International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Reliable Load Balanced Routing Policy using Bayesian Optimization for SDN-IOT Applications

[1]Dr. G. Srinivasan, [2]G. Seshadri Sekhar, [3]P. Suresh

Submitted: 25/01/2024 **Revised**: 03/03/2024 **Accepted**: 11/03/2024

Abstract: In this paper, we propose to design the reliable load balancing routing policy using Bayesian optimization for Software Defined Network (SDN) and Internet of Things (IoT) heterogeneous applications. The main aim of this approach is to determine the best path between the IoT gateway and the target IoT server while satisfying the servers' resource limits as well as Quality of Service (QoS) requirements of traffic like lower delay and higher throughput. For delay-aware traffic, the controller selects the forwarder nodes with lower delay and lower packet queue length. For bandwidth-aware traffic, the link quality, link capacity and noise parameters are estimated and the forwarder nodes with high link capacity and quality with lower noise will be selected. For normal traffic, shortest path will be selected. Then by applying Bayesian optimization, the optimum load balanced path is selected for each type of traffic. By simulation results, it has been shown that the proposed technique enhances the network throughput and minimizes the end-to-end delay.

Keywords: *Software Defined Network (SDN), Internet of Things (IoT), Load balancing, Bayesian optimization. Routing policy*

1. Introduction

Big data, cloud computing, and Internet of Things (IoT) have all aided to significantly boost traffic on conventional networks. The control and data planes of these networks are closely coupled, making them difficult to manage and expensive to operate and maintain. As a cutting-edge and creative networking paradigm, SDN offers programmability with simple network control and management that is required for the upcoming Internet. SDN provides access to the fundamental bearers and the network. In order to meet client needs at a reduced cost, numerous routes and superior approaches can share the carriers thanks to SDN's virtualization capability [1]. The IoT is a cutting-edge idea that enables varied networks to be used by flexible ecosystems. In particular, an IoT framework is constructed by connecting wireless networking devices, wireless sensors, actuators, and RFID-equipped devices to the internet. [2]

The availability and performance of the network are impacted by traffic congestion since SDN architecture is centralised. The SDN control plane has a limited capacity for scaling and permits a high volume of requests that choke the network. SDN gives the network the ability to create the best routing and, consequently, flow forwarding policies, allowing for network-level traffic load balancing [3]. To distribute numerous client requests to resources through multiple servers, network load balancing is used. A high-density SDN employing load balancing was able to produce an effective network performance. [4]. Load balancing divides inbound network packets that enter the network and outbound network packets across the network through several controllers in the older network [5]. An efficient load balancer uses the least amount of electricity possible while optimising network factors including latency, resource utilisation, throughput, and fault tolerance. A dedicated server typically performs load balancing in traditional networks [6] [7].

However, one of the main problems with load balancing is how difficult it is to manage a big network with just one server. The load balancer, on the other hand, takes a long time and has a rigorous maintenance procedure. Although there are several controllers set aside for backup, the scalability and availability difficulties with this technique need to be taken into account. In the event that a controller fails, a different controller that is already in the queue is used to replace the failed controller. [8].

1.1 Motivation and Objectives

The main objectives of this work are

- ✓ Develop a load-balanced routing through SDN for each flow in heterogeneous IoT applications
- ✓ Develop a reliability policy which checks the reliability for better load balancing
- ✓ Select the forwarding nodes based on link quality, link capacity, traffic load and delay.

[1]*Alva's Institute of Engineering and Technology, Solapur - Mangalore Highway, Shobhavana Campus MIJAR, Moodbidri, Mangaluru, Karnataka 574225*
srinivasanphd123@gmail.com
[2]*Sri Venkateswara college of engineering, Tirupati - 517507, AndhraPradesh*
seshadrisekhar.g@svce.edu.in
[3]*Sri Venkateswara college of engineering, Tirupati - 517507, AndhraPradesh ,*
suresh.pb@svcolleges.edu.in

To meet these objectives, we propose to design a reliable and load balancing routing policy using Bayesian optimization for SDN-IoT applications.

2. Related Works

In order to increase reliability and raise QoS performance, Mohammad Riyaz Belgaum et al [1] have presented a framework to take into account in PSO and ACO algorithms with the help of direct and indirect data from the switches.

A Multiple Distributed Controller Load Balancing (MDCLB) technique has been put out by Himanshi Babbar et al [2] to eliminate the load unbalance in the control and forwarding planes. The goal of this variable load balancing is to keep the traffic load balanced while minimising network latency for the controllers on the control plane. The server's specific switch will migrate packets from intra cluster to inter cluster if the threshold value exceeds the load in order to balance the load. The iperf test tool and mininet emulator are used in its Python implementation.

In a hybrid IP/SDN network, Jaime Galán-Jiménez et al. [3] have presented the Hybrid Spreading Load Algorithm (HSLA) heuristic, which simultaneously addresses the issues of traffic balancing by minimizing link utilisation and power utilization. To determine which nodes should be converted from IP to SDN, HSLA is assessed across topologies of various sizes using various techniques. These analyses show that HSLA outperforms competing strategies that only focus on one of the objectives.

A unique SDN architecture that includes topology, BS and controller discovery, link, and virtual routing has been presented by Elham Hajian et al. [9] with the goal of decreasing load imbalance and extending the lifetime. For this, a novel method for load-balanced routing via SDN and virtualization is suggested. It directly monitors the link load information and the network running status. The communication of network status and other pertinent information is decreased by this implementation. The simulation results demonstrate how the proposed method distributes demand across the network and are characterised by energy balancing, which also increases network lifetime.

Ziran Min et al. [10] employed multiple M/M/1 queues to optimise the objectives in dynamic IoT scenarios. They proposed genetic algorithm, a simulated annealing algorithm, and a modified greedy algorithm, to minimize the queueing and processing delays of the requests and balancing the controller loads.

In order to simultaneously balance traffic between IoT servers and satisfy the QoS needs of various IoT services, Ahmadreza Montazerolghaem et al. [11] have presented a unique framework based on SDN. The issue is initially framed as an NP-hard Integer Linear Programming (ILP) paradigm. It is therefore suggested to use a proactive and anticipatory heuristic approach depending on fuzzy logic and time-series analysis. The suggested architecture is then put into practise using an actual testbed that includes an OpenvSwitch, a Floodlight controller, and Kaa servers. Several tests are run under various conditions to assess performance.

To improve the efficiency of mobile devices communicating across a Wi-Fi network, Hind Sounni et al. [12] have presented a new load balancing method based on the SDN. The implementation of their algorithm is made easier by the use of the SDN. Through simulation with mininet-WiFi, they put their suggested algorithm into practise and assessed it. The outcomes show that the performance of the devices and network load balancing provided by our suggested strategy are both enhanced.

3. Proposed Methodology

3.1 Overview

This method takes into account the IoT reference model, which is made up of three layers that communicate with one another via a number of interfaces. Our aim is to determine the optimal path from the IoT gateway to the target IoT server that meets both the traffic's quality of service (QoS) criteria, such as reduced latency and increased throughput, and the servers' resource limitations. The controller chooses the forwarder nodes with the smallest latency and shortest packet queue length for delay-aware traffic. The link quality, link capacity, and noise characteristics are calculated for bandwidth-aware traffic, and the forwarder nodes with high link capacity and quality and lower noise will be chosen. In cases with typical traffic, the shortest way will be chosen.

3.2 System Model

Three layers make up the IoT reference model, and they communicate with one another via various interfaces. Sensing, metering, and gateways are all part of the IoT device layer. These IoT gateways assemble traffic. As a result, by operating the SDN controller, traffic is attained through OpenFlow switches. It is flowing into the IoT servers' layers. The controller contains a broad overview of the network's traffic, resources, and IoT devices.

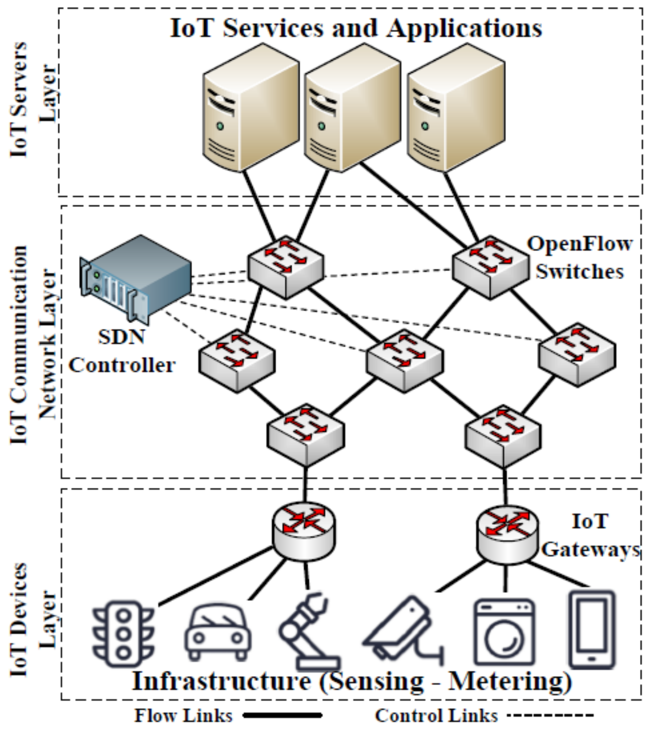

Fig 1 System Model

3.3 Estimation of Metrics

3.3.1 Delay

In the context of distance parameters, delay (D) is used. The forward node is chosen based on how many neighbour nodes hops there are to the base station. For forwarding, the optimal choice is a node with the shortest distance.

3.3.2 Queue Length

The queue length (QL_i) of each node is assessed using the following equation:

$$QL_i^{t+1} = \min\{[QL_i^t + P_{in,i}^t - P_{out,i}^t], Z_i\} \qquad (1)$$

where QL_i^t = ith node’s queue length at time t

$P_{in,i}^t$ *and* $P_{out,i}^t$ = number of ith node’s input and output packets at time t

Z_i = ith node’s buffer size.

$P_{out,i}^t$ is computed as

$$P_{out,i}^t = \min\left\{QL_i^t, \frac{P_{max}}{P_{min}}, \frac{BW_i}{BW_p}\right\} \qquad (2)$$

Where, P_{max} and P_{min} are the maximum and lowest number of packets that node i can send, respectively, depending on its energy; BW_i stands for the node's bandwidth, and BW_p for the packet's bit count.

3.3.3 Link Quality

It is possible to balance the load among nodes by estimating the link quality (LQ) at each node. Both the noise rate and the interference rate (IR) can be used to quantify the quality of the link.

Transmissions on one wireless link may interfere with another as long as they are within each other's range of interference and share the same radio link. Noise may also have an impact on this. The achievable data rate is therefore greatly influenced by transmission interference and noise.

3.3.4 Link Capacity (LC)

The data rate or maximum throughput through a network is referred to as link capacity. Each node measures the throughput of sending a packet to keep track of channel usage. Therefore, while assessing the available bandwidth at the MAC layer, we consider the effects of both congestion and physical issues like fading and interference. It should be emphasised that the successful MAC layer broadcasts are the sole way to determine the bandwidth that is available.

3.3.5 Noise (N)

The noise rate is computed by taking into account the degrees of the first and final nodes in each link.

The degree of link nodes has a direct related to the measurement of noise and interference, which is also referred to as the rate of both.

$I_{i,j} + N_{i,j} \sim \partial(a + b)$

(3)

3.3.6 Objective Function

Objective functions are formed separately for each type of traffic.

The objective function of delay-aware traffic is as follows:

$$OF_1 = v_1 \frac{D}{\max D_i} + v_2 \frac{QL}{\max QLi} \left(\sum_{i=0}^{3} v_i = 1\right) \quad (7)$$

(4)

v_1 and v_2 are weight factors

The objective function of bandwidth-aware traffic is as follows:

$$OF_2 = v_1 \frac{N}{\max N_i} + v_2 \frac{maxLQ}{LQi} + v_3 \frac{maxLC}{LCi} \left(\sum_{i=0}^{3} v_i = 1\right)$$

(5)

v_1, v_2 and v_3 are weight factors

3.4 Types of IoT Traffic

The three categories of IoT traffic in this network are given below:

- Bandwidth-aware
- Delay aware
- Normal

The delay aware traffic chooses least delay path and the bandwidth-aware traffic chooses the high bandwidth path.

The controller chooses the forwarder nodes for delay-aware traffic that have a lower delay [9] and a shorter packet queue. The link quality, link capacity, and noise characteristics are calculated for bandwidth-aware traffic, and the forwarder nodes with high link capacity and quality and lower noise will be chosen.

For normal traffic, shortest path will be selected.

3.5 Bayesian optimization

Bayesian optimisation can be used to successfully resolve global optimisation problems. Global optimisation deals with the issue of choosing an input that minimises or maximises the cost of a certain objective function. An objective function F(λ)'s probabilistic representation is called the surrogate function, which is created by Bayesian optimisation and then successfully searched with an acquisition function. The conditional probability of an occurrence C specified additional occurrence D, $P(C \mid D)$, is determined using

$$P(C \mid D) = P(D \mid C) * P(D)/P(C)$$

(6)

From the conditional probability equation, the marginal probability P(D) is removed for optimization problems.

$$P(C \mid D) = P(D \mid C) * P(C)$$

(7)

ρ_{llh} →likelihood (reverse conditional probability)

ρ_{post}→ posterior (conditional probability)

ρ_{pr}→ prior probability (marginal probability).

Consequently, the Bayes theorem may now be expressed as

$$\rho_{post} = \rho_{llh} * \rho_{pr}$$

(8)

Future search space sampling is guided by the posterior probability, a function that roughly be similar to the objective function. .

The objective function evaluation $F(\lambda_i)$ is first determined using arbitrary exploration space samples $(\lambda_1, \lambda_2, \ldots, \lambda)$. Sequentially gathering the samples and their evaluations yields a set $Z = \{\lambda_i, F(\lambda_i), \ldots \lambda_n, F(\lambda_n)\}$.

The prior and likelihood function are defined in terms of the set Z. The possibility of witnessing the information specified the goal function is the definition of the likelihood function.

$$P(F \mid Z) = P(Z \mid F) * P(Z)$$

(9)

The posterior is updated after the prior and likelihood have been assessed. In order to choose the following sample, x_n, which is given in Equation, the acquisition function, ζ, is then optimised over the Gaussian function (13).

$$\lambda_n = \text{argmax}_\lambda\ \zeta(\lambda \mid Z_{1:n-1})$$

(10)

The Expected Improvement algorithm is used to implement the acquisition function.

$$\zeta(\lambda) = \mathbb{E}[m\ \ (F(\lambda) - F(\lambda^+), 0)]$$

(11)

where $\mathbb{E}$ = expectation operator,

$F(\lambda^+)$ = objective function value of best sample

λ^+ = its location in the search space.

The objective function is then used to evaluate the chosen sample, and the cycle is repeated until the objective function reaches its minimal value. In the event that the observed objective exceeds a certain value, a halting condition is used.

3.6 Process Flow

The steps involved in this process are as follows:

1. Using the (IR), S (gateway node) broadcasts an RREQ packet to D (server) whenever it wishes to send a data packet to D. RREQ packet is received by all nodes within the transmission range.
2. N_i upon receiving the RREQ verifies the C_1 and C_2. (explained in section 3.3)
3. For C1, the controller selects the forwarder nodes with lower delay and lower packet queue length.
4. For C_2, the link quality, link capacity and noise parameters are estimated and the forwarder nodes with high link capacity and quality with lower noise will be selected.
5. For normal traffic, shortest path will be selected.
6. Optimum paths are established for each type of traffic based on the objective functions, by applying Bayesian optimization.

As a result, the optimal route from the IoT gateway to the intended IoT server is found, one that complies with both the traffic's QoS criteria (lower latency and better throughput) and the servers' resource limitations.

4. Simulation Results

The proposed Reliable Load Balancing Routing Policy (RLBRP) is implemented in OpenSwitch 1.3 of NS3. Figure 2 displays the architecture of the simulation. It consists of 5 IoT clients connected with two IoT gateways. The IoT gateways forward the aggregated packets through SDN switches. The SDN controller, finally transmit the packets to 3 IoT servers. From the clients to the server, the data flows are configured. IoT application uses voice data at a constant bit rate (CBR) and file transfer (FTP).

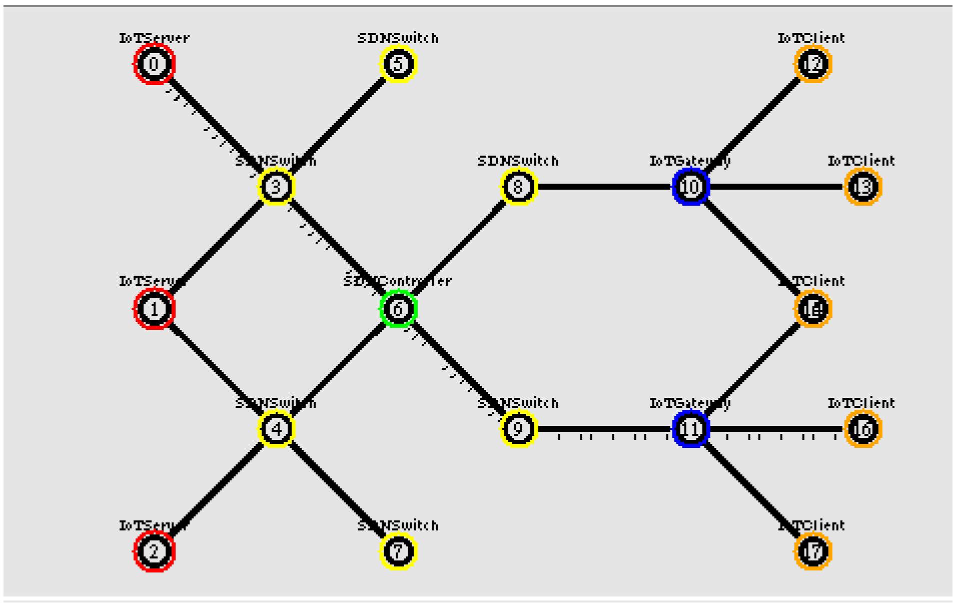

Fig 2 Simulation Topology

4.1 Simulation settings

Table 1 contains a list of the simulation parameters.

Parameter	**Value**
IoT Servers	3
IoT Clients	5
Switches	6
IoT Gateways	2
Controllers	2
IoT traffic models	Constant Bit Rate(CBR) and Exponential (EXP)
Packet Type	Ipv4L3Protocol
Simulation time	50 seconds

Date Rate	100 Mbps
Number of flows	2 to 10
Data transfer rate	250 to 1250Kb

. **Table 1** Simulation Parameters

4.2 Results

The performance of RLBRP is compared with Predictive and Proactive Heuristic mechanism and Fuzzy logic (PPH-Fuzzy) [11]. Throughput is measured A by adjusting the heterogeneous fluxes and traffic rate, which results in an end-to-end latency.

A. Based on Flows

The number of flows is varied in our first experiment to 2, 4, 6, 8, and 10. The end-to-end delay results for changing the flows are displayed in Table 2 and Figure 3.

Flows	**RLBRP (ms)**	**PPH-Fuzzy (ms)**
2	0.404	0.704
4	0.809	1.106
6	1.213	1.647
8	1.618	1.916
10	1.823	2.289

Table 2: Results of Delay for Flows Case

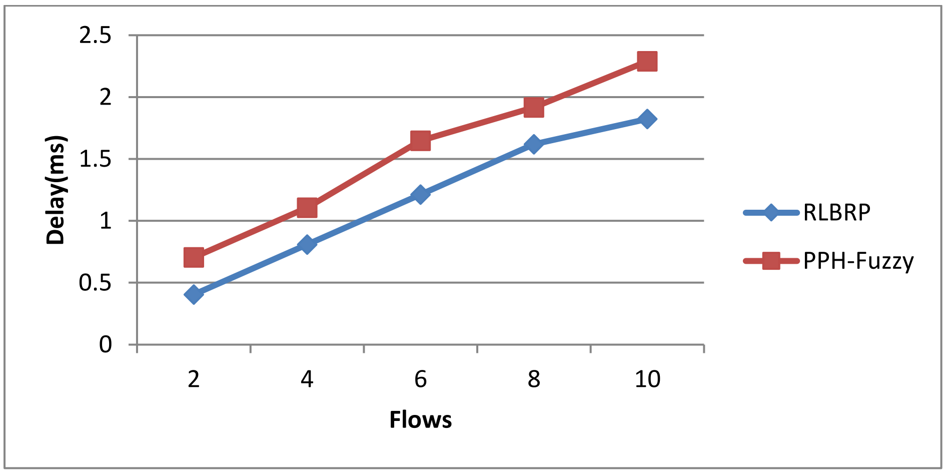

Fig 3 Flows Vs Delay

From Figure 3, we can observe that the delay of our proposed RLBRP is 26% lesser than PPH-Fuzzy.

Table 3 and Figure 4 show the results of throughput for varying the flows.

Flows	**RLBRP (Mb/s)**	**PPH-Fuzzy (Mb/s)**
2	21.74	10.87
4	43.49	17.87
6	65.26	25.75
8	87.05	41.75
10	101.56	68.48

Table 3: Results of Throughput for Flows Case

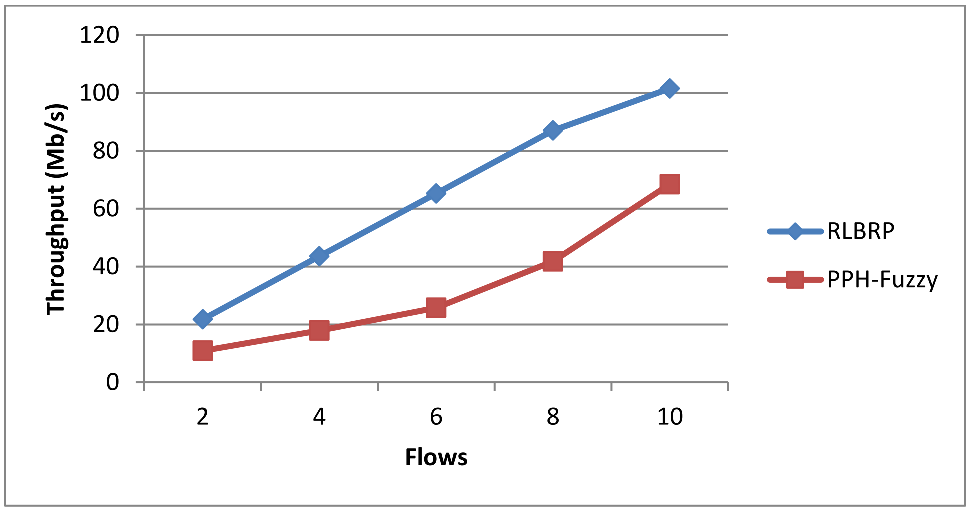

Fig 4: Flows Vs Throughput

From Figure 4, we can observe that the throughput of our proposed RLBRP is 51% higher than PPH-Fuzzy.

B. Based on Rate

The transmission rate is varied to 250, 500, 750, 1000, and 1250 kbps in our second experiment. The end-to-end delay results for changing the rate are displayed in Table 4 and Figure 5.

Rate(Kb)	RLBRP (ms)	PPH-Fuzzy (ms)
250	1.668	1.916
500	1.837	2.507
750	2.245	2.998
1000	2.618	3.391
1250	2.834	3.525

Table 4: Results of Delay for Rate Case

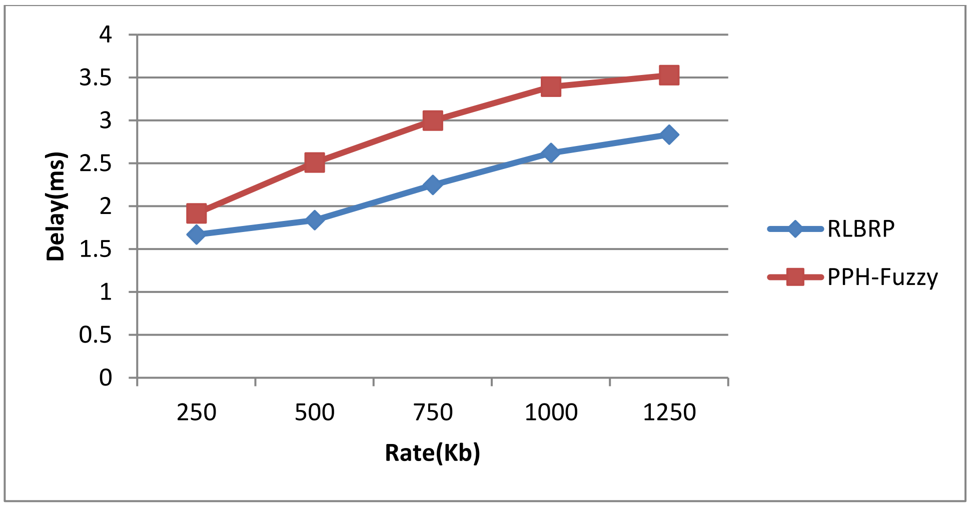

Fig 5: Rate Vs Delay

From Figure 5, we can observe that the delay of our proposed RLBRP is 22% lesser than PPH-Fuzzy.

Table 5 and Figure 6 show the results of throughput for varying the rate.

Rate(Kb)	**RLBRP (Mb/s)**	**PPH-Fuzzy (Mb/s)**
250	21.76	10.87
500	43.51	27.75
750	65.26	41.78
1000	87.01	62.71
1250	108.22	89.22

Table 5: Results of Throughput for Rate Case

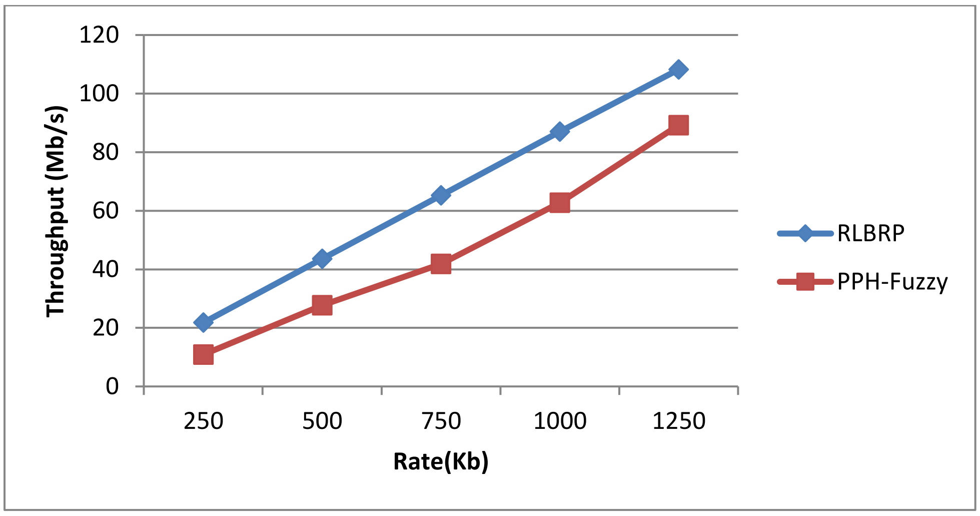

Fig 6 Rate Vs Throughput

From Figure 6, we can observe that the throughput of our proposed RLBRP is 38% higher than PPH-Fuzzy.

5 Conclusion

In this paper, we have proposed the RLBRP for SDN-IoT Heterogeneous Applications. For delay-aware traffic, the controller selects the forwarder nodes with lower delay and lower packet queue length. For bandwidth-aware traffic, the link quality, link capacity and noise parameters are estimated and the forwarder nodes with high link capacity and quality with lower noise will be selected. For normal traffic, shortest path will be selected. The proposed RLBRP is implemented in OpenSwitch of NS3. The performance of RLBRP is compared with PPH-Fuzzy technique. Simulation results show that RLBRP maximizes the network throughput and minimizes the end-to-ennd delay.

References

[1] Mohammad Riyaz Belgaum, Fuead Ali, Zainab Alansari, ShahrulnizaMusa,Muhammad Mansoor Alam and M. S. Mazliham, "Artificial Intelligence Based Reliable Load Balancing Framework in Software-Defined Networks", Computers,Materials & Continua, Vol-7, No-1, 2022.

[2] Himanshi Babbar, Shalli Rani, Divya Gupta, Hani Moaiteq Aljahdali ,Aman Singh and Fadi Al-Turjman, "Load Balancing Algorithm on the Immense Scale of Internet of Things in SDN for Smart Cities", Sustainability, Vol-13, 2021.

[3] Jaime Galán-Jiménez, Marco Polverini, Francesco G. Lavacca, Juan Luis Herrera and Javier Berrocal, "Joint energy efficiency and load balancing optimization in hybrid IP/SDN networks", Annals of Telecommunications (2023) 78:13–31, 2023.

[4] Evans Osei Kofi, Emmanuel Ahene, "Enhanced network load balancing technique for efficient performance in software defined network", PLOS ONE | https://doi.org/10.1371/journal.pone.0284176 April 13, 2023.

[5] Omran M. A. Alssaheli, Z. Zainal Abidin, N. A. Zakaria, Z. Abal Abas, "Software Defined Network based Load Balancing for Network Performance Evaluation", (IJACSA) International Journal of Advanced Computer Science and Applications,, Vol. 13, No. 4, 2022.

[6] Thabo Semong, Thabiso Maupong, Stephen Anokye, Kefalotse Kehulakae,Setso Dimakatso, Gabanthone Boipelo and Seth Sarefo, "Intelligent Load Balancing Techniques in Software Defined Networks: A Survey", Electronics, Vol-9, No-1091, 2020.

[7] Aashish kumar and Darpan Anand, "Load balancing for Software Defined Network using Machine learning", Turkish Journal of Computer and Mathematics Education Vol.12 No.2 (2021), 527- 535, 2021.

[8] Sharathkumar S and N Sreenath, "A Reliable Load Balancing Fault Tolerant Multi-SDN Controller approach in a typical Software Defined Network", EAI Endorsed Transactions on Internet of Things, doi: 10.4108/eai.2-2-2022.173295, 2022.

[9] Elham Hajian, Mohammad Reza Khayyambashi , And Naser Movahhedinia, "A Mechanism for Load Balancing Routing and Virtualization Based on SDWSN for IoT Applications", IEEE Access, Vol-10, 2022.

[10] Ziran Min, Hongyang Sun, Shunxing Bao, Aniruddha S. Gokhale and Swapna S. Gokhale, "A Self-Adaptive Load Balancing Approach for Software-Defined Networks in IoT", IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS) , 2021, DOI: 10.1109/ACSOS52086.2021.00034, 2022.

[11] Ahmadreza Montazerolghaem and Mohammad Hossein Yaghmaee, "Load-balanced and QoS-aware Software-defined Internet of Things", IEEE Internet Of Things Journal, Feb 2020.

[12] Hind Sounni, Najib El kamoun, Fatima Lakrami, "A new SDN-based load balancing algorithm for IoT devices", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 21, No. 2, February 2021, pp. 1209~1217, 2021.