International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-11 Issue-9, August 2022

Analysis of the Performance of Various Load Balancing Algorithms for SDN in Data Center Networks

Sujayanth K Vishwakarma, Pavithra H

Abstract: ***In order to fulfil the demands of modern IT, it is necessary to address the many limitations of current networking infrastructures. Software Defined Network (SDN) is the new networking technique to get around these constraints. Existing Network Infrastructures use static switches, which results in a substandard utilization of the network resources, which is one of the main problems. The main issues with the present networking trend are the slow response and delays. In this research different load-balancing algorithms for Software Defined Network based Data Center network is implemented to overcome mentioned issues. The POX controller is used in the implementation and the Mininet emulator is used to simulate the network functionality. The load balancing algorithms program is written in the Python language, which is also used for creation of the network topology. Finally, Siege HTTP load testing tool is used to test network performance. The testing is focused on Quality of Service pa- rameters transaction rate, average response time and throughput. The results show that the Weighted algorithm outperforms all other considered algorithms.***

Keywords: Load Balancing Algorithms, Software Defined Net-Working, Data Center Network, POX Controller

I. INTRODUCTION

In the past two decades, there have been sufficiently great changes in the network requirements, exponential growth in traffic, and the need for more challenging end-to-end objectives. The primary hosting infrastructures for Internet applications and services are data centers (i.e., social net- works, Internet banking, and multimedia content). In such data center networks, traditional load balancing algorithms use customized hardware devices to divide network traffic among various server copies. Although this method generally produces excellent performance, it is expensive and has a rigid configuration that cannot be dynamically changed in response to the status of the network or other information. Nowadays, software defined networking is used to move alot of data. There is a lot of traffic, which has significantly grown. It is vital to address this kind of congestion issue so that networks can be made more effective.

Manuscript received on 30 July 2022 | Revised Manuscript received on 04 August 2022 | Manuscript Accepted on 15 August 2022 | Manuscript published on 30 August 2022.

***** Correspondence Author

Sujayanth K Vishwakarma*, Department of Computer Science and Engineering, RV College of Engineering Bengaluru (Karnataka), India. E-mail: sujayanthkv.scn20@rvce.edu.in

Pavithra H, Department of Computer Science and Engineering, RV College of Engineering Bengaluru (Karnataka), India. E-mail: pavithrah@rvce.edu.in

The Decoupling of the data and control planes is made possible by software defined networking which enables the network administrators use an external SDN controller to operate their network. It makes use of a unique kind of controller with the ability to manipulate the network's forwarding behavior. The SDN- depicted network architecture is very versatile, and the network administrator can benefit from the programmability of SDN enabled switches.

A. Data Centre Network

The Data Center Network (DCN), which connects all the data center resources, is crucial to the operation of a datacenter. To meet the expanding expectations of Software De- fined Networks, DCNs must be scalable and effective enough to connect tens of thousands or even hundreds of thousands of servers (SDN) which is one of the main problems with data centers, regardless of their many configurations whether they are physical data centers or virtual data centers. For improved performance metrics and traffic load balancing, an OpenFlow Data Center network architecture based on Software Defined Networks is deployed. Network load balancing must be utilized to expand the amount of available bandwidth, maximize throughput, and add redundancy. The capacity to balance traffic across several Internet connections is known as network load balancing. By distributing the available band-width consumed by every LAN user on numerous connections, the usage of total amount of available bandwidth can be improved.

B. Software Defined Network

The SDN architecture is made up of networking element like switches and routers in the data plane. These underlying objects are more conventional forwarding components devoid of any autonomous decision-making software. Using flow rules, the forwarding components are given instructions on what to do with the packets that have been received by the southbound interfaces (eg, OpenFlow). A control plane com- posed of control logic found in controllers and applications is responsible for controlling the items of the data plane. Application developers can use the northbound interface using an API provided by SDN controller. A north- bound interface

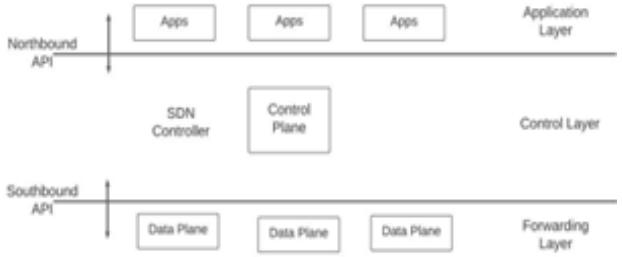

Fig. 1. SDN Architecture

conceptualize the low level instruction by a southbound interface while planning forwarding devices. From Fig. 1, it is clear that SDN is a bottom-up, three layer design that includesthe application layer,the infrastructure layer and the control layer. An incomprehensible and sheer programming interface between the SDN controller and network devices, enables the separation between the control plane and the data plane to be seen. By breaking down the network control problem into digestible parts, this separation of the two planes is crucial for creating networking abstractions and fostering network expansion and revolution. Control plane makes decisions about where and how to direct traffic as well as managing network topology. It has total control over the componentsof the forwarding layer. Applications in the application layer use the capabilities offered by northbound interface to carry out network authorities. Applications for load balancing and routing are both feasible. The most crucial component in this situation is a centralized controller, which will build network configuration based on rules and regulations of the network operator.

II. LITERATURE REVIEW

The weighted response time algorithm is suggested in the study by authors in [1]. The client's request is routed to the server with the lowest weight to response time value under the weighted response time algorithm. According to the study's findings, networks that use weighted response time algorithms have shorter reaction times and higher through- put. To increase throughput, the authors of [2] describe an innovative method that incorporates a load balancer into SDN. To create a network environment, mininet software is utilised. Additionally, the Python programming language and the floodlight controller were employed in the implementation. Using the TCP and UDP protocols, two scenarios which are one before and one after the load balancer were tested and analysed. The results indicate that employing the load bal- ancer with SDN significantly improves network performance. Using an OpenFlow switch, the authors of [3] implement the Random and Round Robin load balancing techniques. When they are compared with random scheduling algorithm, the round robin technique's results for mininet simulations have produced lower packet loss and higher throughput. The load balancing technique for cloud platforms where Virtual Network Functions are made available as a service in the main topic of [4]. They have employed Weighted Round Robin and Least Connection algorithms as the first stage and have carried out an experimental investigation to assess the effectiveness of the two methods. The outcomes demonstrate that in terms of workload allocation and average response time, the weighted round robin algorithm performs better. In [5], authors use a POX controller to develop and evaluate the performance of the Round Robin load balancing approach in a virtual network emulator known as Mininet. This project uses software to accomplish load balancing, in contrast to the conventional hardware-based networking approach. In [6][10] a brief analysis of some load balancing algorithms is provided for services in a software defined network. The authors discuss the various types of load balancing algorithms which can help in better resource utilization and linear service delivery across multiple clients in an SDN environment. On the basis ofvarious parameters, the different load balancing algorithms are contrasted. The performance of two load balancing methods, such as the traffic pattern-based load balancing algorithm and the flow-based load balancing algorithm, with distributed controllers' architecture is compared in [7]. The performance of the load balancing algorithms is evaluated using the Siege HTTP Load Balancing test tool. The outcome demonstrates that the flow-based load balancing algorithm reduces reaction time and increases transaction rate.

III. ALGORITHMS APPLIED TO SDN

In this section, the different algorithm used in this research are presented.

1) Random Load Balancing Algorithm:

Algorithm 1:
Input:- Requesting the available servers list.
Output:- Distribution of requests to servers.
Start
Get the available servers list
When a new request arrives
From the group of available servers, choose a random serverby using
selected server = random. Choice (Available Servers)
Send the requests to the random selected server
End

The above algorithm depicts the application of the Random Load Balancing Algorithm for efficient scheduling of requests from clients and also allocating servers to the customers. The Load Balancer selects one server node at random from a list of available servers to handle client requests.

2) Round Robin Load Balancing Algorithm:

Algorithm 2:
Input: - Requesting the available servers list.
Output: - Efficient distribution of requests to servers.
Start
Get the available servers list
When a new request arrives
From the group of available servers, choose a server by usinglast Serverindex=(self. Last Serverindex +1) / len (Available Servers)
Selected server = Available Servers [lastServerindex].Each server receives requests in a cyclical pattern.
End

Retrieval Number: 100.1/ijitee.G92440811922
DOI: 10.35940/ijitee.G9244.0811922
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The above algorithm depicts the application of the Round Robin Load Balancing Algorithm for scheduling client resource requests and allocating server resources to clients. The most used algorithm is Round Robin since it is simple to use. Each server in this scenario receives the client request ina circular manner. The round-robin distribution of the request among many servers is done by the load balancer.

3) Weighted Round Robin Load Balancing Algorithm:

Algorithm 3:
Input: - Requesting the available servers list.
Output: Efficient distribution of requests to servers.Start
Get the available servers list
When a new request arrives
From the group of available servers, choose a server by using lastServerindex =(lastServerindex +1)/ len(Available Servers)lastServerindex = lastServerindex + weight
Selected server = Available Servers [lastServerindex]Each server receives requests in a cyclical pattern
End

The above algorithm depicts the application for allocating the resource requests from clients and then distributing servers to clients using Weighted Round Robin Load Balancing Algorithm. In this approach we can differentiate between heterogeneous servers. For Example, if Server 1 is havinghigher weight than Server 2, the algorithm will assign more requests to the server with a higher capability of handling load. Here the specific weight is assigned to each server. The Weighted Round Robin is like the Round Robin in sense that the requests are allocated to the servers in cyclical manner only, however, more requests will be sent to the server with higher weights.

4) Least Connection Load Balancing Algorithm:

Algorithm 4:
Input:- Requesting the available servers list.
Output:- distribution of requests to servers.
Start
Get the available servers list
When a new request arrives
Pick one server with minimum connection from the availableserver list
Assign that request to the selected server.
End

The algorithm 4 depicts the application for allocating resourcerequest from clients and assigning different servers to clients using Least Connection Load Balancing Algorithm. Considera situation where, despite the fact that two servers in a cluster have identical specifications, one server can yet become overloaded far more quickly than the other. This can be the result of clients connecting to Server 2 staying connected for a lot longer than clients connecting to Server 1. As a result, Server 2's overall number of connections may increase, but Server 1's connections, which are affected by clients connecting and disconnecting more quickly, would essentially stay the same. As a result, Server 2's resources can deplete more quickly. The Least Connections algorithm fits these circumstances better. The number of active connections that each server has is taken into account by this algorithm. The load balancer will seek to identify the server that has thefewest connections when a client tries to connect, and it will then assign the new connection to that server.

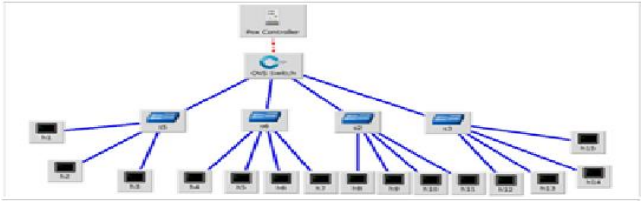

Fig. 2. The simulated 3-level tree Topology

5) Equal Load Balancing Algorithm:

Algorithm 5:
Input:- Requesting the available servers list.
Output: distribution of requests to servers.
Start
Get the available servers list
When a new request arrives
From the list of available servers, choose a server by using lastServerindex = (lastServerindex +1)/ len(Available Servers)lastServerindex = lastServerindex + equal load
Selected server = Available Servers [lastServerindex]
Each server receives requests in a cyclical pattern
End

The above algorithm depicts the application for allocating the resource request from clients and assigning different serverto the clients using Equal Load Balancing Algorithm. In this approach we can differentiate between heterogeneous servers. This algorithm is similar to the Weighted Round Robin in a sense that the equal weight is assigned to each server. The controller assigns the equal number of requests to all the servers in cyclic manner.

IV. IMPLEMENTATION

The objective of this research is the implementation of different load balancing algorithms in the data center network and analyze the network in term of throughput, transaction rate and average response time. The load balancing experi- ment is carried out using the Ubuntu operating system. The network topology is generated using a Mininet and the SDN POX controller [8][9]. The Network topology and the algorithm were developed using the Python programming language. The simulated 3-level tree topology is shown is figure 2 consists of 2 types of switches, Legacy switches at distribution level and OVS Switch which is connected to the Pox Controller.

In order to create the data center topology, Mininet is linkedto the Pox controller via the port 6633. The "Ping all" command is then used to determine whether all hosts in the configured network can be reached. Xming server is used to access the command prompt of each host of the topology. The hosts running the Simple HTTP Server application are depicted as servers and other hosts are depicted as clients. The clients send their request to the SDN Controller's virtual switch IP and Controller distributes the received requests to the different servers based on the Load Balancing techniques.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Curl command is used to generate the GET request from the client. Siege an HTTP Load Testing tool which sends number of requests from concurrent hosts is used to test network performance. The QoS parameters Average Response time, transaction rate and throughput are recorded for the analysis.

V. RESULTS AND ANALYSIS

The outcomes of the simulation of load balancing algorithms in SDN using the Mininet emulator are discussedin this section. The following performance metrics average response time, throughput, and transaction rate are used to compare the effectiveness of various load balancing strategies. Average Response Time: It is calculated by dividing the total time taken to react during the chosen period by the total number of responses during the chosen period. Throughput: It is defined as the number of data packetsthat pass over a communication channel in a certain timeperiod. Transaction rate: The transaction rate is defined as the total volume of flows that the controllers process each second. The Response time and the transaction rate are inversely related. The Siege HTTP Load testing tool is used to test the performance of load balancing algorithm. The Average Re- sponse Time, Throughput and Transaction rate parameters values obtained from the tool are recorded and tabulated as shown in the Table 1. The Comparison of average response time for different load balancing algorithms is shown in fig 3. It is observed that the weighted round robin algorithm has lowest average response time when compared to other algorithms. The Equal load algorithm and Round Robin algorithm has lesser average response time value than Least Connection algorithm. And the Random algorithm has the highest load balancing algorithm than other considered algorithm. The fig 4 shows the Comparison of Transaction Rate for different load balancing algorithms. It is observed that the weighted round robin algorithm has the highest transaction rate.

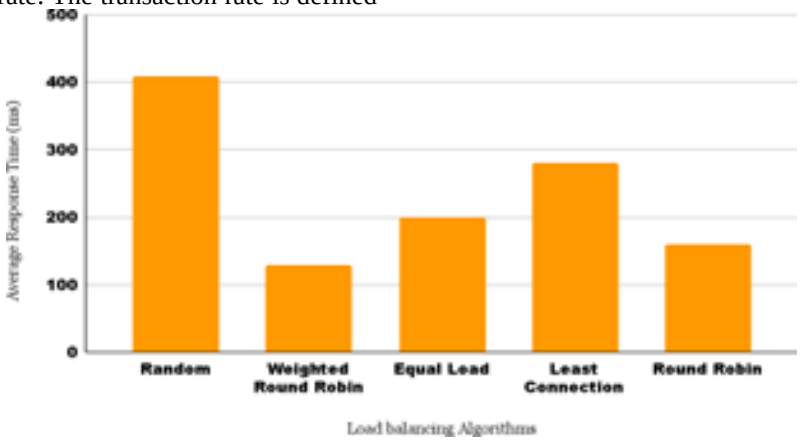

Fig. 3. Comparison of Average Response Time for different load balancing algorithms

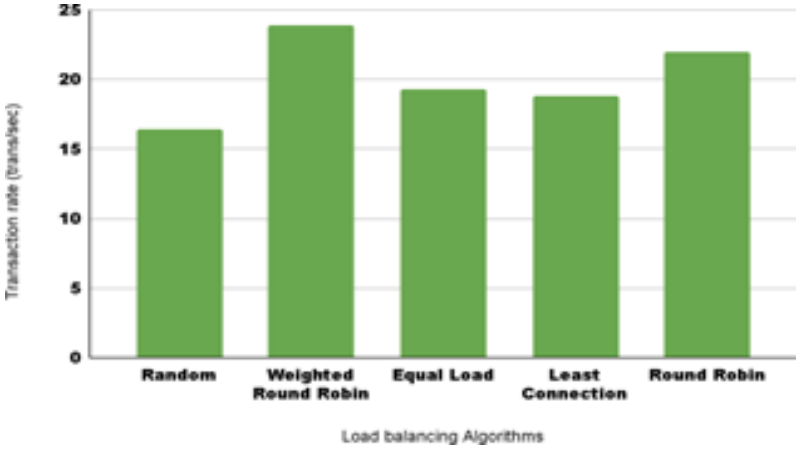

Fig. 4. Comparison of Transaction Rate for different load balancing algorithms

when compared to other algorithms. The Equal load algorithmand Round Robin algorithm has more transaction rate value than Least Connection algorithm. And the Random algorithm has the lowest load balancing algorithm than other considered algorithm. The Comparison of Throughput for different load balancing algorithms is shown in fig 5. It is observed that weightedround robin and equal load balancing algorithms have the highest throughput values. The round robin algorithm has the throughput value lesser than weighted round robin and equal load balancing algorithms. The random algorithm and least connection algorithm have the lowest throughput values. The fig 6 shows the graph of Average response time against different file size. The python script was written to generate 200 GET requests from the client hosts requesting various sizes of data files. The Client program also logs the response time for each GET request. This response time is used for plotting graphs and comparing the response time for various load balancing algorithms.

Retrieval Number: 100.1/ijitee.G92440811922
DOI: 10.35940/ijitee.G9244.0811922
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Table I. Comparison of Average Response Time, Transaction Rate and Concurrency for Different Load Balancing Algorithms

Load Balancing Algorithm	Average Response Time (sec)	Transaction rate (trans/sec)	Throughput (MB/sec)	Concurrency
Weighted Round Robin	0.13	23.91	0.06	3.00
Round Robin	0.16	21.94	0.05	3.48
Equal Load	0.2	19.30	0.06	3.73
Least connection	0.28	18.82	0.04	5.29
Random	0.41	16.46	0.04	6.71

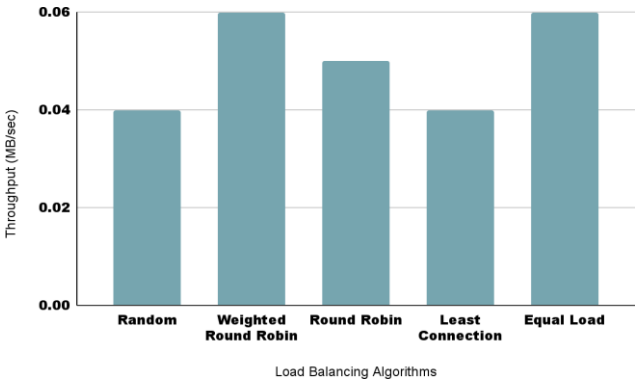

Fig. 5. Comparison of Throughput for different load balancing algorithms

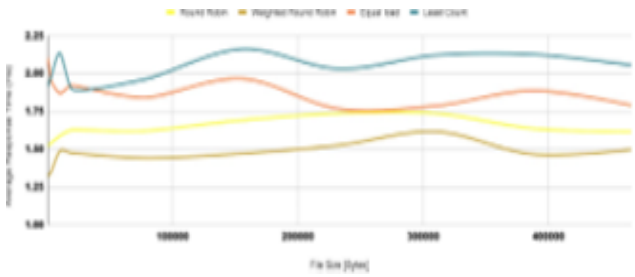

Fig. 6. Average Response Time vs File Size of different load balancing

VI. CONCLUSION

Traditional network management presents more challenges, therefore SDN is an emergent architecture approach that addresses these problems. A very large scale network switches can be regulated with the help of the open and adaptable SDN paradigm. The forwarding information base's centralization enables deterministic end-to-end routing calculations for each flow across topology. SDN has gained a lot of traction, and as it moves closer to a time of data networks which requires a highly flexible load balancing system. In this project, the SDN based POX controller is used to implement the load balancing strategy. The Mininet software is utilised to replicate the SDN based Data Center Network. The Pox controller replicates a number of load balancing strategies, including the least connections approach, weighted round robin, random, and round robin. The Siege HTTP load testing programme is used to evaluate the effectiveness of load balancing methods. The outcomes demonstrate that weighted round robin outperforms every other method presented. The assumption is that the channel bandwidth will remain constant and be error-free.

FUTURE WORK

In this paper, while implementing load balancing algo- rithms it is assumed that all bandwidth have same capacity and they are error free. In future it can be considered as challenge to improve load balancing algorithm. Average re- sponse time, Transaction rate and throughput was consideredas performance metric, but there can be other parameters like bandwidth/channel utilization, Server utilization,..etc. Here, the algorithms are implemented in Mininet emulator, in future it can be deployed in real time scenario and analyze the behavior.

REFERENCES

1. H. Nurwasito and R. Rahmawati, "Weighted Response Time Algorithm for Web Server Load Balancing in Software Defined Network," 2021 3rd International Conference on Electronics Representation and Algorithm (ICERA), 2021, pp. 143- 148, doi: 10.1109/ICERA53111.2021.9538792. [CrossRef]
2. Y. A. H. Omer, A. B. A. Mustafa and A. G. Abdalla, "Performance Analysis of Round Robin Load Balancing in SDN," 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), 2021, pp. 1-5, doi: 10.1109/ICCCEEE49695.2021.9429662. [CrossRef]
3. K. A. Jadhav, M. M. Mulla and D. G. Narayan, "An Effi- cient Load Balancing Mechanism in Software Defined Networks," 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), 2020, pp. 116- 122, doi: 10.1109/CICN49253.2020.9242601. [CrossRef]
4. M. D. B. P. Senevirathne, W. H. Rankothge, N. D. U. Gamage, M. M. T. R. Ariyawansa, H. G. H. Dewwiman and S. A. A. Suhail, "An Expermental Study on Load Balancing for a Software Defined Network based Virtualized Network Functions Platform," 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communica- tion Conference (IEMCON), 2021, pp. 0092-0097, doi: 10.1109/IEM- CON53756.2021.9623144. [CrossRef]
5. K. Rishabh, K. Angadi, K. Chegu, D. S. Harikrishna and S. Ramya, "Analysis of Load Balancing Algorithm in Software Defined Network- ing," 2017 2nd International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS), 2017, pp. 1-4, doi: 10.1109/CSITSS.2017.8447852. [CrossRef]
6. Anish Ghosh, Mrs. T. Manoranjitham "A study on load balancing techniques in SDN" International Journal of Engineering and Technology(IJET),vol 7. pp. 174-177 2018 [CrossRef]
7. Prabakaran, Senthil Ramar, Ramalakshmi. (2021). Software Defined Net- work: Load Balancing Algorithm Design and Analysis. The International Arab Journal of Information Technology. 18. 10.34028/iajit/18/3/7. [CrossRef]
8. Smriti Bhandarkar, Kotla Amjath Khan "Load Balancing in Software-defined Network (SDN) Based on Traffic Volume" Advances in Computer Science and Information Technology (ACSIT) 2015 Volume 2, Number7 pp 72-75
9. L. Liu et al., "An SDN-based Hybrid Strategy for Load Balancing in Data Center Networks," 2019 IEEE Symposium on Computers and Communi- cations (ISCC), 2019, pp. 1-6, doi: 10.1109/ISCC47284.2019.8969673. [CrossRef]

10. C. Hu, Z. Huang and Z. Lin, ”An information transport dynamic load balancing policy based on software defined mobile networks,” 2018 International Conference on Information and Computer Technologies (ICICT), 2018, pp. 83-86, doi: 10.1109/INFOCT.2018.8356845 [CrossRef]

AUTHOR PROFILE

Sujayanth K Vishwakarma, currently pursuing MTech in Computer Network Engineering at R V College of Engineering (RVCE) Bengaluru, Karnataka. Sujayanth K Vishwakarma has completed his B.Tech from Nitte Meenakshi Institute of Technolgy, Bengaluru, Karnataka. Sujayanth K Vishwakarma has done his internship in Firmware Development of Crestron Software Development India Pvt Ltd. The domain Sujayanth K Vishwakarma is currently working under Firmware team wherein it requires understanding the frame work, resolving bugs and implementing new designs. Area of Interest of Sujayanth K Vishwakarma are Internet of Things, Software Defined Networks and Software Engineering. Sujayanth K Vishwakarma has excellent communication skills, learns quickly, and likes applying new ideas to expand his knowledge. Mail id- (sujayanthkv.scn20@rvce.edu.in)

Prof. Pavithra H, is an Assistant professor in Computer Science Engineering Department at the R V College of Engineering, Bengaluru, Karnataka. Prof. Pavithra H has of 10 years of teaching Experience. Area of Interest of Prof. Pavithra H are Software Defined Networks Cloud Computing, Software Engineering. Prof. Pavithra is highly motivated, supportive professor who always encourages students in achieving their goals. Prof. Pavithra has guided many UG and PG students to complete their Research work. Prof. Pavithra has published her research work in many International Journals, International Conferences and National Conference. Prof. Pavithra H has also worked for Samsung with Certificate of Excellence in the consultancy project for the duration of May 2021 to Jan 2022. Mail id- (pavithrah@rvce.edu.in)