demonstrated superior performance showing a significantly higher bandwidth compared to other protocols. This indicates that MLPRS can handle larger volumes of data transmission more efficiently. From Fig.6, it is concluding that MLPRS archived less delay even if it transmits more packets than other protocols.

The model presented in MLPRS significantly outperformed other protocols across both metrics. It not only maximized throughput, crucial for high-speed data transfer in SDNs but also achieved the highest bandwidth, ensuring efficient data transmission capacity. The superior performance of MLPRS can be attributed to its innovative bandwidth management techniques and dynamic routing strategies that optimize resource usage and adapt to network conditions.

Comparing both the studies, MLPRS focuses on real-time SDN topologies and directly addresses real-time network congestion and improves latency whereas LBRCQT [21] is more focused on balancing load and maintaining QoT. MLPRS employs continuous traffic monitoring and ML categorization, offering a more adaptive and flexible solution for dynamic network conditions. LBRCQT [21] while effective uses a centralized approach that may not be as responsive to sudden changes in network traffic.

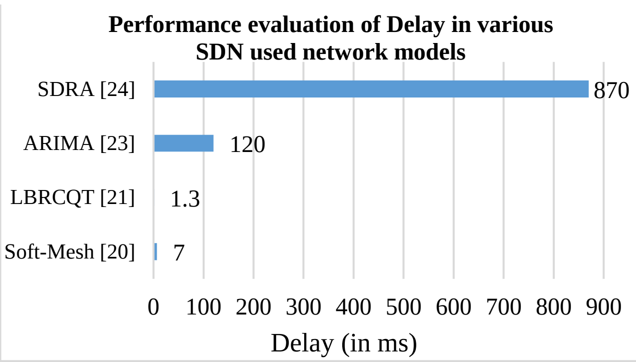

Figure 2. Performance analysis delay in various network models

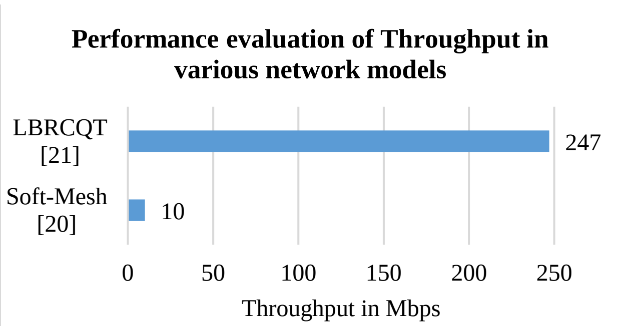

Figure 3. Performance analysis throughput of various network models which uses SDN for improving routing protocols

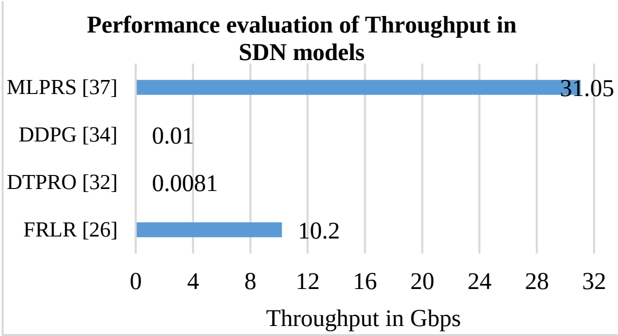

Figure 4. Performance analysis of improved routing SDN models based on throughput

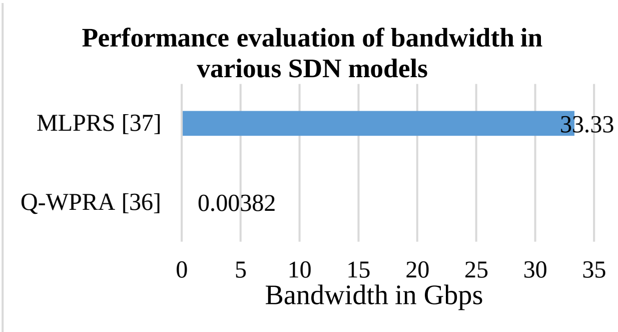

Figure 5. Performance analysis of improved routing SDN models based on bandwidth

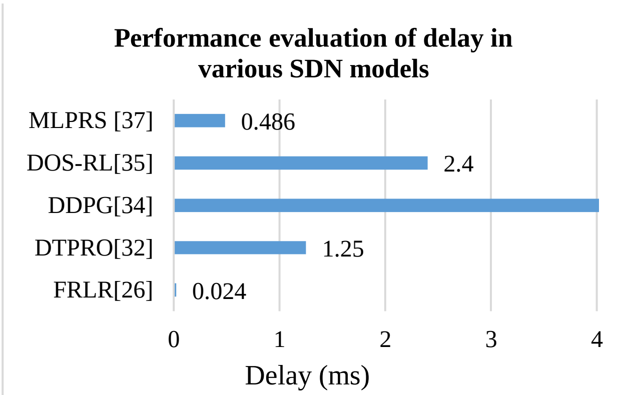

Figure 6. Performance analysis of improved routing SDN models based on delay

Therefore, this comprehensive evaluation indicates that MLPRS [37] is the most effective protocol, offering a well-rounded solution. This makes it the best choice for SDN applications ensuring both high-speed and high-capacity data transmission.

V. Conclusion

In this paper, a comprehensive review was conducted to examine various optimal routing protocols in SDNs with a focus on models incorporating ML techniques. The efficiency of each protocol was assessed based on its advantages, limitations, simulation environment, and performance metrics achieved. From this analysis, it was found that the model leveraging ML for adaptive routing performed exceptionally well compared to other protocols. This was due to its ability to dynamically adjust routing decisions and optimize network performance with lower overhead and more efficient path selection. However, a notable issue with the current ML-based routing model was its higher latency that attributed to the complexity of adapting to real-time changes and the extensive processing required for decision-making. Additionally, the model did not fully address challenges related to data transfer efficiency within SDNs, which affects overall packet delivery rates and network throughput. Future research will focus on refining the ML-based routing model to reduce latency, enhance packet delivery rates, and improve overall network performance in SDNs.

VI. References

[1] Hussain, S & Yuvanesh, S & Yokesh, S. (2023). Revolutionizing Networking: An Exploration of Software-Defined Networking. 1020-1026. doi:10.1109/icacrs58579.2023.10404534.

[2] Faizullah, S., & AlMutairi, S. (2018). Vulnerabilities in SDN due to separation of data and control planes. Int. J. Comput. Appl., 31, 21-24. doi: 10.5120/ijca2018916519

[3] Caria, M., Jukan, A., & Hoffmann, M. (2016). SDN partitioning: A centralized control plane for distributed routing protocols. IEEE Trans. Netw. Serv. Manag., 13(3), 381-393. doi: 10.1109/TNSM.2016.2585759

[4] Ali, J., & Roh, B. H. (2020). An effective hierarchical control plane for software-defined networks leveraging TOPSIS for end-to-end QoS class-mapping. Ieee Access, 8, 88990-89006. doi: 10.1109/Access.2020.2993556

[5] Cabaj, K., Wytrebowicz, J., Kuklinski, S., Radziszewski, P., & Dinh, K. T. (2014). SDN architecture impact on network security. Proc. FedCSIS (Position Papers), Warsaw, Poland, 143-148. doi: 10.15439/2014F473

[6] Dhenakaran, S. S., & Parvathavarthini, A. (2013). An overview of routing protocols in mobile ad-hoc network. Int. J. Adv. Res. Comput. Sci. Softw. Eng., 3(2). doi: 10.4236/cn.2015.72012

[7] Al Mhdawi, A. K., Al-Raweshidy, H., Al-Karkhi, W. J., & Humaidi, A. J. (2024). Micro software-defined control (μSDC): Empowering smart grids with enhanced control and optimization. IEEE Access, 12. doi: 10.1109/ACCESS.2024.3421947

[8] Neghabi, A. A., Navimipour, N. J., Hosseinzadeh, M., & Rezaee, A. (2019). Nature-inspired meta-heuristic algorithms for solving the load balancing problem in the software-defined network. Int. J. Commun. Syst., 32(4), e3875. doi: 10.1002/dac.3875

[9] El-Hefnawy, N. A., Raouf, O. A., & Askr, H. (2022). Dynamic routing optimization algorithm for software-defined networking. Comput. Mater. Continua, 70(1). doi: 10.32604/cmc.2022.017787

[10] Shivendu, A., Dhakal, D., & Sharma, D. (2015). Emulation of shortest path algorithm in software-defined networking environment. Int. J. Comput. Appl., 116(1). doi: 10.5120/20304-2344

[11] Mohanty, S., Sahoo, B., & Behera, S. S. (2024). An assessment of nature-inspired metaheuristic algorithms for resilient controller placement in software-defined networks. Decis. Anal. J., 100501. doi: 10.1016/j.dajour.2024.100501

[12] Han, M. (2018). Optimal routing path calculation for SDN using genetic algorithm. Int. J. Hosp. Inf. Technol., 11(3), 7-12. doi: 10.21742/IJHIT.2018.11.3.02

[13] Sahoo, K. S., et al. (2017). Metaheuristic solutions for solving controller placement problem in SDN-based WAN architecture. Proc. ICETE 2017, Madrid, Spain, 1-6. doi: 10.5220/0006483200150023

[14] Tache, M. D., Păscuțoiu, O., & Borcoci, E. (2024). Optimization algorithms in SDN: Routing, load balancing, and delay optimization. Appl. Sci., 14(14), 5967. doi: 10.3390/app14145967

[15] Aleti, A., & Moser, I. (2016). A systematic literature review of adaptive parameter control methods for evolutionary algorithms. ACM Comput. Surv., 49(3), 1-35. doi: 10.1145/2996355

[16] Lohrasbinasab, I., Shahraki, A., Taherkordi, A., & Jurcut, A. D. (2022). From statistical-to machine learning-based network traffic prediction. Trans. Emerg. Telecommun. Technol., 33(4), e4394. doi: 10.1002/ett.4394

[17] Abdi, A. H., et al. (2024). Security control and data planes of SDN: A comprehensive review of traditional, AI and MTD approaches to security solutions. IEEE Access, 12. doi: 10.1109/ACCESS.2024.3393548

[18] Hark, R., et al. (2019). Preprocessing monitoring information on the SDN data-plane using P4. Proc. IEEE NFV-SDN 2019, Dallas, TX, USA, 1-6. doi: 10.1109/NFV-SDN47374.2019.9040030

[19] Ampratwum, I. (2020). An intelligent traffic classification based optimized routing in SDN-IoT: A machine learning approach. Ph.D. dissertation, Dept. Electr. Comput. Eng., Univ. Ottawa, Ottawa, Canada. doi: 10.20381/ruor-24389

[20] Bano, M., et al. (2021). Soft-mesh: A robust routing architecture for hybrid SDN and wireless mesh networks. IEEE Access, 9, 87715-87730. doi: 10.1109/ACCESS.2021.3089020

[21] Duong, T. V. T. (2021). Load balancing routing under constraints of quality of transmission in mesh wireless network based on software defined networking. J. Commun. Netw., 23(1), 12-22. doi: 10.23919/JCN.2021.000004

[22] Alidadi, A., Arab, S., & Askari, T. (2022). A novel optimized routing algorithm for QoS traffic engineering in SDN-based mobile networks. ICT Express, 8(1), 130-134. doi: 10.1016/j.icte.2021.12.010

[23] Qian, Q., et al. (2022). LoRaWAN network downlink routing control strategy based on the SDN framework and improved ARIMA model. Future Internet, 14(11), 307. doi: 10.3390/fi14110307

[24] Shi, Z., Zhu, J., & Wei, H. (2022). SARSA-based delay-aware route selection for SDN-enabled wireless-PLC power distribution IoT. Alexandria Eng. J., 61(8), 5795-5803. doi: 10.1016/j.aej.2021.11.029

[25] Samadi, R., Nazari, A., & Seitz, J. (2023). Intelligent energy-aware routing protocol in mobile IoT networks based on SDN. IEEE Trans. Green Commun. Netw., 12. doi: 10.1016/j.jksuci.2022.08.005

[26] Li, H., Osmani, A., & Aziz, A. S. A. (2022). A fuzzy-based fast routing algorithm with guaranteed latency-throughput over software defined networks. J. King Saud Univ. Comput. Inf. Sci., 34(10), 8221-8233. doi: 10.1016/j.jksuci.2022.08.005

[27] Zhou, W., et al. (2022). AQROM: A quality of service aware routing optimization mechanism based on asynchronous advantage actor-critic in software-defined networks. Digit. Commun. Netw. doi: 10.1016/j.dcan.2022.11.016

[28] Santana, P., & Moura, J. (2023). A Bayesian multi-armed bandit algorithm for dynamic end-to-end routing in SDN-based networks with piecewise-stationary rewards. Algorithms, 16(5), 233. doi: 10.3390/a16050233

[29] Pathan, M. N., et al. (2024). Priority based energy and load aware routing algorithms for SDN enabled data center network. Comput. Netw., 240, 110166. doi: 10.1016/j.comnet.2023.110166

[30] Riveros-Rojas, G. J., et al. (2024). Energy-and-blocking-aware routing and device assignment in software-defined networking—A MILP and genetic algorithm approach. Math. Comput. Appl., 29(2), 18. doi: 10.3390/mca29020018

[31] Zhang, J., et al. (2024). Routing and scheduling co-design for holistic software-defined deterministic network. Fundam. Res., 4(1), 25-34. doi: 10.1016/j.fmre.2023.11.007

[32] Bouzidi, E. H., et al. (2021). Deep Q-network and traffic prediction based routing optimization in software defined networks. J. Netw. Comput. Appl., 192, 103181. doi: 10.1016/j.jnca.2021.103181

[33] Younus, M. U., Khan, M. K., & Bhatti, A. R. (2021).

Improving the software-defined wireless sensor networks routing performance using reinforcement learning. IEEE Internet Things J., 9(5), 3495-3508. doi: 10.1109/JIOT.2021.3102130

[34] Kim, G., Kim, Y., & Lim, H. (2022). Deep reinforcement learning-based routing on software-defined networks. IEEE Access, 10, 18121-18133. doi: 10.1109/ACCESS.2022.3151081

[35] Godfrey, D., et al. (2023). An energy-efficient routing protocol with reinforcement learning in software-defined wireless sensor networks. Sensors, 23(20), 8435. doi: 10.3390/s23208435

[36] Ke, C. H., Tu, Y. H., & Ma, Y. W. (2023). A reinforcement learning approach for widest path routing in software-defined networks. ICT Express, 9(5), 882-889. doi: 10.1016/j.icte.2022.10.007

[37] Gunavathie, M. A., & Umamaheswari, S. (2023). MLPRS: A machine learning-based proactive re-routing scheme for flow classification and priority assignment. J. Eng. Res., 11(3), 114-122.
doi: 10.1016/j.jer.2023.100075

