ANALYSIS & ENHANCING LOAD BALANCING TECHNIQUE IN SDN ENVIRONMENT USING DIFFERENT ALGORITHMS & ROUTING WITH MULTIPLE SDN

[1]Aakash Ujjainwal, [2]Dr. Sandeep Gupta
[1]Research Scholar, [2]Assistant Professor
[1]ujjainwal341@gmail.com, [2]guptasandeep1093@gmail.com
CSE Department
Sunderdeep Engineering College Ghaziabad
Ghaziabad, UP, INDIA

Abstract

Software-defined networking (SDN) is an innovative approach in networking that allows administrators to initialize, control, manage, and dynamically modify network behavior through open interfaces and the abstraction of lower-level functionality. Traditional network architectures, being static, often fall short in supporting the dynamic and scalable computing and storage needs of modern computing environments. SDN addresses this limitation by decoupling the system responsible for making traffic decisions from the systems that forward this traffic to its destination. Load balancing is a technique used in computer networks to distribute workload across multiple computers, ensuring that tasks are completed within the same time frame. This can be implemented using hardware, software, or a combination of both, and often results in computer server clustering due to the load balancing process. This paper explores various load balancing algorithms that can enhance resource utilization and ensure consistent service delivery to multiple clients in an SDN environment. It examines the analysis and enhancement of load balancing using multiple SDN controllers or addressing limitations in existing hash IP algorithms, such as hash collisions and load redirection. To address these limitations, we propose a new hash IP algorithm, HDW, for network load balancing (NLB) to improve network efficiency, availability, and scalability. This new algorithm is achieved through a constructive merger with the weighted scheduler (WS) technique and dynamic switching of routing paths (DSP). This approach helps reduce delays and jitters and provides some level of security due to the hashing process. Our comprehensive simulations and performance evaluations demonstrate that the proposed HDW algorithm is more efficient compared to other load balancing algorithms in the context of software-defined networking.

***Keywords**: SDN; Open flow; API; FTP; NLB, Floodlight controller, topology, hash IP*

1. INTRODUCTION

Software-defined organizing (SDN) marks a major takeoff from conventional organizing by decoupling the control plane from the information plane in gadgets like switches and switches. This imaginative design in fig-1 employments a centralized controller to oversee all sending components, varying from the customary coordinates model. SDN empowers communication between the control and information planes through southbound interfacing, permitting the control plane to coordinate the information plane's operations. Also, northbound interfacing give broad arrange programmability, encouraging the creation of applications that can robotize organizing assignments and drive innovation. Early SDN models with a single centralized controller experienced a few issues. Such a controller can get to be a bottleneck, ruining productivity and versatility. It, too, presents a noteworthy hazard to arrange accessibility and security since the disappointment of this controller can disturb the whole network's control plane, and a compromised controller can result in a total misfortune of arrange management. To address these issues, multi controller structures have been presented. Utilizing numerous controllers improves excess, as the disappointment of one controller doesn't cripple the whole arrange. This approach moreover progresses security, as different controllers can collaborate to distinguish and confine a compromised controller. Additionally, versatility is improved since the workload can be dispersed among numerous controllers, anticipating any single controller from getting to be a bottleneck. This paper points to give a nitty-gritty diagram of dispersed SDN designs, recognizing between coherently and physically conveyed models. It basically surveys existing writing, distinguishing crevices in talks almost multi controller usage. by tending to these holes, the paper points to contribute to the improvement of more vigorous and versatile SDN plans.

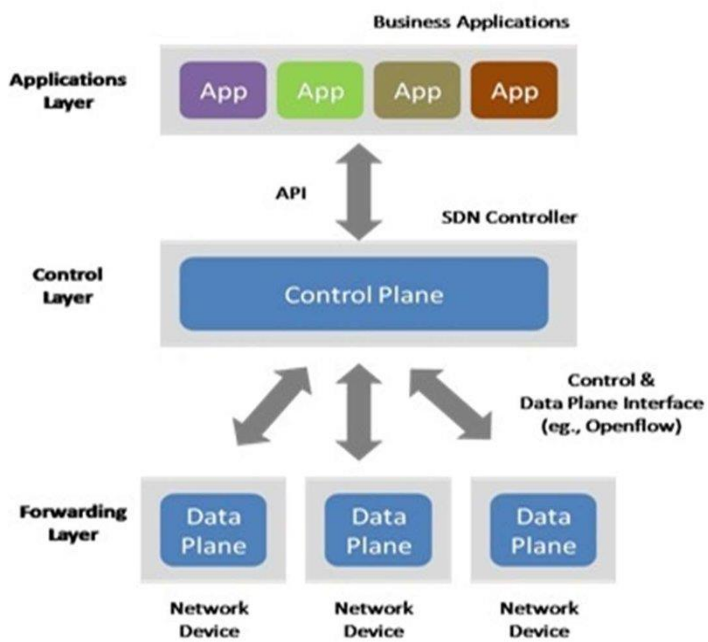

Fig-1

2. Literature Survey

The field of Software Defined Networking (SDN) has seen significant advancements, focusing on load balancing and congestion management to enhance network performance and reliability. Foundational concepts and challenges in SDN, such as the separation of control and data planes, have been discussed by M. Mousa et al. (2016). Hierarchical routing schemes tailored for SDN-VANETs, as proposed by Gao Y. et al. (2018), aim to mitigate traffic congestion and improve network performance. Dynamic load balancing frameworks for server clusters leveraging SDN's centralized control have been introduced by Chiang Mei-Ling et al. (2020). Congestion-aware routing mechanisms using real-time network state information, as discussed by S. Attarha et al. (2017), enhance network efficiency. Comprehensive surveys by Hafeez T. et al. (2018) and Al-Heety O.S. et al. (2020) explore congestion detection and mitigation in data centers and SDN-VANETs, respectively. Z. Chen et al. (2017) address load balancing in high-density SDN-based WiFi networks, while Sahoo K. S. et al. (2020) propose switch migration-based load balancing for multicontroller SDNs in IoT environments. An artificial neural network-based approach for load balancing in SDNs has been presented by S. WilsonPrakash and P. Deepalakshmi (2019), and a systematic review of various load balancing mechanisms is provided by Neghabi A. A. et al. (2018). These studies collectively underscore the potential of SDN to revolutionize network management and highlight the ongoing research to address its challenges.

Methodology: -

In this research paper we have used different type of methods & technique through which we Have described best analysis & enhancing load balancing technique in SDN Environment using Multi controller SDN Network with help of NLB technique ,weighted scheduling technique , Improved hash Ip algorithm, dynamic switching of routing path with multiple controller. The basic overview is understood by this fig-2

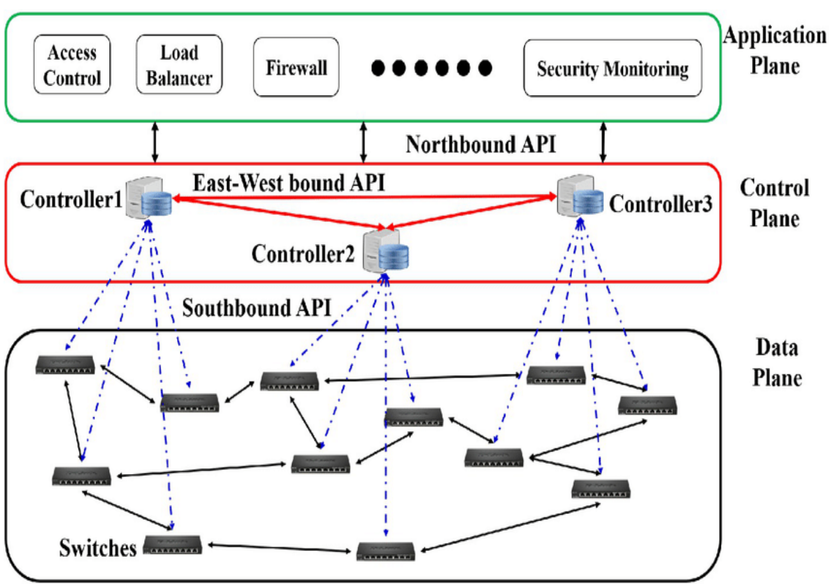

Fig-2

Here, we make the capable NLB method by uniting the made strides Hash IP calculation with a weighted scheduler and Enthusiastic trading of coordinating way (HDW calculation). To move forward security and organize unflinching quality, hash calculation executes cryptographic work to hash data into curiously key [19]. This makes a contrast to assign resources to the right isolated to expect redirection and overabundance on the organize [20].

After that we used here improved hash IP algorithm which also helps to achieve better performance & analysis in load balancing technique in multiple SDN environments & here we used to describe it from different resources.

The improved hash IP algorithm:-

The innovative Hash IP algorithm streamlines network load balancing by eliminating the need for extensive table lookups of streams, a common issue in both dynamic and static algorithms. This algorithm creates a unique hash key by combining the destination and source IP addresses of both the client and server. Unlike earlier Hash IP algorithms, the new version incorporates the calculated weight of packets into the hash key to avoid hash collisions, which were a significant problem previously. This unique hash key is then used to allocate clients to servers. To manage varying bit lengths, the network load balancer employs a secure hash algorithm, using cryptographic functions that include modular additions, compression functions, and bitwise operations to convert data into hash values, as shown in Fig-3. Incoming IP addresses are accompanied by IP header fields that provide essential information such as the source IP, protocols, data, packet size, and checksum. The modern Hash IP algorithm captures the size of packets to calculate their total weight, which is then used to determine the number of request queues on the network. By using a scheduling technique, the Hash IP algorithm ensures that requests are appropriately allocated based on the weight of the packets.

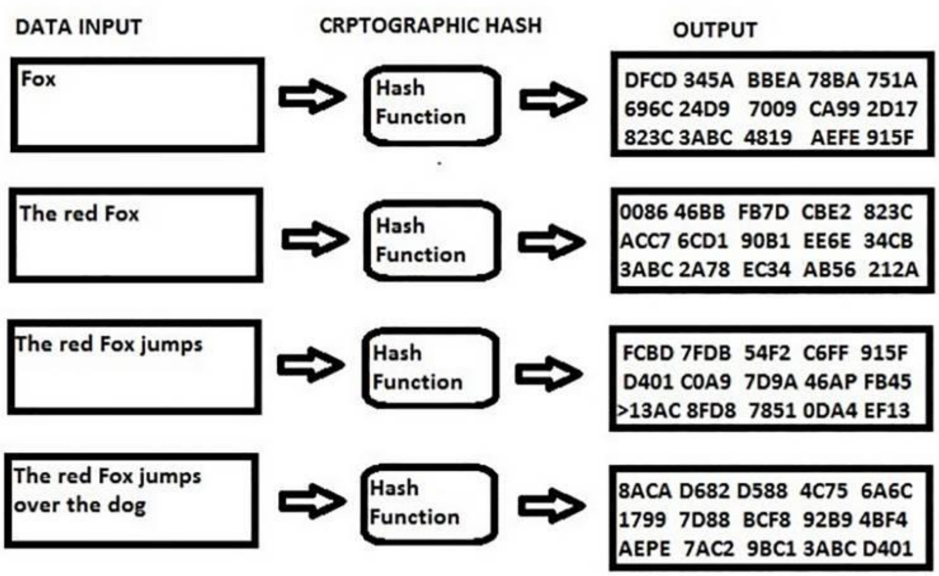

fig-3

Another methodology which we have to use to analysis & enhancing the load balancing technique in SDN technique is weighted scheduling algorithm.

Weighted Scheduling Algorithm:-

This technique helps distribute traffic to the appropriate servers on the network based on their capacity. Determining the actual size of packets (weight) to be transferred on the network was a challenge for most weight scheduling techniques [23]. Our proposed new Hash IP algorithm addresses this limitation. The algorithm calculates the total weight of packets, assisting the weight scheduler in assigning the correct weight to the appropriate servers on the network. These two methods complement each other. The weight scheduling technique first directs heavy traffic to the strongest server, followed by medium servers, and finally to the least powerful ones. Each queue q_i with equal or different data packet sizes has an associated weight w_i . Katevenis, Sidiropoulos , and Courcoubetis proposed the weight scheduling (WS) technique [24], represented by:

$$\frac{w_i}{\sum_{k=1}^{n} w_k}$$

For bandwidth allocation, the total weight and packet size of each incoming request are considered. Given a mean m_i of each queue q_i, the bandwidth allocation is calculated by:

$$\frac{m_i \times w_i}{\sum_{k=1}^{n} m_k \times w_k}$$

For link allocation of each queue q_i their capacity is:

$$\sum_{k=1}^{n} l_i = 1$$

Therefore, the weight is set as:

$$w_i = \frac{l_i}{m_i}$$

The description of symbols for the equations is found in Table 1. For each weight, bandwidth is calculated and a specific link path is assigned to transmit the request on the network, avoiding collisions and improving performance. In Hash tables, we employed insert, search, and remove methods for hashing. The implemented pseudo code methods created a unique way to manage values in the algorithm [25]. These values include the calculated weight of packets, data, and the key formed by the IP addresses. The number of incoming requests (RN) channeled in queues on the network is also added to the hash values to ensure uniqueness.

Table 1

Notation table.

Symbol	Description
w_i	Weight number of every packet on the network
m_i	Mean of calculated weight
$\sum_{k=1}^{n}$	Sum of every weight
l_i	Link path to direct packets
n, k, i	Variable to represent numbers

Dynamic switching of routing path:

Here's a humanized and paraphrased version of the given text:--We also proposed an optimal method for forwarding packets within the routing path of a software-defined network. This approach integrates conventional switches with programmed switches on the same network to ensure efficient packet forwarding without any leakages or disruptions . The technique we employed, Dynamic Switching of a Path (DSP), is resilient to routing failures, conserves energy, and extends the network's lifespan . DSP utilizes Hybrid routing protocols, which address the limitations of both static and dynamic protocols. These Hybrid protocols include:**Interior Gateway Routing Protocols (IGRP):** This protocol transmits data within the network's radius, maintaining an internal link-state routing path and using an enhanced version of IGRP for routing within autonomous systems.**Exterior Gateway Routing Protocol (EARP):** This protocol handles data transmission outside the network's zone radius, as described by Erick Osborn in 2020 .The Hybrid routing protocol dynamically assigns zones to each node and operates during transmissions inside or outside these routing zones . EARP manages outer node routing, while IGRP manages inner node routing. In Fig. 4, OF represents a peripheral node using enhanced IGRP. Nodes K-L are interior nodes also using IGRP. Nodes H and I are in outer zones relative to node G, which transmits data outside the network's radius using the Exterior Gateway Routing Protocol. When data is not available within the network zone, node G uses the Border Gateway Protocol (BGP) in EARP to route data outside the network radius. Fig. 2 illustrates the routing mechanisms of DSP.

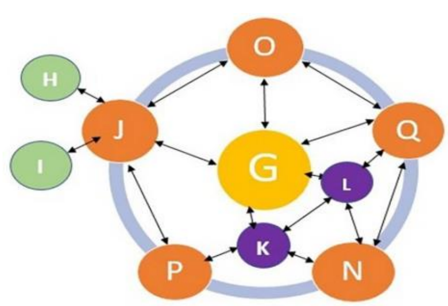

Fig-4

Architectural module of HDW algorithm:

The architectural model of the HDW Algorithm illustrates the interrelationship between the combined techniques and their complementary operations within the network. As shown in Fig. 5, requests are queued as q1, q2, q3, ... qn, in to the allocated server, with each queue having an associated weight wi (representing the number of packets). In a scenario with three servers in the SDN, server C handles the highest traffic loads, followed by server B with medium loads, and server A with the least loads. The dynamic links facilitate the switching of routing paths between servers, enhancing the network's efficiency. The Floodlight controller stands out for its ability to offload traffic on the network more effectively than other Open Flow controllers, as demonstrated during simulations.

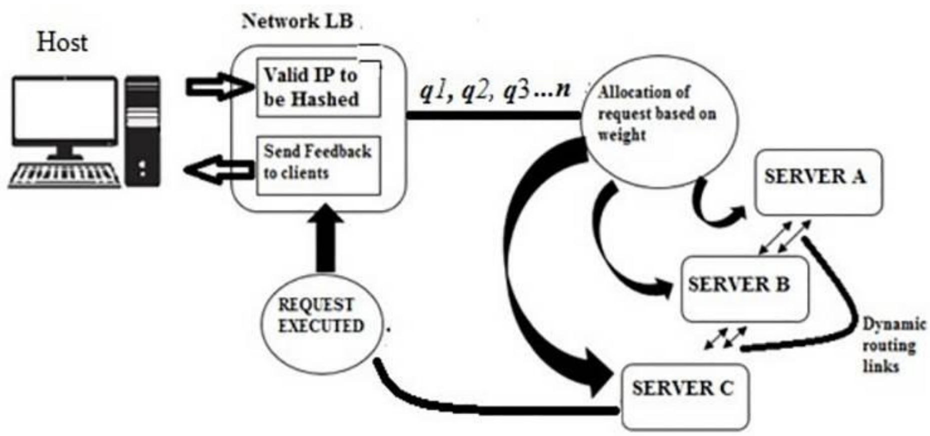

Fig-5

3. Related work

In the above sections methodology we have explained different technique, algorithm & methodology which is proven help in load balancing in a SDN environment so now this research related work we have also use & merge those findings with our finding to get a effective research result .Here we have also used another reference to understand & relate with our work for better load balancing technique with architecture of multi controller software defined network are like a multi controller architecture consists of multiple controllers collaborating to enhance performance and scalability. In the realm of software-defined networking (SDN), multi controller architectures can vary significantly in their features and characteristics. We will explore these variations in the following paragraphs, focusing on the distinctions between logically or physically centralized and distributed architectures, as well as between flat and hierarchical designs. Additionally, we will discuss aspects such as elasticity, controller placement, and inter controller communication. In this analysis we have different scenarios from different resources are like Physically Centralized versus Physically Distributed , Logically Distributed versus Logically Centralized , Flat Architecture versus Hierarchical Architecture , Dynamic Architecture versus Static Architecture, Inter controllers Communication , Placement Problem in Multi controller

Software-Defined Architectur using all these scenarios & above algorithms in methodology on a Single SDN Device in a multiple SDN controller architecture we can get more reliable or fast service as required

4. Result & Analysis

In this section, we assess the performance of our proposed HDW algorithm by comparing it to three existing approaches. These approaches include the distributed load balancing algorithm using Weighted Round Robin (WRR) by Vyakaranal et al. [16], referred to as DWRR; the weighted load balancing with binary node algorithm by Robin et al. [13], referred to as WLB; and the dynamic weighted random selection algorithm by Mei-Ling et al. [3], referred to as DWRS, as shown in Table 4. For fairness, we used identical parameters across all experiments for each algorithm being evaluated.

Table-2Comparative analysis of related algorithms

Research work	Performance Metrics				
	Throughput	Secured	Congestion Control	Availability	Jitter
DWRS [3]	✔	×	✔	✔	0.21
WLB [13]	×	✔	✔	✔	0.35
DWRR [16]	✔	×	×	✔	0.19
Proposed Technique	✔	✔	✔	✔	0.28

The performance metrics are based on quality of service parameters such as throughput, jitter, response time, and network security. Our proposed HDW algorithm demonstrated higher efficiency, as indicated by the superior results compared to other algorithms. Vyakaranal et al.'s distributed load balancing algorithm using Weighted Round Robin (DWRR) [16] achieved high throughput and improved network availability, but it lacked security measures. Robin et al.'s weighted load balancing with binary node (WLB) algorithm [13] provided good security but suffered from lower resource availability, negatively impacting network throughput. Mailing et al.'s dynamic weighted random selection (DWRS) algorithm [3] offered random request selection to balance throughput and availability; however, it did not account for the weight and execution time of requests, which led to issues with congestion and security. Figure 6 presents a comparative graph analyzing the resource utilization efficiency of these algorithms.

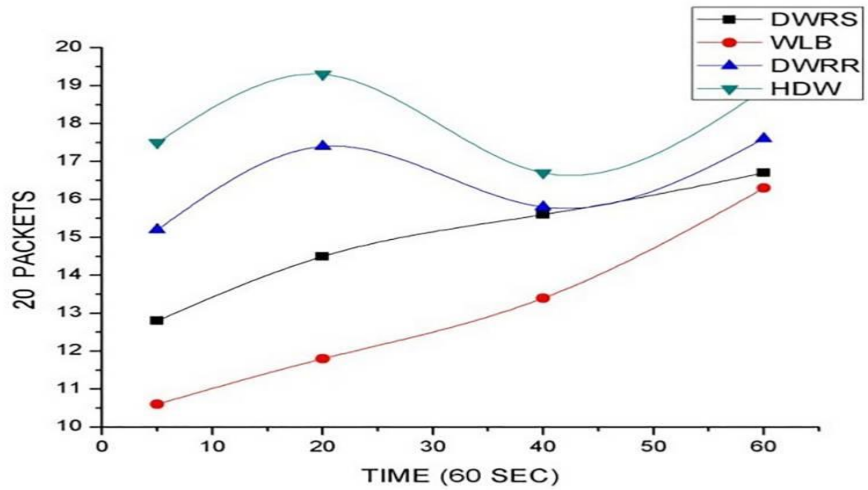

fig-6

The proposed HDW algorithm maximizes throughput to enhance resource availability through an efficient weight scheduling method. By incorporating hash functionality, the HDW algorithm ensures network security. Additionally, it effectively controls congestion by dynamically switching routing protocols. With the help of above reference researches we have found that load balancing is enhanced while using above techniques with multiple SND individually in Combined SDN architecture .which will helps the entire network to get enhanced quality of services on various aspects.

5. Conclusion

In this research paper we have used different kinds of sources & references to analyze & enhance the Load Balancing in SDN environment we have determine the main purpose of this paper is to balancing the load on network environment with the help of using different enhancing technique individually on a single SDN device in a multiple load balancing SDN architecture we have use different algorithms and their graphs & results individually, methodologies and different scenarios from different resources. Here we have determine that's how we can make our SDN architecture better to enhancing quality of services with the help of different algorithms , methods , scenarios & different networking features on Individual SDN device in Multiple SDN Environment . Here we have concludes different aspect & points . In this work, we developed a network load balancing technique for software-defined networking (SDN) by integrating a Hash IP load balancing algorithm with a weighted scheduler and dynamic routing path switching. Our simulation results demonstrated the effectiveness of the floodlight controller, which accurately displayed source and destination IP addresses, scheduled traffic links, and the total number of transmitted packets. The proposed algorithm, combined with the floodlight controller, achieved superior results in terms of throughput, response time, and jitter, indicating significant improvements in scalability, availability, and performance for SDN.The evolution of SDN highlights the shift from centralized to distributed architectures, driven by the need for greater efficiency, scalability, and availability. This paper provided a comprehensive overview of SDN multicontroller architectures, detailing their characteristics and presenting various examples of implemented and experimental solutions. Future research should focus on exploring dynamic network load balancing with different algorithms and SDN controllers, as well as evaluating the new Hash IP algorithm and weighted scheduling method across various network topologies and sizes to identify potential limitations. Network researchers and designers will face challenges in optimizing multicontroller networks, such as developing efficient communication processes, designing robust network architectures, and integrating new applications into the northbound interface that support multiple controllers. Addressing these challenges is crucial for advancing the capabilities and performance of SDN multicontroller environments.

References

1. M. Mousa, A. M. Bahaa-Eldin and M. Sobh, Software Defined Networking concepts and challenges. 11th International Conference on Computer Engineering & Systems (ICCES), Cairo, Egypt, 2016, pp. 79-90.

2. Gao Y., Zhang Z., Zhao D., Zhang Y. and Luo T. A hierarchical routing scheme with load balancing in software defined vehicular ad hoc network. in *IEEE Access*, vol. 6, pp. 73774–73785, 2018. doi: 10.1109/ACCESS.2018.2884708 [CrossRef] [Google Scholar]

3. Chiang Mei-Ling, Cheng Hui- Sheng, Lui Hsien-Yi, Chiang Ching-Yi. SDN-based server cluster with dynamic load balancing and performance improvement *Springer Cluster computing* 24 (1), 537–558 doi: 10.1007/s10586-020-03135-w [CrossRef] [Google Scholar]

4. S. Attarha, K. Haji Hosseiny, G. Mirjalily and K. Mizanian, A load balanced congestion aware routing mechanism for Software Defined Networks, 2017 Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran, 2017, pp. 2206–2210.

5. Hafeez T., Ahmed N., Ahmed B. and Malik WA., Detection and mitigation of congestion in SDN enabled data center networks: A survey in *IEEE Access*, vol. 6, pp. 1730–1740, 2018. doi: 10.1109/ACCESS.2017.2780122 [CrossRef] [Google Scholar]

6. Al-Heety O.S., Zakaria Z., Ismail M., Shakir M. M., Alani S. and Alsariera H. A Comprehensive Survey: Benefits, Services, Recent Works, Challenges, Security, and Use Cases for SDN-VANET, in *IEEE Access*, vol. 8, pp. 91028–91047, 2020. doi: 10.1109/ACCESS.2020.2992580 [CrossRef] [Google Scholar]

7. Z. Chen, S. Manzoor, Y. Gao and X. Hei, Achieving Load Balancing in High-Density Software Defined WiFi Networks, 2017 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2017, pp. 206–211.

8. Sahoo K. S. et al., ESMLB: Efficient Switch Migration-Based Load Balancing for Multicontroller SDN in IoT, in *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5852–5860, July 2020. doi: 10.1109/JIOT.2019.2952527 [CrossRef] [Google Scholar]

9. S. WilsonPrakash and P. Deepalakshmi, Artificial Neural Network Based Load Balancing On Software Defined Networking, 2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS), Tamilnadu, India, 2019, pp. 1–4.

10. Neghabi A. A., Jafari Navimipour N., Hosseinzadeh M. and Rezaee A., Load Balancing Mechanisms in the Software Defined Networks: A Systematic and Comprehensive Review of the Literature, in *IEEE Access*, vol. 6, pp. 14159–14178, 2018. doi: 10.1109/ACCESS.2018.2805842 [CrossRef] [Google Scholar]

11. Belgaum M.R, Alansari Z., Musa S., Alam M. M. and Mazliham M. S., Impact of Artificial Intelligence-enabled Software-defined Networks in Infrastructure and Operations: Trends and Challenges, *International Journal of Advanced Computer Science and Applications(IJACSA)*, 12(1), 2021. 10.14569/IJACSA.2021.0120109 [CrossRef] [Google Scholar]

12. R. Hamed, Mohamed Rizik, B. {Mokhtar IOT Manager: concerns-based SDN Management framework for IOT networks. SPRINGER Proceedings of SAI Intelligent conference 153-167

13. R. Losch, J. Schmidt, N.G Felde Weighted Load Balancing in Distributed Hash Tabeles Digiital Library. Int. Conf.Information Integration and web-based Application, 473-482.

14. S. Kaur, K. Kumar, J. Singh and Navtej Singh Ghumman, Round-robin based load balancing in Software Defined Networking, 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2015, pp. 2136–2139.

15. M. F. Ramdhani, S. N. Hertiana and B. Dirgantara, Multipath routing with load balancing and admission control in Software-Defined Networking (SDN), 2016 4th International Conference on Information and Communication Technology (ICoICT), Bandung, Indonesia, 2016, pp. 1–6.

16. Vyakaranal, S.B., Naragund, J.G. Weighted Round-Robin Load Balancing Algorithm for Software-Defined Network. In: Sridhar, V., Padma, M., Rao, K. (eds) Emerging Research in Electronics, Computer Science and Technology. Lecture Notes in Electrical Engineering, vol 545. Springer, Singapore. 10.1007/978-981-13-5802-9_35 [CrossRef]

17. Wang W., Dong M., Ota K., Wu J., Li J., and Li G., 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. doi: 10.1504/IJCSE.2019.10017882 [CrossRef] [Google Scholar]

18. S. Yilmaz, A. M. Tekalp and B. D. Unluturk, Video streaming over software defined networks with server load balancing, 2015 International Conference on Computing, Networking and Communications (ICNC), Garden Grove, CA, USA, 2015, pp. 722–726.

19. Scott-Hayward S., Natarajan S. and Sezer S., A Survey of Security in Software Defined Networks, in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 623–654, Firstquarter 2016. doi: 10.1109/COMST.2015.2453114 [CrossRef] [Google Scholar]

20. Kang B., Choo H. An SDN-enhanced load-balancing technique in the cloud system. *J Supercomput* 74, 5706–5729 (2018). 10.1007/s11227-016-1936-z [CrossRef] [Google Scholar]

21. Mulla, M.M., Raikar, M.M., Meghana, M.K., Shetti, N.S., Madhu, R.K. Load Balancing for Software-Defined Networks. In: Sridhar, V., Padma, M., Rao, K. (eds) Emerging Research in Electronics, Computer Science and Technology. Lecture Notes in Electrical Engineering, vol 545. Springer, Singapore. 10.1007/978-981-13-5802-9_22 [CrossRef]

22. Shang F., Mao L., and Gong W., Service-aware adaptive link load balancing mechanism for software-defined networking, Future Gener. *Comput. Syst.*, vol. 81, pp. 452–464. 10.1016/j.future.2017.08.015 [CrossRef] [Google Scholar]

23. Louhab F. E. Bahnasse, Oulahyane H. A., Talea M., and Bakali A., Novel SDN architecture for smart MPLS traffic engineering-different Server aware management, *Future Gener. Comput. Syst.*, vol. 87, pp. 115–126. 10.1016/j.future.2018.04.066 [CrossRef] [Google Scholar]

24. P. -J. Chuang and H. -J. Chen, Efficient Load Balancing in Software Defined Networks, 2017 International Conference on Information, Communication and Engineering (ICICE), Xiamen, China, 2017, pp. 526–528.

25. Zhao Y., Liu C., Wang H., Fu X., Shao Q., and Zhang J., Load balancing based multi-controller coordinated deployment strategy in software defined optical networks, *Opt. Fiber Technol.*, vol. 46, pp. 198–204. 10.1016/j.yofte.2018.10.012 [CrossRef] [Google Scholar]

26. Marcos Cabral, Maurico Silveira, and Ryan Urie: Schweitzer Engineering Laboratories, Inc SDN Advantages for Ethernet-based control Journal: Science Direct

27. 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), pp. 289–294.

28. Z. Chen, S. Manzoor, Y. Gao and X. Hei, Achieving Load Balancing in High-Density Software Defined WiFi Networks, 2017 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2017, pp. 206–211.

29. Han T. and Ansari N., A Traffic Load Balancing Framework for Software-Defined Radio Access Networks Powered by Hybrid Energy Sources, in *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 1038–1051, April 2016. doi: 10.1109/TNET.2015.2404576 [CrossRef] [Google Scholar]

30. Tanha M., Sajjadi D., Ruby R. and Pan J., Traffic Engineering Enhancement by Progressive Migration to SDN, in *IEEE Communications Letters*, vol. 22, no. 3, pp. 438–441, March 2018. doi: 10.1109/LCOMM.2018.2789419 [CrossRef] [Google Scholar]