Received May 1, 2020, accepted May 16, 2020, date of publication May 20, 2020, date of current version June 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2995849

A Systematic Review of Load Balancing Techniques in Software-Defined Networking

MOHAMMAD RIYAZ BELGAUM[1], (Graduate Student Member, IEEE), SHAHRULNIZA MUSA[1], MUHAMMAD MANSOOR ALAM[1,2], AND MAZLIHAM MOHD SU'UD[3], (Member, IEEE)

[1]Malaysian Institute of Information Technology (MIIT), Universiti Kuala Lumpur, Kuala Lumpur 50250, Malaysia
[2]Institute of Business and Management (IoBM), Karachi 74900, Pakistan
[3]Malaysian France Institute (MFI), Universiti Kuala Lumpur, Kuala Lumpur 50250, Malaysia

Corresponding author: Shahrulniza Musa (shahrulniza@unikl.edu.my)

ABSTRACT The traditional networks are facing difficulties in managing the services offered by cloud computing, big data, and the Internet of Things as the users have become more dependent on their services. Software-Defined Networking (SDN) has pulled enthusiasm in the integration process of technologies and function as per the user's requirements for both academia and industry, and it has begun to be embraced in actual framework usage. The emergence of SDN has given another idea to empower the focal programmability of the system. Because of the increasing demand and the scarcity of resources, the load balancing issue needs to be addressed efficiently to manage the incoming traffic and resources and to improve network performance. One of the most critical issues is the role of the controller in SDN to balance the load for having a better Quality of Service (QoS). Though there are few survey articles written on load balancing, there is no detail and systematic review conducted in load balancing in SDN. Hence, this paper extends and reviews the discussion with a taxonomy of current emerging load balancing techniques in SDN systematically by categorizing the techniques as conventional and artificial intelligence-based techniques to improve the service quality. The review also includes the study of metrics and parameters which have been used to measure the performance. This review would allow gaining more information on load balancing approaches in SDN and enables the researchers to fill the current research gaps.

INDEX TERMS Artificial intelligence, conventional, load balancing, review, SDN, software-defined networking, systematic.

I. INTRODUCTION

Software is a common word in the vocabulary of all most all the technologies, and its integration with the telecommunications and networking industry leads to emerging technologies like Software Defined Networking (SDN) [1] and Network Functional Virtualization (NFV) [2]. Along with the advantages, it also brings challenges as they are two sides of the same coin. These technologies require the networking components to embrace software within the hardware to manage them. Though the concept of having centralized control over the network is not new, a holistic view of separation of the control plane from the data plane has made the technical geeks gain insight. The capability of SDN to manage a majority of network components functionality efficiently has earned the confidence of service providers. OpenFlow [3] and Path Computation Element [4] are two technologies supporting SDN. OpenFlow is a standard protocol recommended by the Open Networking Foundation (ONF), which separates the control plane from the switch and provides an interface between the control planes in the controller and the data plane in the switch [5] for communication. Path Computation Element (PCE) is supported by the Internet Engineering Task Force (IETF) [6] and is preferred for closed environments like data centres where the path computation is migrated from the networking component to the controller.

Vendor-specific devices and networks have become very difficult to manage due to cross network operational challenges. The heterogeneous devices are flooded with a sea of vendors providing frameworks, platforms, and end solutions to handle the increasing load. These devices function based on hard coding done for specific legacy networking. The routing protocols and the management issues and challenges still have more added overhead on the network. The solution

The associate editor coordinating the review of this manuscript and approving it for publication was Xingwang Li.

providers [7] need an easy approach to adapt a platform by matching customer requirements so that the adaptability completes the process in quick turnaround time and be protocol friendly.

Load balancing is a method for assigning load to network components to optimize network performance and boost QoS. Load balancing techniques, strategies, and algorithms help both the end-users and service providers to assign or transfer the load to improve efficiency. Load balancing helps to predict the traffic bottleneck before happening. The load balancer [8] is like a network traffic control police.

Knowing the fact that load balancing improves the performance of SDN, there is a minimal review conducted in this area, and this prompted the authors to investigate more in this area. This paper mainly aims at examining the various conventional load balancing techniques as well as the artificial intelligence-based load balancing techniques in SDN to identify the issues to be considered in balancing the load efficiently and enhance the performance.

The authors in [9] have conducted a systematic literature review on load balancing techniques in SDN by examining a total of 19 articles published between the years 2013 and 2017. The articles were categorized into deterministic and non-deterministic approaches. Furthermore, the authors in [10] have surveyed 23 articles published between the years 2013 and 2017 based on nature-inspired meta-heuristic algorithms used to balance the load in SDN. The articles were categorized into ant colony optimization, genetic algorithm, particle swarm optimization, greedy, and simulated annealing. There were a significant number of articles published after 2017, and none of them has considered the articles published after that for review. Therefore, this paper conducts a systematic literature review on load balancing techniques in SDN by collecting the articles from various sources published between 2015 and 2019. Also, from Figure 3, it is evident that around 45% of the total number of articles considered for review are published in the years 2018 and 2019. The reason for considering the recent past five years publications is that the advancement in technology helps to propose and implement new techniques to improve the performance. Moreover, during this period, the development in the field of SDN has accelerated, which is going to have a major impact on future development and invention. Therefore, in this review paper, our significant contributions are:

- We present a systematic literature review of load balancing techniques in SDN
- We discuss the taxonomy of current trends in load balancing techniques by focusing on their improvements and limitations.
- We present the metrics used in measuring the performance of current approaches.
- We identify future research works, which forms the recommendations for current and future research.

The objective of this categorization is to provide the basis for future research. The goal of the evaluation is to analyze and understand the prevailing techniques. This is necessary if further viable methods are to be established, which could improve current techniques or benefit from previous studies. A formal statement expanding through the parts is the next tentative brief part of the review. Section II describes the background of SDN architecture. Section III addresses the related reviews in SDN. The methodology adopted in this paper is established in Section IV, while Section V discusses the results and Section VI starts with addressing the research questions, discusses the various articles published by categorizing the current load balancing techniques based on the policy/strategy used, metrics used and finally puts forward the research trends and open issues in the field of SDN during load balancing, while Section VII summarizes the inference and makes suggestions in this direction for further research.

II. BACKGROUND

An Ethernet Switch [11] comprises of two key segments, the switching segment, and the software, or firmware. When a packet touches the base of one of the switch ports, the device confirms from the firmware what to do with the bundle. The firmware at that point investigates the target address on a rundown (called a MAC table), which contains all the devices that are at present associated with the switch and communicates with the equipment from which port to send the bundle out. This is the point where the SDN finds its way to better perform in the network by isolating the brain of the switch called the Control Plane from the forwarding equipment, or Data Plane. By setting this Control Plane onto a separate server rather than inside the real switch, it adds additional functionalities to the device. This new Control Plane programming moved towards becoming what is presently known as an SDN Controller, and the extra functions are executed as SDN applications that plugin to the controller.

This section focuses on the SDN architecture. Load balancing in SDN is done in different ways, either by applying artificial intelligence techniques or by using traditional or modified traditional techniques. Artificial intelligence techniques are rule-based that mimics human characteristics to implement real-time applications in the system. Eventually, the metrics used to assess load balancing efficiency are listed.

A. SDN ARCHITECTURE

SDN architecture has three layers, namely the infrastructure layer, the control layer, and the application layer, which is illustrated in Figure1. Each of these layers performs specific functions and interact with each other using interfaces.

1) INFRASTRUCTURE LAYER

Starting with the infrastructure layer, it has all the physical network elements like a switch, router, openVswitch [12], wireless access point [13], etc. These devices' primary functions are to receive the request from the client and forward the data to the next layer, i.e., to the control layer. The data plane of these network components moves it to the controller

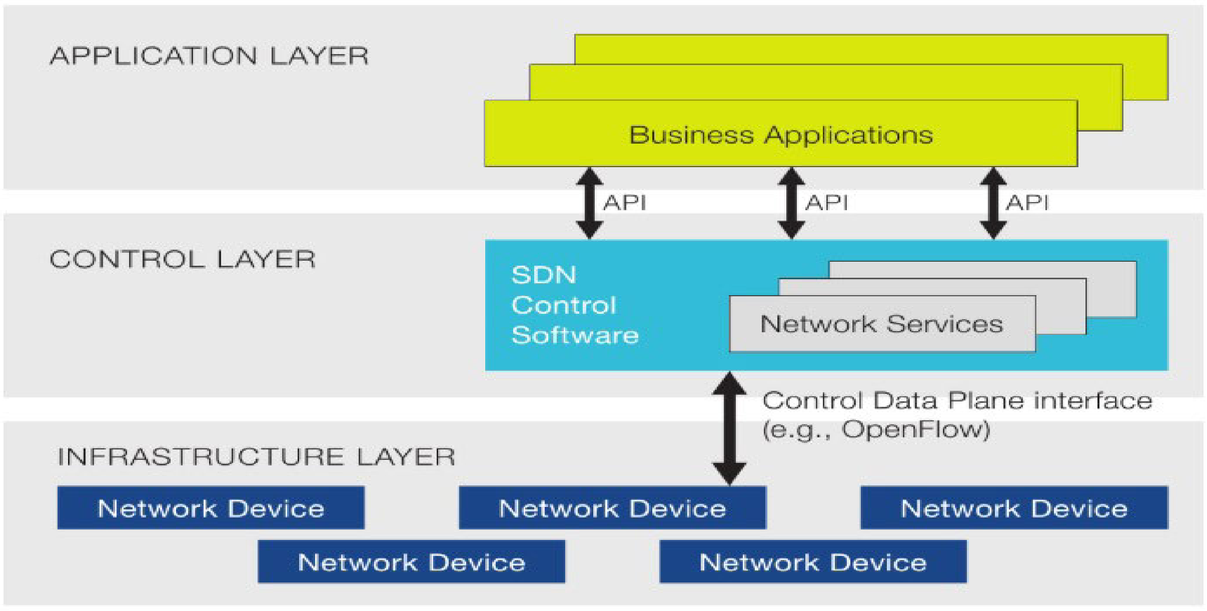

FIGURE 1. SDN architecture [14].

in the flow tables by following the rules. The controller is the key element to define and install the rules in the switches.

2) CONTROL LAYER

The control layer is an intermediary between the lower and upper layers of the architecture. SDN controller [15] is a decision-making module in this layer to balance the load to improve performance. It is also responsible for configuring, managing, and controlling the network elements by communicating with them using flow messages. It offers an abstract and centralized view of the layer of infrastructure. Some of the examples of controllers are ONOS [16], Opendaylight [17], Floodlight [18], Beacon [19], Ryu [20], POX [21], etc. The control layer and the network layer interact using the southbound interface. Some of the southbound interfaces used are OpenFlow [3], Border Gateway Protocol [22] (BGP), SNMP [23], OVSDB [24], NETCONF [25], etc.

3) APPLICATION LAYER

All the end-user applications are in this layer, which has their network requirements and is passed to the control layer through the northbound interface. Some of the northbound interfaces are REST API [26], pyretic [27], Frenetic [28], Procera [29], etc.

SDN also allows full network access via a single programmable controller, regardless of whether the network is in the cloud or physically present. The interaction between the networking device and the controller is done through means of flow messages. Flow messages can be of different categories.

- Group flows: The controller configures the group, and that information is stored in the group tables in the network devices.
- Status: Controller checks the status of network devices like flow_status, port_status, queue_status, group_status, table_status.
- Connection: An echo request and reply message are exchanged between the networking device and controller to verify whether the controller is active or not.
- Asynchronous: Asynchronous messages reach the controller via the networking system from the configured switch to remove the flow rule from the networking device, configure and apply failure, port up / down.

III. RELATED WORK

The ever-growing dependability of the users on cloud services has increased the load on the network. Managing the load has put many challenges for the research community, which grasped their interest to balance the load efficiently using software-defined networking. The current directions to program the network and to automate the process are dragging the researchers and the practitioners in the course of improving load balancing in SDN. Therefore, this review focuses and discusses the current load balancing mechanisms in SDN. The authors in [30] have proposed a mechanism to measure the real-time traffic when the requests are added into the network in passive mode. The increased overheads related to communication cost, controller computation is to be minimized and maintaining accuracy. The proposed techniques were implemented on both fixed and elastic schemas to verify its acceptability by overcoming the challenges related to overhead.

Similarly, to balance the load efficiently, bandwidth is considered as one of the inputs. The growing demand for bandwidth and data transfer is also rising with the number of terminals linked in the network. Therefore, to address this issue, the authors in [31] suggested a load balancing framework for service-oriented SDN-SFC to improve network performance. The request was classified based on the type, and a priority was assigned to each service and then used a heuristic algorithm to decide on taking a specific transmission path among the service function chains available. Furthermore, this technique shortened data transmission time. It improved the degree of load balancing, too, while the other parameters like response time, size of the task, execution time, etc., which could affect the performance of load balancing were paid least importance.

Load balancing gained more importance in software-defined networking as the mechanisms used to affect the QoS provided to the users. QoS can be measured by considering the bandwidth, delay, jitter, and packet loss but not limited to only these. QoS is motivated by resource scheduling, network monitoring, and by routing mechanisms like flow routing, inter-domain routing, etc., and by other QoS oriented mechanisms. The authors surveyed the QoS in software-defined networking in [32] to identify the potential challenges in improving the QoS in SDN. However, the load balancing problem was not investigated here. Improvising the QoS in intradomain has always been the focus rather than inter-domain autonomous system communication QoS provisioning. The main reason behind this is being the network administrators running their proprietary software and are not ready to share their policies. The authors in [33] have conducted a systematic literature review on handling big data applications in the field of healthcare. The vast data sets collected must be efficiently managed and processed by maintaining QoS, at high priority as the data is highly sensitive. In the survey, the authors have highlighted the areas for improvement of big data techniques in healthcare.

A systematic literature review on integrated NFV/SDN was conducted by the authors in [34] to understand the related architectures to address the challenges to improve the architecture designs further. The improvement in the architectural design of integrated NFV/SDN addresses the challenges of reliability, performance, and scalability in SDN. NFV and SDN are supporting technologies striving to provide a standard network solution. Providing network functions to the users at ease is one of the applications of SDN, while NFV utilizes SDN to provide programmable network function. In this review, though many issues of SDN were discussed in future research directions, load balancing was neglected. Without neglecting the prominent technologies related to SDN are cloud computing and Internet of Things (IoT). The authors in [35] have conducted a systematic survey on the integration of cloud computing and IoT. In the review, the authors have highlighted the challenges in the integration by presenting the possibilities for integration in terms of platforms and architectures, and applications. Also, the authors in [36] studied the integration of artificial intelligence in software-defined networking to identify the scope of improving various areas in SDN by AI-based mechanisms. Multiple fields of AI, like machine learning, metaheuristics, and fuzzy inference systems with the subfields in them, were investigated for the inclusion of AI in SDN. Many articles were reviewed, and it was discussed along with the purpose for which AI was included in SDN along with their findings. Very little focus was given on the inclusion of AI for the task of load balancing in SDN.

It is crucial to consider that these studies did not attempt to analyze the existing load balancing approaches and the available problems to be solved. Also, it lacks classification depending on the definition, and the critical role of load balancing within the SDN, specifically in a full literature review. In this article, five questions are formalized in the next portion, which explicitly illustrates the significance of load balancing for SDN, with a focus on the scope of improvements by considering future directions.

IV. METHODOLOGY

This section presents the methodology used in performing a Systematic Literature Review (SLR). Following the guidelines given by the authors in [37], the researchers have done an SLR with a particular focus on load balancing in SDN. Formulating research questions is a part of the SLR, and so the research questions, along with the motivating factors, are mentioned in this section. The articles have been selected from various data sources, as listed below. Furthermore, a specific search strategy has been stratified to get the articles in the domain, which is also discussed here. Later the research articles are selected for review after undergoing thorough scrutiny and after considering the inclusion and exclusion criteria specified below.

To identify the state of the art of load balancing in SDN, the following Table 1 lists the research questions and the motivations to raise such questions.

Research Questions:

Data Sources: The articles from quality publishers like IEEE, Springer, Science Direct, Wiley, ACM digital library, Sage, Inderscience, MDPI, google scholar are collected for review, which is shown in the following Table 2.

Search Strategy: The focus on software-defined networking to improve the network performance and monitoring came into light from 2011, as this technology is much related to cloud computing. Moreover, in the beginning, significant research was not done in this area, so the articles considered for review here is from the past five years, i.e., from 2015. Based on the theme and the proposed research questions, we characterize the seeking watchwords as an initial step to figure the search string.

The researchers likewise considered the search terms ''load balancing,'' ''Software-Defined Networking,'' as the primary keywords. We used the logical operators ''AND'' and ''OR'' for associating the significant watchwords. In the long run, after a few tests, we pick the accompanying search string that gives us the adequate number of related research by considering the keywords mentioned in Table 3 and framed the search string as follows:

TABLE 1. Research questions and motivations.

Questions	Motivations
1. What are the challenges in traditional networks that led to the emergence of SDN?	SDN helps to address the challenges of traditional networks and perform better by utilizing the resources efficiently
2. How is load balancing beneficial for SDN?	It helps to enhance the performance and benefits both the user and the service provider
3. What are the existing strategies, policies, and algorithms used for load balancing in SDN?	Many techniques to ensure load balance in SDN are discussed and compared carefully with other techniques
4. What are the parameters and metrics considered during load balancing in SDN?	It helps to study the parameters affecting load balancing and measure the performance
5. What are the research trends and open issues that are unaddressed in load balancing in SDN?	This article helps the researchers to know the current status and future scope to improve load balancing in SDN

TABLE 2. Database sources.

Publisher	URL
IEEE	http://ieeexplore.ieee.org
Springer	http://link.springer.com
Science Direct	http://sciencedirect,com
Wiley Online Library	http://onlinelibrary.wiley.com
ACM	http://www.acm.org
Sage	http://journals.sagepub.com
Inderscience	http://www.inderscience.com
MDPI	http://mdpi.com
Google Scholar	http://scholar.google.com

Articles Selection Process: The methodology used in the article's selection process starts with framing the research questions. Framing the search string helps in the selection and search process. The articles published in the English language are considered. The PRISMA flow diagram [38] for the scoping review process is followed, which is shown in Figure 2. After finding fitting literary works, a review on load balancing techniques in software-defined networks is done in this research.

The search process closes in all respects by categorizing the load balancing techniques thoroughly to guarantee the completeness of this survey. Most of the articles were screened out since their titles did not apply to the determination criteria, or on the other hand, abstracts were not identified to be considered in this survey.

TABLE 3. List of strings and keywords.

String	Batch1 (B1)	Batch2 (B2)
String1 (S1)	Software-Defined Networking	Load Balancing
String2 (S2)	Software Defined Networking	Load Balance
String3 (S3)	Software-Defined Network	
String4 (S4)	Software Defined Networks	
String5 (S5)	SDN	

Search String: (([B1, S1] OR [B1, S2] OR [B1, S3] OR [B1, S4] OR [B1, S5]) AND ([B2, S1] OR [B2, S2]))

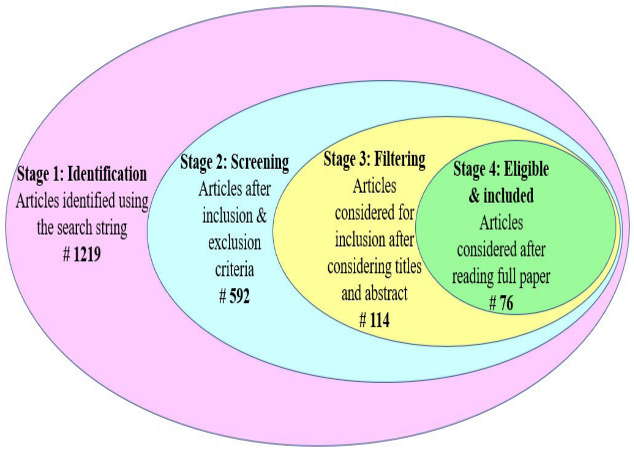
Stage 1: Identification
Articles identified using the search string
# 1219
Stage 2: Screening
Articles after inclusion & exclusion criteria
# 592
Stage 3: Filtering
Articles considered for inclusion after considering titles and abstract
# 114
Stage 4: Eligible & included
Articles considered after reading full paper
# 76

FIGURE 2. Articles selection process for review.

V. RESULTS

As appeared in Figure 2, the underlying inquiry brought about a sum of 1219 articles published between 2015 up to 2019 from different quality publishers, as mentioned in table 1, using the framed search string. For choosing the significant related research, the inclusion and exclusion criteria are connected, which are specified in Table 4, and the articles were reduced to 592 articles. And based on their titles and studies based on their abstracts, the number of selected articles were condensed to 114. From that point onward, 114 articles were studied and entirely checked based on the content matching our categorization of conventional load balancing techniques in SDN or artificial intelligence-based load balancing techniques in SDN, resulting in 76 articles based on the content finally. Based on the set criteria, the essential research articles are selected after the match of title, abstract, and comprehensive published researches for guaranteeing that the outcomes are related to the current research work.

TABLE 4. Inclusion and exclusion criteria.

Inclusion criteria	Exclusion criteria
The study focuses on load balancing in software-defined networks	The study that focuses on other issues in software-defined networks
The articles written only in English are considered	The articles not written in English are not considered
The articles which are published by the above-mentioned scholarly publishers and peer-reviewed journals are considered	The articles which are not published and not peer-reviewed are not considered
The articles published in well-reputed indexed journals and conferences are considered	The articles from white papers, keynote speeches, and editorials are not considered

Inclusion and Exclusion Criteria: From the articles which are selected for review, Figure 3 shows the number of articles published year-wise, which are selected for the study. Figure 4 shows the articles which were published by well-known scholarly publishers between 2015 and 2019. The articles are categorized based on the methods they have used as conventional load balancing techniques and artificial intelligence-based load balancing techniques and publisher. Finally, the number of published articles using conventional load balancing techniques was 49 after elimination. Out of which 24 were from IEEE, 13 were from Springer, five from Elsevier, two from Wiley, one from IET, one from ACM, one from Sage, and two from others as shown in Figure 4. Moreover, the number of articles published based on artificial intelligence was 27. Out of which 13 were from IEEE, five from Springer, two from Elsevier, one from Wiley, one from ACM, and five from others.

VI. DISCUSSION

The literature review has revealed the following facts and findings against each research question.

Q1. What are the challenges in traditional networks that led to the emergence of SDN?

The following are the challenges of traditional networks which are to be addressed using SDN.

Challenges:

- The rules in traditional networks are predefined as to decide on forwarding the packets to the destination. The network operators have a loose control on it, and so all the packets follow the same path. Once congestion [39] arises on the path, all the packets must pass by the same congestion without taking an alternative route even though the alternative path is not congested.
- Lack of visibility into cloud and P2P applications for QoS [40] as the services are limited to port and protocol with policy-based routing [41].
- Lack of application-aware load balancing [42] and intelligent path aware [43] dynamic failover.
- There is no centralized zero-touch provisioning and automation in the traditional networks.
- The high cost of MPLS links [44] and backhauled internet traffic usually do not carry security solutions.
- From the point of network topology, there are some connections between the switches which need many ports than the numbers of servers or end nodes. Adding another degree of network sophistication is the inclusion of virtual switches [45] with increasing complexity.
- In terms of scalability, the customers using IaaS [46] in the cloud perform a wide variety of applications where the traditional VLAN [47] technology does not support as they need logical separation from each other. However, the SDN supports scalable LAN segmentation to function in the cloud environment effectively.

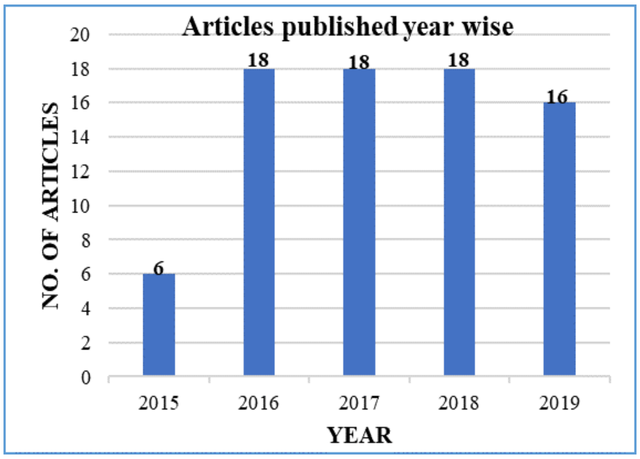

FIGURE 3. Articles selected for review published year wise.

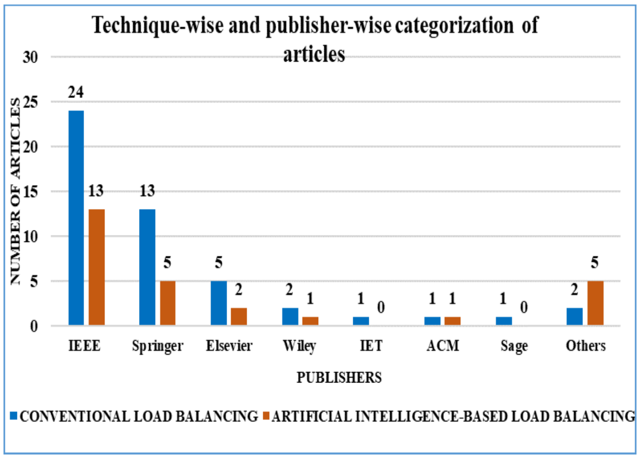

FIGURE 4. Distribution of articles based on technique and publisher-wise.

- In a traditional environment, the network operators cannot easily upgrade the infrastructure and face challenges in real-time as they must meet many constraints. At the same time, SDN can do in a short time with no additional hardware, thereby reducing the CAPEX and OPEX.

SDN addresses the above challenges, and the following are the advantages of SDN and the reasons for its fast acceptability.

Advantages of SDN:

- Flexible and innovative: The networking components in SDN are scalable, making it for the administrator more flexible as the devices are vendor-free. Also, the provision of accessing the multi-vendor elements through a common interface makes it more innovative.
- Decoupling and Abstraction: SDN isolates the network performing functions from the hardware and places them in a centralized environment called controller, where abstraction can be done. The centralized controller supports the efficient management of networks by providing automation and orchestration.
- Centralized management and control: As all the control planes from the switches are moved to a single location, i.e., to a controller, it helps the administrator to have complete traffic control with accessible APIs to effectively handle them, offering a detailed image of the network as a whole. Network administrators may handle traffic loads without difficulty by configuring network code from a single unified controller instead of having multiple machines that are the speciality of Simple Network Management Protocol (SNMP) [48].
- Boosting network visibility: The network visibility allows the network managers to access all the components and helps in the identification and elimination of network errors. SDN caters the administrators to dynamically troubleshoot the delay in traffic flows and adapt to the changing needs.
- Better network security: The security benefits provided by the protocols using SDN are substantially more than the traditional networking protocols.
- Scalability: The switch from legacy technology like Multiprotocol Label Switching (MPLS) 49] to SDN helps to add the users and sites very fast. The performance not only accelerates on-premises but also on the cloud-hosted platforms too.
- Vendor free Environment: Different vendor administrations have their proprietary procedures and functions and are unique. SDN provides a single environment for all the vendors with a standard interface and having centralized control.
- Complexity: SDN reduces the complexity by providing an automated environment to configure and update the networks.
- Programming: The isolation of the control plane from the switches allows having centralized control by programming them and having an abstract view. It enables creating a generalized framework for multiple vendors to have a standard policy. Moreover, it gives scope to test and deploy a new network protocol to improve the functionality effectively.
- Reliability and QoS: Network faults can be easily detected and fixed by unified network monitoring of the system as there is centralized control. Traffic control allows the QoS of network operations to be accomplished.
- Low cost: With existing components and programming them to use has reduced the cost on the part of organizations.

Reasons for Fast Acceptability of SDN:

- Increased cloud usage is the primary reason. With more online connectivity, one crucial concern is that services and data are not managed at the same pace as in the local network today. SDN treats cloud-based services as network devices and thus offers a way of accessing and handling them.
- The other reason is client programmability. OpenFlow right now works with Python and different programming dialects, so the administrator can manage the devices connected and configure them according to the requirements by programming them. Above all this, standards are followed in customizing the network without waiting for the device manufacturers to build those functionalities.

Q2. How is load balancing beneficial for SDN?

This question is answered in two parts by first discussing the need for load balancing in SDN and then explaining the significance of load balancing.

The Need for Load Balancing in SDN: The servers in the network are getting piled with the load as the demand from the users is increasing. So, to provide a better service and fulfilling QoS requirements, the load must be balanced. If this concern is ignored, then it leads to failure of the links and sometimes server crash.

In comparison with the traditional networks, the switches have only data plane with them while separating all the control planes from the switches and moving them to a centralized unit called controller in Software Defined Networks. The extensive technical services provided by the internet are supporting the strategies of Industry 4.0 [50]; most of the traditional industries are moving towards being smart and intelligent. A large amount of data must be compiled and uploaded to the cloud to do so. While to access that data, the networking resources must be allocated dynamically and efficiently for better user satisfaction and better performance. Load balancing is a decision-making module to improve performance in a distributed environment. The load balancer can be either a hardware device or a software module that runs in the control layer of SDN architecture. Load balancing can be static as well as dynamic. In static load balancing, the requirements are predefined, and the rules are already set to follow and fulfil the requirements. However, this type of load balancing is not as efficient as the requirements of the users cannot be static in a real-time environment. Therefore, the need for dynamic load balancing arises. Moreover, dynamic load balancing does not need the requirements before, and the load is distributed based on the changing requirements. The function of a load balancer is the optimization of throughput to improve performance, with low response time, efficiently utilizing the resource with no deadlocks, and not imposing extra overhead on the network. However, with a single controller, there are problems regarding reliability and scalability. So, to overcome this problem, having multiple distributed controllers [51] is a solution when the east-west interface enables those controllers to communicate with each other. The issue of reliability can be resolved as a more significant number of controllers are available and can take over the tasks of a failed controller by the other controllers. In terms of scalability, the networking elements can be added to the times each controller can accommodate. However, the increasing load on each networking element is to be balanced and is an issue to be addressed.

Significance of Load Balancing: The three layers in software-defined networks communicate with each other using interfaces. On the one hand, the network devices present in the infrastructure layer forwards the requests to the control layer. On the other hand, the applications with burdens of various services in the application layer are to fulfil. So, to satisfy both the requirements, the control layer plays a central intelligent role. With the increasing demand of the customers over the cloud services, the number of requests from the clients is increasing, so this puts an increased load on the networking elements to handle them. Management of load raises a concern to efficiently balance the load with the existing infrastructure and gain the satisfaction of the customers by improving the QoS provided to them.

Q3. What are the existing strategies, policies, and algorithms used for load balancing in SDN?

This paper has classified the articles broadly as conventional load balancing techniques and artificial intelligence-based load balancing techniques based on the methods used in the articles selected for review from different sources.

Load Balancing Techniques in SDN: In this review, we explored the load balancing techniques and algorithms implemented by various researchers and sorted them based on the methodologies, which fall in one of these two techniques. Load balancing related problems include the analysis of requests from the network device, defining and mapping the flow tables, server response time, and finding a path to the destination. Based on the above criteria, 76 articles are selected for the review. Each article is discussed in terms of the technique used along with the problem addressed, advantages, and disadvantages. In this section, the findings are reviewed to perform load balancing in SDN by adopting either artificial intelligence-based technique or non-artificial intelligence or conventional technique. Conventional load balancing techniques are classical techniques which rely on symbolic logic and formalism and follows a top-down approach in problem-solving. On the other hand, artificial intelligence-based load balancing techniques are the techniques which mimic the outcome based on the pattern of previous behaviour. The problems are solved using case-based reasoning, rule-based systems, genetic models, swarm intelligence and hybrid models. Artificial intelligence-based load balancing techniques follow a hybrid approach in problem-solving. The objective of this categorization is the base to choose or enhance any one of the techniques in the future to balance the load in software-defined networks.

A. CONVENTIONAL LOAD BALANCING TECHNIQUES

The conventional load balancing techniques are the current techniques in use to balance the load. These techniques use the traditional algorithms for load balancing, and prominent of them include round-robin technique, equal-cost multipath routing protocol, least connections, random techniques, etc. Initially, all selected conventional load balancing techniques with their characteristics are discussed in Table 5, and later, the metrics used to measure the performance are presented in Table 7.

The researchers in the above Table 5 have proposed conventional load balancing techniques to balance the load. The authors in [52] balanced the load with a focus on the controller. The incoming traffic is balanced using Virtual SDN (vSDN) controller duplication and share the load. For example, in a vSDN network, one application in the master controller will periodically assign virtual IPs to other controllers, and the master controller then maps virtual IPs to actual IPs. The addition of vSDN controller plays the role of virtual network function (VNF), where the copy of the controller is the same as that of the first controller, and the existence of both the controllers are made transparent to all the cloud users. While in [94], using the distributed controllers, the proposed mechanism splits the traffic into transmission control protocol (TCP) and user datagram protocol (UDP) and uses a failover mechanism to create high availability environment leading to guaranteed reliability. Furthermore, in [54], when a message from a controller's flow rule arrives in TALON, the flow rule first becomes a physical context, which translates the target switch and the virtual address into a switch and physical address. Then TALON collects the trajectory information for each tenant and assigns the performance by using multi-path calculations, which consider the required throughput and bandwidth available for each link. Finally, flow rules are created based on the calculated paths, and TALON sends messages from the flow rule to the appropriate physical switches. In [71], the load balancer takes over the prioritizing of QoS flow requirements and transfers them to the controllers based on weight coefficients. If there is a connection between the traffic streams and the network, they are classified as two major traffic types: critical (real-time) and non-critical traffic (non-real time). To ensure QoS, the priority procedure provides that the controllers process

TABLE 5. Characteristics of selected conventional load balancing techniques in SDN.

Reference	Policy / Strategy / Algorithm	Problem addressed	Improvements / Achievements	Weakness / Limitations
Ejaz et al. [52]	Controller duplication and migration	Efficiently balanced load using VSDN and VNF	• Delay reduced • Improved Bandwidth utilization • Improved throughput	• Increased complexity • Additional controllers added • High energy consumption and carbon emission
Xu et al. [53]	Switch migration	Serial switch requests and non-serial switch requests load balanced	• Low computation time • Few switch migrations	• Path computation is assumed • Assumption of the cost of installing flow entry
Jin et al. [54]	Tenant throughput allocation	Efficient throughput management	• Improvement in throughput • Traffic requirements satisfied	• Lack of attention on latency • Lack of reliability
Gao et al. [55]	Hierarchical geographical routing protocol	Network congestion and sparse connectivity	• Packet transmission performance improved	• Lack of accuracy • Lack of communication between nodes • Overhead
Hu et al. [56]	All pairs shortest path	Improved load balancing	• Guaranteed quality of transmission • Adapts to mobile network	• Increased computation cost • Overload on controller
Lan et al. [57]	Hierarchical control plane	Static switch controller mapping	• Dynamically balancing load • Few switch migrations	• Underutilization of resources • Topology dependent
Tao et al. [58]	Load balance factor	Controller selection	• Reduction in flow request cost	• Parameters affecting network performance not considered
Gao et al. [59]	Centralized algorithm based on the max utility function	Load balancing in LTE-V networks	• Improving uplink and downlink rate • Load balancing	• Fixed threshold • Performance metrics not considered
Koryachko et al. [60]	Dynamic load balancing	Alignment of load and optimal routes	• Shortest path • Network optimization	• Numerically evaluated
AbdelRahman, A. S., & El-Sisi, A. B. [61]	Three phases algorithm	Overloading in access point and congestion	• Minimal packet loss rate • Improved transfer rate	• Load factor not evaluated
Attarha et al. [62]	Re-routing algorithm	Congestion control and load balancing	• Best backup path • Short running time	• Depends on the transmission rate • Maintenance of backup paths • Other factors not considered
Bremler-Barr et al. [63]	Mbalancer scheme	The need for extra servers	• Better control over traffic • Better load balancing	• Bottleneck on throughput
Chen et al. [64]	Adaptive connection	Load balancing and hand over	• Improved network performance	• Static • Cost ineffective
Chuang, P. J., & Chen, H. J. [65]	Global and partial strategies	Load balance and network stability	• Improvement in QoS	• Overhead
Han, T. and N. Ansari [66]	Distributed algorithm	Load balancing and power consumption	• Minimize communication overhead	• Latency increased
Wang et al. [67]	Dynamic load balancing	Load balancing and flexibility	• Efficient resource allocation Security	• Cost
Fizi, F. S., & Askar, S.[68]	Novel load balancing algorithm	Congestion control and load balancing	• Throughput increased • Packet loss rate reduced	• Complexity in re-routing
Hu et al. [69]	Binary integer problem	Controller placement and load balancing	• Minimized avg delay and worst delay	• Results in local optimum • Cost
Hwang et al. [70]	On-line routing	Optimal path	• Reduces cost • Improves utilization	• Not suitable for multipath TCP connections
Hai et al. [71]	Pre-defined load	Load balancing and	• Minimized communication	• Depends on static

TABLE 5. ***(Continued.)*** **Characteristics of selected conventional load balancing techniques in SDN.**

Reference	Policy / Strategy / Algorithm	Problem addressed	Improvements / Achievements	Weakness / Limitations
	threshold	control plane reliability	overhead • Increased load distribution	load
Ramdhani et al. [72]	Multipath routing algorithm	Traffic flow handling and balancing network load	• Better path decision • Better load balancing	• Congestion • Overhead
Kaur et al. [73]	Round-Robin	Load distribution	• Uniform load distribution	• Not dynamic
Yilmaz et al. [74]	Server load balancing	QoS for video streaming	• Delay reduced • Overload of server	• Other QoS metrics not evaluated
Tu et al. [75]	Middlebox	Load balancing and QoS guarantee	• Improved bandwidth utilization • Latency reduced	• Single point failure
Hu et al. [76]	Switch migration	Multi-controller load	• Efficient load balancing • Efficient migration	• Reliability • Security
Wang et al. [77]	Cross-domain load balancing	Utilization of network and computing resources	• Better load balancing • Communication synchronization	• Security • Prone to attacks
Mulla et al. [78]	Load balancing technique	Traffic segregation and load balancing	• Better load balancing • Improved network QoS • Better traffic segregation	• Overhead • Delay
Vyakaranal, S. B., & Naragund, J. G [79]	Weighted Round Robin	Load balancing	• No starvation • Improved load balancing	• Biased as admin assigns weight
Kang, B., & Choo, H [80]	Intercloud manager	Traffic handling to balance the load	• No system saturation • Improved system performance	• Overhead due to additional messages
Shang et al. [81]	Adaptive link load balancing	Imbalance of load	• Effectively balances network traffic • Increased bandwidth utilization • Achieves QoS	• Overhead
Lin et al. [82]	Dynamic load balancing	Network congestion	• Improved load balancing degree • Improved re-association time	• User priorities not considered • QoS not considered
Ma et al. [83]	Multi-controller load balancing	Load balancing in a multi-controller environment	• Efficient load distribution	• Inefficient use of resources
Song et al. [84]	Flowstealer	Load balancing under burst traffic	• Minimal switch migrations • Better throughput	• Overhead • Security
Yang et al. [85]	Power saving algorithm	Reduction of power consumption and load balancing	• Better load balancing • Reduced power consumption	• QoS not considered
Chen et al. [86]	Dynamic load balancing	Network congestion and load balancing	• Efficient bandwidth utilization • Better load balancing	• Overhead • QoS metrics not considered
Kaur, K., et al. [87]	Least time-based weighted round-robin	Load balancing	• Better response time • Executes more transactions	• Server load not considered
Kaur, S. and J. Singh [88]	Direct routing-based algorithm	Load balancing with minimizing latency	• Better throughput • Better load balancing • Improves response time	• Suitable for a single controller
Priyadarsini et al. [89]	Self-adaptive load balancing scheme	Switch migration and load balancing	• Minimum switch migrations • Better throughput	• Cost • Delay not considered
Zhao et al. [90]	Controller placement load balancing	Load balancing on multiple controllers	• Reduction of load difference • Reduced migration cost • Better load balancing	• Scalability
Zhong et al. [91]	Server response time	Load balancing and efficient resource utilization	• Better load balancing • Better resource utilization	• Energy efficiency not considered • Overhead
Rangisetti et al.[92]	QoS aware load balancing	Load balancing in LTE networks	• Reduces overload • Better network performance	• Delay and jitter not considered
Boero et al. [93]	Load balancing by rerouting	Load balancing by meeting deadlines	• Better load balancing • Reduced switch migrations	• Network congestion
Gasmelseed, H. and R. Ramar [94]	Traffic pattern-based load balancing	Load balancing and availability issues	• Avoids single point failure • Improved reliability	• Increased cost • Increased delay

TABLE 5. ***(Continued.)*** **Characteristics of selected conventional load balancing techniques in SDN.**

Reference	Policy / Strategy / Algorithm	Problem addressed	Improvements / Achievements	Weakness / Limitations
			• Better controller management • Better load balancing	
Chakravarthy, V. D. and B. Amutha [95]	Simple path algorithm	Load balancing in data centre networks	• Successful packets exchange • Reduced packet loss • Better load balancing	• Delay not considered • Throughput not measured
Sahoo, K. S. and B. Sahoo [96]	Controller adoption and migration	Load balancing and switch and controller selection	• Better load balancing • Reduced migration cost • Improved QoS	• Suitable for static load • Priorities not considered
Raza et al. [97]	Dynamic load balancing	Load balancing in proxy mobile IPv6	• Better load balancing • Improved uplink and downlink	• Low scalability • Low availability • QoS not considered
Yao et al. [98]	Double threshold load balancing	Load balancing with scalability	• Reduced load on the super controller • Reduced jitter	• Other QoS metrics not considered
Shi et al. [99]	Multi-controller deployment algorithm	Load balancing with controller deployment	• Efficient load balancing • Reduced delay	• Other QoS metrics not considered • Security
Wang et al. [100]	Novel multi-controller placement	Load balancing through clustering and controller placement	• Efficient load balancing • Reduced latency	• Other QoS metrics not considered • Isolation reduces connectivity

critical traffic before non-critical traffic based on necessary data on the packet header. To identify the unloaded controller and concurrently call switch relocation, the authors suggested strategies for controller adaptation and migratory decision using the medium network load in [96]. A migration index factor is used to pick the target controller that shows the difference between migration cost and variance in load among idle controllers. To handle the traffic coming from communication-intensive applications, the authors in [75] tackle this difficult problem by adding a programmable middlebox that can spread traffic equally. The middlebox is a Clos network architecture that utilizes SDN to maximize capacity utilization, thus maintaining QoS. To learn the traffic distribution and serve, the Middlebox SDN controller collects information from switches and servers and processes them intending to improve bandwidth utilization and minimizing latency.

Moreover, the controller placement technique proposed in [58] describes first the overall flux function costs that take into account switching weights, routing costs from a switch to the controller, and routing costs for intercontrollers. Next, with a known number of controllers, a controller-based load balance factor is proposed, which results in the positioning and linear function of the load balance factor and total flow demand cost of the controllers. Some studies on the control system placement problem have now been carried out in SDN from various areas such as delay, capacity, reliability, and load balance. The issue of placement of the controller from delay and load balance is discussed in [69]. The authors have conducted a study for a specific topology, to minimize the average delay or the worst delay of all control paths, to select the places of multiple controllers regarding load balance. The authors in [90] suggested that multiple controllers are a solution to reduce latency between controller and switch. However, balancing the load on such controllers and deploying them at correct locations is a challenge that was addressed by them using a hierarchical control architecture. The controller and switch placements were adjusted to balance the load better. According to simulative findings, the proposed technique proves to be efficient in terms of migration costs and balancing the load.

Furthermore, the multi-controller deployment algorithm proposed in [99] balances the load among the controllers placed in different regions. Moreover, the controller placement was decided based on the least number of switches present under the controller within an area considering the topology. In [83], the mechanism used by the authors was the implementation of a hierarchical control plane, both a meta control plane and a local control plane in a multi-controller SDN environment for optimizing the processing performance. The authors in [100] proposed a novel algorithm for multi-controller placement based on different parameters. Initially considering latency, clusters are formed, then the isolated nodes are eliminated to maintain connectivity, and then load balancing has been carried out.

Considering switch migration, some researchers have proposed techniques to balance the load. In [53], two methods were introduced, one for the requests to process which do not arrive in a serial manner named BalCon and the other called BalConPlus to handle different types of requests. In either case, the migration of switches is minimal and reduces the migration cost and balances the load effectively. The authors in [84] proposed Flow Stealer, the lightweight method for load balancing distributed SDN controllers. Flow Stealer uses a low-cost flow stealing method by stealing flow events, which allows idle controllers to temporarily sharing workloads with overloaded controllers. The proposed method not only reacts more quickly to network traffic changes, but it also reduces the switching frequency. Flow Stealer often involves flow theft and turn movement to respond to traffic congestion and long-term shifts in behaviour. The researchers in [76] focus on minimizing cost and time of migration and proposes an algorithm for migrating optimally. The calculated values are compared with the threshold to trigger the proposed technique. In [91], the authors presented a new way of load balancing SDN networks to improve load balance by reducing the response time of the server. The higher the load on the cluster, the longer it takes to respond, the higher the servers in a server cluster have several similar performances and offer the same service. The longer the response time, the larger the load. Based on the duration of the response, the load balance problem in the server cluster is solved. The authors in [60], [67], [82], [86], [97] have used dynamic load balancing techniques to perform load balancing and optimize the use of resources by finding the shortest path to reach the destination and also improve the QoS performance.

With the advent of energy harvesting technologies, green energy can be used by Base Stations (BS) to reduce power demand on the grid. The traffic load balance is essential for mobile grids with high BS density to take advantage of small cell base stations (SCBs) capacity. Green energy use should be incorporated into traffic load balancing strategies as a performance measure to use harvested energy fully. The authors in [66] proposed a distributed algorithm to balance the load and minimize the overhead by utilizing green energy. For the load balance of the LTE and Wi-Fi integrated network, an appropriate network coordination mechanism is required. The proposed technique in [85] provides a right base station selection (e.g., LTE evolved node BS or Wi-Fi access points) for user equipment (UE) for load balance using the access network discovery and selection function (ANDSF). The ANDSF is integrated with software-defined networking (SDN) to make the ANDSF network more programmable, flexible, and dynamically manageable.

Furthermore, an ANDSF (PSA) power-saving algorithm is proposed for the proper assignment of network resources to UEs and reduced Wi-Fi access points (APs) power consumption. In [92], a centralized LTE RAN (SD-LTE-RAN) framework and new QoS Aware Load Balance (QALB) algorithm are proposed to address load imbalance problems. The QALB algorithm takes several neighbour cells, UEs, and QoS profiles and expected outputs of neighbouring cells into consideration in making load-balance choices. The proposed mechanism [98] adopts the distribution and centralization methods and design a double threshold load allocation by using hybridflow in a wide-area wireless network environment within multiple controllers to balance the load effectively. To optimize the resource allocation, the authors in [77] proposed an Extensive Messaging and Presence Protocol (XMPP) based cross-domain load balancing mechanism(CDLB) for SDN in the Cloud data centre. Unlike the polling method, the proposed scheme is an XMPP-based push model that prevents the wasting of network and computer resources in a broadly distributed network environment. The proposed system requires all controllers on the centralized control level to communicate in real-time, utilizing XMPP and the XMPP publication / subscribe extension, the same compatible global network details. The authors in [80] introduced an SDN-enhanced Inter-cloud Manager (S-ICM) that assigns cloud-based network flows. S-ICM contains two primary parts: surveillance and decision-making. For monitoring, S-ICM uses an SDN control message which observes and collects data and decides on network delay in packet measurements.

Round-robin is a technique based on the queuing model. The incoming requests are assigned to the server for a fair amount of time and avoid the problem of starvation. It incurs overhead as the requests which do not complete execution within the quantum of time allocated has to be saved and wait for the turn on a cycle basis. Sometimes the priority requests also have to wait for the resources to be allocated. The authors in [73], [78], [79], [87] have either used the method or modified versions directly by adding weights or considering the least time needed to perform load balancing.

Based on routing and re-routing, various techniques have been proposed by the researchers to balance the load. The proposed method in [68] employs re-routing if the use of the bandwidth falls below and the loss reaches a specified value, the algorithm begins and balance the load by re-routing some traffic to a new path with lower use of the bandwidth. In [88], the return request from the webserver to the application is not included in the load balancer. This ensures that the server explicitly reacts to the request, thus improving performance. The authors in [74] proposed a framework across an OpenFlow to enhance video streaming service quality, where the controller program manages the complex load of the system and dynamically re-routes to low loaded servers when surge situation is observed. The authors in [59] proposed two new heterogeneous LTE vehicle networks (LTE-V HetNets) to improve the transmission performance of beacon messages. The study focuses on LTE-V HetNets with the mechanism for uplink-downlink disconnection. The software-defined network (SDN) architecture is built into LTE-V to achieve centralized control. The SDN framework proposes the centralized usefulness function-based algorithm that allows global load equilibrium to be achieved through the self-adjustment of the number of vehicles connected to each base station. The problem caused by overburdening or congestion in a specific access point (AP) is alleviated in [61]. It depends on the mobile station being forced to transfer to another AP in the overlapping area between APs. It works in three phases; AP load calculation, AP load monitoring, and forced delivery phases. Furthermore, in [55], a hierarchical geographical routing protocol was proposed by the authors based on the concept of SDN in vehicular ad hoc networks to identify the vehicle density in each grid and evaluate the cost function to balance the load with minimal cost path. Since individual routing adjustment strategies often take a remarkable amount of time in the software-defined network (SDNs) to achieve load balance and network stability, the authors in [65] provides two different routing adjustment strategies-the global and partial, for reducing the time consumption. It takes less time to achieve the desired load balance because, as opposed to individual strategies, it rearranges the optimal path once. In a significantly reduced period compared with different approaches, the global strategy will, at the same time, adjust the routings to achieve the desired load balance. Network congestion is a result of increasing load on the network and degrades the QoS. So re-routing the traffic is a solution, where the authors in [56], [62], [70], [72], [93], [95] have used and modified the flow rules to improve the degree of load balancing.

The authors in [57] adopt a load balancing mechanism based on the hierarchical control plane. Considering the controller factor and switch factor, the total load of each controller is estimated. Then the proposed mechanism moves the load from the high-loaded controller to a quick charge controller with switch migration. In [63], the authors suggested MBalancer, a simple load balance program for Memcached, that can be integrated seamlessly into Memcached software-defined networks (SDN) architectures. MBalancer functions as an SDN device that duplicates hotkeys on a variety of (or all) servers. The SDN controller updates the SDN transmission tables with SDN ready-to-use load balancing functionality.

The researchers in [81] used information allocation flow requests strategy, considering the current load and propagation delay of idle controllers. The idle controller is assigned part of the flow request information with a minimum of flow delay and propagation requests. The overloaded controller triggers the load balancing based on the magnitude of the table to avoid incoherence of network status. Secondly, an adaptive service-sensitive load balance mechanism is designed to determine the service types through the northbound interface and periodically observe the network state. The adaptive load-balancing algorithm is suggested, and the connection weight dependent on QoS-aware is implemented, which tests the maximum efficiency of the link by collecting the QoS in real-time. In [64], the authors proposed an SD-WiFi network load balance scheme called adaptive connection and hand-off (ACH), which considers the overload and hand-out process together to balance the load. Moreover, in [89], the authors suggested a self-adapting load balancing (SALB) scheme, which dynamically manages load between multiple controllers through multiple moving switches from source to goal controllers. The critical feature in the framework is the efficient transfer of load under high loads while also considering the difference between switches and having control.

B. ARTIFICIAL INTELLIGENCE BASED LOAD BALANCING TECHNIQUES

Artificial Intelligence-based techniques use a metaheuristic approach to solve real-world problems. The sub-areas of artificial intelligence include deep learning, neural network, natural language processing, knowledge representation, reasoning (logical and probabilistic), and decision making with search, planning, and decision theory. Initially, all selected artificial intelligence-based load balancing techniques with their characteristics are discussed in Table 6, and later, the metrics used to measure the performance are presented in Table 8.

Artificial intelligence-based load balancing techniques provide better learning abilities and foster decision making in SDN. The various techniques used by the researchers, as shown in Table 6, states the policy used in balancing the load along with the problem addressed as well as the advantages and disadvantages of each of the techniques proposed. In this section, a description of how each of the techniques was used will be discussed.

Some of the researchers have used a heuristic approach in migrating the switches from highly loaded controllers to low loaded controllers while fulfilling the task of load balancing. The authors in [101] present a heuristic approach to solving the migration problem. Swap movements and shifts are integrated into a search scheme. The value to both immigration and exit controllers is measured at every move. In comparison to current methods, when a switch migration is not feasible, the suggested algorithm will not halt the search. Instead, it searches for more complex movements such as switching two keys to boost more performance. In [126], the authors proposed an SDN based publish/subscribe (SDNPS) system that can build and finalize subject-related overlays efficiently and without redundancies, based on a global topology overview, to disseminate events. It organizes and codes the subjects into binary strings, as a Huffman tree so that SDN-configurable switches can run filtering and forwarding events to reduce end-to-end latency. This hierarchical organization of the huffman tree allows the construction and storage of overlays, which lessen the computing process time and space complexity.

The researchers in [104] used optimization technique for load balancing and also minimizing congestion running in four phases. Initial two stages create sub-topology, which improves the performance by reducing the space. In the third phase, the process to balance the load is implemented, and the last phase does a post-processing task of creating paths and injecting corresponding flows in the switches. The online controller load balancing (OCLB) technique proposed in [102] focuses on balancing the load by reducing the average response time of the controller. The switch migrations are done considering the distribution of real-time application

TABLE 6. Characteristics of selected artificial intelligence-based load balancing techniques in SDN.

Reference	Policy / Strategy / Algorithm	Problem addressed	Improvements / Achievements	Weakness / Limitations
Al-Tam, F. and N. Correia [101]	Heuristic approach	Load balancing and switch migration	• Better load balancing • Optimal controller selection	• Centralized and suffers from single point failure • Migration cost not evaluated • Underutilization of bandwidth
Zhang et al. [102]	Online controller load balancing	Load balancing and scalability	• Minimum switch migrations • Reduced response time • Better load balancing	• Other QoS metrics not considered • Requests from diverse flow not considered
Pietrabissa et al. [103]	Game theory	Load balancing and congestion	• Improved scalability • Minimum communication • No single point failure	• Increased SDN proxies • Fixed convergence properties
Farhoudi et al. [104]	Heuristic algorithm with optimization	Load balancing and network congestion	• Better load balancing • Reduced response time • Increased throughput	• Lacks reliability • Lacks scalability • Network performance not measured
Wang et al. [105]	Greedy method	Migration considering the tradeoff between migration cost and load balance rate	• Improves elasticity of SDN • Efficient switch migration	• Overhead • Security
Wang et al. [106]	NP-Hardness	Load balancing and route deployment	• Reduced control load • Improved throughput	• Other QoS metrics not considered
Mahlab et al. [107]	Optimization using genetic algorithm	Optimal load distribution	• Better link utilization • Minimized cost	• Results in local optimum
Yu et al. [108]	Deep Neural Network and Backpropagation	Load balancing with efficient path selection	• Better load balancing • Optimal path • Flow forecasting	• Overhead • Limited parameters used
Cimorelli et al. [109]	Game theory & Wardrop equilibrium	Load balancing across multiple controllers	• Reduced latency • Improved throughput	• Results in local optimum • Cluster overhead
He et al. [110]	Modified Particle Swarm Optimization	Load balancing	• Better load balancing • Reduced delay • Improved QoS	• Increased energy consumption • Security • Degree of load balancing not evaluated
Liao et al. [111]	Binary tree	Load balancing by adjusting flow table rules	• Better load balancing • Maximize server usage	• Complex • Topology dependent • QoS metrics not used
Sathyanarayana, S., & Moh, M. [112]	Ant colony optimization	Load balancing with network performance	• Reduced latency • Improved throughput	• Overhead
Yang et al. [113]	Neural Networks and K-means	Load balancing with a prediction of flow requests	• Better controller selection • Applied on cloud services • Better load balancing	• Energy consumption • Inaccurate as prediction used
Manzoor et al. [114]	Support vector machines and Fuzzy based Particle Swarm Optimization	Load balancing and scalability	• Better load balancing • Better throughput • Reduced latency	• Flow categorized based on priority • Static • Overhead
Li et al. [115]	Fuzzy logic	Load balancing	• Better load balancing • Efficient performance in communication	• Overhead • Frequent sleep and restart of the server
Zhao et al. [116]	Rounding based route joint deployment	Load balancing and optimization of flows	• Reduced control load • Better load balancing • Reduced flow rules • Minimized deployment delay	• Static
Duan et al. [117]	Queuing model	Load balancing and forwarding delays	• Fits for 5G networks • Better load balancing	• Application sensitive • Other QoS metrics not considered
Wang et al. [118]	Rounding algorithm with NP-Hardness	Load balancing and scalability	• Better load balancing • Better QoS	• Load increases beyond a threshold

TABLE 6. *(Continued.)* **Characteristics of selected artificial intelligence-based load balancing techniques in SDN.**

Reference	Policy / Strategy / Algorithm	Problem addressed	Improvements / Achievements	Weakness / Limitations
			• Better controller utilization	
Xu et al. [119]	NP-Hard	Load balancing with flow re-routing overhead	• Better load balancing • Reduced overhead	• Static traffic • Limited controllers
Sahoo et al. [120]	Karush-Kuhn-Tucker	Load balancing with optimal controller selection	• Better load balancing • Increased end-user utility • Better QoS	• Communication overhead • Priorities
Manzoor et al. [121]	Analytical hierarchical process	Load balancing with flow request processing	• Better load balancing • Increased throughput • Reduced delay • Increased successful request delivery	• Few parameters used for AHP
Xue et al. [122]	Genetic-Ant colony optimization	Load balancing	• Reduced transmission time • Reduced packet loss rate • Optimal search	• Overhead • Few parameters considered
Bao et al. [123]	Fuzzy logic	Load balancing and scalability	• Better load balancing • Improved network performance	• Overhead • Static load
Rofie et al. [124]	Dijkstra's algorithm	Load balancing with the best path	• Better load balancing • Better network performance • Best path	• Results in local optimum
Lemeshko, O. and O. Yeremenko [125]	Two-level fast re-routing	Load balancing	• Better load balancing • Reduced computing power • Reduced cost	• Addition resources • Other QoS metrics not considered
Wang et al. [126]	Heuristic algorithm	Load balancing and multicasting	• Better load balancing • Avoidance of imprudent forwarding • Improved inter-cluster routing	• Overhead • Static load
Sahoo et al.[127]	Decision making based on TOPSIS	Load balancing with switch migration	• Efficient switch migration • Better load balancing • Reduced overhead • Reduced migration time	• Traffic demands not considered • Fit for small scale environment

with a dependable parameter aimed at decreasing response time. The optimal solution is dependent on the conditions for termination and migration where OCLB determines. The authors in [107], genetic-based optimization technique to minimize the load imbalance in software-defined elastic optical networks, and reduce the cost of service delay. It works in three phases. Initially, the optimizer selects the available solutions. Secondly, the defragmentation algorithm verifies the feasibility of each solution, and finally, the power budget algorithm evaluates the validity. The optimization techniques are merged with other methods to give the best solutions like in [112], the authors merged the ant colony optimization technique with dynamic load balancing technique with each of them serving their purpose. The dynamic load balancing technique is used to find the least loaded server, and the optimization technique is used to find the optimal path to reach the server.

Furthermore, the authors in [122] utilized the strength of a genetic algorithm with ant colony optimization to generate the best solutions in handling the load imbalance and convergence latency. Another optimization technique is the particle swarm optimization (PSO), in which the solution convergence rate is fast. The authors in [110] have used a modified PSO to balance the load in the wireless Internet of Vehicles. The purpose of this technique is to efficiently find the vehicular nodes when the remaining energy is less so that the power consumption is reduced, and the delivery of information is reliable. Karush-Kuhn-Tucker (KKT) conditions are the necessary constraints that are to be met to guarantee a limited local minimum optimum. The authors in [120] used the KKT conditions to find the optimal controller in the process of balancing the load. Demand and supply curve based SDN (DSSDN) is used to know the load factors, and then KKT conditions are used by OpenFlow device to improve response time and make optimal decisions efficiently. The authors in [114] use a support vector machine (SVM) to classify the traffic based on priority. Markov Chain Model (MCM) is used to predict the load of controllers and using Type-2 Fuzzy based Particle Swarm Optimization (TFPSO) optimal controller is selected. The researchers in [116] have considered hybrid routing by joint optimization problem and first proved it to be NP-Hard. Then Rounding-based Route Joint deployment (RRJD) algorithm is used to solve the problem and improve the network performance. The authors in [125], proposed a two-level fast re-routing technique to solve the mixed-integer linear programming (MILP) optimization problem and puts forth the primary and secondary paths and ensuring load balancing.

The authors used three functionalities, which include path selection, significant nodes, and flow prediction using intelligent algorithms in [108]. Deep Neural Networks (DNNs) and Q-learning are used for path selection. Network traffic information is complicated and cannot be distributed according to a specific probability. So DNNs are appropriate than the probability to process traffic. The results of DNNs are used to find an optimal route. Then the significant nodes are identified, and later flow rules are predicted to prevent abnormal flow blocking off the path.

Similarly, the authors in [113] used a backpropagation neural network and K-Mean cluster to train the network and forecast if the user would access the networking device in the future or not. The queuing model used in [117] to verify whether or not the flow rule is present in the switch to reach the destination for a new request. The offload manager (OM) in this technique predicts the future paths by executing the proposed algorithm considering the delay threshold. Using the binary tree concept, the authors adjust the networking elements load in [111]. Wild card rules are used to adjust the IP address and are distributed to different regions. Then a dynamic algorithm is run to update the flow table considering the server utilization. The researchers in the SDN system utilizes a form of load balancing to provide network congestion management in [124] by finding the best path using Dijkstra's algorithm.

Fuzzy logic is another subset of artificial intelligence, which is used to solve the load balancing problem in SDN. The authors in [115] used a fuzzy function to analyze the parameters affecting the server load. SDN control feature is used for monitoring server data in the entire network and program the virtual server functions. Servers freeze and restart to have an increased performance, which balances the load and energy consumption dynamically. The authors proposed a similar kind of technique in [123] by measuring the performance using other metrics and ensuring load balancing with improved network performance.

The authors in [106] explicitly describe the load balance and low delay route deployment problems for the control connection and illustrate NP-hardness to improve QoS by taking the control link constraints and other data plane restrictions in SDN into consideration. The issue of performing controller load balancing and link load balancing is formulated as NP-Hard in [118]. The rounding-based algorithm is proposed to solve the problem and provide better scalability and reduce the load. Similarly, other authors in [119] suggest partial flow statistics collection (PFSC) problem to be NP-Hard. Load balancing is done to reduce the overhead caused due to flow re-routing and obtain an optimal solution by considering the quality of flow statistics. The authors in [121] use the Analytical Hierarchical Process (AHP) method to assign the flow to each controller, considering various constraints. The controllers are organized as global and local. The global controller handles cluster formation and the local controller handles local device load and clustering is updated timely.

Game theory is another subset of artificial intelligence used to handle social situations among competing entities. This concept has been used by the authors in [109] to balance traffic across a cluster of SDN controllers. The proposed algorithm trains the network to learn the appropriate flow rate from switch to controller in attaining Wardrop equilibrium and converges to stable policies resulting in balancing the load. Similarly, in [103], the authors proposed a load balancing approach which is noncooperative and learns Wardrop equilibrium. The SDN Proxy receives the requests, and these proxies forward the request to the controller using the proposed method. Better load balancing is achieved by migration, which is one of the solutions but involves overhead. The authors in [105] have used a greedy method to have an efficient switch migration, thereby balancing the load. Whenever there is a load imbalance, it is triggered by a metric, and then the migration is done by evaluating the tradeoff between migration costs and load variation. The authors in [127] have proposed a framework to balance the load in SDN using switch migration. The target controller selection is made considering multi-criteria. Decision analysis is conducted by using a method called ''Technique for Order Preference by Similarity to an Ideal Solution'' to identify the controller nearest to the ideal solution to migrate the load.

Q4. What are the parameters and metrics considered during load balancing in SDN?

In this segment, we discuss the metrics and criteria used for the study of load balance efficiency in the current studies. It helps to assess and know its comparative performance and consider the advantages and disadvantages of it, and to equate it to previous methods. Such criteria apply to consistency parameters. During the review, 21 metrics have been identified and are discussed below. The metrics used by different researchers in each of their techniques used by conventional load balancing and artificial intelligence-based load balancing are listed in Table 7 and Table 8, respectively.

Metrics Used for Assessing the Load Balance Efficiency in SDN: For load balance in the SDN, the following qualitative criteria are described.

1. Throughput: The rate of successful requests processed in a unit of time from the source to the destination an as throughput[52].

$$Throughput = \sum\nolimits_{requestsi}^{n} (timet) \quad (1)$$

2. Response time: It is the time difference that the request was sent, and when it addresses the start of processing [53], [128].

$$Response_{time} = abs_{request\ i}(sub_{time} start_{time}) \quad (2)$$

where sub_{time} is the time at which the request is submitted, and $start_{time}$ is the time at which the request starts to process.

3. Execution time: It is the difference of time at which the request completes or finishes the processing and the time the request starts to process [80].

$$Execution_{time} = request_i(finish_{time} start_{time}) \quad (3)$$

where $finish_{time}$ is the time at which the request has been completed processing, and $start_{time}$ is the time at which the request starts processing.
4. Round Trip Time (RTT): Round trip time is the time taken by a packet to move from source to destination and back [71]. RTT is a necessary performance measurement element because it is the time to wait until an ACK is transmitted before a segment is retransmitted. If the estimated time for the round trip is below the actual time of the round trip, segments are transmitted earlier than the authentic phase or if the corresponding ACK has spread through the network. If the duration of the round trip is too high, timeouts are longer than necessary and are therefore less successful.
5. Resource utilization: It is measured as the efficient use of the resource to process the request, and the objective is to maximize the most important resource in processing the request [129]. The resources can be either CPU or memory or Input/Output or network or bandwidth.

$$ResourceUtilization = \frac{\sum_{i\ requesti}^{n} Execution_{time}}{MaxrequestExecution_{time}} \quad (4)$$

6. Delay: Time taken by a packet to move from node to node is called delay. There are different types of delays like communication delay [130], routing delay [131], processing delay [132], migration delay [133].
7. Packet delivery ratio: The ratio of the total number of packets received at the target to the total number of packets sent at the source [55] is termed as packet delivery ratio.
8. End to End delay or Latency: End to End delay [134] is the time taken by the packet to transit from the source to destination in a network. It includes the transmission delay, propagation delay, and switch delay caused by the switch while forwarding the packet.

$$EED = TD_{sr} + PD + TD_{dt} + SD \quad (5)$$

where EED is the end to end delay, TD_{sr} is the transmission delay at the source, PD is the propagation delay, TD_{dt} is the transmission delay at the destination and SD is the delay at the switch.
9. Energy consumption: The amount of energy consumed by each node in the network to process a request irrespective of whether the processing is a success or a failure is called energy consumption [66], [135], [136]. Energy consumption is minimized if load balancing is done effectively.
10. Packet loss rate: The total number of packets not reached by the number of packets sent to the destination is called packet loss. The rate at which the packets loss [137] is known as a packet loss rate. The objective is always to have a low packet loss rate to ensure efficiency.
11. Degree of load balancing: It is a metric used to measure the distribution of load on the networking element. This metric can be measured using several indices like Jain's fairness index [82], [106] and load balance rate [96], [105], [113].
12. Number of migrations: It is the number of times the packet is shifted from one switch to another in the path of reaching the destination. To have an efficient performance of communication, the number of migrations should be less [53], [84], [102].
13. Migration cost: Migration costs [105] includes the exchange of message costs and load costs between the networking elements. To deliver the packet to the destination, some switch needs to communicate with the other switch, which is termed as message costs and to have a balanced load between switches or controllers, the cost incurred to transfer load between switches or controllers is termed as load cost.
14. Overhead: Excess cost, time, space incurred during communication is termed as overhead [81].
15. Uplink/downlink rate: During the process of communication between the networking elements, the speed at which the packets are transmitted is said to be uplink/downlink rate [106].
16. Threshold miss probability: Duan *et al.* in [117], have defined a metric called threshold miss probability, to measure the delay in 5G heterogeneous networks. The likelihood of a WiFi network not meeting the latency requirements of service is specified as a threshold miss probability. The threshold error likelihood should be reduced to improve the performance at the end of the customer.
17. Peak load ratio: The maximum traffic load on each link is termed as a peak load ratio [119].
18. Jitter: The difference in the latency of packet transmission from one networking element to the other is known as jitter. When there is congestion in the network, jitter is increased [120].
19. Re-association time: In the process of WiFi load balancing based on SDN, the authors in [82] have defined a metric called re-association time, where it is the time taken to associate the client stations to a lightly loaded access point (AP) decided by the controller. A huge re-association time results in massive data transfer, which consumes enormous computing power by the SDN controller.
20. Availability: Availability [94] is characterized as the positive communication ratio of controllers and servers that exclude servers and control errors.

$$Availability = \frac{S_f + Time\_out}{\Sigma C_t} \quad (6)$$

In the above equation, S_f is the socket failure, *Time_out* is the time spent by the controller waiting for the

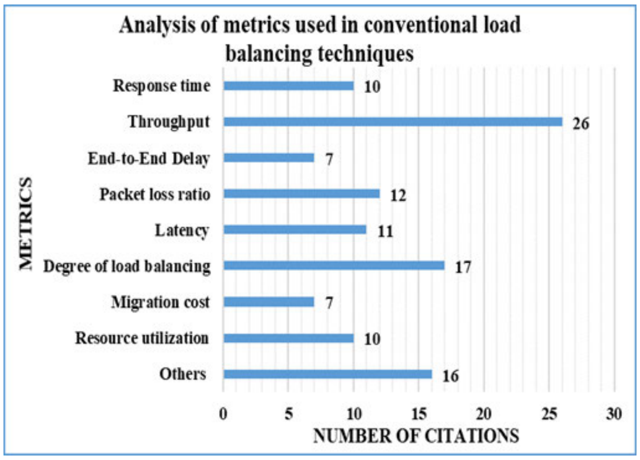

FIGURE 5. Analysis of metrics used in conventional load balancing techniques.

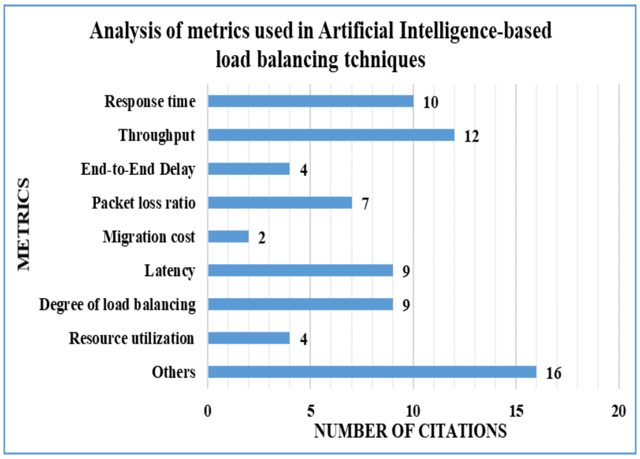

FIGURE 6. Analysis of metrics used in artificial intelligence-based load balancing techniques.

message, and C_t is the number of attempts made for connections.

21. Concurrency: It is defined as the number of mutual communications provided by the server, as given by the authors in [94]. The function of concurrency decreases with the decrease in server efficiency.

This study focuses on the qualitative metrics having an impact on load balancing in SDN. The metrics like response time, throughput, end-to-end delay, packet loss ratio, latency, degree of load balancing, migration cost, resource utilization are used as qualitative metrics. From the conventional load balancing techniques, considered in our study, it is evident from Figure 5 that the metrics considered are qualitative. It shows the number of times each metric was used to measure the performance in different conventional load balancing techniques. The count of response time is 10; throughput is 26; the end-to-end delay is 7; packet loss ratio is 12; latency is 11; the degree of load balancing is 17; migration cost is 7; resource utilization is 10, and others contribute to a total of 16. Similarly, from the artificial intelligence-based load balancing techniques, considered in our study, it is also evident from Figure 6 that the metrics are qualitative. The count of response time is 10; throughput is 12; the end-to-end delay is 4; the packet loss ratio is 7; migration cost is 2; latency is 9; the degree of load balancing is 9; resource utilization is 4, and others contribute to a total of 16 as shown in Figure 6.

Additionally, Figure 7 shows the number of times each metric was used in each year in different conventional load balancing techniques. Similarly, Figure 8 shows the number of times each metric was used in each year in artificial intelligence-based load balancing techniques.

Q5. What are the research trends and open issues that are unaddressed in load balancing in SDN?

SDN has been facing challenges of tremendous scale off late. Though there are several industries like banking, insurance, manufacturing, healthcare, central / state government, transportation, education, customer services, etc., telecommunications stand on the top as it connects all the industries mentioned above by allowing to network and communicates with each other. SDN is deployed in different ways in the

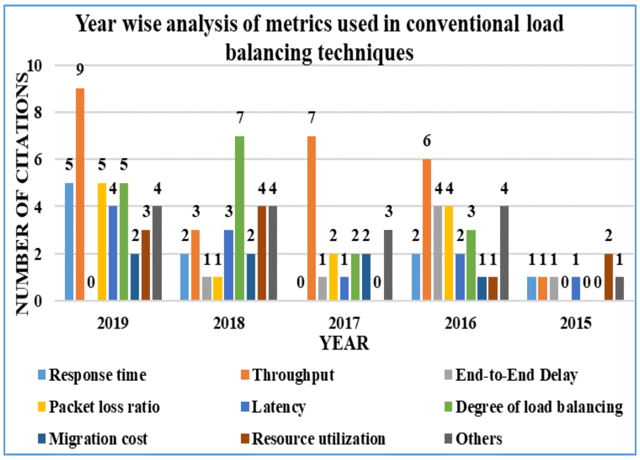

FIGURE 7. Year-wise analysis of metrics used in conventional load balancing techniques.

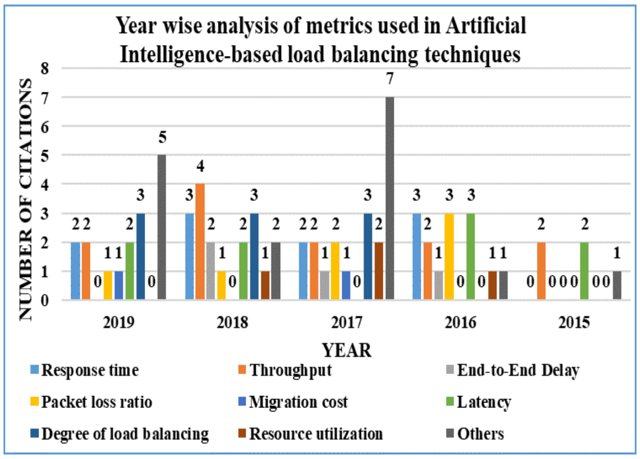

FIGURE 8. Year-wise analysis of metrics used in artificial intelligence-based load balancing techniques.

industry. However, the central concept is to manage the network function with a centralized controller keeping the forwarding plane with the hardware device itself.

Trends and Open Issues:

The following are some of the trends and issues in SDN.

➢ Further SDN functions are responsible for the administration, monitoring, and engagement of a network transfer cluster within a distributed architecture. A failure of one controller can be taken over by the other to keep the networking function active by reducing the risk of failure, but still, there is an overhead of shifting a load of the failed controller to the active ones. The degree of overhead reduction in migration of load has not been considered in the current techniques, which can be explored in the future. And in an environment where there is only one centralized controller, the failure of risk is higher as the entire network comes down with the failure of the controller.

➢ The load balancing on the SDN controller and the lag of contact between controller and switches are a severe challenge to SDN. In an environment where there is one centralized controller, the load increases significantly with the increase of users or applications accessing the services. Very little research has been done with a focus on traffic-aware load balancing, and none of the researchers have considered the requests coming from delay-sensitive data applications. The connection establishment requests and the exception of the traffic increase on the controller cause high communication delay, which can be considered as future direction to reduce it.

➢ Another challenge from the point of load balancing is that the load should be distributed considering the efficiency of the switch. The load migrations from one network element to others is done as the intermediate element is nearer to the destination. However, this results

TABLE 7. Metrics used in conventional load balancing techniques in SDN.

Reference	Response time	Throughput	End-to-End Delay	Packet loss ratio	Packet delivery ratio	Latency	Round trip time	Execution time	Degree of load balancing	No. of migrations	Energy consumption	Migration cost	Resource utilization	Overhead	Uplink/Downlink rate	Threshold miss probability	Peak load ratio	Jitter	Re-association time	Availability	Concurrency
Ejaz et al. **[52]**		*				*			*				*								
Xu et al. [**53**]	*																				
Jin et al. **[54]**		*																			
Gao et al. **[55]**		*			*	*				*											
Hu et al. **[56]**				*					*				*								
Lan et al. **[57]**		*							*												
Tao et al. **[58]**									*			*									
Gao et al. **[59]**															*						
Koryachko et al. **[60]**						*						*									
AbdelRahman, A. S., & El-Sisi, A. B. **[61]**				*											*			*			
Attarha et al. **[62]**		*		*																	
Bremler-Barr et al. **[63]**		*	*																		
Chen et al. **[64]**		*							*												
Chuang, P. J., & Chen, H. J. **[65]**		*										*									
Han, T. and N. Ansari **[66]**			*								*										
Wang et al. **[67]**		*																			
Fizi, F. S., & Askar, S.**[68]**		*		*																	
Hu et al. **[69]**			*						*												
Hwang et al. **[70]**		*		*																	
Hai et al. **[71]**							*		*					*							
Ramdhani et al. **[72]**		*	*	*																	
Kaur et al. **[73]**	*	*																			
Yilmaz et al. **[74]**						*							*								

TABLE 7. ***(Continued.)*** **Metrics used in conventional load balancing techniques in SDN.**

Reference	Response time	Throughput	End-to-End Delay	Packet loss ratio	Packet delivery ratio	Latency	Round trip time	Execution time	Degree of load balancing	No. of migrations	Energy consumption	Migration cost	Resource utilization	Overhead	Uplink/Downlink rate	Threshold miss probability	Peak load ratio	Jitter	Re-association time	Availability	Concurrency
Tu et al. **[75]**			*										*								
Hu et al. **[76]**	*	*							*			*									
Wang et al. **[77]**		*		*		*															
Mulla et al. **[78]**		*		*		*															
Vyakaranal, S. B., & Naragund, J. G **[79]**		*				*							*								
Kang, B., & Choo, H **[80]**	*							*													
Shang et al. **[81]**		*				*							*								
Lin et al. **[82]**									*										*		
Ma et al. **[83]**		*																			
Song et al. **[84]**		*																			
Yang et al. **[85]**									*		*										
Chen et al. **[86]**						*							*								
Kaur, K., et al. **[87]**	*	*																			
Kaur, S. and J. Singh **[88]**	*	*										*									
Priyadarsini et al. **[89]**		*		*					*					*							
Zhao et al. **[90]**									*			*									
Zhong et al. **[91]**	*												*								
Rangisetti et al.**[92]**		*							*												
Boero et al. **[93]**				*		*															
Gasmelseed, H. and R. Ramar **[94]**	*	*		*																*	*
Chakravarthy, V. D. and B. Amutha **[95]**	*								*				*								
Sahoo, K. S. and B. Sahoo **[96]**	*	*		*					*			*						*			
Raza et al. **[97]**			*																		
Yao et al. **[98]**																		*			
Shi et al. **[99]**						*			*												
Wang et al. **[100]**			*						*				*								

in high response time as the load difference, as well as the traffic estimation, was not considered in the previous research. Therefore, future work in this direction is very interesting.

➢ Messages are shared between switches and controllers during the process of load balancing. During this process, the messages are not free of threats, and there is every possibility that the messages get attacked by various threats. So, to handle this challenge, there should be some secured co-operative mechanism for proving the authenticity of messages in the platform to manage them smartly, which has not been considered in the current techniques and is another future direction.

➢ The OpenFlow rules updating problem is another exciting challenge to address as the rules need to be updated based on the dynamic network behaviour. Fault-tolerant rules are to be developed to support this robust approach. Also, the impact of default rules to act in forwarding the flow table to controllers and software switches increases the processing delay. Various versions of OpenFlow support flat table, multi-level table, and pipeline processing, and to take the best advantage, the issue of a specific rule to be placed in a flow table must be given attention. Therefore, the combination of prediction of flow rules and the controller placement problem from past experiences is another direction for future work to be addressed to optimize network performance.

➢ Energy efficiency has grabbed the attention of the researchers while performing load balancing in SDN. The topology framing and cluster formation play a significant role while the communication is carried out

TABLE 8. **Metrics used in artificial intelligence based load balancing techniques in SDN.**

Reference	Response time	Throughput	End-to-End Delay	Packet loss ratio	Packet delivery ratio	Latency	Round trip time	Execution time	Degree of load balancing	No. of migrations	Energy consumption	Migration cost	Resource utilization	Overhead	Uplink/Downlink rate	Threshold miss probability	Peak load ratio	Jitter	Re-association time	Availability	Concurrency
Al-Tam, F. and N. Correia **[101]**		*						*	*	*											
Zhang et al. **[102]**	*								*	*											
Pietrabissa et al. **[103]**		*	*																		
Farhoudi et al. **[104]**	*	*																			
Wang et al. **[105]**	*							*	*			*									
Wang et al. **[106]**				*					*						*						
Mahlab et al. **[107]**		*							*												
Yu et al. **[108]**			*	*																	
Cimorelli et al. **[109]**	*					*															
He et al. **[110]**			*	*																	
Liao et al. **[111]**	*																				
Sathyanarayana, S., & Moh, M. **[112]**	*	*		*		*															
Yang et al. **[113]**						*			*												
Manzoor et al. **[114]**		*	*						*												
Li et al. **[115]**	*												*								
Zhao et al. **[116]**		*												*	*						
Duan et al. **[117]**		*				*										*					
Wang et al. **[118]**	*								*												
Xu et al. **[119]**					*												*				
Sahoo et al. **[120]**		*		*		*												*			
Manzoor et al. **[121]**		*				*															
Xue et al. **[122]**				*			*														
Bao et al. **[123]**	*	*																			
Rofie et al. **[124]**						*							*								
Lemeshko, O. and O. Yeremenko **[125]**													*								
Wang et al. **[126]**		*		*		*		*					*								
Sahoo et al. **[127]**	*					*			*			*		*							

between the nodes. So, an efficient cluster formed along with cluster head rotation can be considered further while the intra-cluster and inter-cluster communication are performed, to reduce energy consumption.

➢ With the growing demand for SDN virtualization, cloud services have attracted the attention of industry and academia. Network Function Virtualization replaces the networking elements with the software-based functions on giant volume servers at the data centres. The question of how and where to place these network functions to enable NFV to the cloud users to balance the load efficiently has not been taken into consideration in the present works and can be considered as future work.

➢ Another challenge can be from the point of the data plane because traditional switches may not easily support SDN due to evolving standards. Therefore, a hybrid approach can be used in this scenario by partially implementing SDN in accessing the network, while the other part is still doing a core network. To implement SDN immediately, the SDN enabled platforms can be in actual switches while the traditional platforms are in intermediate nodes without supplementing the entire network. Furthermore, an interface to communicate between them is another future direction.

➢ Very little research has been done in load balancing using artificial intelligence-based techniques. In contrast, a combination of two or more techniques resulting in a hybrid approach can be used effectively in balancing the load as a future direction, making it very interesting.

- ➢ The researchers in the current techniques have used a few parameters to access the QoS, but not all. So, as a future direction, all parameters can be considered to measure the QoS efficiently.
- ➢ The application of SDN architectural principles to a WAN network is termed as Software Defined Wide Area Network (SD-WAN). And the challenge it has put forth is to manage thousands of sites ensuring service delivery and scalable security by compelling and providing cost-effective technology for the enterprises and the service providers. With massive centralization and inefficient load balancing mechanism, both the users and service providers are struggling, as observed from the current research. So, as a future direction, while load balance proliferation in SDN is essential, the mechanism must be optimized.

Limitations of This Review: This review has considered a limited number of databases, Journals, and Conferences. Further, a limited number of keywords, strings have been used for the searching of literature. The articles published before 2015 have not been included in this review. Mainly, this review focuses on only load balancing issues and not on any other related issues in SDN.

VII. CONCLUSION AND RECOMMENDATIONS

This paper provides a systematic review of load balancing techniques and algorithms used by different researchers. The selected papers are categorized into conventional and artificial intelligence-based load balancing techniques based on the methodology used to solve the load balancing problems in SDN. This paper also discusses the issues addressed, strategies used, and solution results. Based on various studies considered in this study, we find that multiple techniques did not consider some essential criteria and that enhancing the efficiency of the existing techniques is crucial. This study allows network administrators, service providers, and end-users to undertake additional research in the future to improve the efficiency in load balancing in SDN.

Software-defined networking is an emerging architecture that should efficiently balance the load for better network performance and improving QoS. Further work on the target distribution of services with finite resources is needed. Also, extensive research on load balancing schemes concerning the green optimization of the data centre is recommended. This review aims to provide a possible path for further study in load balancing in Software-defined networking.

REFERENCES

[1] Y. Zhang, L. Cui, W. Wang, and Y. Zhang, "A survey on software defined networking with multiple controllers," *J. Netw. Comput. Appl.*, vol. 103, pp. 101–118, Feb. 2018.

[2] B. Yi, X. Wang, K. Li, S. K. Das, and M. Huang, "A comprehensive survey of network function virtualization," *Comput. Netw.*, vol. 133, pp. 212–262, Mar. 2018.

[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[4] J. L. Roux, *Path Computation Element (PCE) Communication Protocol (PCEP)*. Fremont, CA, USA: IETF, 2009.

[5] (2020). *Open Networking Foundation*. Accessed: Jan. 10, 2020. [Online]. Available: https://www.opennetworking.org/sdn-definition/

[6] M. Handley, O. Bonaventure, and U. C. de Louvain, *Internet Engineering Task Force (IETF) A. Ford Request for Comments: 6824 Cisco Category: Experimental C. Raiciu*. 2013.

[7] A. Tsuchiya, F. Fraile, I. Koshijima, A. Ortiz, and R. Poler, "Software defined networking firewall for industry 4.0 manufacturing systems," *J. Ind. Eng. Manage.*, vol. 11, no. 2, pp. 318–333, 2018.

[8] D. E. Eisenbud, "Maglev: A fast and reliable software network load balancer," in *Proc. 13th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2016, pp. 523–535.

[9] A. A. Neghabi, N. J. Navimipour, M. Hosseinzadeh, and A. Rezaee, "Load balancing mechanisms in the software defined networks: A systematic and comprehensive review of the literature," *IEEE Access*, vol. 6, pp. 14159–14178, 2018.

[10] A. A. Neghabi, N. J. Navimipour, M. Hosseinzadeh, and A. Rezaee, "Nature-inspired meta-heuristic algorithms for solving the load balancing problem in the software-defined network," *Int. J. Commun. Syst.*, vol. 32, no. 4, p. e3875, Mar. 2019.

[11] P. T. Congdon, P. Mohapatra, M. Farrens, and V. Akella, "Simultaneously reducing latency and power consumption in OpenFlow switches," *IEEE/ACM Trans. Netw.*, vol. 22, no. 3, pp. 1007–1020, Jun. 2014.

[12] B. Pfaff, "The design and implementation of open vswitch," in *Proc. 12th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2015, pp. 117–130.

[13] O. B. Karimi, J. Liu, and J. Rexford, "Optimal collaborative access point association in wireless networks," in *Proc. INFOCOM IEEE Conf. Comput. Commun.*, 2014, pp. 1141–1149.

[14] Open Networking Foundation. (2020). *SDN Architecture ONF TR-502*. Accessed: Mar. 3, 2020. [Online]. Available: https://www.opennetworking.org/wp-content/uploads/2013/02/TR_SDN_ARCH_1.0_06062014.pdf

[15] Y. E. Oktian, S. Lee, H. Lee, and J. Lam, "Distributed SDN controller system: A survey on design choice," *Comput. Netw.*, vol. 121, pp. 100–111, Jul. 2017.

[16] P. Berde, "ONOS: Towards an open, distributed SDN OS," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw. (ACM)*, 2014, pp. 1–6.

[17] J. Medved, R. Varga, A. Tkacik, and K. Gray, "OpenDaylight: Towards a model-driven SDN controller architecture," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Jun. 2014, pp. 1–6.

[18] V. B. Harkal and A. Deshmukh, "Software defined networking with floodlight controller," *Int. J. Comput. Appl.*, vol. 975, no. 3, p. 8887, Jul. 2016.

[19] D. Erickson, "The beacon openflow controller," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2013, pp. 13–18.

[20] J. H. Cox, S. Donovan, R. J. Clarky, and H. L. Owen, "Ryuretic: A modular framework for RYU," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Nov. 2016, pp. 1065–1070.

[21] S. Kaur, J. Singh, and N. S. Ghumman, "Network programmability using POX controller," in *Proc. Int. Conf. Commun., Comput. Syst. (ICCCS)*, vol. 138, 2014, pp. 134–138.

[22] S. Papadopoulos, K. Moustakas, A. Drosou, and D. Tzovaras, "Border gateway protocol graph: Detecting and visualising Internet routing anomalies," *IET Inf. Secur.*, vol. 10, no. 3, pp. 125–133, May 2016.

[23] M. Slabicki and K. Grochla, "Performance evaluation of CoAP, SNMP and NETCONF protocols in fog computing architecture," in *Proc. IEEE/IFIP Netw. Operations Manage. Symp. (NOMS)*, Apr. 2016, pp. 1315–1319.

[24] F. Volpato, M. P. D. Silva, and M. A. R. Dantas, "OFQuality: A quality of service management module for software-defined networking," *Int. J. Grid Utility Comput.*, vol. 10, no. 2, pp. 187–198, 2019.

[25] M. Dallaglio, N. Sambo, F. Cugini, and P. Castoldi, "Control and management of transponders with NETCONF and YANG," *J. Opt. Commun. Netw.*, vol. 9, no. 3, p. B43, 2017.

[26] F. Haupt, F. Leymann, A. Scherer, and K. Vukojevic-Haupt, "A framework for the structural analysis of REST APIs," in *Proc. IEEE Int. Conf. Softw. Archit. (ICSA)*, Apr. 2017, pp. 55–58.

[27] B. He, L. Dong, T. Xu, S. Fei, H. Zhang, and W. Wang, "Research on network programming language and policy conflicts for SDN," *Concurrency Comput., Pract. Exp.*, vol. 29, no. 19, p. e4218, Oct. 2017.
[28] N. Foster, R. Harrison, M. J. Freedman, C. Monsanto, J. Rexford, A. Story, and D. P. Walker, "Frenetic: A network programming language," *ACM SIGPLAN Notices*, vol. 46, no. 9, pp. 279–291, 2011.
[29] A. Voellmy, H. Kim, and N. Feamster, "Procera: A language for high-level reactive network control," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, 2012, pp. 43–48.
[30] H. Tahaei, R. Salleh, S. Khan, R. Izard, K.-K.-R. Choo, and N. B. Anuar, "A multi-objective software defined network traffic measurement," *Measurement*, vol. 95, pp. 317–327, Jan. 2017.
[31] W.-C. Chien, C.-F. Lai, H.-H. Cho, and H.-C. Chao, "A SDN-SFC-based service-oriented load balancing for the IoT applications," *J. Netw. Comput. Appl.*, vol. 114, pp. 88–97, Jul. 2018.
[32] M. Karakus and A. Durresi, "Quality of service (QoS) in software defined networking (SDN): A survey," *J. Netw. Comput. Appl.*, vol. 80, pp. 200–218, Feb. 2017.
[33] A. Pashazadeh and N. J. Navimipour, "Big data handling mechanisms in the healthcare applications: A comprehensive and systematic literature review," *J. Biomed. Informat.*, vol. 82, pp. 47–62, Jun. 2018.
[34] M. S. Bonfim, K. L. Dias, and S. F. L. Fernandes, "Integrated NFV/SDN architectures: A systematic literature review," *ACM Comput. Surv.*, vol. 51, no. 6, p. 114, 2019.
[35] M. N. Jahantigh, A. M. Rahmani, N. J. Navimirour, and A. Rezaee, "Integration of Internet of Things and cloud computing: A systematic survey," *IET Commun.*, vol. 14, no. 2, pp. 165–176, Jan. 2020.
[36] M. Latah and L. Toker, "Artificial intelligence enabled software-defined networking: A comprehensive overview," *IET Netw.*, vol. 8, no. 2, pp. 79–99, Mar. 2019.
[37] S. K. Boell and D. Cecez-Kecmanovic, "On being 'systematic' in literature reviews," in *Formulating Research Methods for Information Systems*. London, U.K.: Palgrave Macmillan, 2015, pp. 48–78.
[38] M. D. J. Peters, C. M. Godfrey, H. Khalil, P. McInerney, D. Parker, and C. B. Soares, "Guidance for conducting systematic scoping reviews," *Int. J. Evidence-Based Healthcare*, vol. 13, no. 3, pp. 141–146, Sep. 2015.
[39] T. Hafeez, N. Ahmed, B. Ahmed, and A. W. Malik, "Detection and mitigation of congestion in SDN enabled data center networks: A survey," *IEEE Access*, vol. 6, pp. 1730–1740, 2018.
[40] F. Li, J. Cao, X. Wang, and Y. Sun, "A QoS guaranteed technique for cloud applications based on software defined networking," *IEEE Access*, vol. 5, pp. 21229–21241, 2017.
[41] M. Tanha, D. Sajjadi, R. Ruby, and J. Pan, "Traffic engineering enhancement by progressive migration to SDN," *IEEE Commun. Lett.*, vol. 22, no. 3, pp. 438–441, Mar. 2018.
[42] H. Mekky, F. Hao, S. Mukherjee, Z.-L. Zhang, and T. V. Lakshman, "Application-aware data plane processing in SDN," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, 2014, pp. 13–18.
[43] Y.-J. Chen, Y.-H. Shen, and L.-C. Wang, "Traffic-aware load balancing for M2M networks using SDN," in *Proc. IEEE 6th Int. Conf. Cloud Comput. Technol. Sci.*, Dec. 2014, pp. 668–671.
[44] G. A. Mazhin, M. Bag-Mohammadi, M. Ghasemi, and S. Feizi, "Multi-layer architecture for realization of network virtualization using MPLS technology," *ICT Exp.*, vol. 3, no. 1, pp. 43–47, Mar. 2017.
[45] P. Emmerich, D. Raumer, F. Wohlfart, and G. Carle, "Performance characteristics of virtual switching," in *Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2014, pp. 120–125.
[46] R. Ghosh, F. Longo, F. Frattini, S. Russo, and K. S. Trivedi, "Scalable analytics for IaaS cloud availability," *IEEE Trans. Cloud Comput.*, vol. 2, no. 1, pp. 57–70, Jan. 2014.
[47] S. Chidambaram, A. A. Abou-Emara, H. S. Bhasin, N. P. Saraiya, J.-J. Yeh, and C. D. Youngworth, "Storage traffic communication via a switch fabric in accordance with a VLAN," Oracle Int. Corp., Redwood City, CA, USA, Tech. Rep. US8743872B2, 2014.
[48] P. P. Ray, "An introduction to dew computing: Definition, concept and implications," *IEEE Access*, vol. 6, pp. 723–737, 2017.
[49] A. Bahnasse, F. E. Louhab, H. A. Oulahyane, M. Talea, and A. Bakali, "Novel SDN architecture for smart MPLS traffic engineering-diffServ aware management," *Future Gener. Comput. Syst.*, vol. 87, pp. 115–126, Oct. 2018.
[50] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2018.
[51] A. A. Dixit, F. Hao, S. Mukherjee, T. V. Lakshman, and R. Kompella, "ElastiCon: An elastic distributed SDN controller," in *Proc. 10th ACM/IEEE Symp. Archit. Netw. Commun. Syst. (ANCS)*, Oct. 2014, pp. 17–27.
[52] S. Ejaz, Z. Iqbal, P. A. Shah, B. H. Bukhari, A. Ali, and F. Aadil, "Traffic load balancing using software defined networking (SDN) controller as virtualized network function," *IEEE Access*, vol. 7, pp. 46646–46658, 2019.
[53] Y. Xu, M. Cello, I.-C. Wang, A. Walid, G. Wilfong, C. H.-P. Wen, M. Marchese, and H. J. Chao, "Dynamic switch migration in distributed software-defined networks to achieve controller load balance," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 515–529, Mar. 2019.
[54] H. Jin, G. Yang, B.-Y. Yu, and C. Yoo, "TALON: Tenant throughput allocation through traffic load-balancing in virtualized software-defined networks," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2019, pp. 233–238.
[55] Y. Gao, Z. Zhang, D. Zhao, Y. Zhang, and T. Luo, "A hierarchical routing scheme with load balancing in software defined vehicular ad hoc networks," *IEEE Access*, vol. 6, pp. 73774–73785, 2018.
[56] C. Hu, Z. Huang, and Z. Lin, "An information transport dynamic load balancing policy based on software defined mobile networks," in *Proc. Int. Conf. Inf. Comput. Technol. (ICICT)*, Mar. 2018, pp. 83–86.
[57] W. Lan, F. Li, X. Liu, and Y. Qiu, "A dynamic load balancing mechanism for distributed controllers in software-defined networking," in *Proc. 10th Int. Conf. Measuring Technol. Mechatronics Autom. (ICMTMA)*, Feb. 2018, pp. 259–262.
[58] P. Tao, C. Ying, Z. Sun, S. Tan, P. Wang, and Z. Sun, "The controller placement of software-defined networks based on minimum delay and load balancing," in *Proc. IEEE 16th Intl Conf Dependable, Autonomic Secure Comput., 16th Intl Conf Pervas. Intell. Comput., 4th Intl Conf Big Data Intell. Comput. Cyber Sci. Technol. Congr.(DASC/PiCom/DataCom/CyberSciTech)*, Aug. 2018, pp. 310–313.
[59] Y. Gao and T. Luo, "A load balancing scheme for supporting safety applications in heterogeneous software defined LTE-V networks," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–5.
[60] V. Koryachko, D. Perepelkin, and V. Byshov, "Approach of dynamic load balancing in software defined networks with QoS," in *Proc. 6th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2017, pp. 1–5.
[61] A. S. AbdelRahman and A. B. El-Sisi, "Dynamic load balancing technique for software defined Wi-Fi networks," in *Proc. 12th Int. Conf. Comput. Eng. Syst. (ICCES)*, Dec. 2017, pp. 289–294.
[62] S. Attarha, K. H. Hosseiny, G. Mirjalily, and K. Mizanian, "A load balanced congestion aware routing mechanism for software defined networks," in *Proc. Iranian Conf. Electr. Eng. (ICEE)*, May 2017, pp. 2206–2210.
[63] A. Bremler-Barr, D. Hay, I. Moyal, and L. Schiff, "Load balancing memcached traffic using software defined networking," in *Proc. IFIP Netw. Conf. (IFIP Networking) Workshops*, Jun. 2017, pp. 1–9.
[64] Z. Chen, S. Manzoor, Y. Gao, and X. Hei, "Achieving load balancing in high-density software defined WiFi networks," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Dec. 2017, pp. 206–211.
[65] P.-J. Chuang and H.-J. Chen, "Efficient load balancing in software defined networks," in *Proc. Int. Conf. Inf., Commun. Eng. (ICICE)*, Nov. 2017, pp. 526–528.
[66] T. Han and N. Ansari, "A traffic load balancing framework for software-defined radio access networks powered by hybrid energy sources," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 1038–1051, Apr. 2016.
[67] W. Yong, T. Xiaoling, H. Qian, and K. Yuwen, "A dynamic load balancing method of cloud-center based on SDN," *China Commun.*, vol. 13, no. 2, pp. 130–137, Feb. 2016.
[68] F. S. Fizi and S. Askar, "A novel load balancing algorithm for software defined network based datacenters," in *Proc. Int. Conf. Broadband Commun. Next Gener. Netw. Multimedia Appl. (CoBCom)*, Sep. 2016, pp. 1–6.
[69] Y. Hu, T. Luo, W. Wang, and C. Deng, "On the load balanced controller placement problem in software defined networks," in *Proc. 2nd IEEE Int. Conf. Comput. Commun. (ICCC)*, Oct. 2016, pp. 2430–2434.
[70] R.-H. Hwang and H.-P. Tseng, "Load balancing and routing mechanism based on software defined network in data centers," in *Proc. Int. Comput. Symp. (ICS)*, Dec. 2016, pp. 165–170.
[71] N. T. Hai and D.-S. Kim, "Efficient load balancing for multi-controller in SDN-based mission-critical networks," in *Proc. IEEE 14th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2016, pp. 420–425.

[72] M. F. Ramdhani, S. N. Hertiana, and B. Dirgantara, ''Multipath routing with load balancing and admission control in software-defined networking (SDN),'' in *Proc. 4th Int. Conf. Inf. Commun. Technol. (ICoICT)*, May 2016, pp. 1–6.

[73] S. Kaur, K. Kumar, J. Singh, and N. S. Ghumman, ''Round-robin based load balancing in software defined networking,'' in *Proc. 2nd Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2015, pp. 2136–2139.

[74] S. Yilmaz, A. M. Tekalp, and B. D. Unluturk, ''Video streaming over software defined networks with server load balancing,'' in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2015, pp. 722–726.

[75] R. Tu, X. Wang, J. Zhao, Y. Yang, L. Shi, and T. Wolf, ''Design of a load-balancing middlebox based on SDN for data centers,'' in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2015, pp. 480–485.

[76] T. Hu, J. Lan, J. Zhang, and W. Zhao, ''EASM: Efficiency-aware switch migration for balancing controller loads in software-defined networking,'' *Peer-to-Peer Netw. Appl.*, vol. 12, no. 2, pp. 452–464, 2019.

[77] W. Wang, M. Dong, K. Ota, J. Wu, J. Li, and G. Li, ''CDLB: A cross-domain load balancing mechanism for software defined networks in cloud data centre,'' *Int. J. Comput. Sci. Eng.*, vol. 18, no. 1, pp. 44–53, 2019.

[78] M. M. Mulla, M. Raikar, M. Meghana, N. S. Shetti, and R. Madhu, ''Load balancing for software-defined networks,'' in *Emerging Research in Electronics, Computer Science and Technology*. Singapore: Springer, 2019, pp. 235–244.

[79] S. B. Vyakaranal and J. G. Naragund, ''Weighted round-robin load balancing algorithm for software-defined network,'' in *Emerging Research in Electronics, Computer Science and Technology*. Singapore: Springer, 2019, pp. 375–387.

[80] B. Kang and H. Choo, ''An SDN-enhanced load-balancing technique in the cloud system,'' *J. Supercomput.*, vol. 74, no. 11, pp. 5706–5729, Nov. 2018.

[81] F. Shang, L. Mao, and W. Gong, ''Service-aware adaptive link load balancing mechanism for software-defined networking,'' *Future Gener. Comput. Syst.*, vol. 81, pp. 452–464, Apr. 2018.

[82] Y.-D. Lin, C. C. Wang, Y.-J. Lu, Y.-C. Lai, and H.-C. Yang, ''Two-tier dynamic load balancing in SDN-enabled Wi-Fi networks,'' *Wireless Netw.*, vol. 24, no. 8, pp. 2811–2823, Nov. 2018.

[83] Y.-W. Ma, J.-L. Chen, Y.-H. Tsai, K.-H. Cheng, and W.-C. Hung, ''Load-balancing multiple controllers mechanism for software-defined networking,'' *Wireless Pers. Commun.*, vol. 94, no. 4, pp. 3549–3574, Jun. 2017.

[84] P. Song, Y. Liu, T. Liu, and D. Qian, ''Flow stealer: Lightweight load balancing by stealing flows in distributed SDN controllers,'' *Sci. China Inf. Sci.*, vol. 60, no. 3, 2017, Art. no. 032202.

[85] S.-N. Yang, C.-H. Ke, Y.-B. Lin, and C.-H. Gan, ''Mobility management through access network discovery and selection function for load balancing and power saving in software-defined networking environment,'' *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, p. 204, Dec. 2016.

[86] Y. Chen, W. Chen, Y. Hu, L. Zhang, and Y. Wei, ''Dynamic load balancing for software-defined data center networks,'' in *Proc. Int. Conf. Collaborative Comput., Netw., Appl. Worksharing*. Cham, Switzerland: Springer, 2016, pp. 286–301.

[87] K. Kaur, S. Kaur, and V. Gupta, ''Least time based weighted load balancing using software defined networking,'' in *Proc. Int. Conf. Adv. Comput. Data Sci.* Singapore: Springer, 2016, pp. 309–314.

[88] S. Kaur and J. Singh, ''Implementation of server load balancing in software defined networking,'' in *Information Systems Design and Intelligent Applications*. New Delhi, India: Springer, 2016, pp. 147–157.

[89] M. Priyadarsini, J. C. Mukherjee, P. Bera, S. Kumar, A. H. M. Jakaria, and M. A. Rahman, ''An adaptive load balancing scheme for software-defined network controllers,'' *Comput. Netw.*, vol. 164, Dec. 2019, Art. no. 106918.

[90] Y. Zhao, C. Liu, H. Wang, X. Fu, Q. Shao, and J. Zhang, ''Load balancing-based multi-controller coordinated deployment strategy in software defined optical networks,'' *Opt. Fiber Technol.*, vol. 46, pp. 198–204, Dec. 2018.

[91] H. Zhong, Y. Fang, and J. Cui, ''Reprint of 'LBBSRT: An efficient SDN load balancing scheme based on server response time,''' *Future Gener. Comput. Syst.*, vol. 80, pp. 409–416, Mar. 2018.

[92] A. K. Rangisetti, T. V. Pasca S., and B. R. Tamma, ''QoS aware load balance in software defined LTE networks,'' *Comput. Commun.*, vol. 97, pp. 52–71, Jan. 2017.

[93] L. Boero, M. Cello, C. Garibotto, M. Marchese, and M. Mongelli, ''BeaQoS: Load balancing and deadline management of queues in an OpenFlow SDN switch,'' *Comput. Netw.*, vol. 106, pp. 161–170, Sep. 2016.

[94] H. Gasmelseed and R. Ramar, ''Traffic pattern-based load-balancing algorithm in software-defined network using distributed controllers,'' *Int. J. Commun. Syst.*, vol. 32, no. 17, p. e3841, Nov. 2019.

[95] V. D. Chakravarthy and B. Amutha, ''A novel software-defined networking approach for load balancing in data center networks,'' *Int. J. Commun. Syst.*, p. e4213, Nov. 2019.

[96] K. S. Sahoo and B. Sahoo, ''CAMD: A switch migration based load balancing framework for software defined networks,'' *IET Netw.*, vol. 8, no. 4, pp. 264–271, Jul. 2019.

[97] S. M. Raza, D. Park, Y. Park, K. Lee, and H. Choo, ''Dynamic load balancing of local mobility anchors in software defined networking based proxy mobile IPv6,'' in *Proc. 10th Int. Conf. Ubiquitous Inf. Manage. Commun.*, 2016, pp. 1–4.

[98] H. Yao, C. Qiu, C. Zhao, and L. Shi, ''A multicontroller load balancing approach in software-defined wireless networks,'' *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 10, p. 454159, 2015.

[99] S. Jiugen, Z. Wei, J. Kunying, and X. Ying, ''Multi-controller deployment algorithm based on load balance in software defined network,'' *J. Electron. Inf. Technol.*, vol. 40, no. 2, pp. 455–461, 2018.

[100] Q. Wang, L. Gao, Y. Yang, J. Zhao, T. Dou, and H. Fang, ''A load-balanced algorithm for multi-controller placement in software-defined network,'' *Mech. Syst. Control Formerly Control Intell. Syst.*, vol. 46, no. 2, pp. 72–81, 2018.

[101] F. Al-Tam and N. Correia, ''On load balancing via switch migration in software-defined networking,'' *IEEE Access*, vol. 7, pp. 95998–96010, 2019.

[102] S. Zhang, J. Lan, P. Sun, and Y. Jiang, ''Online load balancing for distributed control plane in software-defined data center network,'' *IEEE Access*, vol. 6, pp. 18184–18191, 2018.

[103] A. Pietrabissa, L. R. Celsi, F. Cimorelli, V. Suraci, F. D. Priscoli, A. Di Giorgio, A. Giuseppi, and S. Monaco, ''Lyapunov-based design of a distributed wardrop load-balancing algorithm with application to software-defined networking,'' *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 5, pp. 1924–1936, Sep. 2019.

[104] M. Farhoudi, P. Habibi, and M. Sabaei, ''Server load balancing in software-defined networks,'' in *Proc. 9th Int. Symp. Telecommun. (IST)*, Dec. 2018, pp. 435–441.

[105] C. Wang, B. Hu, S. Chen, D. Li, and B. Liu, ''A switch migration-based decision-making scheme for balancing load in SDN,'' *IEEE Access*, vol. 5, pp. 4537–4544, 2017.

[106] P. Wang, H. Xu, L. Huang, J. He, and Z. Meng, ''Control link load balancing and low delay route deployment for software defined networks,'' *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2446–2456, Nov. 2017.

[107] U. Mahlab, P. E. Omiyi, H. Hundert, Y. Wolbrum, O. Elimelech, I. Aharon, K. S. Z. Erlich, and S. Zarakovsky, ''Entropy-based load-balancing for software-defined elastic optical networks,'' in *Proc. 19th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2017, pp. 1–4.

[108] C. Yu, Z. Zhao, Y. Zhou, and H. Zhang, ''Intelligent optimizing scheme for load balancing in software defined networks,'' in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, Jun. 2017, pp. 1–5.

[109] F. Cimorelli, F. D. Priscoli, A. Pietrabissa, L. R. Celsi, V. Suraci, and L. Zuccaro, ''A distributed load balancing algorithm for the control plane in software defined networking,'' in *Proc. 24th Medit. Conf. Control Autom. (MED)*, Jun. 2016, pp. 1033–1040.

[110] X. He, Z. Ren, C. Shi, and J. Fang, ''A novel load balancing strategy of software-defined cloud/fog networking in the Internet of vehicles,'' *China Commun.*, vol. 13, pp. 140–149, Nov. 2016.

[111] W.-H. Liao, S.-C. Kuai, and C.-H. Lu, ''Dynamic load-balancing mechanism for software-defined networking,'' in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Jul. 2016, pp. 336–341.

[112] S. Sathyanarayana and M. Moh, ''Joint route-server load balancing in software defined networks using ant colony optimization,'' in *Proc. Int. Conf. High Perform. Comput. Simulation (HPCS)*, Jul. 2016, pp. 156–163.

[113] C.-T. Yang, S.-T. Chen, J.-C. Liu, Y.-W. Su, D. Puthal, and R. Ranjan, ''A predictive load balancing technique for software defined networked cloud services,'' *Comput.*, vol. 101, no. 3, pp. 211–235, 2019.

[114] S. Manzoor, X. Hei, and W. Cheng, ''A multi-controller load balancing strategy for software defined WiFi networks,'' in *Proc. Int. Conf. Cloud Comput. Secur.* Cham, Switzerland: Springer, 2018, pp. 622–633.

[115] G. Li, T. Gao, Z. Zhang, and Y. Chen, ''Fuzzy logic load-balancing strategy based on software-defined networking,'' in *Proc. Int. Wireless Internet Conf.* Cham, Switzerland: Springer, 2017, pp. 471–482.

[116] G. Zhao, L. Huang, Z. Li, and H. Xu, "Load-balancing software-defined networking through hybrid routing," in *Proc. Int. Conf. Wireless Algorithms, Syst., Appl.* Springer, 2017, pp. 96–108.
[117] X. Duan, A. M. Akhtar, and X. Wang, "Software-defined networking-based resource management: Data offloading with load balancing in 5G HetNet," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, no. 1, p. 181, Dec. 2015.
[118] H. Wang, H. Xu, L. Huang, J. Wang, and X. Yang, "Load-balancing routing in software defined networks with multiple controllers," *Comput. Netw.*, vol. 141, pp. 82–91, Aug. 2018.
[119] H. Xu, X.-Y. Li, L. Huang, Y. Du, and Z. Liu, "Partial flow statistics collection for load-balanced routing in software defined networks," *Comput. Netw.*, vol. 122, pp. 43–55, Jul. 2017.
[120] K. S. Sahoo, M. Tiwary, B. Sahoo, R. Dash, and K. Naik, "DSSDN: Demand-supply based load balancing in software-defined wide-area networks," *Int. J. Netw. Manage.*, vol. 28, no. 4, p. e2022, Jul. 2018.
[121] S. Manzoor, X. Hei, and W. Cheng, "Towards dynamic two-tier load balancing for software defined WiFi networks," in *Proc. 2nd Int. Conf. Commun. Inf. Syst. (ICCIS)*, 2017, pp. 63–67.
[122] H. Xue, K. Kim, and H. Youn, "Dynamic load balancing of software-defined networking based on genetic-ant colony optimization," *Sensors*, vol. 19, no. 2, p. 311, Jan. 2019.
[123] Y. Bao, Y. Li, L. Ma, and W. Chen, "Synchronous automatic training for wearable sensors via knowledge distillation: Poster abstract," in *Proc. 18th Int. Conf. Inf. Process. Sensor Netw.*, Montreal, QC, Canada, Apr. 2019, pp. 303–304.
[124] S. A. M. Rofie, I. Ramli, K. N. Redzwan, S. M. M. Hassan, and M. S. B. Ibrahim, "OpenFlow based load balancing for software-defined network applications," *Adv. Sci. Lett.*, vol. 24, no. 2, pp. 1210–1213, Feb. 2018.
[125] O. Lemeshko and O. Yeremenko, "Enhanced method of fast re-routing with load balancing in software-defined networks," *J. Electr. Eng.*, vol. 68, no. 6, pp. 444–454, Nov. 2017.
[126] Y. Wang, Y. Zhang, and J. Chen, "SDNPS: A load-balanced topic-based Publish/Subscribe system in software-defined networking," *Appl. Sci.*, vol. 6, no. 4, p. 91, Mar. 2016.
[127] K. S. Sahoo, D. Puthal, M. Tiwary, M. Usman, B. Sahoo, Z. Wen, B. P. S. Sahoo, and R. Ranjan, "ESMLB: Efficient switch migration-based load balancing for multi-controller SDN in IoT," *IEEE Internet Things J.*, to be published.
[128] H. Zhong, Y. Fang, and J. Cui, "LBBSRT: An efficient SDN load balancing scheme based on server response time," *Future Gener. Comput. Syst.*, vol. 68, pp. 183–190, Mar. 2017.
[129] A. Blenk, A. Basta, and W. Kellerer, "HyperFlex: An SDN virtualization architecture with flexible hypervisor function allocation," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 397–405.
[130] T. Li, H. Zhou, H. Luo, Q. Xu, and Y. Ye, "Using SDN and NFV to implement satellite communication networks," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Jul. 2016, pp. 131–134.
[131] H. Zhang and J. Yan, "Performance of SDN routing in comparison with legacy routing protocols," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery*, Sep. 2015, pp. 491–494.
[132] G. S. Aujla, N. Kumar, A. Y. Zomaya, and R. Ranjan, "Optimal decision making for big data processing at edge-cloud environment: An SDN perspective," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 778–789, Feb. 2018.
[133] S. Q. Zhang, P. Yasrebi, A. Tizghadam, H. Bannazadeh, and A. Leon-Garcia, "Fast network flow resumption for live virtual machine migration on SDN," in *Proc. IEEE 23rd Int. Conf. Netw. Protocols (ICNP)*, Nov. 2015, pp. 446–452.
[134] T. Zhang and B. Liu, "Exposing end-to-end delay in software-defined networking," *Int. J. Reconfigurable Comput.*, vol. 2019, pp. 1–12, Mar. 2019.
[135] J. Son, A. V. Dastjerdi, R. N. Calheiros, and R. Buyya, "SLA-aware and energy-efficient dynamic overbooking in SDN-based cloud data centers," *IEEE Trans. Sustain. Comput.*, vol. 2, no. 2, pp. 76–89, Apr. 2017.
[136] A. Fernandez-Fernandez, C. Cervello-Pastor, and L. Ochoa-Aday, "Achieving energy efficiency: An energy-aware approach in SDN," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–7.
[137] G. Shang, P. Zhe, X. Bin, H. Aiqun, and R. Kui, "FloodDefender: Protecting data and control plane resources under SDN-aimed DoS attacks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2017, pp. 1–9.

MOHAMMAD RIYAZ BELGAUM (Graduate Student Member, IEEE) received the master's degree with a specialization in computer applications, in 2001, and the M.Eng. degree with a specialization in computer science and engineering, in 2006. He is currently pursuing the Ph.D. degree in information technology with Universiti Kuala Lumpur with a focus on software defined networks.

He is an Oracle Database 11G Administrator Certified Professional and an Oracle Database 11G Program with PL/SQL Certified Associate. He has vast teaching experience across various countries with much focus on research. He has authored more than 30 research papers published in journals and peer-reviewed conferences. His areas of interest include cloud computing, mobile ad hoc networking, wireless sensor networks, the Internet of Things, and bytecode analysis.

SHAHRULNIZA MUSA received the master's degree in integrated research study and the Ph.D. degree in communication network security from the Faculty of Electrical and Electronic Engineering, Loughborough University, U.K., in 2005 and 2008, respectively. He is currently the Deputy President in charge of the academic and technology with Universiti Kuala Lumpur (UniKL). He started his career at UniKL since its establishment, in 2002. Apart from teaching and post-graduate supervision, he is also active in software project consultation and development in business application, enterprise resource planning (ERP), and customer relation management (CRM). His research interests are in cybersecurity, the IoT application, the IoT security, bigdata analytics, and SDN.

MUHAMMAD MANSOOR ALAM received the M.E. degree in systems engineering, the M.Sc. degree in computer science, the Ph.D. degree in computer engineering, and the Ph.D. degree in electrical and electronic engineering.

He is an active Researcher in the field of telecommunication and network. He has authored more than 60 research articles published in ISI indexed journals, as book chapters, and in peer-reviewed conferences. He is also an author of the book *Study Guide of Network Security* copyrighted by Open University Malaysia and The Open University of Hong Kong. He is also an active Reviewer for the ISI indexed journal *Pertanika Journal of Science and Technology* (JST)).

MAZLIHAM MOHD SU'UD (Member, IEEE) received the Ph.D. degree in computer engineering from the Université de La Rochelle, in 2007, and the master's degree in electrical and electronics engineering from the University of Montpellier, in 1993. Since 2013, he has been the President/CEO of Universiti Kuala Lumpur, Malaysia.

• • •