International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6618
***Article Received**: 05 March 2023 **Revised**: 06 April 2023 **Accepted**: 26 April 2023*

Dynamic Weighted Round Robin Approach in Software-Defined Networks Using Pox Controller

[1]Buhyavarapu Manasa, [2]Dr. A. Ramesh Babu
[1]Research Scholar in Computer Science, Chaitanya Deemed to be University.
[2]Professor, Dept of Computer Science, Chaitanya Deemed to be University.

Abstract: Load balancing is important in solving over-load traffic problems in the network. Therefore, it has been among the first appealing applications in Software Defined Networking (SDN) networks. Numerous SDN-based load-balancing approaches have been recommended to enhance the performance of SDN networks. However, network control could be more manageable in large networks with hundreds of switches and routers. The SDN is a unique way of building, controlling, and developing networks to modify this unpleasant situation. The major concept of SDN contains logically centralizing network management in an SDN controller, which manages and observes the behaviour of the network. Numerous load-balancing approaches are known, such as Round Robin (RR), random policy, Weighted randomized policy (WRP), etc. Every load-balancing policy approach has some benefits and detriments. This paper developed an advanced load-balancing algorithm, a dynamic weighted round-robin (DWRR), and ran it on the top of the SDN controller. Then we calculate the result of our proposed load-balancing approach by comparing it with the current round-robin (RR) and weighted round-robin (WRR) approaches. Mininet tool is utilized for the investigation, and the controller utilized as the control plane is named the POX controller.

Keywords: Software Defined Networking, Distributed Load balancing, SDN controller, round robin, dynamic weighted round robin.

I. INTRODUCTION

Nowadays the internet connects people everywhere around and has completed it possible for everyone from everywhere. Although traditional IP networks are widely recognized, they are challenging to configure and monitor due to their predefined policies. Reshaping society in response to failures, flaws, masses, and ups and downs is daunting. Traditional networks use embedded hardware and software to route traffic through routers and switches. They are included by upright linking both the controller and information planes. Creating our custom algorithms is impossible in traditional load balancers, as they can be unique to the resource and cannot be programmed.

Data Center Network (DCN)

A data center network (DCN), which connects all intermediate record assets, is essential to the operation of an information center. DCNs necessity be ascendable and environmentally friendly enough to join tens of hundreds or dozens of servers (SDNs) to meet the growing expectations of SDN. Load balancing is one of the greatest serious issues in analytics centers, irrespective of their multiple outlines, whether physical data facilities or digital analytics centers. In most cases, an OpenFlow stats network architecture based on SDN is implemented to obtain better metrics to balance average throughput and tourist load. The network load balancing should be used to extend the accessible bandwidth and increase performance and redundancy load. The ability to pin visitors to more than one internet connection is called as network load balancing. By distributing the bandwidth each LAN user provides across more than one connection, this feature balances the network bands, including the Internet, email, etc., thus optimizing the available bandwidth.

Software Defined Network (SDN)

SDN is an inventive version that facilitates network processing and management by introducing useful low-grade inferences. It permits network managers to respond quickly to business development changes because of topological or insurance needs. In SDN, unlike in conventional networks, average changes in throughput and capacity are monitored and controlled by software packages that live in controllers rather than male or female devices configured with administrator assistance. The SDN framework contains basic schemas. The manipulative plan is responsible for the facts transmitted to the network. It includes all the software for good governance programmed into the controller server. The information layer is especially accountable for sending points. It has network gadgets such as controllers and routers. The planes communicate via OpenFlow, a common protocol for SDN fabric conversations.

SDN [1] specializes in dividing the control plane that controls the network and log levels on which traffic flows, allowing network administrators to lock site visitors with international central control. The OpenFlow protocol transmits by removing administration procedures and network devices

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6618
***Article Received**: 05 March 2023 **Revised**: 06 April 2023 **Accepted**: 26 April 2023*

from single-handling aircraft controllers and identifying them in a centralized logical controller. The controller generally manages design administration for SDN-enabled devices, delivers presentation and responsibility control, and comprehends network topology. Loaded with the above data, it can develop connection needs based on selected necessities, including Quality of service, and execute cross-device hyperlink administration. SDN also presents programmability and new inferences within the network that facilitate community development. Network virtualization has benefits, such as dynamically rotating up and down and adjusting for typical software use cases. In addition, it can install safety regulations in every network [2].

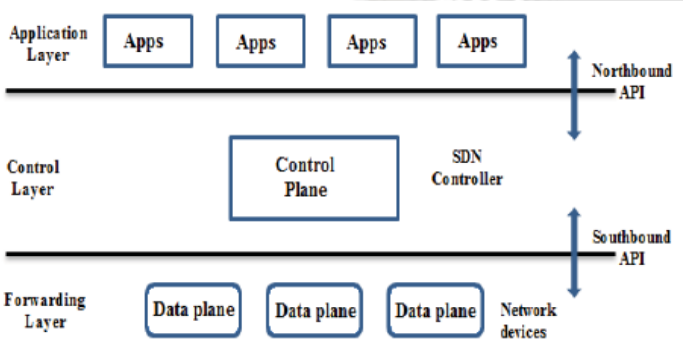

Fig.1 SDN Architecture

As shown in figure 1, the SDN is divided into three-layered versions: an application layer, an infrastructure layer, and a processing layer. It can identify the partitioning of most control and information levels via a translucent and obvious programming interface among the devices within the network and the SDN controller. This division among the two degrees is essential for producing abstract reasoning in networks by dividing the network control problem into capacity features, donating to the network's upward momentum and evolution. The control plane controls what the community looks like and decides where and how we should push visitors. We have entire control over the redirect layer agents. The software layer includes programs that utilize the power provided by the north interface to implement community power.

SDN can be operated to link registry centers with public cloud operators developing a dynamically scalable hybrid cloud network. To break the layer boundary, the control that SDN follows now consists of packet forwarding within the switch section and link change at the data link level. Based on the recognition of each instant community and consumer-demarcated strategies, SDN enables actual central management of the network with the ability to gain instant popularity in the community [3]. Specifically, SDN allows centralized control over the internet and response control with data swapped among private layers within community materials. With properly designed and accurately intended centralized algorithms, several of the favoured performance issues can be dealt with efficiently. Demonstrates new answers to everyday issues like balanced packet routing, realistic traffic arrangement, QoS support, forced green booting, and congestion management for dropped site visitors. The exemption can be quickly developed and implemented to confirm and verify its achievement in improving the network's overall performance [4]. Load balancing is a portable computer networking technology for distributing the workload across more than one computer system or group of laptops, community links, first-processing devices, disk drives, or other properties.

Our main contribution to this paperwork is as follows:

- The assessment of a server load-based load-balancing resolution for SDN.
- We have performed the DWRR method, namely SDN's dynamic weighted round-robin load balancing approach.
- We take as initial all the parameters, such as link delay and link speed. Conferring to various speeds, we assigned different delays to each link.
- The Mininet emulation tool is employed to test the proposed load-balancing models.
- Finally, we compared our proposed method with existed round-robin approach.

II. REVIEW OF LITERATURE

In today's network, a single server cannot handle all buyer requests because the traffic volume is enormous. The primary ability of load balancing is to allocate the buyer's load across multiple servers. Traditional load balancers have been used for this reason. Nonetheless, the biggest difficulties with these load balancers are the unprogrammable stylists. SDN-based load balancers are now fully deployed. We can try a stupid open stream device on a robust load balancer by writing an SDN program (in combination with a load balancer).

Several papers discuss the software of SDN in communication networks. The primary technical competencies of the precise special features are described in [5]. Various models have been proposed to enhance community slippage, solve network congestion by converting occasional float paths into glide streams, and acquire load balancing between distinct links. Most of the classifications of load balancing within the analysis are based, more often than not, on the capabilities of load-balancing methods. According to those classifications, load-balancing algorithms are classified as static or dynamic, centralized or decentralized.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6618
Article Received: 05 March 2023 Revised: 06 April 2023 Accepted: 26 April 2023

Arkar Soe Linn et al. [6] The SDN is a new system at the upward thrust inside the age of facts generation. SDN is greater flexible and programmable than a traditional network. One critical way to use SDN is as a server load-balancing approach. The load on servers improves each day with the usage of the Internet. Therefore, this load may need to be set up on the servers to provide green offerings to forestall customers without delay. In our proposed gadget, the weighted random, round robin, and spherical robin load balancing techniques are realized using an OpenFlow switch related to a fully Python-based SDN-based POX controller.

Sujayanth et al. [7] This paper implemented a single set of load-balancing regulations for the entire SDN-based backbone to address these issues. POX is lobbying for an SDN implementation and software from the Mininet network simulation software. The implementation of load-balancing rules includes Python to produce farm topologies. The test focused on the overall best global transaction provider (QoS) performance, hybrid reaction time, and transaction fees.

Garima Sinha et al. [8] This paper aims to recommend stable and fast load-balancing rules to users of multiplication help and to evaluate the overall performance of algorithms designed to use popular load-balancing algorithms. The proposed guidelines will not forget cloud lifetime, expected downtime for tasks by a virtualization system, and virtual machine uptime dwellings to allocate incoming demand for virtual machines fairly and proficiently. The response time of the EWRR approach is much lower than in the evaluation of the different techniques. The EWRR obtained the best score rating for the RR, Throttled, ACO, and Hybrid reaction conditions of 0, 77, 2.20, 8.31, 20.82, and 100, respectively.

Genetu Y. Basena et al. [9] Load balancing is necessary to properly allocate rack requests to the offloaded server and dynamically maintain load distribution among server companies. In traditional IP networks, maintaining load balancing is impossible and is no longer scalable due to poor visibility of the global topology through routing agents. However, SDN can make significant determinations of any topological alternative in the shortest possible time. To address the above task, we proposed a new server interface load-balancing technology that enables an efficient and environmentally friendly server processing machine for OpenFlow SDN. Experimental results on the Ryu driver and the Mininet emulator show poor overall performance on modern hardware.

Omran M et al. [10] This check proposes an SDN-based Total Load Balancing implementation. It presents preconfigured servers in the farm that accept a packet of Internet Protocol (IP) statistics from a couple of customers in the same area for audiences and requests. The experiments were completed with the use of Mininet™ and have been based on numerous possibilities (state of affairs a, state of affairs b, situation c) of the network topologies. The parameters utilized to assess load balancing in SDN are throughput and delay. The consequences specified that situation (a) provides excessive average performance, eventualities (b) and (c) provide little jitter values, and case (c) provides the least postponement. Impact SDN offers adaptive multipath for brilliant direction detection for better network overall performance.

Şeyma Aymaz et al. [11] This paper developed into a new generation of networking that separates processing and recording planes, enables network scheduling, and responds fast to converting conditions via imparting a global view of the network. To triumph over the restrictions of previous community infrastructure, The SDN sits for load balancing above the coping with level. Load balancers distribute a load of most of the compromised servers based totally on proprietary technology. Random and round bots are several weight-balancing strategies. In this announcement, those strategies that run on top of the POX driver and characteristic evaluation of Wireshark usage have been finished.

Taufik Hidayat et al. [12] Using load balancing in a network is desirable if clients access the network actively and widely. One of the intentions is to allow social imbalances to emerge. A Round Robin (RR) rule set can be useful to community load balancing due to its simple set of policies for scheduling strategies that can provide the overall performance of the technical charts. The authors use a Scientific Literature Summary (SLR) approach that can be implemented to select criteria through object searches to preserve the identity that emerges. The SLR is divided into five phases: question apparatus, criteria selection, presentation selection, observation outcome selection, and appropriate evaluation. With an SLR, the files are expected to meet the standards, and you can see how amazing it is.

III. LIMITATIONS IN THE PREVIOUS WORK

- Although comfy to execute, round-robin DNS has several dangers, together with those emerging from report caching inside the DNS order itself, in addition to consumer-facet discourse caching and the style of which can be problematic to address.
- If a provider at one of the discourses inside the list yields, the DNS will keep to present out that deal with, and clients will nonetheless endeavor to get the non-purposeful assistance.
- Due to it does not remember transaction time, network congestion, and server load, it plays most accurately for

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6618
***Article Received**: 05 March 2023 **Revised**: 06 April 2023 **Accepted**: 26 April 2023*

help with a couple of consistently dispersed associates to servers of equal volume. Then, it simply does load distribution.

- Large overhead is the number one hindrance of the earlier paper.
- The hyperlink delay parameter is not evaluated in this preceding work.

IV. PROPOSED METHODOLOGY

This paper proposed a dynamic weighted round-robin (DWRR) load-balancing method and solved the limitations in the previous work. This work evaluates all the parameters, like link delay and speed. According to various rates, we designated various delays to every link. The server with the most elevated speed link represents a minor delay and contains the most recommendations.

Dynamic Weighted Round Robin (DWRR)

A DWRR approach is much like a round-robin in that the way orders are allocated to nodes remains repeated, albeit with some wind. The node with the best specification may be proportionally cut up for a better variety of requests.

But how can the weight balancer know which node has the highest potential? We assign "weights" to every node when configuring a load balancer. The node with the highest specification needs to get the quality weight of the path.

Typically, we specify weights that can be proportional to real capacities. So, as an example, if server 1's abilities are five instances more than that of server 2, we will assign it a weight of five and server two a weight of 1.

Thus, when clients input, the first 5 may be allotted to node one and the sixth to node 2. If more customers enter, the same institution can comply. The seventh, 8th, ninth, 10th, and eleventh will all go to server 1, 12 to server 2, and so on.

Capacity is not usually the only foundation for selecting a Dynamic Weighted Round Robin (DWRR) algorithm. Sometimes we want to apply it if, for instance, we need a server to get considerably fewer connections than a similarly successful server because the first server handles business-important applications. We also don't need it to result easily overloaded

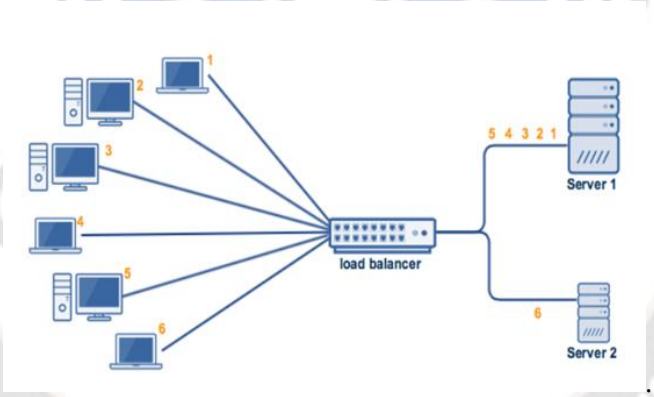

Fig.2 Load balancer with multiple servers

PROPOSED WORK ARCHITECTURE

DWRR Approach:

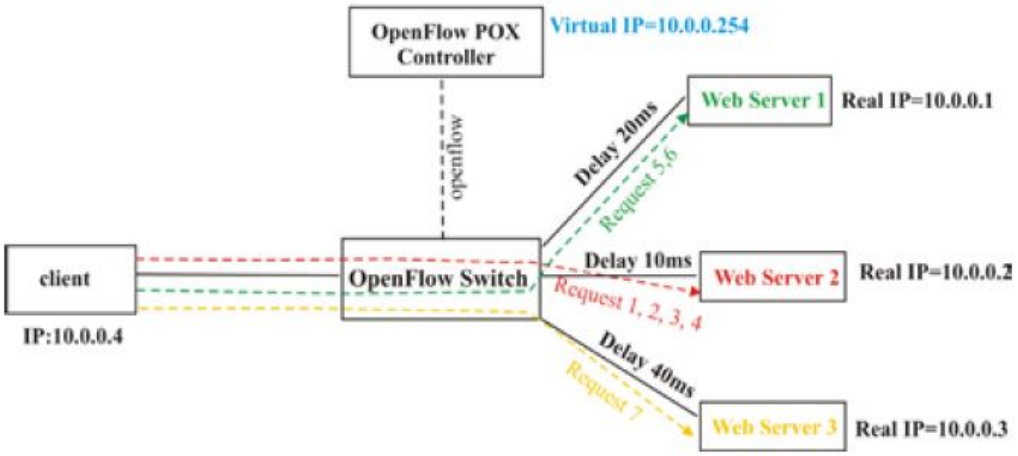

Fig.3 Proposed DWRR Topology

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6618
***Article Received**: 05 March 2023 **Revised**: 06 April 2023 **Accepted**: 26 April 2023*

Algorithm: Dynamic Weighted Round Robin

Input: Requesting for set of available servers.

Output: Efficient request allocation to servers.

Begin

Get list of servers

When a new client request comes in Pick a server from the list of servers

Using

$$lastServerindex = \frac{lastServerindex+1}{len(Available\ Servers)lastServerindex}$$

$$= lastServerindex + weight\ chosen\ server = Available\ Servers[lastServerindex]$$

Request is forwarded to every server in a cyclic fashion

End

The above-suggested method demonstrates the performance of planning forward appeals from client nodes and designating servers to consumer nodes operating a round-the-dynamic weighted round-robin load-balancing algorithm. In this method, we will distinguish among the heterogeneous servers. For the pattern, if server 1 has a higher weight than server 2, the algorithm will allocate higher propositions to the server with better load-handling ability. Here a distinct weight is allocated to every server. A DWRR is similar to a round-robin (RR) in that appeals are assigned occasionally to the servers. However, larger requests can be allocated to the server with the best conditions.
DWRR works in such a way that the server framework no longer considers instances of executing functions properly; However, consider assigning a service company with the highest uptime. This ensures that the load is distributed evenly across the servers, which reduces the standard response time.

Two variants, V_{max} and V_{min} are utilized to estimate the threshold variety. Threshold for every VM is considered as follows

$$T_{min} = \frac{c}{c} * V_{min} * L * n$$

Where,

Where V_{min} indicates the minimum variance, L signifies the total capacity of a node, and n signifies the total number of VMs. T_{min} denotes the minimum threshold value of a VM.

$$T_{max} = \frac{c}{c} * V_{max} * L * n$$

Here, T_{max} signifies the maximum threshold value of a $VM\ and\ V\_(max)$ signifies the maximum variance

V. EXPERIMENTAL SETUP

The research used the Mininet tool to design the network topology to check the proposed DWRR technique called the Dynamic Weighted Round Robin (DWRR) Load Balancing approach. These network topologies contained three servers, one client, a POX controller, and an OpenFlow switch topology. The host h4, having IP deals with 10.0.0.4, performs as a client. The hosts h1, h2, and h3 have IP addresses 10.0.0.1, 10.0.0.2, and 10.0.0.3 as web servers. On host h4, we set up the "siege" load testing device. On hosts h1, h2, and h3, we completed net servers. There are links among the servers and the OpenFlow switch, and we selected a 20,10,40 ms delay on a different hyperlink. Behind that, we designated the dynamic weight according to the delay. The server with the quickest delay holds more weight (visitors) than other servers having more delay.

As demonstrated in Figure. 3, the link put off a number of the server h2, and the Openflow transfer is 10 ms, h1 and Openflow controller is 20 ms, and h3 and Openflow switch is forty ms. As per this approach, the h2 server contains different dynamic weights than various servers. For instance, requests 1,2,3,4 are handled via h2. The following h1 server takes dynamic weight much less than the h2 server but more than the h3 server. For instance, requests 5,6 are controlled via the h1 server. The server has a different delay than others managing with the least weight. For example, the h3 server handles seven for the most straightforward request. The weight that we allotted is dynamic and consistent with delay. Unless all the appeals aren't finished, this manner will resume.

Table.1 SDN setup description and parameters

Description	**Tools used**
Network Emulation	Mininet
SDN Controller Type	Distributed
SDN Controller	POX
Number of Servers	10
Testing Time	20-80 sec

VI. EXPERIMENT EVALUATION

We compare our proposed “dynamic weighted round-robin load balancing (DWRR) algorithm” with the previous round-robin (RR) load balancing model and Weighted round-robin

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6618
***Article Received**: 05 March 2023 **Revised**: 06 April 2023 **Accepted**: 26 April 2023*

(WRR) based on Average Response time (sec), Throughput (MB/sec), Transaction Rate (trans/sec), and Concurrency. The mininet tool is utilized to develop a load transmitted to the servers. The total load generated by the Mininet equals the number of simultaneous users produced by the number of submissions by individual users. For example, if we have 20 simultaneous users, every sending five recommendations, the total load or the number of requests equals 100. With the growth in concurrent users, the complete no of requests also improved

Table.2 Comparison between Average response time, Throughput, concurrency, and Transaction rate for various Load balancing methods

Load Balancing Algorithm	Average Response Time(sec)	Transaction rate(trans/sec)	Throughput (MB/sec)	Concurrency
Dynamic Weighted Round Robin	0.13	23.91	0.06	3.00
Weighted Round Robin	0.15	21.94	0.05	3.48
Round Robin	0.16	19.30	0.04	3.73

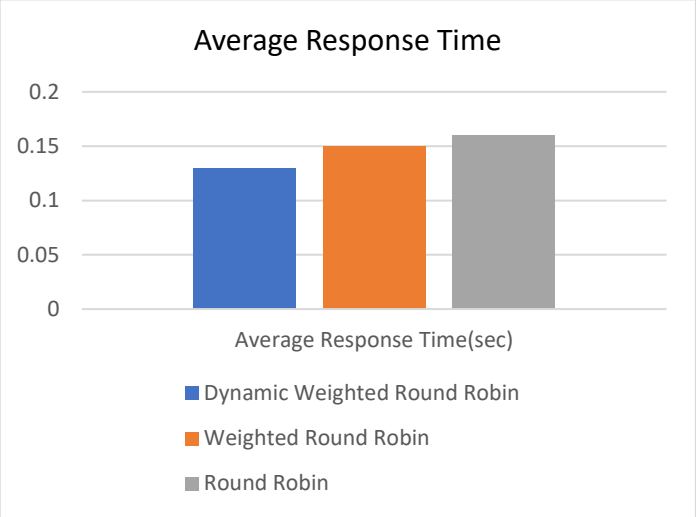

Fig.4 The Average Response time comparison among various load balancing algorithms

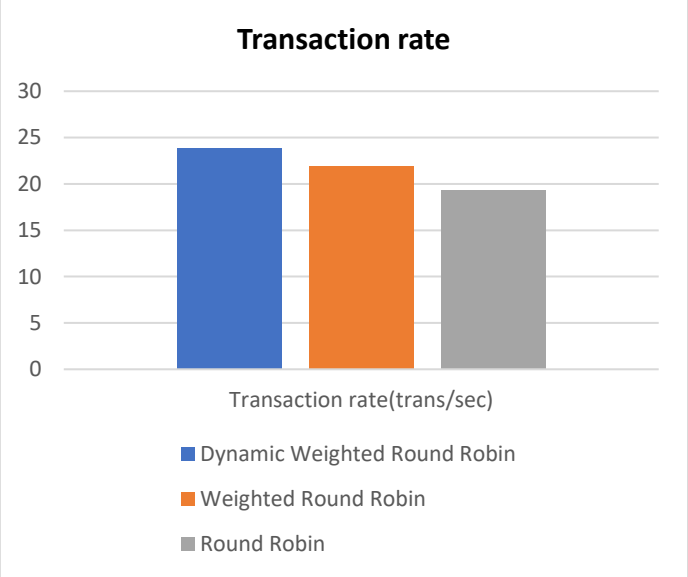

Fig.5 Comparison of Transaction rate between various load balancing algorithms

Figure 4 presents the results of analysing the average response time across special client domains for each load-balancing algorithm inside an experiment. Due to the more range of clients, the average response time may be higher throughout every algorithm. However, the quantity of difference varies in line with each rule set. Therefore, the total performance of our proposed algorithm shows a minimum average reaction time compared to others.

Figure 5 presents the results obtained by analysing the transaction rate computed with the help of different varieties of customers for every load-balancing algorithm in the test. Also, concurrent users are considered at the x-axis, and transaction rates (trans/sec) are considered at the y-axis. Looking at the outcomes, we can say that the transaction rate is higher in our DWRR model than in the current Round Robin and Weighted Round Robin algorithms.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 5
DOI: https://doi.org/10.17762/ijritcc.v11i5.6618
***Article Received**: 05 March 2023 **Revised**: 06 April 2023 **Accepted**: 26 April 2023*

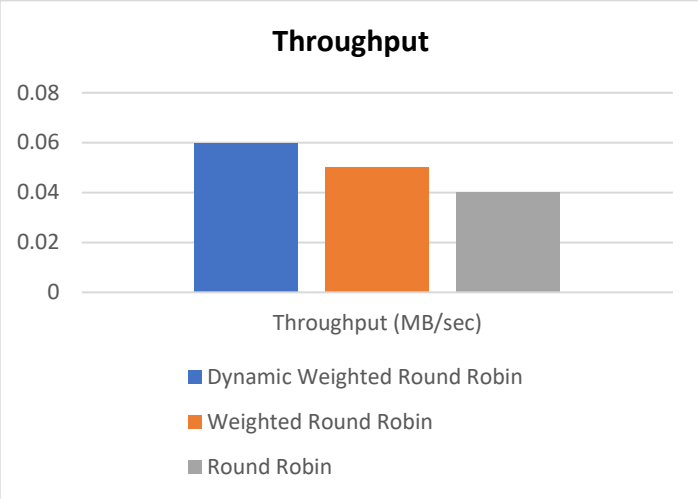

Fig.6 Comparison of Throughput between various load balancing algorithms

Figure 6 indicates the performance of several load-balancing algorithms. Using this graph evaluation, the proposed dynamic weighted round-robin (DWRR) load balancing provides better performance than previous load balancing techniques for WRR and RR.

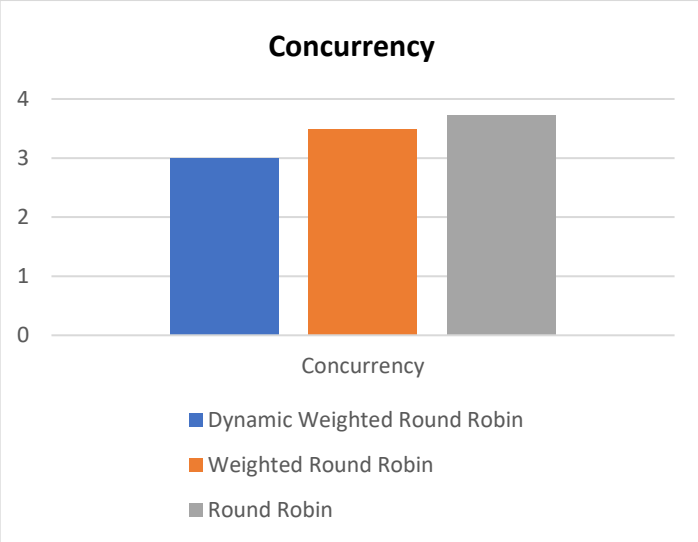

Fig.7 Comparison of Concurrency between various load-balancing algorithms

As illustrated in figure 7, the suggested method shows better concurrency when compared with the existing methods.

VII. CONCLUSION

The state-of-the-art shows that SDN load balancing is approvingly suggested due to its lightweight, stable, reliable, and scalability features. This paper has implemented a dynamic weighted round-robin (DWRR) algorithm and runs it on the top of the SDN controller. The investigations were conducted using the Mininet tool, and we allocated a load to the servers providing the delay. With the empirical results, we can say that the suggested dynamic weighted round-robin (DWRR) method is more acceptable in all performance parameters, like Average Response time, transaction rate, throughput, and concurrency, than the previous algorithms of round-robin and weighted round-robin.

REFERENCES

[1] Linn Arkar Soe, Thawda Win Su, 2019, "Server Load Balancing in Software Defined Networking ", pp.261-265.
[2] Diamond S and B Lin, 2020, "Network Optimization for Unified Packet and Circuit Switched Networks,", pp. 159-180.
[3] Jimson E and Hijazi M, 2020, "A Survey on the Architecture, Application, and Security of Software Defined Networking,", pp. 100289.
[4] Genetu Y. Basena, "Controller Load Balancing in Software-Defined Networking", pp.144-148, 2022.
[5] M Omran A, Zainal Abidin, 2022, "Software Defined Network based Load Balancing for Network Performance Evaluation", pp.117-124.
[6] Aymaz Samet, Ercüment Öztürk, 2019, "An Analysis of Load Balancing Strategies with Wireshark in Software Defined Networks", IEEE.
[7] Hidayat Taufik, Rahutomo Mahardiko, "Load Balancing Network by using Round Robin Algorithm: A Systematic Literature Review", DOI:10.15575/JOIN.V4I2.446.
[8] Li, Jun, et al, 2014, "An effective path load balancing mechanism based on SDN.", IEEE.
[9] Ali akbar neghab and rezaee1Ali, 2018, "Load Balancing Mechanisms in the SDN: A Systematic and Comprehensive Review of the Literature" IEEE.
[10] K Nisar and Khan S, 2020, "A Survey on the Architecture, Application, and Security of Software Defined Networking,", pp. 100289.
[11] S Prabakaran. and R Ramar, 2019, "Stateful Firewall-Enabled SDN with Distributed Controllers: A Network Performance Study,", pp. e4237.
[12] Wu Z and L.Wang, 2020, "Toward Building Video Multicast Tree With Congestion Avoidance Capability in SDN,", pp. 162-169.
[13] K Sharif, M Karim, 2020, "A Comprehensive Survey of Interface Protocols for Software Defined Networks," *JNCA*, vol. 156, pp.102563.
[14] Senthil P and Ramalakshmi R, 2020, "Flow Based Proactive Prediction Load Balancing in Stateful Firewall Enabled SDN with Distributed Controllers,", pp. 8337-8355.