Computer Science, Engineering and Technology
Vol: 2(4), December 2024
REST Publisher; ISSN: 2583-9179 (Online)
Website: https://restpublisher.com/journals/cset/
DOI: https://doi.org/10.46632/cset/2/4/1

An Extensive Analysis of Current Software Defined Network Optimization Techniques

Neha Bharani
SOC IPS Academy, Indore, Madhya Pradesh, India.
**Corresponding author: neha4bharani@gmail.com*

Abstract: *In the modern, technologically advanced era, the use of devices connected to the Internet has increased exponentially. This has led to a significant increase in Internet activity. Network scalability, low levels of service, and the need to manage a large number of clients on a single server (which can lead to denial of service (DOS) attacks) are just a few of the consequences of increased traffic if you are a load balancer, . A method proposed in the literature. Although load balancers are popular, studies have shown that they have a few drawbacks, such as not being programmable and exclusive to a single vendor to overcome these issues and the subsequent increase in Internet traffic saw the emergence of Software Defined Networking (SDN) as a paradigm shift. Software defined networking (SDN) enables programmable load balancers and gives customers the freedom to develop and implement their own load-balancing algorithms this exam covers the origins of software-defined networking (SDN) and Open Flow, with impact on weight and density so included. In this article, we will go through various SDN load balancing techniques. This review builds on existing research challenges, their answers, and possible future research directions. Load balancing in intelligent software-defined (SDN) algorithms is typically performed using mathematical tools. Finally, we detail the metrics used to evaluate the performance of the algorithms.*

Keyword: *Open Flow, Dynamic Load Balancing Problems, Software Defined Network*

1. INTRODUCTION

Over the past decade, various innovations have attracted much attention. These include cloud computing, social media platforms, the Internet of Things (IoT), and sophisticated mobile applications. The quality of service (QoS) requirements of these technologies cause performance, scalability and flexibility problems caused by old backbone networks All of these problems are long gone and are vendor specific. Flexible network management is not possible with traditional vendor-specific data plane control plane systems. Currently, no airline allows vendors, at the customer's request, to change the policy to move those items. When flights of either gender specific are used, human supervision is required. Websites and Internet services are hosted by cloud platforms.[1]

Performance, dynamic management, and QoS were critical to the delivery of these services. To overcome the shortcomings of conventional networks, researchers have suggested two methods: software defined networking (SDN) [2] and network functions virtualization (NFV) [3]. The design and infrastructure of your network can be virtualized in software and used on various devices with SDN and NFV. Both ideas use software abstraction to circumvent the issues of conventional networks. Furthermore, SDN increases the network's programmability by allowing programs to define and modify its architecture and functionality [4].

Physical network devices are separated by software-defined networking (SDN), which shifts all decisions to the control plane (see Figure 2). All network intelligence, packet forwarding, and network management policies and procedures are embedded in the control plane Scalability and robust administration of the network is made possible by SDN design Centralized network control is enabled by an SDN controller. Simplifying network configuration makes network management vendor-neutral and allows for less complex network devices.

A potentially revolutionary approach to network technologies such as cloud computing and big data applications is software-defined networking (SDN) [6]. SDN has proven to be far superior to traditional networks. The evolving network architecture known as software-defined networking (SDN) has the potential to rethink the limitations of existing architecture. Compared to a real forwarding plane, data handling planes and packet forwarding plane solutions that use this rate are among the best. The system can be dynamically controlled using any programming language that supports SDN. Due to its versatility, it is essential for searching client requests, which in turn enables management of big data cloud computing platforms [7].SDN-based networks while using conventional big data analytics techniques in cloud computing and software-defined networking (SDN).

The process of efficiently distributing traffic in network devices, or load balancing, is an important aspect of network design and management [8 Load balancing is considered effective when it can reduce energy consumption time improving fault tolerance, continuity, delay, and resource management. In traditional networks, specialized servers handle most of the load balancing operations (see Figure 1). Dynamic load balancing server is installed to meet the constantly changing requirements of the network [9]. The focus of our research is on several SDN load balancing strategies.

Load balancing is a method of dividing the workload among multiple components so that no one is overburdened [10]. The ultimate goal is traffic optimization, resource efficiency, and response time reduction. In network management, load balancing has historically been done by allocating network traffic to specific pieces of hardware. While this approach is expensive and cannot dynamically change the system based on real-time data, it generally works well. Software-defined networking (SDN) is a great way to balance load in the cloud because it adapts to the needs of the business. SDN controls every device in a cloud computing environment. [11]

Because of its complexity, SDN can be difficult to get right. The authors of the paper emphasize that further research is needed in this area, although load balancing in SDN can improve SDN performance This is the first proper load balancing (LB) study to examine how Various AI techniques impact SDN performance, if you like in SDN. In the last decade, several scholars have proposed load balancing methods using mathematical models. Each of these approaches has different advantages and disadvantages depending on the project and network environment. By examining the load balancing scheme in data centers, this study seeks to provide a comprehensive analysis by how it is used

What follows is the structure of the rest of the paper: In Section I, the effects of software-defined networks on load balancing in the cloud are discussed. In section II, we discussed the characteristics of the load balancing for SDN architecture and respective layers. Section III provides an in-depth evaluation of the load balancing system for a selection of selected articles. The findings are presented in Section IV. Section V concludes the study..

2. BACKGROUND

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

a) SDN Architecture : Because of its complexity, SDN can be difficult to get right. The authors of the paper emphasize that further research is needed in this area, although load balancing in SDN can improve SDN performance if you are interested in software-defined networking (SDN) and this is the case the first on this topic. SDN focuses on a way to define routing decisions without the need for proprietary hardware using common configurable software. This is done by physically separating two separate layers, the data plane and the control plane. For a long time, switches and routers were the only physical components that could be used to create network architectures, the data plane and the control plane remained unified Network administrators had little control over packet forwarding and for this reason, it is possible for a network to have multiple control planes for each device, even if they are incompatible with each other [12]. A single high-capacity server can establish control plane rules when using a software-defined network. This server controls the packet rules for each network device. Allowing access from a particular IP address or blocking traffic from a particular port are just a few of the other possible restrictions. Networks are assembled using SDN architecture in three separate but interconnected layers, as shown in Figure 1. Each layer has a specific function, and communication between them is infrequent.

The application layer is ideally its base layer. Applications can interact with the layer below and show how web flows behave thanks to the software. The load balancing described in the article will apply at this level. The SDN controller, also known as the control plane, is located in the middle layer. Its ability to communicate in both directions is one of its unique characteristics; This allows you to abstract routing devices and modify their flow at a higher level.

A number of free SDN controllers are available. Several controller names are shown in Figure 1. Below that lies the layer of network devices, the data plane. Receive, process, and update data packages as directed by the controller. Internal statistics are used to perform general statistics. Besides providing a direct indication of behaviour and network requirements, the interaction between the application layer and the controller (also known as the Northbound API) often gives an unrealistic view of the network through a protocol like Open Flow connecting to lower levels.

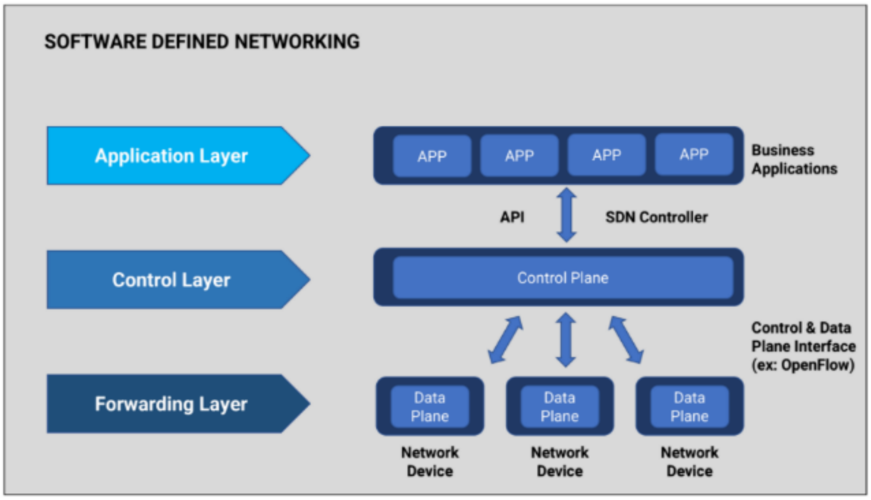

FIGURE 1. The Traditional SDN Architecture

b) Load Balancing Algorithm in SDN: Typically, the performance of a distributed system is improved by LB technology, which efficiently distributes workloads, client inputs, and requests among available network resources this approach can reduce overall usage, latency, and network overhead . . . . There are two ways to use it: hardware and software. The combination of virtual resource and software-defined network (SDN) in IoMT can improve load distribution and energy efficiency [15]. Load balancing algorithms can be implemented in a variety of ways, including static, dynamic, and hybrid [16]. Static methods rely heavily on the initial data of the system. Unpredictable user behaviour and static load balancing rules may prevent some networks from using static load balancing (STL) techniques. However, dynamic strategies are more effective in load balancing than static ones because they use a pre-designed load balancing strategy [17] A well-designed load balancing strategy can reduce overloading so up, improve productivity, and reduce response time and packet loss.

To achieve dependable results, it is crucial to incorporate several features, also referred to as qualitative metrics, while doing comparisons. These include latency, energy usage, packet delivery ratio, scalability, and more [19].

One of the main problems with distributed computing is load balancing. The load may be due to a network latency, a memory constraint, or a CPU limit. Enhancing the structure of the distributed system is a continuous activity that necessitates distributing the work across its different nodes. As a result, it can prevent situations in which the hubs of the network are either overloaded or under loaded. Load balancing is a strategy that makes sure the burden is distributed evenly so that the running assignment may be finished without putting too much strain on the processor pool or frame hub.

There are essentially two types of load balancing algorithms discussed here:
1. Static Algorithm
2. Dynamic Algorithm

1. Static Algorithm: Static load balancing strategies include an even distribution of resources during client-server communication. An entirely manual control mechanism is present in this type of program. Assigning initial tasks to individual processors is symbolized by the master CPU. There is never a change in which central processing unit (CPU) is responsible for what. Consequently, the work is started off strong by the central processor. The CPUs in the slave process decide what to do next and report back to the master with their results. No matter how many CPUs are given to a task, each one will always finish. Static load balancing methods reduce overall performance time for critical programs while minimizing the possibility of processor communication delays. This is not an attempt to prevent anything. [20]

The following traits of static load balancing techniques:
1. Cut down on contact delays because less communication means a longer execution time.
2. Improved processing and response times are provided by weighted algorithms.
3. Distribution load balancing techniques based on the load at the time the node is chosen prior to execution starting.
4. Static approaches yield more consistent and dependable outcomes when they are applied uniformly and in similar and consistent circumstances.
5. It is not too difficult to predict how static methods will behave.
6. It is difficult to predict future loads due to the arrival and processing conditions of the loads.

2. Dynamic Algorithm: For optimal load balancing over the whole network or architecture, the dynamic algorithm actively seeks out and prioritizes the lightest server. It can't improve internal framework mobility without this continual contact with the network. In this case, the current framework status is used to decide the load management options. To achieve better load balancing in distributed computing, it is possible to enhance some subjective measures. [21]

The dynamic distribution of pre-programmed load balancer configurations is the reason why dynamic load balancing solutions outperform their static counterparts. Proper load balancing is crucial to keep resources from being overloaded while also optimizing response time, throughput, consumption, and scalability. You can implement Dynamic Load Balancing in one of three ways: non-distributed, distributed, or semi-distributed.

3. EXAMINE OF LOAD DISTRIBUTION TECHNIQUES

Adaptive load balancers utilize intelligence to determine which node in a network farm can process a data packet arriving at its destination. This can only be accomplished with algorithms that were specifically built to handle load dispersion. Algorithms vary greatly according to whether a load is distributed on the application layer or the network layer. Algorithm selection, which impacts performance and business continuity, is influenced by the efficiency of load distribution systems.

Finding the best load balancing methods is essential for optimizing the performance of any network, including SDN. The goal of intelligent load balancing across several servers, networks, clouds, and controllers has been the subject of several published approaches for optimizing network traffic.

In certain cases, improving the control plane's functionality requires decentralizing the SDN controller. Due to decentralization, the control plane usually has more than one controller. We discuss load balancing in the following scenario where the control plane is comprised of many SDN controllers.

In [22], the reliable and load-balance-aware multi-controller deployment (RLMD) system is published. When the control and data planes separate in multi-controller SDN networks, the authors claim that RLMD's principal objective is to fix the issues of controller traffic load disparity and control plane accuracy. In order to guarantee reliable and efficient controller installations and equitable distribution of controller traffic loads, RLMD employs a progressive approach. The authors also came up with a method called multiple domain partitioning (MDP) that uses the load balancing rate of the controller and the node attract ability to match switches and controllers. When compared to other conventional solutions, their technique greatly enhanced the reliability of the control plane and balanced the allocation of controller traffic.

To tackle the well-documented problem of load balancing in the control plane, the authors of [23] focused on a distributed controllers architecture for the software-defined networking (SDN) control plane that can dynamically balance its traffic load. Researchers were able to fill a knowledge gap about the placement and matching of controllers with switches by resolving the challenge of managing many controllers. By efficiently dividing the switch requests among multiple controllers, these procedures avoid congestion.

ElastiCon is proposed in [24] as a strategy based on switch migration. ElastiCon ensures that the network always runs at peak performance, regardless of traffic patterns, by periodically distributing the workload among a pool of controllers. The potential for ElastiCon to cause unexpectedly high network overhead is one of its downsides. In [25], we can see a few of related studies that demonstrate how the SDN controller may be made more efficient. Finding the optimal migration switch to provide load balancing was the primary focus of these analyses. This method keeps the network's performance from dropping by lowering the load on overworked controllers.

A load balancing approach based on switch groups of multiple controller architecture was devised by the authors of [26] to enhance the efficiency of the load balancing process in terms of time. The proposed solution uses a switch selection algorithm and a target controllers selection algorithm to fix the problem of load oscillation among controllers.

A survey on software-defined networks was conducted by the authors in [27]. In their review essay, they covered the history, fundamental concept, and problem areas of SDN in relation to traditional networks. The writers delve deeper on eight fundamental SDN concerns. Research into the future of cloud computing, the NaaS paradigm, the transition to SDN networks, and countless more ideas is still in its infancy. The benefits of software-defined networking (SDN) and its performance-enhancing features were highlighted in [28], who reviewed many studies pertaining to SDN. The authors survey the literature on recent SDN studies, outlining the pros and cons of these studies. Their research indicates that SDN switch devices may be made cheaper, that resource utilization is improved, and that the network is more scalable. Nevertheless, neither the commercial development of SDN nor the implementation's design defects were examined by the researchers.

While discussing the effects on the network, the authors in [29] focused on the problems caused by load imbalance. Various load balancing solutions were described and organized in the article. The pros and cons of this proposal were considered. There was a gap in their study regarding the performance optimization method and the distributed SDN architecture scenario. For network monitors, the method proposed in [30] works. They proved that over-monitoring a network slows it down. If you want to pick switches and network traffic efficiently, the authors recommend using the network monitor approach.

For software-defined networking, [31] analyzed 19 publications in detail. In their review study, the authors classified load balancing systems as either deterministic or non-deterministic. Regarding various load distribution benchmarks of their methods, the complexities and limitations of these methodologies were also investigated. In addition, we have checked the algorithms' validity. Adding a failure mechanism unique to the SDN environment could have broadened the scope of the study.

The researchers in [32] did a thorough investigation into SDN load balancing strategies. Load balancing strategies can be grouped into three distinct types. They can exhibit either dynamic or static behaviours. This algorithm must meticulously examine the load's real-time changes. Distributed and dynamic decision-making is the main premise of load balancing algorithms. Throughput maximization with latency and reaction time minimization is the main objective of the distributed and dynamic load balancing approach. Type of traffic, priority, and network slicing are all areas where research is lacking.

Scalable software management of the data center's network is achieved through the use of a defined network concept. The multi-controller approach was used for large networks in [33]. They have thoroughly examined the pros and cons of utilizing multiple controllers in their research. Heuristics, game theory, and linear programming were among the methods discussed by the researcher when it came to multi-controller load balancing. Concerns about dependability, safety, and consistency go unmentioned in the study. According to a survey in [34], the Controller Placement Problem (CPP) was investigated. The authors examined CPP modelling, CPP formulation, and related metrics in their survey. An optimal and heuristic-based approach to address CPP. Levels of CPP in large networks and NFV conduction are areas that could use more investigation in the future. As far as SDN load balancing surveys are concerned, there is no methodical procedure. Systematic reviews concerning multi-controllers and dynamic controllers are not conducted by another researcher..

4. RESULTS

In this section, we will also discuss some unanswered questions and remarks regarding intelligent load balancing techniques for SDN sampling. Some metrics, such delay and quality of service, are essential to load balancing and are characteristics of successful load balancing schemes. Not surprisingly, all or nearly all of the researches have employed these two criteria. The two measures that researchers utilize the least in the chosen journals are fairness and interference minimization. Most of the time, a technique is created to intelligently divide network traffic based on a preset statistic. Furthermore, if a system is able to balance traffic flow, it does not need to ensure fairness or avoid interference. We are concerned about the technique's effectiveness.

Remember that the metrics used depend on the goal and the methods used to measure the effectiveness of the solution. Most of the researchers in the sample test the proposed methods using Minette as a model. The argument is that when combined with Open Flow and SDN, Mininet becomes an effective network design, test and specification tool. 20% of the time they use Iperf to build network traffic, 10% they use OMNET++, and 5% they use other tools like Matlab, Python,.Net, and MaxiNet. Mininet was combined with other tools such as Iperf and MaxiNet to create the necessary SDN network architecture. Furthermore, [35] combined OMNET++ with Python. Because encouragement could only be learned through mistakes, the authors combined deep learning with reinforcement learning strategies. SDN shows how it gives the network administrator the freedom to create network management strategies that are not restricted by device or equipment manufacturers, by allowing the use of multiple tools.

The following are some of the limitations of the various approaches:
a) Most of them are not for use in wireless networks;
b) They have not taken failure mechanisms into account;
c) No one has applied machine learning (ML) approach; d) more time-consuming algorithms;
e) Most of them are not implemented in real large-scale networks; and
f) No one has presented security mechanism.
g) Matlab and the emulation tool Mini Net are used to assess the majority of methods.

5. OPEN ISSUES

a) The above analysis shows that there are many unresolved issues with SDN-based data center load balancing processes. Here we have discussed several incomplete research projects. A brief review of research on enhancing the stability, scalability, and reliability of software-defined network controllers. These can have a significant impact on the efficiency of the data centre network. Several issues related to load balancing strategies in data centres require further investigation. Each QoS parameter is not evaluated in a single way.

The full review reveals the following research gaps. It is also important to use a number of dynamic controllers to evaluate the design. A few open questions are as follows. There hasn't been any investigation into the proper management of resource utilisation for improved performance.

b) Low scalability, low dependability, and flexibility concerns with the control plan affect SDN-based load balancing.

c) The data center does not take into account dynamic traffic scheduling or congestion control.

d) When making load balancing judgments, there isn't a separate system to take into account multiple QoS characteristics as throughput, scalability, flexibility, cost, complexity, and bandwidth.

6. CONCLUSION

According to comprehensive research on load balancing techniques, SDN improves data centre network performance. This can find an overview and analysis of SDN Data Centre load balancing here. The advantages and disadvantages of weighting methods used by different researchers are discussed in our review. We also examine other operating parameters to evaluate the way in which load balance is affected. Journals published from 2015 to 2019 were counted for this study. After ACM, IEEE publishes the majority of articles (approximately 51.72 percent). Most researchers used mini nets as simulators to investigate different mechanisms. Improved performance, reduced latency, better utilization of network resources, higher flexibility, and many other benefits are all delivered by learning methods.

REFERENCES

[1]. Laizhong Cui, F. Richard Yu, and Qiao Yan. “When big data meets Software-Defined networking: SDN for big data and big data for SDN” IEEE Network 30, 1 (January 2016), 58– 65.DOI: http://dx.doi.org/10.1109/MNET.2016.7389832.

[2]. Xuekai Du, Zhihui Lv, Jie Wu, Chengrong Wu, and Shi Chen.” PDSDN: A policy-driven SDN controller improving the scheme for multi-tenant cloud data centre environments” In Proceedings of the 2016 IEEE International Conference on ServicesComputin (SCC’16).387–394.DOI: http://dx.doi.org/10.1109/SCC.2016.57.

[3]. Yuhua Xu, Zhe Sun, Zhixin Sun “SDN-based Architecture for Big Data Network” International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC).

[4]. Jose Ordonez-Lucena, Pablo Ameigeiras, Diego Lopez “Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges”, IEEE Communications Magazine (Volume: 55, Issue: 5, May 2017).

[5]. Qingxia Chen, Renchao Xie, Tao Huang, Jiang Liu, Yunjie Liu “Software Defined Networking, Caching and Computing Resource Allocation with Imperfect NSI”, IEEE Global Communications Conference GLOBECOM 2017 - 2017

[6]. Diego Kreutz, Fernando M. V. Ramos, Paulo Esteves Veríssimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig, “Software-Defined Networking: A Comprehensive Survey” 0018-9219, IEEE Vol. 103, No. 1, January 2015

[7]. Wenfeng Xia, Yonggang Wen, Chuan Heng Foh, Dusit Niyato, and Haiyong Xie “A Survey on Software-Defined Networking” 1553-877X, 2014 IEEE VOL. 17, NO. 1, first quarter 2015

[8]. Mao Qilin, Shen WeiKang “A Load Balancing Method Based on SDN” 978-1-4673-7143-8/15, 2015 IEEE, Seventh International Conference on Measuring Technology and Mechatronics Automation

[9]. Renuga Kanagavelu, Khin Mi Mi Aung “Software-Defined Load Balancer in Cloud Data Centers” ICCIP '16, November 26-29, 2016, Singapore, Singapore © 2016 ACM. ISBN 978-1-4503-4819-5/16/11

[10]. Lin Li, Qiaozhi Xu “Load Balancing Researches in SDN: A Survey” 978-1-5090-3025-5/17 ©2017 IEEE

[11]. Byungseok Kang, Hyunseung Choo “An SDN-enhanced load-balancing technique in the cloud system “DOI 10.1007/s11227-016-1936-z © Springer Business Media New York 2016

[12]. Ammar Muthanna, Abdelhamid A. Ateya, Maria Makolkina, Anastasia Vybornova, Ekaterina Markova, Alexander Gogol and Andrey Koucheryavy “SDN Multi-controller Networks with Load Balanced” ACM Woodstock conference, June 26-27, 2018, Amman, Jordan\isbn{978-1-4503-48447/17/07}

[13]. Shan-Hsiang Shen “An Efficient Network Monitor for SDN Networks” Sigmetrics’18, Irvine, CA, USA © 2018 ACM, Vol. 46, No. 2, September 2018

[14]. Ali Akbar Neghabi, Nima Jafari Navimipour, Mehdi Hosseinzadeh*, Ali Rezaee “Load balancing mechanisms in the software-defined networks: a systematic and comprehensive review of the literature” 2169-3536 © 2018 IEEE. DOI 10.1109/ACCESS.2018.2805842, IEEE Access

[15]. Mustafa Hasan Al Bowarab, Nurul Azuma Zakaria, Z. Zainal Abidin “Load Balancing Algorithms in Software Defined Network “International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-7, Issue-6S5, April 2019

[16]. Mohamed I. Hamed, Basem M. ElHalawany, Mostafa M. Fouda and Adly S. Tag Elden “A Novel Approach for Resource Utilization and Management in SDN” 978-1-5386-4266-5/17 © 2017 IEEE

[17]. Abderrahime Filali, Abdellatif Kobbane, Mouna Elmachkour, Soumaya Cherkaoui “SDN Controller Assignment and Load Balancing with Minimum Quota of Processing Capacity” 978-1-5386-3180-5/18 ©2018 IEEE

[18]. Farah Chahlaoui, Mohammed RAISS, Hamza DAHMOUNI “Performance Analysis of Load Balancing Mechanisms in SDN Networks” NISS’19 March 27-29, 2019, Rabat, Morocco © 2019 ACM. ISBN 978-1-4503-6645-8/19/03

[19]. Tao Wang, Fangming Liu, Senior Member, IEEE, and Hong Xu, Member, IEEE, ACM "An Efficient Online Algorithm for Dynamic SDN Controller Assignment in Data Center Networks" IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 25, NO. 5, OCTOBER 2017
[20]. Tao Hu, Zehua Guo, Thar Baker, Julong Lan "Multi-controller Based Software-Defined Networking: A Survey" DOI 10.1109/ACCESS.2018.2814738,2169-3536 (c) 2018 IEEE
[21]. Wenjing Lan, Fangmin Li, Xinhua Liu, Yiwen Qiu "A Dynamic Load Balancing Mechanism for Distributed Controllers in Software Defined Networking "2157-1481/18/ ©2018 IEEE DOI 10.1109/ICMTMA.2018.00069
[22]. Gokay Saldamli, Harsha Sanjeeva, Karthik Siddalingaapa, Ruthra Vadivel Murugesan, Levent Ertaul" A Secure Collaborative Module on Distributed SDN" 2019 10th International Conference on Information and Communication Systems (ICICS) 978-1-7281-0045-6/19/ ©2019 IEEE
[23]. Kshira Sagar Sahoo, Bibhudatta Sahoo" CAMD: a switch migration-based load balancing framework for software-defined networks" ISSN 2047-4954 IET Netw., 2019, Vol. 8 Iss. 4, pp. 264-271.
[24]. Tamal Das, Vignesh Sridharan and Mohan Gurusamy "A Survey on Controller Placement in SDN" DOI 10.1109/COMST.2019.2935453, IEEE 1553-877X © 2019 IEEE
[25]. Sudhir Kumar M, P Chakravarthi, B Jagadeesh, M sai Prakash "Load balancing in cloud computing" International Journal of Engineering & Technology, 7 (1.1) (2018) 306-310
[26]. Tu-Liang Lin 1, Chen-Hao Kuo 1, Hong-Yi Chang 1, *, Wan-Kun Chang 2, Yi-Ying Lin 2" A Parameterized Wildcard Method based on SDN for Server Load Balancing" 2016 International Conference on Networking and Network Applications 978-1-4673-9803-9/15 © 2015 IEEE DOI 10.1109/NaNA.2016.74
[27]. Alex M. R. Ruelas, Christian Esteve Rothenberg "A Load Balancing Method Based on Artificial Neural Networks for Knowledge-defined Data Center Networking" IFIP LANC 2018, October 2018, São Paulo, Brazil © 2018 Association for Computing Machinery. ACM ISBN 978-1-4503-5922-1/18/10
[28]. Federico Cimorelli, Francesco Delli Priscoli, Antonio Pietrabissa, Lorenzo Ricciardi Celsi, Vincenzo Suraci, and Letterio Zuccaro "A Distributed Load Balancing Algorithm for the Control Plane in Software Defined Networking" 24th Mediterranean Conference on Control and Automation (MED) June 21-24, 2016, Athens, Greece 978-1-4673-8345-5/16/ ©2016 IEEE
[29]. Akbar Majidi, Xiaofeng Gao, Shunjia Zhu, Nazila Jahanbakhsh, Guihai Chen "Adaptive Routing Reconfigurations to Minimize Flow Cost in SDN-Based Data Center Networks" ICPP 2019, August 5–8, 2019, Kyoto, Japan © 2019 Association for Computing Machinery. ACM ISBN 978-1-4503-6295-5/19/08.
[30]. Anish Ghosh 1 *, Mrs T. Manoranjitham 1 "A study on load balancing techniques in SDN" International Journal of Engineering & Technology, 7 (2.4) (2018) 174-177 © 2018
[31]. Cui Chen-xiao 1 and Xu Ya-bin 1, 2 "Research on Load Balance Method in SDN" International Journal of Grid and Distributed Computing Vol. 9, No. 1 (2016) ISSN: 2005-4262 IJGDC
[32]. Tariq Emad Ali 1, Ameer Hussein Morad 2, Mohammed A. Abdala 3 "Load Balance in Data Center SDN Networks "International Journal of Electrical and Computer Engineering (IJECE) Vol. 8, No. 5, October 2018, ISSN: 2088-8708
[33]. C. Fancy, M. Pushpalatha, Pushpa "Experimentation of Traditional Load Balancing Algorithms in Software Defined Network" International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1S4, June 2019
[34]. Prabhjot Kaur 1, Jasmeen Kaur Chahal 2 and Abhinav Bhandari 3 "Load Balancing in Software-Defined Networking: A Review" Asian Journal of Computer Science and Technology ISSN: 2249-0701 Vol.7 No.1, 2018
[35]. Umme Zakia and Hanene Ben Yedder "Dynamic Load Balancing in SDN-Based Data Center Networks" 978-1-5386-3371-7/17 ©2017 IEEE October 2017.
[36]. Dr.Abhay Kothari 1, Neha Bharani 2 "Software Quality Measurement using complexity Analysis of Various software Engineering phases" Test Engineering and Management Pages ISSN: 0193-4120 Pages No.9437-9447.
[37]. Dr.Abhay Kothari 1, Neha Bharani 2 Evaluation of Software Complexity Based on Neural Network and Machine Learning Techniques: A Review Jul 2021.