Software-Defined Networking's Adaptive Load Optimisation Technique for Multimedia Applications with Multi-Controller Utilisation

Pardeep Singh Tiwana[1,2]
Computer Science & Engineering
[1]Chandigarh University, Gharuan, Mohali, Punjab
[2]Chandigarh Engineering College-CGC Landran, Mohali, Punjab
tiwana07763@gmail.com
ORCID: 0000-0002-0449-5982

Jaspreet Singh
Computer Science & Engineering
Chandigarh University, Gharuan
Mohali, Punjab
cec.jaspreet@gmail.com

2023 9th International Conference on Signal Processing and Communication (ICSC) | 979-8-3503-8320-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSC60394.2023.10441149

***Abstract*— Modern networking settings' growing need for multimedia applications has sparked research into cutting-edge methods to improve network efficiency and resource utilization. A potential model that decouples the control and data planes to allow for centralized management and programmability has been developed called software-defined networking (SDN). The adaptive load optimization technique for multimedia applications with multi-controller utilization in SDN is presented in this research as being novel. The key to this method is its adaptive load optimization mechanism, which intelligently distributes network resources according to current traffic patterns and application needs. The method detects network congestion and proactively redistributes loads across network segments using machine learning and optimization techniques. For multimedia applications, this guarantees optimum resource utilization, reduces latency, and improves Quality of Service (QoS). The suggested approach shows its efficiency in improving the performance of multimedia applications through comprehensive simulations and testing in a variety of network conditions. Results show better throughput, decreased latency, and effective resource utilisation when compared to conventional static load balancing methods. Even in big and complicated network installations, the multi-controller architecture demonstrates its value by maintaining stability and responsiveness.**

Keywords—SDN, Multimedia applications, multi-controller, adaptive load optimization, resource allocation, congestion.

I. INTRODUCTION

The flexibility, scalability, and adaptability of traditional ways of managing and controlling network infrastructure have been constrained in the constantly changing world of computer networking. The Software Defined Networking (SDN) paradigm, which redefines how networks are created, implemented, and maintained, comes into play in this context. SDN is a ground-breaking idea that divides a network architecture's control plane from its data plane [13]. SDN decouples these operations from data packet forwarding and the decision-making process for routing and traffic management, in contrast to traditional networks where network equipment (routers, switches, etc.) handles both [14]. By centralising and abstracting network control via software-driven processes, this separation is made possible. The programmability concept is the foundation of SDN. Network administrators may achieve previously unheard-of levels of automation and customization by consolidating management in a software-based controller, allowing them to dynamically manage and customize network behavior using programmable interfaces. This change enables organizations to efficiently adjust their network infrastructure to changing workloads, application requirements, and security demands by eliminating the need for intricate and time-consuming reconfigurations of individual network devices.

A. SDN Composition

A revolutionary method for creating and administering computer networks that places an emphasis on flexibility, programmability, and centralized management is known as software-defined networking (SDN) architecture. Network managers may govern and coordinate network behavior using software-defined controllers because to SDN's fundamental separation of the network control plane from the data plane. The agility, scalability, and adaptability that this design fosters make it especially effective in contemporary network settings with fluctuating workloads and a wide range of applications.

Data Plane: The data plane is in charge of sending data packets, which make up network traffic, between network components like switches and routers. Data plane devices are managed by the central SDN controller in an SDN architecture.

Control Plane: The control plane manages network policies, configuration settings, and routing choices in conventional networking. In SDN, the SDN controller, which dynamically controls and configures network devices, abstracts and centralizes the control plane.

SDN Controller: At the center of the SDN architecture is the SDN controller. It is a piece of software that is in charge of assigning configuration instructions to network devices, enforcing policies, and making high-level judgments regarding network traffic[15]. Network administration is made possible by controllers by enabling programmable interfaces and providing a centralized view of the network.

Southbound APIs are interfaces that connect the SDN controller to the data plane's underlying network devices. They provide the controller with access to switches, routers, and other networking devices, allowing the controller to manage device behavior and govern traffic flow.

Northbound APIs: These interfaces let higher-level applications and network management systems access the SDN controller's capabilities. Northbound APIs give programmers the tools they need to design unique apps that can connect with and govern the network according to their own requirements.

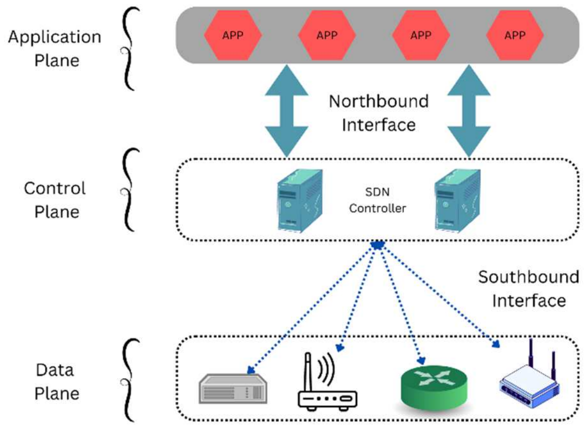

Fig. 1: Layout of Software-Defined Network

B. Research Aim & Objectives

In the framework of Software-Defined Networking (SDN), the main objective of this research is to design and examine an Adaptive Load Optimisation Technique for Multimedia Applications with Multi-Controller Utilisation. With regard to dynamic multimedia traffic patterns, fluctuating Quality of Service (QoS) needs, and the necessity for effective resource allocation in contemporary networking systems, this study aims to solve these issues. Designing, implementing, and testing a novel SDN-based strategy with the capacity to adaptively distribute network resources in order to improve the performance and scalability of multimedia applications while preserving ideal QoS levels is the key objective.

- We have used user selection to lower the traffic rate and remove the less energy-efficient users from the network in order to improve QoS.
- We have kept switches at a threshold level that, when exceeded, reroutes the surplus flows to other entities in order to increase the QoE, response time, processing speed, and decrease the buffer rate.
- Hybrid load balancing is used to maintain a balanced load across controllers and gain improved precision in load balancing. Finally, we have connected the edges with controllers that aid in scaling to decrease the communication time between the data plane and control plane.

C. Research Motivation

The rising importance of multimedia applications in today's connected environment is what inspired this research. Virtual reality experiences, online gaming, and video conferencing are just a few of the services that fall under the umbrella of multimedia apps. A strong and responsive network infrastructure is essential to enable these apps as they grow more and more intertwined into our personal and professional life.

Traditional network designs sometimes find it difficult to satisfy the special requirements put forward by multimedia applications. These apps have dynamic traffic patterns that, if poorly managed, can quickly overload network resources and damage user experiences. In addition, they frequently need strict Quality of Service (QoS) standards to guarantee continuous, trouble-free operation. It has been difficult to meet these objectives using traditional network models.

The idea behind this study is that Software-Defined Networking (SDN) has the ability to fundamentally alter how networks are managed and optimised for multimedia applications. SDN offers a viable framework for managing QoS and dynamic resource allocation by offering centralised programmability and separating the control and data planes. Realising this promise, however, necessitates the creation of cutting-edge methods that can adapt to the dynamic nature of multimedia traffic and effectively use network resources.

This research's multi-controller utilisation component is also driven by the requirement for SDN scalability. Traditional single-controller designs may fail to deliver the necessary responsiveness and fault tolerance as networks get bigger and more complex. If properly developed, a multi-controller strategy can reduce the load of control and improve network stability.

This study intends to investigate, develop, and evaluate an adaptive load optimisation technique for multimedia applications with multi-controller utilisation in SDN in light of these driving factors. By doing this, it hopes to create networking technologies that can accommodate the changing needs of multimedia applications, thereby raising the calibre and dependability of multimedia experiences in our networked society.

D. Paper Organization

The paper is further divided into a number of sections, which are defined as follows: The state of the art and its limitations are exemplified in Section II. Section III identifies the main issues with load balancing in an SDN system. The suggested adaptive load optimisation model technique, which combines mathematical equations, pseudocode, and algorithm processes, is demonstrated in Section IV. The simulation setup, comparative findings, and a summary of the research are all included in Section V of the proposed work. The model is concluded in Section VI, which also outlines areas for future research.

II. LITERATURE SURVEY

By moving the switches to an underloaded controller in this study, load balancing between many controllers was accomplished. In order to achieve this, they used the double deep Q network (DDQN), a deep neural network and reinforcement learning technique, to do away with the extra computation a single DQN made and used in the global controller. While the second DQN (target) predicts the Q values of the targeted domain and is trained by the replay memory method, the first DQN (learning) was used to choose the switch migration and to intimate the system settings. Finally, a greedy algorithm is used to analyse each packet in messages received to arrive at the best outcome[1]. Srisam et al [2] By reducing the maximum load/burden on the overloaded controller, the authors of this research have balanced the load among SDN controllers. Here, the load on the controller was calculated by taking into account various metrics, including the number of packets in each message the controller received, the communication latency, and the processing time. In this study, a load minimization-based optimisation approach was applied to decrease the processing time required by the integer linear programming problem (ILP). Here, by transferring the flows from the old controller to the new controller, ILP was employed to find the optimal solution for load imbalance issues. By estimating the load on the controller using LSTM (long term prediction) and autoregressive integrated moving average (short term prediction) time series methods, the authors of this study accomplished switch migration prior to the occurrence of the event. On the other hand, the forwarding layer devices that need to be migrated were described as a nonlinear binary issue, and its NP-hard challenges are alleviated by utilising an RL method known as win stay loss shift. Here, the global controller predicts the load of the controller based on the past load history of each controller and the requests given by the switches/packet in messages [3].

Adekoya et al [4] study, a brand-new architecture built on switch migration is suggested to deal with the massive fluxes entering the network to solve load balancing issues. By taking into account the average load and variation of the controllers, this framework maps the underutilized controllers. Three sub-modules, including load estimate, forwarding device selection, and optimum controller selection, were implemented in each controller. An ideal controller is picked by taking into account the variance computed from the group of underloaded controllers, and at the same time, the forwarding device with the highest volume of flow requests is chosen for migration. Yeo et al [5] achieve an equitable load among dispersed controllers in this study, the authors chose a switch migration strategy based on reinforcement learning. This requires migrating the switch, and the targeted controller was selected based on the parameter of the resource utilization ratio of various resource categories of both controllers and switches. Additionally, in order to determine the precise controller for load balancing in terms of migration, forwarding devices served as the RL agent, and all switch migration options were taken into account based on the smallest action sets (number of controllers). In this study, a paradigm for accomplishing load balancing was provided by anticipating the demand on controllers and identifying the advantages of moving switches sooner. The observation framework, event triggering framework, and action execution framework are the three main sub-frameworks that are implemented at the control layer. In the first framework, switches' packet-level communications are taken into account, while in the second framework, Taylor's equation is utilized to forecast the load behaviors of SDN controllers. Additionally, a benefit evaluation is done here, which indicates the least amount of loss caused by a migration device. In order to create a balanced load among the controllers, the job of the controller is ultimately modified through migration [6]. To reduce the time that passes between the data and control planes, authors have developed a method based on flexible load balancing by predicting the state of the connections. An upgraded version of the LSTM type of recurrent neural network, which was deployed at the application layer, was used in this case to anticipate the link status. In order to train the LSTM model, the controller first keeps track of the behaviors of the data plane and saves this information in a database. Additionally, a greedy method known as Dijkstra was used to estimate the optimum communication channel among the entities. The controllers also install flows depending on the projected values (Dijkstra weight) [7]. Lai et al [8] Switch migration has been used by the writers of this study to distribute the load across different controllers more evenly. For this, two distinct controllers simultaneously controlled the packet and message flows of a forwarding device. For aiding reasons, this technique selects a severely loaded controller and a lightweight controller, fixes them together as a pair, and calculates the migrating time. They were able to retain the controllers at a certain threshold while also lowering the cost of switch migration and lowering switch oscillation conditions. In this work, load balancing is accomplished with the aid of a preference-based solution architecture, which selects the best controller by taking the distance for accepting migrating flows into account. For this, the workload of each controller is used to fix a dynamic threshold. The target controller, on the other hand, was carefully picked based on many load-oriented factors including the number of devices under a controller, CPU, and incoming packet rate. Additionally, a hierarchy function was employed to provide the percentage of each attribute (qualitative and quantitative) specified. Eventually, migration is performed on the forwarding device with the lowest load [9]. By moving switches from overloaded controllers to underloaded controllers, the authors of this study have developed a framework for balancing the load across dispersed controllers. In order to lessen switch migration issues, many agents-based reinforcement learning (DRL) techniques was implemented at the controller layer. Additionally, switch migration is carried out when the workload exceeds the predetermined threshold, and additional factors like memory, bandwidth, and CPU utilization are also taken into account [10]. By choosing the overloaded controller and underloaded controller by hand, the authors of this study present a framework based on experience and prediction to solve the load balancing problems that occur in a multi-controller environment. The load state was ascertained by considering the call arrival ratio of switches and the packet loss ratio of controllers (prediction). Here, controllers recorded the number of calls that were made at each interval, and using this information, the packet loss rate was forecasted. In addition, three sub-modules called severely loaded controller data, target controller estimation, controller & switch mapping, and new connection setup were active to provide load balancing [11]. In this study, load balancing across distributed controllers was accomplished by dynamically moving the switches with the fewest flow requests, an artificial fish swarm, and an improved ant colony optimization method. By gathering topological information, the best controller to accept the migration was selected based on these two optimization techniques. On the other side, the commuting salesman problem was used to identify switch migration issues. In the end, the authors took into account a number of factors to calculate the loads, including the delay of the forwarding device and controller and the total number of flow requests [12].

Table I. Comparison of Technqiues

Techniques	**Approaches**	**Controller Selection Criteria**	**Load Estimation Methods**
Double Deep Q Network (DDQN)	Switch migration based on Q value predictions, historical load	Reinforceme nt learning, historical load data,	LSTM, Autoregressiv e Integrated Moving

	data, and packet analysis	packet analysis	Average, Long-term and Short-term predictions
Load estimation,Optimum controller selection	Mapping underutilized controllers, Variance computation, Forwarding device with highest flow requests	Variance computation, Volume of flow requests	Load estimation modules, LSTM for load prediction
Reinforcement Learning, Resource Utilization Ratio	Switch migration based on resource utilization ratio, RL agent for forwarding device selection	Resource utilization ratio, Historical load data	Taylor's Equation, Benefit evaluation
Flexible Load Balancing, LSTM-based Prediction	LSTM for link status prediction, Greedy method for optimal communication channel	Historical data of data plane behaviors	LSTM for link status prediction, Dijkstra Algorithm
Preference-based Solution Architecture, Load Thresholds, Dynamic Thresholds	Pairing severely loaded and lightweight controllers, Dynamic threshold calculation	Workload, Number of devices, CPU, Incoming packet rate	Dynamic thresholds, Hierarchy function
Multi-Agent Reinforcement Learning (DRL)	DRL techniques at controller layer, Threshold-based migration	Workload, Memory, Bandwidth, CPU utilization	Threshold-based workload analysis
Experience-based, Prediction-based	Manual selection of overloaded and underloaded controllers, Load state prediction	Call arrival ratio of switches, Packet loss ratio of controllers	Historical data, Prediction models
Improved Ant Colony Optimization	Switch migration based on optimization methods, Commuting Salesman Problem for migration routes	Topological information, Delay of forwarding device and controller, Total number of flow requests	Optimization methods, Commuting Salesman Problem

III. Problem Statement

A new age of digital communication and entertainment has arrived because of the explosion of multimedia applications, including video streaming, online gaming, virtual conferences, and augmented reality. These applications provide considerable problems to traditional network designs because to their unpredictable traffic patterns, diverse Quality of Service (QoS) needs, and latency sensitivity[16]. Traditional networking paradigms frequently have trouble effectively managing and optimizing network resources to meet the special requirements of multimedia traffic, which results in subpar user experiences and the underuse of resources [17].

In the context of multimedia applications, Software-Defined Networking (SDN) has become a possible remedy for conventional networking's drawbacks. The opportunity to centralize control, give programmability, and enable dynamic resource allocation and traffic-condition adaptation is made possible by SDN. However, a number of significant obstacles still exist:

Dynamic traffic management: The traffic loads produced by multimedia applications are unexpected and fluctuating. The flexibility necessary to effectively handle these changing traffic patterns in real-time is sometimes lacking in current SDN systems [17][18].

Quality of Service (QoS) Optimisation: For multimedia applications, maintaining constant QoS levels is crucial. Realising good QoS for different kinds of multimedia traffic while maintaining effective resource allocation is still a difficult task[19].

Network Scalability: Single-controller SDN designs may encounter scalability issues as networks develop to support an increasing number of users and devices. The challenge is to retain performance and responsiveness while scaling SDN across big and complicated networks.

Controller Coordination: For network stability and dependability in a multi-controller SDN environment, establishing smooth coordination between controllers and effectively distributing control decisions is crucial[20][21].

To provide a novel adaptive load optimization technique for multimedia applications inside SDN, with an emphasis on multi-controller utilization in particular, is the issue statement for this project. This method should provide effective resource allocation, dynamic traffic management, and QoS optimization while also guaranteeing the scalability and synchronization of numerous controllers. The goal is to improve multimedia applications' overall performance, dependability, and quality in contemporary networking contexts.

IV. Proposed Adaptive Load Optimization Model

The major goal of the proposed work is to balance the load on the various SDN controllers. In order to achieve effective load balancing, many technologies, including edge computing and network function virtualization, have offered their assistance. Three levels make up this research project: the user plane (U-Plane), where SDN network users participate, the data plane (D-Plane), where open flow switches are present, and the control plane (C-Plane), where intelligent controllers operate. Three main procedures are used to make the research work novel.

V. Implementation & Results

The simulation outcome of the suggested study was conducted using a Network Simulator (NS-3.26), which significantly improves the research efficacy. The suggested model is evaluated using several performance criteria, demonstrating that our approach achieves improved performance. The present study aims to conduct a comparative analysis between the proposed model and current approaches, namely MARVEL and Round Robin. The primary objective of this study is to optimize and augment load distribution across many controllers, consequently leading to improvements in reaction time and processing speed. The proposed study demonstrates improved performance across many performance parameters, including total load, average packet loss, switch migration, and throughput.

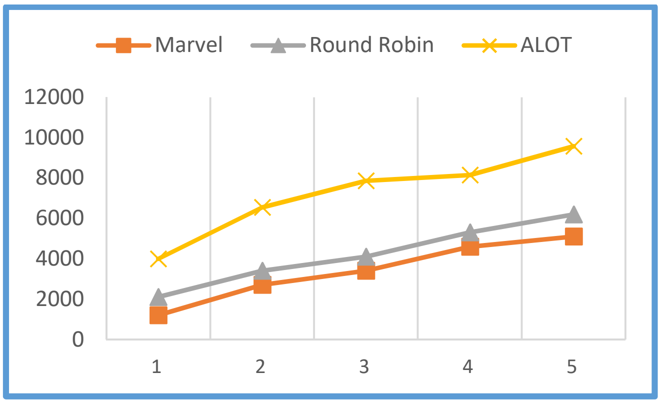

Fig. 2: Total Load vs Number of Arrived Users

In the above figure 2 shows the total load analyzed by the number of arrived users on the MARVEL, Round robin, and ALOT technique, and its shows the ALOT technique is far better than the other three models.

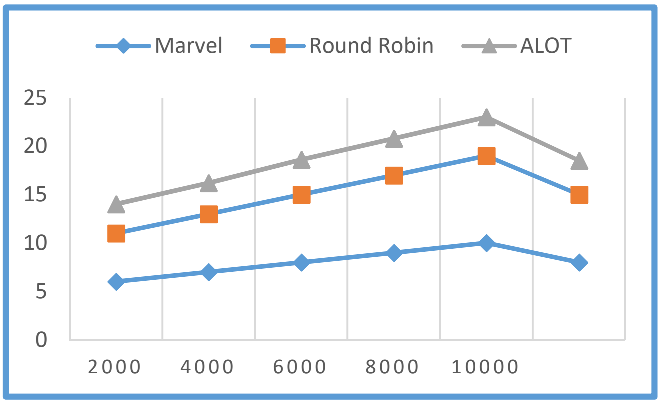

Fig 3: Average Packet Loss vs incoming Load

In Figure 3 ALOT technique performance is evaluated on the basis of average packet loss vs the incoming load with the MARVEL and Round Robin technique there is the A LOT performing the low packet ratio as compared to the other models.

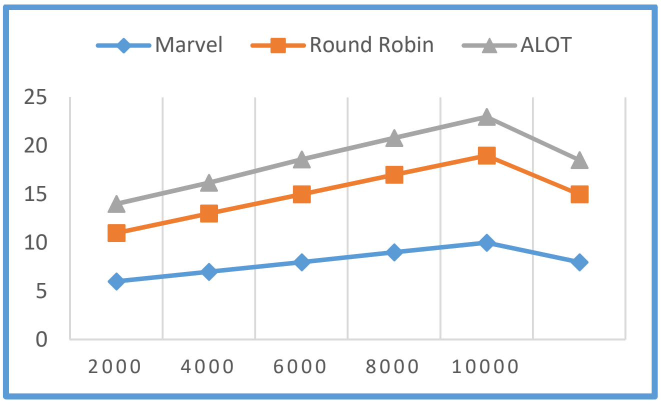

Fig 4: Incoming Load vs Number of Migration

In Figure 4 shows the A LOT model handle the load versus number of switched migration in the network and the above graph shows the A LOT performa better than the MARVEL and Round robin model.

VI. Conclusion

This article presents the ALOT model, a unique method to load optimization in software-defined networking that was developed by the authors. The performance of the ALOT model is assessed by a comparison study using well-established methods like as MARVEL and Round Robin. These methods are used to evaluate how efficiently the ALOT model manages load management. The findings provide compelling evidence that the ALOT model is preferable in terms of both the average packet loss and the frequency of switch migrations. Despite this, there are still areas for improvement that need to be addressed, such as improving the network's capacity for clear and efficient communication.

References

[1] Xiang, Min, et al. "Deep Reinforcement Learning-based load balancing strategy for multiple controllers in SDN." e-Prime-Advances in Electrical Engineering, Electronics and Energy 2 (2022): 100038.

[2] Srisamarn, Ukrist, Lapas Pradittasnee, and Nattapong Kitsuwan. "Resolving load imbalance state for SDN by minimizing maximum load of controllers." Journal of Network and Systems Management 29.4 (2021): 46.

[3] Filali, Abderrahime, et al. "Preemptive SDN load balancing with machine learning for delay sensitive applications." IEEE Transactions on Vehicular Technology 69.12 (2020): 15947-15963.

[4] Adekoya, Oladipupo, Adel Aneiba, and Mohammad Patwary. "An improved switch migration decision algorithm for SDN load balancing." IEEE Open Journal of the Communications Society 1 (2020): 1602-1613.

[5] Yeo, Sangho, et al. "Achieving balanced load distribution with reinforcement learning-based switch migration in distributed SDN controllers." Electronics 10.2 (2021): 162.

[6] Zhong, Hong, et al. "Assessing Profit of Prediction for SDN controllers load balancing." Computer Networks 191 (2021): 107991.

[7] Chen, Junyan, et al. "ALBLP: adaptive load-balancing architecture based on link-state prediction in software-defined networking." Wireless Communications and Mobile Computing 2022 (2022): 1-16.

[8] Lai, Wei-Kuang, et al. "TSSM: Time-sharing switch migration to balance loads of distributed SDN controllers." IEEE Transactions on Network and Service Management 19.2 (2022): 1585-1597.

[9] Ider, Masoud, and Behrang Barekatain. "An enhanced AHP–TOPSIS-based load balancing algorithm for switch migration in software-defined networks." The Journal of Supercomputing 77 (2021): 563-596.

[10] Sun, Penghao, et al. "MARVEL: Enabling controller load balancing in software-defined networks with multi-agent reinforcement learning." Computer Networks 177 (2020): 107230.

[11] Banerjee, Anuradha, and DM Akbar Hussain. "EXPRL: experience and prediction based load balancing strategy for multi-controller software defined networks." International Journal of Information Technology (2020): 1-15.

[12] Fu, Yanfang, et al. "Multi-Controller Load Balancing Algorithm for Test Network Based on IACO." Symmetry 13.10 (2021): 1901.

[13] Li, Zhuo, et al. "SDN controller load balancing based on reinforcement learning." 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS). IEEE, 2018.

[14] Wang, Chuan'an, et al. "A switch migration-based decision-making scheme for balancing load in SDN." IEEE Access 5 (2017): 4537-4544.

[15] P. S. Tiwana and J. Singh, "Load Optimization Accessions, Ramification the QoS in Software Defined Networking," 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), Greater Noida, India, 2023, pp. 1013-1017, doi: 10.1109/AISC56616.2023.10085328.

[16] Wadhwa, Heena, and Rajni Aron. "TRAM: Technique for resource allocation and management in fog computing environment." The Journal of Supercomputing 78.1 (2022): 667-690.

[17] Long, Hui, et al. "LABERIO: Dynamic load-balanced routing in OpenFlow-enabled networks." 2013 IEEE 27th international

conference on advanced information networking and applications (AINA). IEEE, 2013.

[18] Zakia, Umme, and Hanene Ben Yedder. "Dynamic load balancing in SDN-based data center networks." 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE, 2017.

[19] Chahlaoui, Farah, Mohammed Raiss El-Fenni, and Hamza Dahmouni. "Performance analysis of load balancing mechanisms in SDN networks." Proceedings of the 2nd International Conference on Networking, Information Systems & Security. 2019.

[20] H. Selvi, G. Gür and F. Alagöz, "Cooperative load balancing for hierarchical SDN controllers," 2016 IEEE 17th International Conference on High Performance Switching and Routing (HPSR), Yokohama, Japan, 2016, pp. 100-105, doi: 10.1109/HPSR.2016.7525646.

[21] W. Chen, C. Chen, X. Jiang and L. Liu, "Multi-Controller Placement Towards SDN Based on Louvain Heuristic Algorithm," in IEEE Access, vol. 6, pp. 49486-49497, 2018, doi: 10.1109/ACCESS.2018.28679