Scientific Research Publishing

Journal of Computer and Communications, 2024, 12, 49-66
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Review of Load Balancing Mechanisms in SDN-Based Data Centers

Qin Du, Xin Cui, Haoyao Tang, Xiangxiao Chen

School of Computer Science and Technology, Shandong University of Technology, Zibo, China
Email: 3098732609@qq.com

How to cite this paper: Du, Q., Cui, X., Tang, H.Y. and Chen, X.X. (2024) Review of Load Balancing Mechanisms in SDN-Based Data Centers. *Journal of Computer and Communications*, **12**, 49-66.
https://doi.org/10.4236/jcc.2024.121004

Received: December 11, 2023
Accepted: January 12, 2024
Published: January 15, 2024

Abstract

With the continuous expansion of the data center network scale, changing network requirements, and increasing pressure on network bandwidth, the traditional network architecture can no longer meet people's needs. The development of software defined networks has brought new opportunities and challenges to future networks. The data and control separation characteristics of SDN improve the performance of the entire network. Researchers have integrated SDN architecture into data centers to improve network resource utilization and performance. This paper first introduces the basic concepts of SDN and data center networks. Then it discusses SDN-based load balancing mechanisms for data centers from different perspectives. Finally, it summarizes and looks forward to the study on SDN-based load balancing mechanisms and its development trend.

Keywords

Software Defined Network, Data Center, Load Balancing, Traffic Conflicts, Traffic Scheduling

1. Introduction

In recent years, the continuous expansion of Data Center Networks (DCN) scale, coupled with the dynamic evolution of network requirements, has resulted in a perpetual increase in the volume of data requiring transmission and processing. This has led to progressively stringent demands on network bandwidth. Traditional DCNs, designed for data forwarding and control, rely on specific network devices. However, these devices necessitate exceedingly complex protocols, making it particularly challenging for engineers to deploy, maintain, manage, and scale network operations. This challenge hinders the ability to meet the escalating demands of networks and effectively address various issues arising in dy-

namic network environments [1].

Software Defined Network (SDN) is a new network architecture that decouples the data plane and control plane in traditional networks through OpenFlow and provides communication links between the control and data layers [2]. The controller has a global perspective in the network. it can control all network devices and has an open, programmable interface so administrators can manage the network flexibly and achieve more granular traffic control and function customization. These characteristics not only reduce the difficulty and complexity of management but also meet the individual needs of more users for network services. Traditional DCNs are unable to achieve load balancing (LB) due to the absence of a global network structure and resource view. In contrast, SDN with its global perspective and open programmable interfaces, enables more effective and flexible LB, meeting the evolving needs of data center development.

Despite the improvements in network performance that LB can bring to SDN-based data centers, further research is still necessary. For instance, in data center networks, two types of traffic may coexist simultaneously [3], Ignoring the characteristics of this traffic could lead to network load imbalance [4]. This issue can be addressed by judiciously allocating network resources to avoid congestion and conflicts in network traffic. Additionally, dynamic scheduling and adjustment of network traffic based on specific business requirements and network conditions can be implemented to reduce transmission latency in data centers. Therefore, this paper will comprehensively introduce SDN-based data center LB mechanisms from multiple perspectives, discussing the significant trends and challenges of SDN LB. This will provide valuable insights to researchers aiming to enhance SDN performance.

The structure of this study can be organized as follows. First section 2 elucidates the architecture of Software-Defined Networking (SDN). Then section 3 delves into the application scenarios of SDN-based data center load balancing mechanisms, encompassing aspects such as traffic classification, traffic prediction, traffic table management, and traffic scheduling. Then section 4 expounds on recent trends and prospective research directions in this domain. Finally, the study culminates with conclusions presented in Section 5.

2. Software Defined Network Architecture

SDN is a network architecture that provides network administrators with greater flexibility and control by separating network control from the underlying hardware. The emergence of this architecture has fundamentally addressed the issues of rigidity and lack of flexibility inherent in traditional network architectures, enabling networks to dynamically and flexibly adjust and manage based on application requirements. The SDN architecture is divided into three layers: data layer, control layer, and application layer, as shown in **Figure 1**.

- Data layer: It includes network devices such as switches and hosts. The primary function of the switch is to collect network data and pass it to the controller in

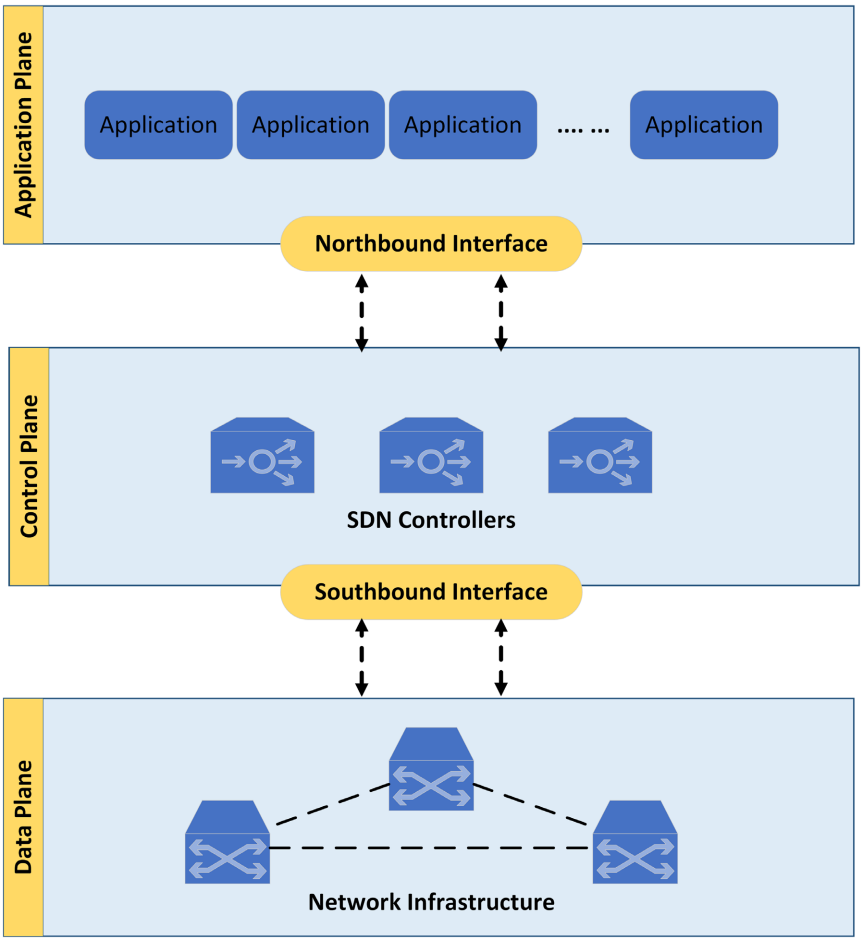

Figure 1. SDN architecture.

the control layer through the southbound API.

- Control layer: It serves as the "brain" of the SDN architecture. The controller on the control layer has a global view of the network and can control network devices globally. Interacting with the application layer through the northbound API and controlling the network devices on the data layer through the southbound API.
- Application layer: Comprising multiple applications designed to meet user requirements, the application layer is managed by network administrators. They interact with the controller through northbound APIs to enable more user-friendly operations.

3. Review of Load Balancing Mechanisms in SDN-Based Data Center

This study will introduce SDN-based data center mechanisms from five aspects: traffic classification, prediction, table management, scheduling, and others.

3.1. Traffic Classification

By employing traffic classification to identify distinct types of traffic, and subse-

quently managing and optimizing them based on their characteristics and requirements, it is possible to enhance the management of network resources, prevent conflicts, and thereby achieve LB.

Several studies have proposed using optimization algorithms to solve the traffic conflict problem. Hamdan *et al.* [5] utilized classifiers on SDN controllers and switches to detect elephant traffic (ET) and filter out most of the candidate ETs for accurate and effective detection. However, this approach may face scalability and real-time operation challenges in the large-scale networks.

Abdollahi *et al.* [6] applied the backpack model to model link bandwidth and incoming traffic and assigned service type values to each traffic through an SDN controller. Particle Swarm Optimization (PSO) algorithm is used to optimize the knapsack problem for more efficient network traffic management and better performance. Nonetheless, the complexity of this algorithm and the difficulties in tuning it could restrict its application in dynamic network environments.

Khairi *et al.* [7] proposed a method to identify and classify conflicting traffic using machine learning (ML) algorithms, which can accurately detect and classify different types of conflicting traffic based on the characteristics of the traffic rules, such as priority, action, protocol, and IP source address. However, it necessitates extensive data for training and must address the challenge of real-time model updating.

Xie *et al.* [8] analyzed the features of packets to pre-identify ETs. Then, critical features of the first n packets are extracted to predict their traffic size. This method not only improves the management efficiency of network traffic but also helps to optimize the network performance. However, challenges remain in the accuracy of early predictions and processing speed within high-speed networks.

Sun *et al.* [9] determined the basic threshold based on statistical rules. If the number of packet's bytes (PB) exceeds the threshold, it is a suspicious ET. Otherwise, the number of times the PB value does not exceed the threshold is counted in T time. If the maximum tolerance value is not reached, it is an actual ET. Otherwise, the ET produces degradation, and it is removed from the set. Incremental learning training data is also introduced as a way to improve the adaptability of the dynamic data model. However, the threshold setting and strategies for handling deteriorated flows require further refinement.

Diel *et al.* [10] used a supervised data classification algorithm to detect real-time direct current congestion and categorize it into four main classes. It then used actor-critic reinforcement learning (RL) to find better Transmission Control Protocol (TCP) parameters. This methodology, while innovative, is not without its challenges, particularly in terms of data dependency and the complexity of the algorithm.

3.2. Traffic Forecast

By forecasting traffic, proactively managing resource allocation and LB, conflicts arising from sudden traffic fluctuations can be avoided.

Liu *et al.* [11] used the Deep residual network architecture to predict the occupancy of network links in the next time window. Then, based on the predicted network conditions, the algorithm automatically selects the best traffic routing to realize efficient network traffic management. However, this approach may face issues of complexity and adaptability.

Begam *et al.* [12] proposed a multiple regression-based search (MRBS) algorithm to select servers for traffic prediction through regression analysis. However, this method might exhibit limited accuracy in handling complex network data.

Nougnanke *et al.* [13] combined an intelligent buffering scheme with an online network optimization algorithm. A time prediction model was also designed using random forest regression based on the constructed dataset. The model can accurately predict performance, thus providing an efficient and accurate solution for data centers, yet its implementation is complex and relies on specific datasets.

Hai-Anh *et al.* [14] applied the Long Short-Term Memory algorithm and its variants to the original Backward Congestion Notification algorithm to increase the prediction capability of the initial link congestion. The algorithm is then implemented in a software-defined DCN.

Xie *et al.* [8] pre-identified ETs by capturing the stateless characteristics of the first packet of network traffic arriving at a switch and extracting the characteristics of the first n packets to predict the size of the ET. This approach helps to optimize network performance while improving the efficiency of network traffic management.

Shaikh *et al.* [15] trained a Bayesian network model through RL to predict network congestion accurately. The efficient, intelligent, and accurate framework can significantly improve network performance and stability.

However, these methods necessitate extensive training and computational resources, and the training process itself can be quite intricate."

Wei *et al.* [16] used Deep Learning (DL) techniques to categorize traffic types and predict traffic size finely. Compared with traditional methods, the algorithm has significant advantages in traffic scheduling, can better adapt to the dynamically changing network environment, and improves the network performance and resource utilization of data centers. However, it may also face challenges in practical deployment and data handling.

3.3. Traffic Table Management

In SDN, traffic tables serve as crucial components guiding the behavior of data traffic. Managing and optimizing traffic table rules contribute to achieving LB.

Gonzalez-Diaz *et al.* [17] utilized the octet of the MAC address to encode address format identifiers that enable the identification of frame types. They used two or three zone identifiers to locate servers and switches in a typical topology. They also used one or more traffic IDs to identify specific traffic or traffic types, which enables fine-grained traffic management, sophisticated quality of service (QoS) management, and advanced resource allocation. However, this metho-

dology may rely on specific hardware environments and pose scalability challenges in large-scale networks.

Zhou *et al.* [18] avoided traffic conflicts by dynamically updating the source address validation information at downstream network ingress ports, employing a fine-grained two-layer structure to flexibly match traffic entries, adding a priority-based validation mechanism, and designing a state partitioning and transition module in order to optimize the network performance under anomalous conditions and to improve the performance and security of the network. Nonetheless, this approach could complicate network management and pose challenges in real-time data processing.

Shen *et al.* [19] analyzed the life cycle of traffic table entries and observed the actual workload of DCNs by building a queuing theory-based traffic table state estimation model to find out the probabilistic characteristics of routing policies and calculate the critical parameters in the model. Finally, the optimal traffic table states obtained from the model are utilized to improve routing decisions, avoid traffic conflicts, and improve network performance. This approach provides new ideas and methods for traffic table state estimation and routing decisions in SDN switches, although its effectiveness may be limited by the model's accuracy and understanding of actual workloads.

Zhang *et al.* [20] determined whether there is a conflict between policies based on the service bandwidth requirement and the capacity of the network function. Next, conflict resolution is transformed into an optimization problem to maximize the number of policies that satisfy the service initiator's intent and minimize the bandwidth loss for high-priority users. However, it's worth noting that this method may entail complex computations and pose challenges in balancing interests among diverse users in practical applications.

Liang *et al.* [21] constructed an SDN traffic rule conflict detection knowledge graph to store network information, including traffic rules, and then built production rules based on the definitions of traffic rule conflicts (single-table conflicts and multi-table conflicts). This approach makes the generative rules easy to read and modify and also helps us better understand and optimize the performance and behavior of SDN networks and improve the reliability and stability of the network. However, this method might require the maintenance of a large set of rules and data, with additional efforts required for updates and modifications.

3.4. Traffic Scheduling

Traffic scheduling is a crucial aspect in network communication, achieving LB through the rational allocation and transmission of data traffic within the network. Subsequently, traffic scheduling methods will be introduced, including those based on heuristic algorithms, intelligent algorithms, and other approaches.

3.4.1. Traffic Scheduling Based on Heuristic Algorithms

In recent years, routing strategies based on heuristic algorithms (HA) have

found widespread application in addressing traffic scheduling issues, with the majority of these strategies adopting ant colony optimization (ACO) algorithm. However, ACO exhibit certain limitations. Consequently, numerous have been conducted to enhance and improve ACO. Zhu *et al.* [22] aimed to maximize the average link bandwidth utilization. They transformed the traffic scheduling problem into an integer linear programming model and solved the rerouting paths for large traffic by redefining the pheromone updating method in the ACO. This approach not only improves the utilization of link bandwidth but also helps to reduce the risk of traffic conflicts. However, it may encounter performance challenges when applied to significantly large-scale networks. Zheng *et al.* [23] considered the performance parameters of network links and servers and followed the principle of selecting links and servers with low utilization. ACO is used to find the global optimal solution by designing an evaluation method for the server and link modules. At the same time, Kent's Chaos model is used to interfere with the ant colony's transfer probability to enhance the algorithm's solving ability and robustness. However, this approach could increase the algorithm's complexity and computational load.

In addition, research has explored improvements to ACO by integrating them with other algorithms. Dai *et al.* [24] improved the Differential Evolution (DE) by optimizing the objective function to compute multiple available candidate paths, which are used as initialized global pheromones for the ACO, and then find the globally optimal routes for forwarding the ET. However, this method might require additional computational resources and time to find the optimal solution.

PSO is also a common method in traffic scheduling. Ma *et al.* [25] evaluated the degree of matching between the bandwidth demand of network traffic and the current link bandwidth resources by establishing an objective function. Then, the shortest set of paths is found, and the particle aggregation degree is defined to determine whether the algorithm falls into a local optimum. Finally, the optimized PSO selects the best scheduling path for network traffic from the shortest path. This method boasts rapid adaptation to network environmental changes and effective traffic scheduling. Nonetheless, it may encounter challenges with local optima in highly dynamic network environments. Abdollahi *et al.* [6] work by constructing a network knapsack model that compares link bandwidth to knapsack capacity and incoming traffic to items. The model accurately obtains the service value type by analyzing the traffic size and the SDN controller's decision. In order to find the optimal scheduling path, the model uses PSO to optimize the knapsack problem. Forwarding the selected and unselected traffic is the first choice in the optimization process. However, challenges may arise in terms of adaptability and efficiency under real network conditions.

Researchers have also explored and adopted novel. HA. Xu *et al.* [26] utilized the global exploration and local mining capabilities of the spider monkey optimization (SMO) to use the link idle rate as the fitness value for each path and

introduced adaptive weights to evaluate and update the paths dynamically. The path with the most negligible link occupancy is selected as the optimal forwarding path. Ye *et al.* [27] proposed an integrated optimization method with generalizability, combining adaptive LB and heuristic path selection (ILBPS). After connecting the prediction results and calculating the link weights, this method designs a heuristic path selection method (ABCSP) based on the Artificial Bee Colony (ABC) algorithm to compute the optimal paths, and then, these optimal paths will be sent to the data plane. However, these novel algorithms may require extensive experimentation and tuning to ensure effectiveness under various network conditions.

Furthermore, the multipath routing and LB algorithms for SDN based on ABC [28] and the Bird Migration Algorithm [29] proposed by Perepelki *et al.* both apply the algorithms to network topology instances employing path encoding to find the shortest path from the source node to the destination node. It provides a new idea to solve the problem of applying HA to SDN.

The application of HAs in traffic scheduling has expanded from traditional approaches to a variety of novel algorithms. These new algorithms optimize search processes by simulating various natural phenomena or behavioral patterns, offering fresh perspectives and solutions for addressing intricate traffic scheduling problems.

3.4.2. Traffic Scheduling Based on Intelligent Algorithms

In the current complex network environment, the increasing attention to traffic scheduling based on intelligent algorithms is driven by the rapid development of artificial intelligence (AI) and ML.

Traffic scheduling based on RL involves training intelligent agents to learn optimal routing strategies through interactive processes with the environment, aiming to achieve LB, reduce latency, and minimize packet loss. Balakiruthiga *et al.* [30] proposed a model consisting of three independent Q Reinforcement Learning agents responsible for controller localization, routing, and LBs, enabling fast and intelligent decision-making. However, it may necessitate a complex coordination mechanism to ensure effective cooperation among the agents. Gao *et al.* [31] captured historical traffic matrices and monitored link utilization through an SDN controller. RL components are then utilized for dynamic prediction, and traffic scheduling decisions are output accordingly. Nevertheless, reliance on historical traffic data may limit the algorithm's effectiveness in rapidly changing network environments. Yang *et al.* [32] proposed an RL-based approach to the problem, stating that current solutions mainly focus on selecting storage nodes but neglect the storage method. They delicately constructed the storage problem as a joint optimization problem of data storage and traffic management. Then, a solution design based on multi-intelligent body Q-learning is proposed. However, the complexity of implementation may pose challenges in environments with limited storage resources.

Deep Reinforcement Learning (DRL) based traffic scheduling combines deep

neural networks (DNN) and RL that can deal with complex, nonlinear problems and is more advantageous when dealing with complex network environments. Zeng *et al.* [33] combined Graph Convolutional Neural Network (GCN) and DRL. Due to the ability of GCN to sense the network state and topology, it can be used to realize LB by making intelligent routing decisions. The algorithm can sense the network topology information and make superior decisions compared to AI-based algorithms. However, the complexity of the deep learning model might result in significant computational burdens and require extensive training data.

Liu *et al.* [34] reconfigured cache and bandwidth resources by quantifying the contribution scores of cache and bandwidth in reducing delay. A routing scheme with a resource reorganization state is also proposed to adaptively route appropriately according to the network state. Yet, its reliance on precise resource assessment may make accurate implementation challenging in real-world network settings.

Xu *et al.* [35] fully utilized the known parametric quantities of Distributed DL traffic to discover congestion and reroute them via an ECMP-like hash function to compensate for the bandwidth of slower traffic. This scheme can effectively balance the network load and improve network performance and stability. However, frequent updates and adjustments might be necessary in highly dynamic network environments.

Zhao *et al.* [36] first designed the multicast tree state matrix, link bandwidth matrix, link delay matrix, and link packet loss rate matrix as the state space of the DRL agent. Action selection strategies for four scenarios were developed to add these links to the current multicast tree. Then, two forms of single-step and final reward functions are designed to guide the intelligent body in making decisions and constructing the optimal multicast tree. However, the complexity of the decision-making process could reduce computational efficiency.

Sharma *et al.* [37] trained a Bayesian network to predict congestion through RL and made optimal decisions based on the predictions. They also effectively dealt with the congestion problem by automatically adjusting the parameter weights of the controller. However, this mechanism for auto-adjusting parameter weights might require continual adjustment under varying network conditions.

Gao *et al.* [38] utilized in-band network telemetry (INT) to collect the internal state of devices across the network and adopt an energy-saving algorithm based on DRL to make quick decisions on whether to open or close network device ports based on the real-time network state. The trained system can adaptively adjust its energy-saving strategy when the DCN topology changes. However, the high demand for adaptability could pose challenges in environments with frequent network topology changes.

Wu *et al.* [39] used hop count, criticality, and cost as the main factors. Also, using SDN architecture and reinforcement learning deep Q-network (RLDQ), an intelligent multilayer traffic scheduling traffic is proposed to obtain the cur-

rent optimal global routing policy based on the real-time traffic demand in the network. However, practical implementation may face challenges like complexity, high computational resource demands, and the need for continuous updates and maintenance to adapt to changing network environments and technologies.

In addition, there are traffic scheduling based on RL and HA. Kandil *et al.* [40] combined an exploration strategy with simulated annealing. The Agent utilizes network traffic and statistics to learn optimal routing paths, and the algorithm can significantly reduce energy consumption and meet QoS requirements. This strategy not only improves network performance but also helps to achieve more efficient network traffic management. However, it requires extensive experimentation and debugging to adapt to various network conditions.

3.4.3. Other Methods of Traffic Scheduling

In addition to traffic scheduling methods based on heuristic and intelligent algorithms, there are also several alternative approaches. Xu *et al.* [41] calculated the set of candidate paths for reachable paths between source and destination hosts, considered path hop count and bandwidth utilization comprehensively, and evaluated the candidate paths using a fuzzy logic model to select the optimal path. However, the accuracy of this fuzzy logic model may depend on the appropriateness of parameter settings, which could require adjustments in varying network environments. Huang *et al.* [42] rescheduled the ET passing through congested links to minimize the controller overhead. They also proposed an algorithm that dynamically adjusts the polling period according to the current network load and a probability-based path selection algorithm to improve the network throughput further. Yet, this approach may necessitate precise congestion monitoring and complex scheduling algorithms. Chang *et al.* [43] proposed a label-switching-based routing where they route the ET to the least congested path while using proactive methods such as ECMP to route other traffic. Each edge switch runs an ET detection algorithm to reduce the response time for heavy traffic. They also utilize the idea of INT to gather congestion information. This method minimizes the traffic table entries in the switches and speeds up the lookup process in the switches. This technique, however, might require substantial support for in-band telemetry technologies.

Some researchers have made improvements to segmented routing techniques. Li *et al.* [44] constructed a computational model for path weights by combining multiple factors. The optimal path of K shortest paths is selected for data forwarding. Zhou *et al.* [45] implemented the forwarding table construction for segmented routes using a secondary traffic table while using the algorithm for data traffic with non-uniform weighting in conjunction with link utilization and hop counts to compute the optimal path. These methods offer flexibility in adjusting routing strategies according to network conditions but may confront challenges in computational complexity and real-time updates. Gao *et al.* [46] designed a probabilistic scheduling algorithm by constructing a global node

probability matrix to elect the optimal path for traffic forwarding. However, this approach might require accurate information about the global network state and could face scalability issues in practical deployment.

Additionally, there are researchers who have made enhancements to network protocols. Liu *et al.* [47] solved the problem of overloading a single controller by deploying multiple controllers for distributed monitoring and collecting and storing congestion information across the network. This structure improves network scalability regarding storage overhead of congestion state information and does not lose any necessary information. However, this may require highly consistent controller collaboration and complex congestion information processing mechanisms. Ha *et al.* [48] found all the available paths using the Deep First Search (DFS) algorithm and calculated the overhead of each path based on link utilization. They also rescheduled the traffic based on the fairness factor. Maintaining fairness in traffic scheduling might be challenging in practice, especially in situations of frequent network state changes, necessitating continuous adjustments to fairness strategies.

3.5. Other Strategies

In addition to the above methods, there are other methods to achieve LB.

Zaher *et al.* [49] scheduled a portion of the traffic based on the available bandwidth through a distributed sampling technique. Send another portion of the traffic to the controller. Periodically polling edge switches for network information. As well as re-scheduling only the ET to mitigate sampling overhead and packet conflicts associated with Equal-Cost Multipath Routing (ECMP) [50]. However, reliance on regular polling may introduce latency issues in information relevancy, and the specialized treatment of elephant flows necessitates precise traffic identification mechanisms.

Gutiérrez *et al.* [51] minimized the Flow Completion Time (FCT) of mouse traffic and the high throughput of ETs without starvation. It does not require a priori information, and the host reduces the FCT of the MT by carrying information in the packet that allows the forwarding device to adjust the traffic scheduling process to avoid traffic conflicts dynamically. However, this may necessitate more complex packet processing logic.

Tang *et al.* [52] input traffic features into the training of actor neural networks. The mapping between traffic and routing policies is learned through RL. After offline training, it is possible to quickly infer the diversion strategy and determine the diversion ratio on the SDN switch. This helps to optimize network traffic routing in hybrid SDN and avoid traffic conflict problems, although its applicability may be limited in rapidly changing network environments due to the extensive offline training required, although its applicability may be limited in rapidly changing network environments due to the extensive offline training required.

Ahmad *et al.* [53] proposed a Packet-In filtering mechanism that achieves the

goal of reducing CPU and memory load and avoiding traffic conflicts by classifying, filtering, and forwarding Packet-In messages combined with the introduction of a Packet-In listener module and information retrieval using a REST API. This helps to improve the efficiency and reliability of the whole system. However, it might require precise message processing mechanisms, as well as increased management and monitoring of system resources.

Liu *et al.* [54] modeled historical traffic using empirical pattern decomposition methods and adopted different treatment strategies for different congestions. When transient congestion occurs, it performs local LB through alternative forwarding paths, while when persistent congestion occurs, it adaptively determines the appropriate timescale in the control plane to perform global LB. This approach could effectively avoid traffic conflicts and improve the performance and reliability of the whole network, but may depend on the accuracy of historical data and the adaptability of the model.

In addition, researchers have proposed a novel ET filtering strategy. Gao *et al.* [46] used a redefined PSO algorithm to filter traffic. It helps to reduce the ET overhead, avoided traffic conflicts, and achieve LB. However, this might require a balance between the algorithm's optimization efficiency and computational complexity.

4. Trends and Challenges

The trends and challenges of SDN-based data centers LB have not been thoroughly investigated. This section discusses some trends and challenges derived from the literature review.

- In DCNs, two types of traffic coexist [5]. With the continuous growth of network traffic and the increasing complexity of network structures, the unfiltered or improperly handled "ET" in the network may consume significant network resources, leading to issues such as network congestion, traffic conflicts, and latency, thereby causing an imbalance in network loads. Current research primarily focuses on preventing network congestion and addressing traffic conflicts to achieve LB, which is a prominent direction for future research.
- In recent years, some studies have widely adopted intelligent algorithms, particularly RL, due to its ability to interact continuously with the environment and improve strategies. It has shown remarkable performance in addressing network LB issues, including its application in enhancing routing [30] [31] [32]. However, this approach requires a substantial amount of training data to learn effective strategies, potentially resulting in unstable or suboptimal decisions during the learning process. Some studies have attempted to combine RL with DL [33]-[39] to leverage the advantages of DL in feature extraction and pattern recognition. However, this approach typically relies on a large amount of labeled data for offline training, consuming significant computational resources and time. The rapid changes in the network environment

may also limit its adaptability to real-time scenarios. Additionally, some studies have applied HAs to LB in SDN [22]-[29]. While HAs can make rapid decisions based on real-time information, they often fall short in long-term planning and adapting to dynamic network environments. Furthermore, a recent study [40] proposes a strategy that combines RL and HAs to achieve more efficient network traffic management. Therefore, integrating the adaptive learning capability of RL with the immediate responsiveness of HAs to achieve a high-accuracy and highly adaptive strategy represents a crucial direction for future research.

- Most current studies have not explored algorithms for LB detection, making the exploration of new load detection methods a direction for future research.
- Only a partial number of studies have considered energy-saving issues in networks, contributing to improving the ubiquity and efficiency of existing LB mechanisms. Thus, considering energy-saving concerns in current LB mechanisms is a significant direction for future research.
- Additionally, only a small fraction of studies has considered how to apply LB techniques to various environments. Therefore, determining how to apply these techniques to practical network environments while ensuring their reliability and robustness is another direction for future research.

5. Conclusion

This paper explores five aspects of SDN's LB mechanisms, namely traffic classification, prediction, table management, scheduling and others. These approaches aim to optimize network performance, prevent traffic conflicts, and reduce data transmission latency. In the context of traffic scheduling, methods based on HAs, intelligent algorithms, and other approaches are discussed. These strategies leverage different algorithms and technologies to optimize traffic allocation and scheduling for faster and more accurate responses, particularly those based on intelligent algorithms. However, research on LB mechanisms in DCNs based on SDN technology is still in the exploratory stage. Compared to more mature fields, there are still numerous technical issues and shortcomings. These methods exhibit various drawbacks, including challenges in handling bursty traffic, dynamically changing controller loads, and overlooking certain services. Therefore, further exploration demands more time and effort.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Zhang, J. and Li, N. (2022) Research on Load Optimization of Software-Defined Network Based on Delay and Load. 2022 *IEEE 2nd International Conference on Networking, Communications and Information Technology* (*NetCIT*), Manchester, 26-27 December 2022, 254-259.

[2] Li, S., Xin, Z., Xu, X. and Zhang, Z. (2023) Load Balancing Algorithm of SDN Controller Based on Dynamic Threshold. 2023 *IEEE* 3*rd Asia-Pacific Conference on Communications Technology and Computer Science* (*ACCTCS*), Shenyang, 25-27 February 2023, 517-520. https://doi.org/10.1109/ACCTCS58815.2023.00105

[3] Liu, Z.P., Ren, S.S., Li, M., Wang, X.P. and Li, X.F. (2021) Software Defines Dynamic Traffic Scheduling Scheme for Network Data Center. *Journal of Jilin University*: *Engineering and Technology Edition*, **51**, 1040-1047.

[4] Wang, Y.M., Wang, X., Dong, Y., Zhang, S.H. and Shi, X.L. (2020) Data Center Traffic Scheduling Strategy Based on Fibonacci Tree Optimization Algorithm. *Journal on Communications*, No. 6, 112-127.

[5] Hamdan, M., Khan, S., Abdelaziz, A., Sadiah, S., Shaikh-Husin, *et al.* (2021) DPLBAnt: Improved Load Balancing Technique Based on Detection and Rerouting of Elephant Flows in Software-Defined Networks. *Computer Communications*, **180**, 315-327. https://doi.org/10.1016/j.comcom.2021.10.013

[6] Abdollahi, S., Deldari, A., Asadi, H., Montazerolghaem, A. and Mazinani, S.M. (2021) Flow-Aware Forwarding in SDN Datacenters Using a Knapsack-PSO-Based Solution. *IEEE Transactions on Network and Service Management*, **18**, 2902-2914. https://doi.org/10.1109/TNSM.2021.3064974

[7] Khairi, M.H.H., Ariffin, S.H.S., Latiff, N.M.A.A., Yusof, K.M., Hassan, M.K., Al-Dhief, F.T. and Hamzah, M. (2021) Detection and Classification of Conflict Flows in SDN Using Machine Learning Algorithms. *IEEE Access*, **9**, 76024-76037. https://doi.org/10.1109/ACCESS.2021.3081629

[8] Xie, S.X., Hu, G.Y., Xing, C.Y. and Liu, Y.Q. (2023) Online Elephant Flow Prediction for Load Balancing in Programmable Switch Based DCN. *IEEE Transactions on Network and Service Management*. https://doi.org/10.1109/TNSM.2023.3318752

[9] Sun, X. and Yang, G. (2022) Research on Load Balancing Strategy of Data Center Based on Yen Algorithm in SDN. 2022 *IEEE* 5*th Advanced Information Management*, *Communicates*, *Electronic and Automation Control Conference* (*IMCEC*), Vol. 5, 1243-1247. https://doi.org/10.1109/IMCEC55388.2022.10020139

[10] Diel, G., Miers, C.C., Pillon, M.A. and Koslovski, G.P. (2023) RSCAT: Towards Zero Touch Congestion Control Based on Actor-Critic Reinforcement Learning and Software-Defined Networking. *Journal of Network and Computer Applications*, **215**, Article ID: 103639. https://doi.org/10.1016/j.jnca.2023.103639

[11] Liu, Y., Zhang, J., Li, W., Wu, Q. and Li, P. (2021) Load Balancing Oriented Predictive Routing Algorithm for Data Center Networks. *Future Internet*, **13**, Article No. 54. https://doi.org/10.3390/fi13020054

[12] Begam, G.S., Sangeetha, M. and Shanker, N.R. (2022) Load Balancing in DCN Servers through SDN Machine Learning Algorithm. *Arabian Journal for Science and Engineering*, **47**, 1423-1434. https://doi.org/10.1007/s13369-021-05911-1

[13] Nougnanke, K.B., Labit, Y., Bruyere, M., Ferlin, S. and Aïvodji, U. (2021) Learning-Based Incast Performance Inference in Software-Defined Data Centers. 2021 24*th IEEE Conference on Innovation in Clouds*, *Internet and Networks and Workshops* (*ICIN*), Paris, 1-4 March 2021, 118-125. https://doi.org/10.1109/ICIN51074.2021.9385546

[14] Hai-Anh, T.R.A.N., Souihi, S. and Mellouk, A. (2021) Towards a Novel Congestion Notification Algorithm for a Software-Defined Data Center Networks. 2021 *IFIP/IEEE International Symposium on Integrated Network Management* (*IM*), Bordeaux, 18-20 May 2021, 99-106.

[15] Shaikh, M.R.R. (2023) Bayesian Network Based Optimal Load Balancing in Software

Defined Networks. 2023 *IEEE International Conference on Emerging Smart Computing and Informatics* (*ESCI*), Pune, 1-3 March 2023, 1-5. https://doi.org/10.1109/ESCI56872.2023.10099730

[16] Wei, Z., Li, Q., Zhu, K., Zhou, J., Zou, L., Jiang, Y. and Xiao, X. (2022) DiffTREAT: Differentiated Traffic Scheduling Based on RNN in Data Centers. *IEEE Transactions on Cloud Computing*, **3**, 2407-2419. https://doi.org/10.1109/TCC.2022.3206593

[17] Gonzalez-Diaz, S., Marks, R., Rojas, E., De La Oliva, A. and Gazda, R. (2021) Stateless Flow-Zone Switching Using Software-Defined Addressing. *IEEE Access*, **9**, 68343-68365. https://doi.org/10.1109/ACCESS.2021.3077955

[18] Zhou, Q., Yu, J. and Li, D. (2021) A Dynamic and Lightweight Framework to Secure Source Addresses in the SDN-Based Networks. *Computer Networks*, **193**, Article ID: 108075. https://doi.org/10.1016/j.comnet.2021.108075

[19] Shen, G., Li, Q., Shi, W., Jiang, Y., Zhang, P., Gu, L. and Xu, M. (2022) Modeling and Optimization of the Data Plane in the SDN-Based DCN by Queuing Theory. *Journal of Network and Computer Applications*, **207**, Article ID: 103481. https://doi.org/10.1016/j.jnca.2022.103481

[20] Zhang, Q., Chen, J., Gao, D. and Wang, X. (2022) Intent-Based Service Policy Conflict Management Algorithm. 2022 *IEEE/CIC International Conference on Communications in China* (*ICCC*), Foshan, 11-13 August 2022, 19-24. https://doi.org/10.1109/ICCC55456.2022.9880783

[21] Liang, S. and Su, J. (2022) Detection of SDN Flow Rule Conflicts Based on Knowledge Graph. In: Quan, W., Ed., *International Conference on Emerging Networking Architecture and Technologies*, Springer Nature, Singapore, 93-104. https://doi.org/10.1007/978-981-19-9697-9_8

[22] Zhu, S.X., Long, Y.F., Sun, G.L. and Li, C.F. (2022) Improved Ant Colony Algorithm for Network Flow Scheduling in SDN Data Center. *Journal of Harbin University of Science & Technology*, **27**, 1-7.

[23] Zheng, H., Guo, J., Zhou, Q., Peng, Y. and Chen, Y. (2023) Application of Improved Ant Colony Algorithm in Load Balancing of Software-Defined Networks. *The Journal of Supercomputing*, **79**, 7438-7460. https://doi.org/10.1007/s11227-022-04957-8

[24] Dai, R.R., Li, H.H. and Fu, X.L. (2022) Data Center Flow Scheduling Mechanism based on Differential Evolution and Ant Colony Optimization Algorithm. *Journal of Computer Applications*, **42**, 3863.

[25] Ma, S.Q., Tang, H., Li, Y. and Lei, Y.J. (2021) A Traffic Scheduling Strategy Based on Particle Swarm Optimization in Data Center Network. *Telecommunication Engineering*, **61**, 865-871.

[26] Xu, H.L., Yang, G.Q. and Jiang, Z.J. (2021) Data Center Adaptive Multi-Path Load Balancing Algorithm Based on Software Defined Network. *Journal of Computer Applications*, **41**, 1160-1164.

[27] Ye, Z., Sun, G. and Guizani, M. (2023) ILBPS: An Integrated Optimization Approach Based on Adaptive Load-Balancing and Heuristic Path Selection in SDN. *IEEE Internet of Things Journal*. https://doi.org/10.1109/JIOT.2023.3309832

[28] Perepelkin, D. and Nguyen, T. (2022) Research of Multipath Routing and Load Balancing Processes in Software Defined Networks Based on Artificial Bee Colony Algorithm. 2022 *IEEE ELEKTRO* (*ELEKTRO*), Krakow, 23-26 May 2022, 1-6. https://doi.org/10.1109/ELEKTRO53996.2022.9803416

[29] Perepelkin, D., Ivanchikova, M. and Nguyen, T. (2023) Research of Multipath Routing and Load Balancing Processes in Software Defined Networks Based on Bird Migration Algorithm. 2023 *IEEE International Russian Smart Industry Conference* (*SmartIndustryCon*), Sochi, 27-31 March 2023, 247-252. https://doi.org/10.1109/SmartIndustryCon57312.2023.10110788

[30] Balakiruthiga, B. and Deepalakshmi, P. (2021) (ITMP)-Intelligent Traffic Management Prototype Using Reinforcement Learning Approach for Software Defined Data Center (SDDC). *Sustainable Computing*: *Informatics and Systems*, **32**, Article ID: 100610. https://doi.org/10.1016/j.suscom.2021.100610

[31] Gao, Y., Gao, X. and Chen, G. (2022) MetisRL: A Reinforcement Learning Approach for Dynamic Routing in Data Center Networks. In: Bhattacharya, A., *et al.*, Eds., *International Conference on Database Systems for Advanced Applications*, Springer International Publishing, Cham, 615-622. https://doi.org/10.1007/978-3-031-00126-0_44

[32] Yang, W., Qin, Y. and Yang, Z. (2022) A Reinforcement Learning Based Data Storage and Traffic Management in Information-Centric Data Center Networks. *Mobile Networks and Applications*, **27**, 266-275. https://doi.org/10.1007/s11036-020-01629-w

[33] Zeng, X., Wu, L., Li, Z. and Jing, Y. (2021) Deep Reinforcement Learning with Graph Convolutional Networks for Load Balancing in SDN-Based Data Center Networks. 2021 *IEEE* 18*th International Computer Conference on Wavelet Active Media Technology and Information Processing* (*ICCWAMTIP*), Chengdu, 17-19 December 2021, 344-352.

[34] Liu, W.X., Cai, J., Chen, Q.C. and Wang, Y. (2021) DRL-R: Deep Reinforcement Learning Approach for Intelligent Routing in Software-Defined Data-Center Networks. *Journal of Network and Computer Applications*, **177**, Article ID: 102865. https://doi.org/10.1016/j.jnca.2020.102865

[35] Xu, Z., Lu, Y., Li, J., Ma, X. and Qian, L. (2022) Tailor: Datacenter Load Balancing for Accelerating Distributed Deep Learning. 2021 *IEEE* 9*th International Conference on Advanced Cloud and Big Data* (*CBD*), Xi'an, 15-17 October 2021, 45-50. https://doi.org/10.1109/CBD54617.2021.00017

[36] Zhao, C., Ye, M., Xue, X., Lv, J., Jiang, Q. and Wang, Y. (2022) DRL-M4MR: An Intelligent Multicast Routing Approach Based on DQN Deep Reinforcement Learning in SDN. *Physical Communication*, **55**, Article ID: 101919. https://doi.org/10.1016/j.phycom.2022.101919

[37] Sharma, A., Tokekar, S. and Varma, S. (2023) Meta-Reinforcement Learning Based Resource Management in Software Defined Networks Using Bayesian Network. 2023 *IEEE* 3*rd International Conference on Technology*, *Engineering*, *Management for Societal Impact Using Marketing*, *Entrepreneurship and Talent* (*TEMSMET*), Mysuru, 10-11 February 2023, 1-6. https://doi.org/10.1109/TEMSMET56707.2023.10150107

[38] Gao, M., Pan, T., Song, E., Yang, M., Huang, T. and Liu, Y. (2022) Power-Aware Traffic Engineering for Data Center Networks via Deep Reinforcement Learning. *GLOBECOM* 2022-2022 *IEEE Global Communications Conference*, Rio de Janeiro, 4-8 December 2022, 6055-6060. https://doi.org/10.1109/GLOBECOM48099.2022.10001013

[39] Wu, G. (2022) Deep Reinforcement Learning Based Multi-Layered Traffic Scheduling Scheme in Data Center Networks. *Wireless Networks*, 1-12. https://doi.org/10.1007/s11276-021-02883-w

[40] Kandil, M., Awad, M.K., Alotaibi, E.M. and Mohammadi, R. (2022) Q-Learning and

Simulated Annealing-Based Routing for Software-Defined Networks. 2022 *IEEE International Conference on Computer and Applications* (*ICCA*), Cairo, 20-22 December 2022, 1-10. https://doi.org/10.1109/ICCA56443.2022.10039651

[41] Yang, X., *et al.* (2020) An Effective Routing Mechanism Based on Fuzzy Logic for Software-Defined Data Center Networks. 2020 *IEEE 6th International Conference on Computer and Communications* (*ICCC*), Chengdu, 11-14 December 2020, 1793-1798. https://doi.org/10.1109/ICCC51575.2020.9344964

[42] Huang, B. and Dong, S. (2020) An Enhanced Scheduling Framework for Elephant Flows in SDN-Based Data Center Networks. 2020 *IEEE Symposium on Computers and Communications* (*ISCC*), Rennes, 7-10 July 2020, 1-7. https://doi.org/10.1109/ISCC50000.2020.9219688

[43] Chang, Y.K., Wang, H.Y. and Lin, Y.H. (2021) A Congestion Aware Multi-Path Label Switching in Data Centers Using Programmable Switches. 2021 *IEEE International Conference on Networking, Architecture and Storage* (*NAS*), Riverside, 24-26 October 2021, 1-8. https://doi.org/10.1109/NAS51552.2021.9605422

[44] Li, Y, Tang, H, and Ma, S.Q. (2021) Multi-Path Scheduling Algorithm Based on Segment Routing in SDN. *Application Research of Computers*, **38**, 1514-1519.

[45] Zhou, J.X., Zhang, Z.P. and Zhou, N. (2020) Load Balancing Technology of Segment Routing Based on CKSP. *Computer Science*, **47**, 256.

[46] Gao, X.C., Liu, W., Wang, Q.L. and Zhang, X. (2023) SDN-Based Hybrid Segmented Routing Probabilistic Flow Scheduling Mechanism. *Application Research of Computers*, **40**, 3382-3387.

[47] Liu, Y., Gu, H., Zhou, Z. and Wang, N. (2022) RSLB: Robust and Scalable Load Balancing in Software-Defined Data Center Networks. *IEEE Transactions on Network and Service Management*, **19**, 4706-4720. https://doi.org/10.1109/TNSM.2022.3192133

[48] Ha, N.V., Tuan, T.A. and Nguyen, T.T.T. (2022) Fairness Enhanced Dynamic Routing Protocol in Software-Defined Networking. 2022 *9th IEEE NAFOSTED Conference on Information and Computer Science* (*NICS*), Ho Chi Minh City, 31 October-1 November 2022, 111-116. https://doi.org/10.1109/NICS56915.2022.10013394

[49] Zaher, M., Alawadi, A.H. and Molnár, S. (2021) Sieve: A Flow Scheduling Framework in SDN Based Data Center Networks. *Computer Communications*, **171**, 99-111. https://doi.org/10.1016/j.comcom.2021.02.013

[50] Zuo, P. and Shu, Y.A. (2021) Dynamic Multi-Path Load Balancing Method Based on Feedforward Neural in DCN. *Computer Engineering*, **47**, 113-119.

[51] Gutiérrez, S.A., Botero, J.F. and Branch-Bedoya, J.W. (2023) An Adaptable and Agnostic Flow Scheduling Approach for Data Center Networks. *Journal of Network and Systems Management*, **31**, Article No. 12. https://doi.org/10.1007/s10922-022-09701-4

[52] Tang, Q., Yang, R. and Guo, Y. (2023) Reinforcement Learning with Contrastive Unsupervised Representations for Traffic Engineering in Hybrid SDN. 2023 *IEEE 15th International Conference on Communication Software and Networks* (*ICCSN*), Shenyang, 21-23 July 2023, 118-122. https://doi.org/10.1109/ICCSN57992.2023.10297329

[53] Ahmad, S. and Mir, A.H. (2023) Protection of Centralized SDN Control Plane from High-Rate Packet-In Messages. *International Journal of Information Security*, **22**, 1197-1206. https://doi.org/10.1007/s10207-023-00685-z

[54] Liu, S., Liu, J. and Xia, B. (2023) Adaptive Timescale Load Balancing Routing Algorithm for LEO Satellite Network. 2023 *IEEE/CIC International Conference on Communications in China* (*ICCC*), Dalian, 10-12 August 2023, 1-5. https://doi.org/10.1109/ICCC57788.2023.10233496