A Detailed Analysis of Modern Load Balancing Methods in Software Defined Networks

Prerita Kulkarni[1*], Nitika Vats Doohan[2], Rajesh Nagar[3]

[1]Department of Computer Science, Medicaps University Indore,India,
[2,3] Institute of Computer Application,Sage University Indore,India
[1*]prerijoshi801@gmail.com, [2]nitika.doohan@gmail.com, [3]Errajes973@gmail.com

Abstract. In these days of smart technology, the use of internet-connected devices has increased dramatically. This has led to a sharp rise in online activity. The necessity to manage a single server with several clients, which might lead to denial of service (DoS) assaults, restricted service availability, and issues with network scalability are only a few of the negative effects of the growing traffic. A solution that has been proposed in the literature to handle these issues is to utilize a load balancer in combination with many servers. Research indicates that load balancers do have a few disadvantages despite being widely used. Among these include the fact that they are exclusive to one manufacturer and cannot be programmed. SDN software-defined networking emerged as a paradigm shift to address these issues and the resulting spike in internet traffic. Software-defined networking (SDN) enables programmable load balancers by granting customers the autonomy to create and employ custom load balancing algorithms. This inquiry aims to investigate SDN and OpenFlow from the ground up, with a particular focus on how they impact load balancing. This paper compares several SDN load balancing approaches based on open research questions, suggested fixes, and potential future paths. Smart load balancing methods for SDN are often designed using mathematical models and simulators. Furthermore, we describe in detail the key performance metrics for these algorithms

Keywords: Software Defined Network, Dynamic Load Balancing Problem, OpenFlow, Network Function Virtualization, Quality of Services First Section

1. Introduction

Many people have shown interest in a number of inventions that have emerged in the last ten years. These include cloud computing, social networking sites, IoT, and cutting-edge smartphone apps. These systems struggle with performance, scalability, and flexibility while handling data and satisfying quality of service requirements because of the antiquated backbone networks. As of right now, no airline allows switching systems, therefore passengers are unable to request transfer providers. Using an aircraft designated for a certain gender requires human supervision. Services from different cloud platforms host websites and online services.[1] The performance, dynamic management, and service quality criteria were critical to the provision of these services. Software-defined networking (SDN) [2] and network functions virtualization (NFV) [3] are two approaches. that experts have suggested as a fix for the issues with

S. Bhalerao et al. (eds.), *Proceedings of the International Conference on Recent Advancement and Modernization in Sustainable Intelligent Technologies & Applications (RAMSITA-2025),* Advances in Intelligent Systems Research 192,
https://doi.org/10.2991/978-94-6463-716-8_72

conventional networks. Software-defined networking (SDN) and network function virtualization (NFV) enable you to operate numerous devices and virtually recreate the physical components of your network. Software abstraction is used in both approaches to get around the problems with traditional networks. Additionally, by enabling programs to define and alter the network's structure and operation, software-defined networking, or SDN, increases the network's programmability [4].

The control plane in software-defined networking (SDN) isolates the physical network devices by making all the decisions. You can locate it in Figure 1. The control plane, which is also where the network's intelligence resides, is where policies and protocols for overseeing the network and forwarding packets are implemented. SDN's architecture allows for robust management and network scalability [5]. An SDN controller can aid in centralizing network management. Using less complicated network devices is made possible by simplifying the network design, which also makes the network control vendor-independent. Software-defined networking, or SDN, has the potential to improve a number of network engineering technologies, such as cloud computing and massive data processing [6]. SDN has proven to be significantly better than traditional networks in a number of areas. The novel architecture of software-defined networking (SDN) can reconsider the constraints of existing configurations. When compared to actual forwarding planes, data control plane and packet forwarding plane solutions that make use of this rank among the best. You can use any programming language that supports software-defined networking to control the system dynamically. Its adaptability makes it possible to monitor big data and cloud computing platforms, which is why it is essential for on-demand client requests [7]. SDN-based networks are produced by combining cloud computing, software-defined networking (SDN), and big data analytics techniques.

The process of effectively distributing traffic among network devices, or load balancing, is a crucial component of network design and management [8]. A load balancer is effective if it can lower power consumption without sacrificing fault tolerance, throughput, latency, or resource utilization. In traditional networks, dedicated servers handle most of the load balancing (see Figure 1). Installing dynamic load balancing servers now allows the network to adapt to its constantly changing demands [9]. Various SDN load balancing techniques are the primary focus of our analysis. The technique of load balancing distributes the effort among several resources so that no resource is overworked [10]. The primary goals are traffic optimization, reaction time reduction, and throughput optimization. Traditionally, load balancing has been implemented in network administration by distributing network traffic across dedicated hardware components. This approach usually works well, but it's pricey and doesn't let you change configurations dynamically depending on data in real-time. Because it changes on the fly to meet the demands of the company, software-defined networking (SDN) is the best solution for load balancing in the cloud. In a cloud computing setting, SDN manages every single device.[11]

Most data centers now require more than 40% of their electricity for cooling. Data centers use a lot of energy for more than just cooling. To ensure that customers' data and cloud services are available 24/7 from any device, wherever in the world, data centers are designed with hyper-redundancy in mind. In the case of a system failure, this guarantees that user experiences will remain uninterrupted. As it expands, its environmental impact increases, despite the fact that the executives, technicians, and engineers responsible for building its infrastructure strive to find a happy medium between sustainability and profitability. Here comes the significance of efficient load balancing in SDN based datacenters so as to provide an energy efficient and sustainable ecosystem for data processing.

Due to its complexity, a thorough investigation of SDN might be difficult. The authors of the paper assert that additional research is necessary in this field, despite the fact that load balancing in SDN can enhance SDN performance. This study examines the effects of different artificial intelligence (AI) strategies on load balancing (LB) and is the first in-depth examination of LB in the context of software-defined networking (SDN). Many scholars have put forth a number of load balancing solutions based on various mathematical models during the past ten years. Each of these strategies has advantages and disadvantages that vary depending on the task at hand and the network's condition. The purpose of this study is to survey load balancing systems used in data centers in order to offer a thorough analysis.
This research seeks to deliver an in-depth analysis by reviewing load balancing solutions deployed in data centers utilizing SDN network architecture. What follows is the structure of the rest of the paper: In Section I, there is a discussion of how software-defined networks affect cloud Load balancing. In Section II, we examined the characteristics of load balancing within the SDN architecture and its corresponding layers. Section III provides an in-depth evaluation of the load balancing system for a selection of selected articles. The findings are presented in Section IV. Section V brings to an end the study.

2. Background

In this section, we have covered the architecture of SDN, characteristics of SDN and load balancing algorithm based on SDN.

2.1 SDN Architecture

A Network Architecture Based on Software In software-defined networking, rather of relying on proprietary hardware, routing decisions are defined by generic, adaptable software. This is accomplished by physically separating the data plane from the control plane. Data planes and control planes used to be one and the same, and switches and routers were the sole physical components that could be used to identify a network's architecture. Managing the packet forwarding and routing of individual components became unfeasible for network managers due to the manufacturer-implemented layer of control. Together, it and the element's control plane became one. Because of this, it is possible for a network to have several, mutually exclusive control planes for a single device [12]. Thanks to software-defined networking, a central high-throughput server can handle all of the network devices' control plane rules. Additional examples of possible constraints include allowing or disabling communication from a certain port or IP address. The SDN architecture stacks networks in three separate but interconnected tiers (Figure 1). Layers seldom share data with one another since they all have separate purposes.

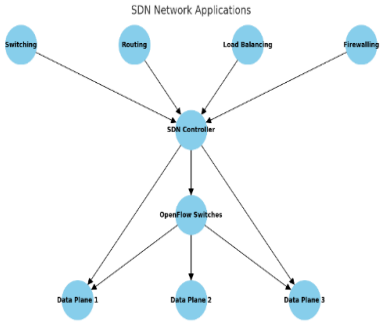

Fig. 1. The Traditional SDN Architecture

The application layer becomes the base layer when it is fully expanded. Programs have the ability to manage the data flow across a network and interact via software with lower-level components. On this tier, the load balancer for the paper will function. The SDN controller, which is located in the intermediate layer, is frequently referred to as the control plane. One of its special features is that it can change the higher layer routing devices' flows and abstract them from one another because of its bidirectional communication capabilities. There are a lot of SDN controllers that are open source. In Figure 1, a number of controller names are displayed. Beneath it lies a layer of network devices known as the data plane. It receives, processes, and changes data packets in accordance with controller orders. Internal counters are used in general statistics. The controller, also known as the Northbound API, and the application layer usually work together to provide a more thorough representation of the network's requirements and behavior in addition to a high-level overview of the network [13]. Flowable control over device flows is made possible by the controller, sometimes referred to as the Southbound API, which uses an OpenFlow-like protocol to interface with lower-level layers.

2.2 Load Balancing Algorithm in SDN

LB technologies often improve the overall performance of distributed systems by effectively distributing workloads, incoming customers, and requests among the available network resources [14]. Using this approach, throughput, latency, and network overload may all be reduced. There are two ways it might be used: in hardware or software. Load distribution and energy efficiency may be improved by integrating software-defined networking (SDN) with virtual resources in the Internet of medical technologies [15]. For the application of load balancing algorithms, several methods are available, such as static, dynamic, and hybrid ones [16]. The initial system information is a major determinant of the static techniques. Unpredictable user behavior and static load balancer rules may prevent some networks from using static load balancing (STL) techniques. However, because dynamic techniques take use of load balancers' pre-programmed patterns, they outperform static ones in terms of load balancing [17]. An effective load-balancing strategy can lead to decreased response times and packet loss ratios, better resource utilization, and overload protection. It can also improve the network's packet delivery ratio, lifespan, scalability, and dependability. Comparing and contrasting the

load-balancing techniques is necessary to choose the best solution to the load-balancing problem and to understand the advantages and disadvantages of each mechanism [18]. To guarantee accurate findings, it is crucial to incorporate a variety of features, sometimes known as qualitative metrics, in comparisons, such as latency, energy usage, packet delivery ratio, scalability, etc. [19]. One of the main problems with distributed computing is load balancing. The load may be coming from a CPU restriction, memory limit, or network latency. Distributing the work across the distributed system's several nodes is a continuous process that improves the system's structure. As a result, it can prevent situations where the hubs on the network are either overcrowded or underloaded. A technique called load balancing ensures that the effort is dispersed effectively, allowing the running assignment to be completed without overtaxing the processor pool or creating a frame hub surplus.

Here, load balancing algorithms mostly fall into two categories as discussed below:

2.2.1. The Static Algorithm

Static load balancing systems provide the benefit of evenly allocating resources throughout client-server communication. The control mechanism for this kind of software is fully manual. The process of assigning individual processors their particular jobs upon startup is known as the master CPU. The CPU that handles a given job always has the same identity. As a result, the work is finished quickly by the central processor. The slave process's central processing units (CPUs) make decisions and communicate their findings to the master process. Whatever the number of jobs allotted to a CPU, it will always complete at least one of them. Static load balancing techniques help important applications operate quicker and with less latency. They also lessen the chance of processor communication delays. Because of this, no attempt is made to stop anything from happening [20].

The static load balancing approach has taken care of the following features:

1. Reduce the time it takes to communicate; a longer execution time is correlated with less communication.
2. Weighted algorithms provide faster responses.
3. Methodical techniques to load balancing that determine which available node to use for a task by looking at its current load.
4. When static operations are performed regularly under comparable and stable conditions, the results are more reliable and consistent.
5. It is easy to forecast the behavior of static methods.
6. Because of the arrival and processing conditions of the loads, it is challenging to forecast future loads.

2.2.2 The Dynamics algorithm

To provide proper load balancing throughout the whole network or architecture, the dynamic algorithm locates the lightest server and gives it priority. Because dynamic load balancing distributes preprogrammed load balancer settings in real time, it performs better than static load balancing [21]. Effective load balancing is crucial to avoiding resource overload and optimizing consumption, scalability, reaction time, and throughput. There are three possible implementations for dynamic load balancing: semi-distributed, distributed, and non-distributed.

3. Task Assignment

In order to identify which nodes in a network farm are capable of processing data packets as they arrive at their intended destination, adaptive load balancers use intelligence. Only

algorithms designed to manage load dispersion can do this. There is an appreciable difference in the techniques depending on whether the load is distributed via the network layer or the application layer. Algorithm selection, which is determined by load distribution systems' efficacy, affects performance and business continuity. Maximizing the efficiency of any network, including SDN, requires identifying the optimal load balancing methods. The objective of various published methods for optimizing network traffic is to provide intelligent load balancing among several controllers, servers, networks, and clouds. Decentralizing the SDN controller could be necessary in some circumstances to improve the control plane's functioning. The decentralization causes the control plane to frequently have a large number of controllers. We study load balancing in the scenario where a large number of SDN controllers make up the control plane. Published in [22], the RLMD system stands for dependable and load-balance-aware multi-controller deployment. According to the authors, controller traffic load imbalance and control plane accuracy are the main problems that RLMD aims to solve in multi-controller SDN networks with separate data and control planes. The RLMD technique is complex because it ensures fair distribution of traffic loads and efficient and robust controller installations. Another technique the authors developed, called multiple domain partitioning (MDP), matches switches and controllers based on the controller's capacity to recruit nodes and balance load. Their method greatly improved control plane dependability and balanced controller traffic distribution as compared to other traditional systems. In order to address the well-documented issue of control plane load balancing, the authors of [23] concentrated on an architecture for the software-defined networking (SDN) control plane that uses distributed controllers and has the capacity to dynamically balance its traffic load. By resolving the problem of managing several controllers, researchers closed a knowledge gap on the positioning and matching of controllers with switches. To reduce congestion, these algorithms efficiently divide up the switch requests across several controllers.

ElastiCon is suggested in [24] as a switch migration method. To maintain optimal network performance, ElastiCon evenly distributes the workload among a number of controllers regardless of traffic patterns. One of ElastiCon's drawbacks is that it could cause incredibly high network overhead. Several relevant papers in [25] show potential ways to increase the efficiency of the SDN controller. The main objective of these evaluations was to determine which migration switch with load balancing was appropriate. This technique partially relieves controllers of their responsibilities, preventing a decline in network performance. To improve load balancing's time efficiency, the inventors of [26] came up with a method based on switch groups of multiple controller architecture. The suggested method addresses the issue of load oscillation among controllers by combining a switch selection methodology with a target controllers selection algorithm. A survey in the field of software-defined networks (SDN) was carried out by the authors of [27]. In their review, they spoke about the background, tenets, and difficulties of SDN and compared it to traditional networks. The authors address eight important SDN issues in further depth. We have just begun to scratch the surface of the many ideas, which include the evolution of software-defined networks, the Naas paradigm, and the future of cloud computing. After examining a substantial amount of related research, the authors of [28] described the benefits of software-defined networking (SDN) and its performance-enhancing features. The authors' brief summary of the advantages and disadvantages of recent SDN research is provided by their literature review. Their findings suggest that SDN switch devices may see reduced costs, improved resource utilization, and advances in network scalability. Nevertheless, the researchers did not look at the implementation's design issues or the

commercial expansion of SDN.The authors of [29] spoke about the implications on the network and concentrated on the issues brought on by load imbalance. The report included an orderly description of many load balancing strategies. We thought about the benefits and drawbacks of this tactic. Their analysis was lacking in two important areas: the performance optimization approach and the distributed SDN architecture scenario. One can utilize the methods described in [30] to network monitors. They demonstrated how excessive monitoring might cause a network to function more slowly. The network monitor technique is recommended by the authors for the optimal choice of switches and network traffic.

In order to determine software-defined networking, [31] studied nineteen papers in detail. The authors distinguished between deterministic and non-deterministic load balancing systems in their evaluation of the literature. We also looked at the complexity and limits of these approaches in relation to other load distribution benchmarks. Additionally, we have verified that the algorithms are proper. It may have expanded the scope of the study to include a failure mechanism specific to the SDN environment. In [32], researchers conducted a comprehensive analysis of SDN load balancing techniques. The following are the three main categories of load balancing strategies. They might exhibit dynamic or static behavior. This approach needs to continuously monitor and assess changes in the load. The fundamental idea behind load balancing algorithms is distributed, dynamic decision making. Reducing latency and reaction time while increasing throughput is the goal of the distributed and dynamic load balancing technique. There is not much study on the following topics: preference, network slicing, and traffic sorting. A predefined network model enables scalable software administration of the data center's network. For large networks, the multi-controller approach was employed in [33]. The benefits and drawbacks of using several controllers have been carefully considered. When discussing techniques for multi-controller load balancing, the researcher made brief mentions of heuristics, linear programming, and game theory. The study does not address concerns regarding dependability, safety, or consistency. The authors of [34] explore the Controller Placement Problem (CPP) and consider its modelling, formulation, and pertinent metrics, as well as optimum and heuristic techniques to solve it. Further studies might focus on CPP levels and NFV conduction in big networks. There is no set procedure for conducting SDN load balancing surveys. There has never been a comprehensive review of dynamic controllers or multi-controllers by another researcher.
In reality, Software-Defined Networking is not without its shortcomings, but today it has found hundreds of applications in various fields and environments, with proven demonstrations of its possible contribution towards creating a more efficient, flexible, and manageable network. Typical applications include Data Center Management, Network Function Virtualization (NFV), Enterprise Networks, Wide Area Networks (WANs), 5G and IoT Networks, Cloud Networking, Smart Cities and Smart Infrastructure, etc.

4. Notable literature survey

Software-Defined Networking (SDN) has come to be regarded as a new paradigm in the advancement of modern-day networks that promises centralized control, dynamic resource management, and enhanced programmability in networks. In spite of the above assertion, however, much of the literature on such a topical issue has provided great insight into the capability as well as the possible applications of SDN; some key shortfalls have continued to impede further progress, as well as practical deployments in the area and these include such

issues as a lack of standardization, scalability challenges, deficient security analyses, and few implementations in the field.

Energy efficiency is one of the most important traits when it comes to networking in large-scale data centers or cloud environments. However, the literature in SDN has been largely neglecting this aspect. All proposed solutions are based on the improvement in performance and functionality rather than power consumption efficiency. Therefore, such resource utilization is inefficient. Energy-aware SDN frameworks have been developed for a couple of years but they are still in their infancy and have not been widely validated. Following are the most notable contributions in the field of SDN by the researchers over the last decade.

Literature Survey

Literature review is shown in Table 1

Table 1. Literature Review

Paper Details	Method	Advantage	Limitation
Farah Chahlaoui, Mohammed RAISS, Hamza DAHMOUNI (2019) Performance Analysis of Load Balancing Mechanisms in SDN Network	Controller and Switch Based Load Balancing	1.Efficient 2.Transparent 3.OpenFlow complaint	1.Lack of realistic model 2.Inadequate evaluation of overhead
Tao Wang, Fangming Liu, Senior Member, IEEE, and Hong Xu (2017) An Efficient Online Algorithm for Dynamic SDN Controller Assignment in Data Center Networks.	OFLoad, Open Flow-based two-stage design framework.	1. effective use of resources, 2. offloading that minimizes the size of the forwarding table.	1.No real network model. 2.Low accessibility.
Tao Hu, Zehua Guo, Thar Baker, Julong Lan (2018) Multi-controller Based Software Defined Networking	SMCLBRT: Response-time-based SDN multiple controllers load balancing techniques	1The findings of the simulation confirm that migration can begin beforehand. 2.Minimize controller overloading. .	1.Reduce the quantity of tests. 2. The majority of load distribution patterns are not covered by the simulation.
Wenjing Lan, Fangmin Li, Xinhua Liu, Yiwen Qiu(2018)A Dynamic Load Balancing Mechanism for	Problem of dynamic load balancing (DLBP) using a greedy approach	1.Lower the highest possible link utilization. 2 Performance gets better.	1.Fit for the genuine large-scale network. 2.Low accessibility.

Distributed Controller in SDN			
Gokay Saldamli, Harsha Sanjeeva, Karthik Siddalingaapa, Ruthra Vadivel Murugesan, Levent Ertau (2019)A Secure Collaborative Module on Distributed SDN	Adaptively Adjusting and Mapping Controllers (AAMcon)	1.Reduce the amount of time waiting. 2.Strong and durable. 3. Excellent Work	1.Future network architecture will be information-centric networking (ICN). 2.There will be less need for many controllers
Tamal Das, Vignesh Sridharan and Mohan Gurusamy (2019) A Survey on Controller Placement in SDN	Application of Deep Neural Network to Virtualized Network Function	1.Enhance the quality of service. 2. Sort out the issue with the increased load.	1.The VNF migration issue is not addressed. 2. It is possible that fewer load balancing parameters were used.
Alex M. R. Ruelas, Christian Esteve Rothenberg (2018) A Load Balancing Method Based on Artificial Neural Networks for Knowledge-defined Data Center Networking	Dynamic load management with flow priority depending on SDN.	1.Decreased deterioration 2. A larger bandwidth. 3. Increase the use of links.	1Take into account node utilization. 2. Not appropriate for the busy network

5. Result

The system under examination is transparent, open, and effective; it makes good use of its resources and uses strategies like offloading to minimize the size of the forwarding table. However, several problems still need to be resolved, such as an unrealistic network model, imprecise overhead assessment, and inadequate accessibility. The system is still not flawless when it comes to controlling VNF migration and utilizing the proper load balancing settings, especially in busy or large-scale networks, even if the findings indicate that link utilization and controller overloading have improved. While information-centric networking (ICN) with fewer controllers may be the way of the future, a number of critical issues, including load distribution and node use, still need to be resolved to provide dependable performance in a variety of network scenarios. A freshly established technique often yields intelligent traffic division on the basis of a specified statistic. Furthermore, a system may not always require impartiality and immunity to outside intervention in order to perform. We fear that if interference reduction is not given more priority, the technique's efficacy may gradually deteriorate. This is occurring because they have the ability to tamper with networks whenever they want. Other metrics like run time, latency, and utilization that are referenced in the literature are not covered in this article. Recall that the objective and the methods for assessing the effectiveness of the solution

dictate the metrics that should be applied. Most of the researchers in the sample test their proposed approaches using mininets as emulations. This is due to the fact that mininets become extremely effective tools for interactive development, testing, and presentation when they are combined with OpenFlow and SDN. Network traffic is created by Iperf twenty percent of the time, and supplemental tools including MATLAB, Python, .Net, and Maxi Net five percent of the time. Mininet was used in combination with other tools like Iperf and Maxi Net to do the necessary SDN network simulations. Moreover, [35] combined Python. The authors blended deep learning and reinforcement learning methodologies since they could only learn from failures and rewards. The fact that SDN allows multiple tool use demonstrates how it liberates the network administrator to create network management strategies unrestricted by the equipment or tool vendors.

Here are a few instances of issues with the different strategies:

This is due to several reasons, including the following: (a) algorithms that are not applicable to wireless networks; (b) failure mechanisms are not taken into account; (c) there is no machine learning approach; (d) more time-consuming algorithms; (e) many of these algorithms are not utilized in actual large-scale networks; and (f) there is no security mechanism. Most approaches are evaluated using an emulator called Mini Net and MATLAB.

6. Open Issues

The previous assessment demonstrates that there are a lot of unanswered questions regarding SDN-based data center load balancing procedures. We discussed a few incomplete research projects here. a concise overview of pertinent research on enhancing the consistency, scalability, and reliability of software-defined network controllers. These could have a significant impact on the data center's network's effectiveness. Numerous load balancing-related data center difficulties require more investigation. Not every QoS parameter is taken into account by one approach. The complete analysis highlights the following research gaps. It's crucial to assess the design with a variety of dynamic controllers. A few open-ended queries are as follows:

A) No investigation has focused on the proper management of resource utilization for improved performance.
B) Low scalability, low dependability, and flexibility problems with the control plan affect SDN-based load balancing.
C) The data center does not take dynamic traffic scheduling or congestion control into account.
D) When determining load balancing, there isn't a separate system to take into account different QoS qualities including throughput, scalability, flexibility, cost, complexity, and bandwidth.
Further research must strive for a formalization of theoretical and practical aspects with a framed and standardized benchmarking to build further sound and viable SDN solutions. Only under such advancements shall SDN present as a fully promoted transformative impact on modern networking.

7. Conclusion

The increasing dependence on AI and ML processes, as well as data centers that house these cutting-edge digital technologies, is posing serious environmental challenges. These data centers need vast quantities of power for data processing and server operations, particularly for computationally intensive jobs like machine learning training. Load balancing in data centers plays a very significant role in energy efficient and sustainable data processing and storing tasks

SDN improves data center networking performance, according to a review of load balancing techniques. This article provides an overview and assessment of load balancing in software-defined networks (SDN) data centers. We go over the benefits and drawbacks of several load balancing techniques used by the researchers in this survey. We also examine other performance metrics to evaluate the load balancing approach. Most studies used Mini Net as an emulator to compare the various approaches. The study techniques provide several benefits, including enhanced resilience, reduced latency, higher performance, and optimal use of network resources. A few issues with the systems stated above include inadequate support for big networks, inadequate resource and failure management, flawed machine learning, missing quality-of-service measures, and a dearth of academic research on energy efficiency. Apart from the current restricted supply of dynamic multi-controllers, future studies might look at different methods of load balancing in data center networks.

References

1. Cui, L., Yu, F.R., Yan, Q.: Integrating Big Data with Software-Defined Networking: Leveraging SDN for Big Data and Utilizing Big Data for SDN. IEEE Network 30(1), 58–65 (2016).
2. Du, X., Lv, Z., Wu, J., Wu, C., Chen, S.: PDSDN: Enhancing Multi-Tenant Cloud Data Center Environments with a Policy-Driven SDN Controller. In: IEEE International Conference on Services Computing (SCC'16), pp. 387–394 (2016).
3. Xu, Y., Sun, Z., Sun, Z.: SDN-Based Architecture for Big Data Networks. In: International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC) (2016).
4. Ordonez-Lucena, J., Ameigeiras, P., Lopez, D.: Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges. IEEE Communications Magazine 55(5) (2017).
5. Chen, Q., Xie, R., Huang, T., Liu, J., Liu, Y.: Resource Allocation for Software-Defined Networking, Caching, and Computing with Imperfect NSI. In: IEEE Global Communications Conference (GLOBECOM) (2017).
6. Kreutz, D., Ramos, F.M.V., Veríssimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-Defined Networking: A Comprehensive Survey. IEEE Transactions on Network and Service Management 103(1) (2015).
7. Xia, W., Wen, Y., Foh, C.H., Niyato, D., Xie, H.: A Survey on Software-Defined Networking. IEEE Communications Surveys & Tutorials 17(1) (2015).
8. Qilin, M., WeiKang, S.: A Load Balancing Method Based on SDN. In: IEEE International Conference on Measuring Technology and Mechatronics Automation, pp. 978-1-4673-7143-8/15 (2015).

9. Kanagavelu, R., Aung, K.M.: Software-Defined Load Balancer in Cloud Data Centers. In: ICCIP, Singapore (2016).

10. Li, L., Xu, Q.: Load Balancing Researches in SDN: A Survey. IEEE (2017).

11. Kang, B., Choo, H.: An SDN-Enhanced Load-Balancing Technique in the Cloud System. Springer Business Media, New York (2016).

12. Muthanna, A., Ateya, A.A., Makolkina, M., Vybornova, A., Markova, E., Gogol, A., Koucheryavy, A.: SDN Multi-Controller Networks with Load Balanced. ACM Woodstock Conference, Amman, Jordan (2018).

13. Shen, S.-H.: An Efficient Network Monitor for SDN Networks. Sigmetrics 46(2), 2018.

14. Neghabi, A.A., Navimipour, N.J., Hosseinzadeh, M., Rezaee, A.: Load Balancing Mechanisms in Software-Defined Networks: A Systematic and Comprehensive Review. IEEE Access (2018).

15. Al Bowarab, M.H., Zakaria, N.A., Abidin, Z.Z.: Load Balancing Algorithms in Software Defined Network. International Journal of Recent Technology and Engineering 7(6S5) (2019).

16. Hamed, M.I., ElHalawany, B.M., Fouda, M.M., Elden, A.S.T.: A Novel Approach for Resource Utilization and Management in SDN. IEEE (2017).

17. Filali, A., Kobbane, A., Elmachkour, M., Cherkaoui, S.: SDN Controller Assignment and Load Balancing with Minimum Quota of Processing Capacity. IEEE (2018).

18. Chahlaoui, F., RAISS, M., DAHMOUNI, H.: Performance Analysis of Load Balancing Mechanisms in SDN Networks. NISS (2019).

19. Wang, T., Liu, F., Xu, H.: An Efficient Online Algorithm for Dynamic SDN Controller Assignment in Data Center Networks. IEEE/ACM Transactions on Networking 25(5) (2017).

20. Hu, T., Guo, Z., Baker, T., Lan, J.: Multi-Controller Based Software-Defined Networking: A Survey. IEEE Access (2018).

21. Lan, W., Li, F., Liu, X., Qiu, Y.: A Dynamic Load Balancing Mechanism for Distributed Controllers in Software Defined Networking. IEEE (2018).

22. Saldamli, G., Sanjeeva, H., Siddalingaapa, K., Murugesan, R.V., Ertaul, L.: A Secure Collaborative Module on Distributed SDN. In: IEEE International Conference on Information and Communication Systems (ICICS) (2019).

23. Sahoo, K.S., Sahoo, B.: CAMD: A Switch Migration-Based Load Balancing Framework for Software-Defined Networks. IET Networks 8(4), 264-271 (2019).

24. Das, T., Sridharan, V., Gurusamy, M.: A Survey on Controller Placement in SDN. IEEE Communications Surveys & Tutorials (2019).

25. Kumar, S.M., Chakravarthi, P., Jagadeesh, B., Prakash, M.S.: Load Balancing in Cloud Computing. International Journal of Engineering & Technology 7(1.1), 306–310 (2018).

26. Lin, T.L., Kuo, C.H., Chang, H.Y., Chang, W.K., Lin, Y.Y.: A Parameterized Wildcard Method Based on SDN for Server Load Balancing. In: IEEE International Conference on Networking and Network Applications, pp. 978-1-4673-9803-9 (2015).

27. Ruelas, A.M.R., Rothenberg, C.E.: A Load Balancing Method Based on Artificial Neural Networks for Knowledge-Defined Data Center Networking. IFIP LANC, São Paulo, Brazil (2018).

28. Cimorelli, F., Delli Priscoli, F., Pietrabissa, A., Ricciardi Celsi, L., Suraci, V., Zuccaro, L.: A Distributed Load Balancing Algorithm for the Control Plane in Software Defined Networking. In: 24th Mediterranean Conference on Control and Automation (MED), IEEE (2016).

29. Majidi, A., Gao, X., Zhu, S., Jahanbakhsh, N., Chen, G.: Adaptive Routing Reconfigurations to Minimize Flow Cost in SDN-Based Data Center Networks. ICPP, Kyoto, Japan (2019).

30. Ghosh, A., Manoranjitham, T.: A Study on Load Balancing Techniques in SDN. International Journal of Engineering & Technology 7(2.4), 174–177 (2018).

31. Chen-xiao, C., Ya-bin, X.: Research on Load Balance Method in SDN. International Journal of Grid and Distributed Computing 9(1) (2016).

32. Ali, T.E., Morad, A.H., Abdala, M.A.: Load Balance in Data Center SDN Networks. International Journal of Electrical and Computer Engineering 8(5) (2018).

33. Fancy, C., Pushpalatha, M., Pushpa: Experimentation of Traditional Load Balancing Algorithms in Software Defined Network. International Journal of Recent Technology and Engineering 8(1S4) (2019).

34. Kaur, P., Chahal, J.K., Bhandari, A.: Load Balancing in Software-Defined Networking: A Review. Asian Journal of Computer Science and Technology 7(1) (2018).

35. Zakia, U., Yedder, H.B.: Dynamic Load Balancing in SDN-Based Data Center Networks. IEEE (2017).