KLE Technological University
Creating Value
Leveraging Knowledge
KLE TECH
BVB Campus, Vidyanagar, Hubballi – 580031, Karnataka, INDIA.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Minor Project Report

On

Load Balancing In Fog Network

submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Manisha Belagal	01FE19BCS207
Mehar Anjum Soudagar	01FE19BCS208
Devyansh Agrawal	01FE19BCS245
Chetan N Shet	01FE19BCS255

Under the guidance of

Dr.Vijeth Kotagi

School of Computer Science and Engineering

KLE Technological University, Hubballi

2021-2022

BVB Campus, Vidyanagar, Hubballi – 580031, Karnataka, INDIA.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2021-22

CERTIFICATE

This is to certify that project entitled Load Balancing in Fog Network is a bonafied work carried out by the student team Manisha Belagal 01FE19BCS207,Mehar Anjum Soudagar 01FE19BCS208,Devyansh Agrawal 01FE19BCS245,Chetan N Shet 01FE19BCS255, in partial fulfillment of the completion of 6th semester B. E. course during the year 2020 – 2021. The project report has been approved as it satisfies the academic requirement with respect to the project work prescribed for the above said course.

Guide Name	Head, SoCSE
Dr.Vijeth Kotagi	Dr. Meena S. M.

External Viva-Voce

Name of the examiners	Signature with date
1 ____________________	____________________
2 ____________________	____________________

Acknowledgement

We would like to thank our faculty and management for their professional guidance towards the completion of the project work. We take this opportunity to thank Dr. Ashok Shettar, Vice-Chancellor, Dr. N.H Ayachit, Registrar, and Dr. P.G Tewari, Dean Academics, KLE Technological University, Hubballi, for their vision and support.

We also take this opportunity to thank Dr. Meena S. M, Professor and Head, SoCSE for having provided us direction and facilitated for enhancement of skills and academic growth.

We thank our guide Dr.Vijeth Kotagi,Assistant Professor, SoCSE for the constant guidance during interaction and reviews.

We extend our acknowledgement to the reviewers for critical suggestions and inputs. We also thank Project Co-Ordinator Mr. Uday N.Kulkarni, Mr. Guruprasad Konnurmath and all the reviewers for their support during the reviews and course of completion.

We express gratitude to our beloved parents for constant encouragement and support.

Manisha Belagal - 01FE19BCS207
Mehar Anjum Soudagar - 01FE19BCS208
Devyansh Agrawal - 01FE19BCS245
Chetan N Shet - 01FE19BCS255

ABSTRACT

Communications and computation needs are growing rapidly all over the world and this has brought about a great evolution where Internet of Things is coming into the picture. Almost all applications today are connected to the Internet which has resulted in massive production of data and hence requires rapid computation. Bringing the computation closer to the edge devices is necessary to reduce the delay in request processing, fog computing helps to do so. Fog nodes receive many requests and the amount of data is also constantly increasing, the fog nodes might be overloaded, so effective load balancing should be carried out to ensure the best performance of the fog nodes. Furthermore, as the volume of data is exponentially increasing in IoT networks, it is inevitable to design an efficient load balancing algorithm to distribute the load across fog nodes in a fog-enabled IoT network. Here, we propose a Q-Learning based dynamic load balancing for fog-enabled IoT networks to make optimal node selection decisions.

Keywords : *Fog Computing, Load balancing, Q learning, Internet of Things.*

CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

With an ever increasing number of IoT devices, and latency constrained applications it is essential to move away from cloud computing and towards processing requests closer to the edge devices. With fog computing, computations have become closer to physical devices connected to the internet. Furthermore, as the volume of data is exponentially increasing in IoT networks, it is inevitable to design an efficient load balancing algorithm to distribute the load across fog nodes in a fog-enabled IoT network. Using load balancing, the incoming requests in the fog environment are distributed among available fog nodes so that no fog nodes are overloaded. Through this mechanism, it is possible to maximize performance, throughput, and utilization of resources while minimizing response times, costs, and power consumption. In this project we propose a Q-Learning based dynamic load balancing framework for fog-enabled IoT network to make optimal node selection decisions for offloading the incoming requests.[1] [2] [3]

1.1 Motivation

Since the beginning of the 21st century, cloud computing has emerged as a paradigm for moving computing and data storage resources between geographically dispersed data centers. Through this mechanism, performance, throughput, and utilization of resources can be maximized while response times, costs, and energy consumption are minimized. With the advancement in wireless technologies and computing, the number of devices connecting to the Internet are growing at an exponential rate, as a result of which Cloud Computing had become a preferred choice that provides a cost-effective solution to high processing and huge data storage needs. With over 50 billion devices connected to the Internet and an unprecedented amount of data, handling such a huge amount of data with traditional computing models like cloud computing is difficult. However, with the recent expansion of IoT services, there are several growing challenges associated with cloud computing, including long communication latency's, privacy breaches, and related traffic loads that negatively impact its performance.

The fog computing paradigm encompasses computing resources located between edge devices and the cloud in order to bring computation closer to IoT devices. By bringing computing power and networking power to the network edge, fog computing extends the cloud paradigm to extend IoT closer to end-users and connected devices. Load balancing is an important aspect of fog networks and is also essential to avoid situations which leads to some overloaded fog nodes. Load balancing aims at distributing workloads to a set of reliable servers in the system. Therefore designing an effective load balancing algorithm in distributed fog networks is necessary.

1.2 Literature Review

Paper Name: Survey of Fog Computing: Fundamental, Network Applications, and Research Challenge[4]
Authors: Mithun Mukherjee, Lei Shu, Di Wang
Published Date: 2018
Learning: This survey provides an outline of fog computing architecture. Fog computing can process delay-sensitive requests from end-users efficiently as compared to cloud computing. Energy consumption is reduced. Resource utilization is increased due to fog computing, it also ensures better performance with respect to bandwidth and delay. Intensive overview of state of-the-art network applications is shown in this paper. And Fog nodes is reliable towards IOT networks and its vast markets. The basic aim of this research work is to provide application to the network and fundamentals of the fog networking. Also this work focus on advancing the efforts on trending issues.

Paper Name: Managing Fog Networks using Reinforcement Learning Based Load Balancing Algorithm[5]
Authors: Jung-yeon Baek, Georges Kaddoum, Sahil Garg, Kuljeet Kaur , and Vivianne Gravel
Published Date: 28 Jan 2019
Learning: The paper proposes a method which uses SDN-based Fog architecture along with Q-learning. A centralized controller is used which has the knowledge of the entire network. The controller controls the fog nodes. Fog nodes are used to reduce the request processing time and also to lower the overload probability. Markov decision process is used which helps a node to optimally select nodes to which it can offload its requests to. To ensure that the optimal action is selected ϵ -greedy algorithm is used. A better performance is obtained using this algorithm than some of the already existing methods.

Paper Name: ReTra: Reinforcement based Traffic Load Balancer in Fog based Network[6]
Authors: Divya.V , R. Leena Sri
Published Date: July 8 2019
Learning: Greater use of IoT devices and reliance on the Internet lead to a massive amount of data being generated. As a result, traditional architectures comprising the cloud and networking devices failed to enable real-time decisions. With fog computing and software-defined networking (SDN), traditional cloud architectures are more efficient. Furthermore, the refined architecture cannot enable the application to reach its full potential without proper load balancing, given the increasing data bombardment from various sources. This paper focuses on creating a real-time fog computing environment by using SDN. A reinforcement learning-based approach to load balancing has been proposed to use this platform. The algorithm adapts to the nature of the network in order to balance the load so that resources are available to the maximum.

Paper Name: Deep Reinforcement Learning for Joint Offloading and Resource Allocation in Fog Computing[7]
Authors: Wenle Bai , Cheng Qian
Published Date: 2021
Learning: This paper proposes an offloading strategy and resource allocation algorithm for the offloading of computing in fog networks, which aims at minimizing the cost of the system. The algorithm is based on deep reinforcement learning (DRL). By splitting the total offloading action into several sub-actions, the multi-agent setup overcomes the dimensional explosion problem of traditional DRL methods. Different fog nodes and users are compared with different cost variables in order to determine the varying cost. Results demonstrate that this method can significantly reduce costs.

Technical challenges that would be addressed

- Dynamically adjusting the load in the network.
- Increasing the response time for the users.
- Bringing the end devices closer to the computing nodes.
- Reducing the request processing time.

1.3 Problem Statement

To design and implement an effective distributed load balancing algorithm in a fog network.

Description: In a fog network as shown in Figure 1.1, each fog node is connected to various IoT devices which are transmitting data continuously. Our task is to design a dynamic load balancing algorithm that will calculate the overload probability of a fog node and make optimal node selection decisions using Q-learning. The requests are then offloaded to the selected nodes.

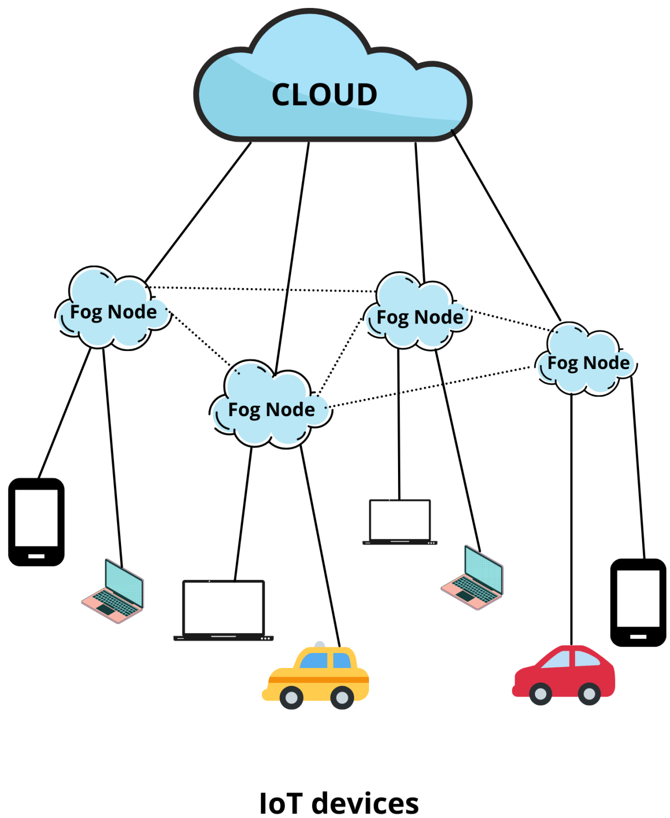

Figure 1.1: IoT Enabled Fog Network

1.4 Applications

- Self-Driving cars.
- Healthcare industry.
- Smart Cities.
- Real-time data analytics.
- Sports broadcast.
- Education field.

1.5 Objectives and Scope of the project

. This section defines the objectives that are planned to be achieved and the scope of the project.

1.5.1 Objectives

- To minimize overloading of fog nodes.
- To make effective dynamic offloading decisions.
- To determine the number of requests to be offloaded.
- To minimize request processing time and hence its latency.

1.5.2 Scope of the project

Fog nodes are considered to have limited computing resources, while the cloud is considered to have unlimited resources.A service request can be processed by offloading packets to neighboring fog nodes or the cloud if the node is overloaded, thus distributing the load across the network.The arrival rate of requests to the fog nodes follows Poisson distribution.

Chapter 2

REQUIREMENT ANALYSIS

2.1 Functional Requirements

- The arrival rate and service rate follow Poisson distribution and exponential distribution respectively.
- If a fog node is overloaded,it should learn to dynamically select neighbouring nodes to offload data.
- Usage of Q learning algorithm to select optimal nodes to offload to.
- The latency constraint of each request must be satisfied.

2.2 Non Functional Requirements

- Algorithm must be scalable.
- The load must be distributed across the network.
- The availability of the system must be 99.99%.
- A fog node should learn without interference of user.

Chapter 3

SYSTEM DESIGN

The proposed load balancing algorithm uses Q learning which consists the following tuples

- F : set of fog nodes
- M : set of IoT devices
- d_m : maximum allowed delay for m^{th} IoT device $\forall m \in M$
- t_m : actual processing time for m^{th} IoT device $\forall m \in M$
- S : set of states,where each state represents the load on fog node f $\in$ F
  - s1 : 0 – 0.6 (low load state)
  - s2 : 0.6 – 0.8 (medium load state)
  - s3 : $>$ 0.8 (high load state)
- A : set of actions defined as A $= \{$(P(B) $- \phi$) $\cup$ a$_c$ $\}$, where B $= \{$ $b_i : i = 1\ to\ |F|\}$ and a_c is action representing offloading to cloud
- R(s , a) : Reward function is calculated as, R(s , a) $=$ tanh(D) , where D $=$ d_m - t_m

3.1 Architecture Design

The system is designed in accordance with Figure 3.1. As we can see Client-server architecture is used. Here, the servers represent the fog nodes, and the sensors represent the IoT devices. All fog nodes are connected to each other and each fog node has several IoT devices connected to it. Each fog node is also connected to the cloud.

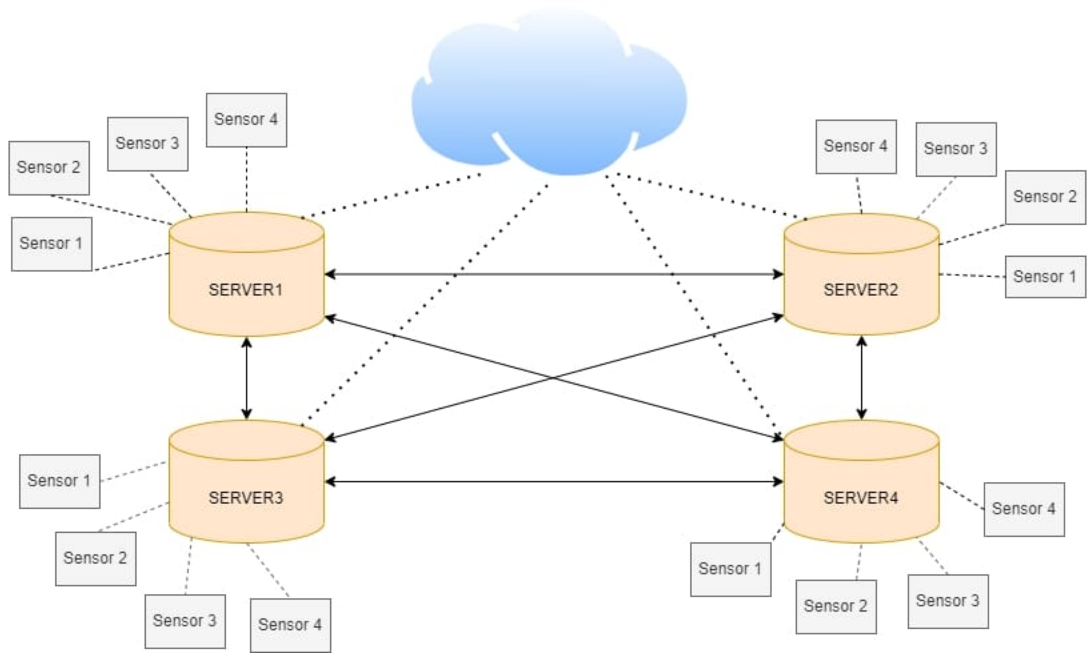

Figure 3.1: Client and Server Architecture

3.2 Dataset Design

The dataset used in this project is a live dataset collected from the campus using the prototype shown in Figure 3.2. The data is displayed in Figure 3.3, it contains information about humidity, temperature, hydrogen levels, carbon monoxide levels, and the air quality index obtained from the sensors.

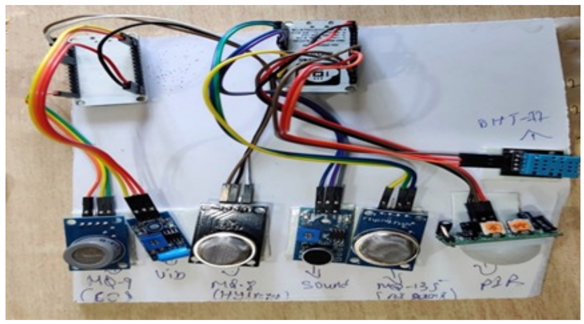

Figure 3.2: IOT devices to collect dataset

	A	B	C	D	E	F	G
1	id	Date	Time	Temp	Humi	MQ135	MQ9
2	1	02/16/22	15:21:50	26	46	560	495
3	2	02/16/22	15:22:52	28	46	610	677
4	3	02/16/22	15:23:55	28	46	497	674
5	4	02/16/22	15:24:57	28	46	476	672
6	5	02/16/22	15:25:57	27	47	448	666
7	6	02/16/22	15:26:59	29	48	437	673
8	7	02/16/22	15:28:00	27	48	426	670
9	8	02/16/22	15:29:01	29	48	423	666
10	9	02/16/22	15:30:02	27	48	401	663
11	10	02/16/22	15:31:06	29	47	408	643
12	11	02/16/22	15:32:08	27	48	405	535
13	12	02/16/22	16:25:48	26	47	547	658
14	13	02/16/22	16:26:49	28	48	466	567
15	14	02/16/22	16:27:50	28	48	446	644
16	15	02/16/22	16:32:00	26	48	349	649
17	16	02/16/22	16:33:01	29	47	415	646
18	17	02/16/22	16:34:01	29	47	412	639
19	18	02/16/22	16:35:04	29	47	403	636
20	19	02/16/22	16:36:07	29	47	413	624
21	20	02/17/22	16:37:07	29	47	405	637
22	21	02/17/22	16:38:09	29	46	393	627
23	22	02/17/22	16:39:09	29	47	393	626
24	23	02/17/22	16:40:11	29	47	422	622
25	24	02/17/22	16:41:12	30	46	416	618
26	25	02/17/22	16:42:13	30	46	401	626
27	26	02/17/22	16:43:14	30	45	410	624

Figure 3.3: Dataset

Chapter 4

IMPLEMENTATION

In this chapter, we describe the algorithm and its implementation in detail.
To find the value of an action in a particular state, Q learning is used. It's a model-free reinforcement learning algorithm.Here state represents current environment of the task and actions are something that the Q learning agent uses to change the states.
Here there are three states S1, S2, S3 as defined is system model and for n fog nodes there are 2^n possible actions.

For n=3, 8 actions are possible which are as follows:
1: offload to node 1
2: offload to node 2
3: offload to node 3
1,2: offload to nodes 1,2
1,3: offload to nodes 1,3
2,3: offload to nodes 2,3
1,2,3: offload to nodes 1,2,3
a_c: offload to the cloud
Therefore A= $\{ 1 , 2 , 3 , \{ 1 , 2 \} , \{ 1 , 3 \} , \{ 2 , 3 \} , \{ 1 , 2 , 3 \} , a_c\}$ represents the action set

4.1 Proposed Algorithm

Algorithm 1 Reinforcement based Load Balancing

Input: λ_f (Arrival rate of the packets in the fog node f ∈ F following Poisson distribution)
α (Learning rate)
ε (Exploration policy)
γ(discount factor)
μ(Processing rate of the server following exponential distribution)
d_m (Allowed latency constraint of a sensor)

Output: Fog nodes selected for offloading and the percentage of packets served within latency constraint

1. Q(s , a)← $0, \forall s \in S, \forall a \in A$ //initialize Q values states and actions.
2.Repeat
3. | choose action a ∈ A based on epsilon greedy algorithm
4. | offload packets to neighbours based on chosen action a.
5. | Receive D from neighbours
6. | Calculate reward R(s,a) = tanh(D)
7. | Update Q table using Q(s,a) =α[R(s , a)+$\gamma\times$ max Q(s‘,*)]+(1-α)Q(s , a)
8. | move to next state s ← s‘

Chapter 5

RESULTS AND DISCUSSIONS

The packets arriving from IoT devices to the fog nodes follow Poisson distribution, since we cannot predict the exact number of packets arriving.The graph in Figure 5.1 shows the variation of number of packets with probability mass function.

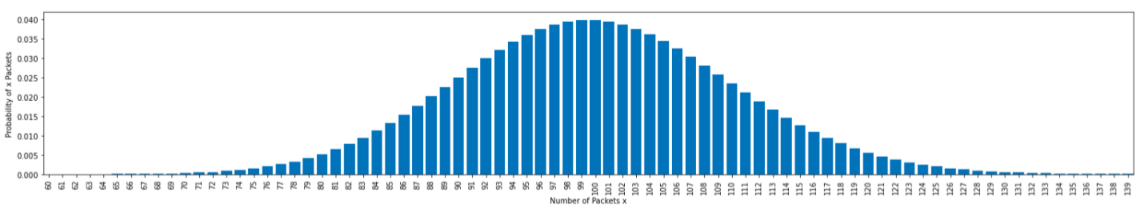

Figure 5.1: Poisson Distribution

The Table 5.2 contains the values of the maximum expected future rewards for each state.Based on this table, we will choose the best course of action by considering the maximum Q-value for each state.At every fog node, 2^n actions are available to choose from.The plot of Q-values for corresponding actions taken is shown in Figure 5.3

Q-Table

	Q-values				
State Low	-0.04390961	0.23672758	0.55096896	0.38417212	0.38432548
State Medium	0.53863242	0.53890643	0.53973244	0.66459545	0.46592737
State High	0.53909914	0.53912702	0.55108362	0.41072062	0.49914651

Figure 5.2: Q-table

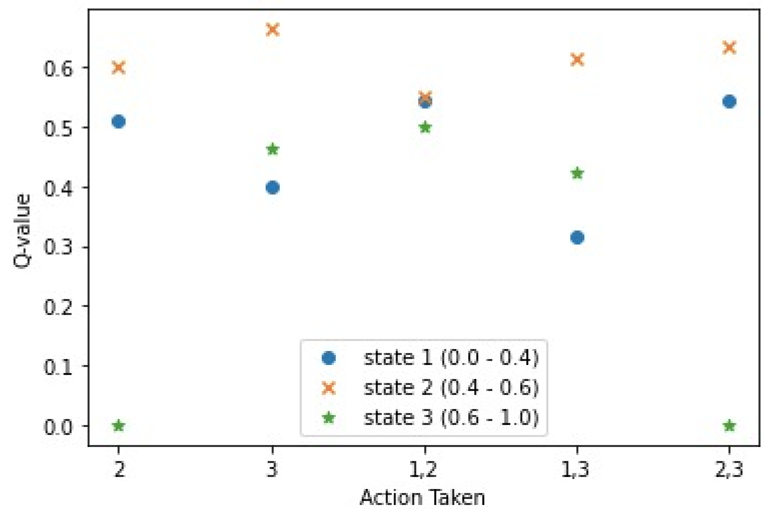

Figure 5.3: Q-value vs Action taken

The algorithm was tested for 1500 iterations.Figure 5.4, 5.5, 5.6 represents the graph plotted for the reward values against number of iterations for states S1,S2,S3 respectively.

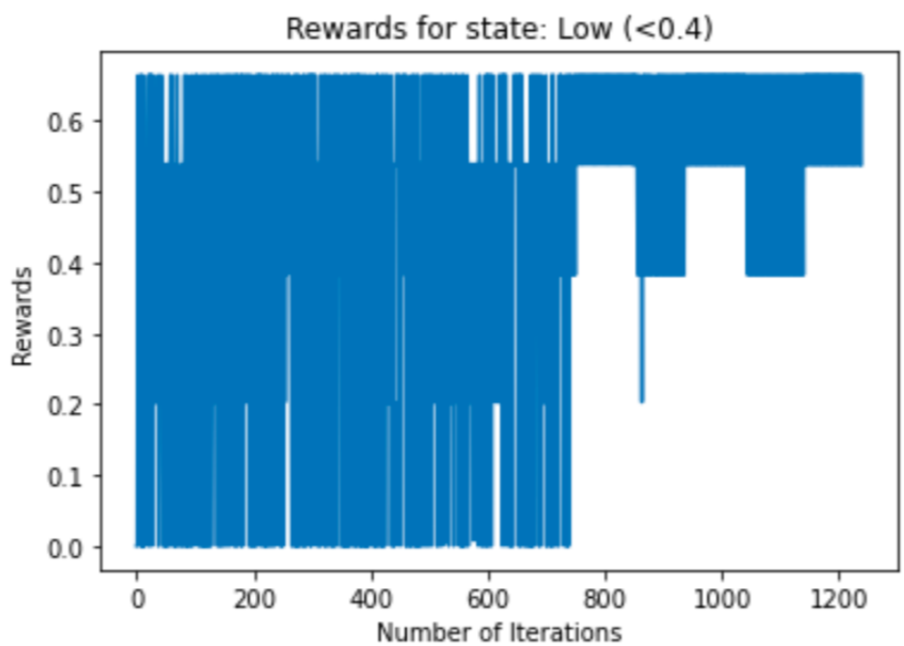

Figure 5.4: Rewards vs No.of Iterations for state S1

The graph shown in Figure 5.7 represents the time taken to process the packets at each iteration. It is observed that the minimum time required by the system for the packets to be processed is 1.95 seconds at the 200^{th} iteration and the maximum time is 2.1 seconds at the 700 and 900^{th} iterations.

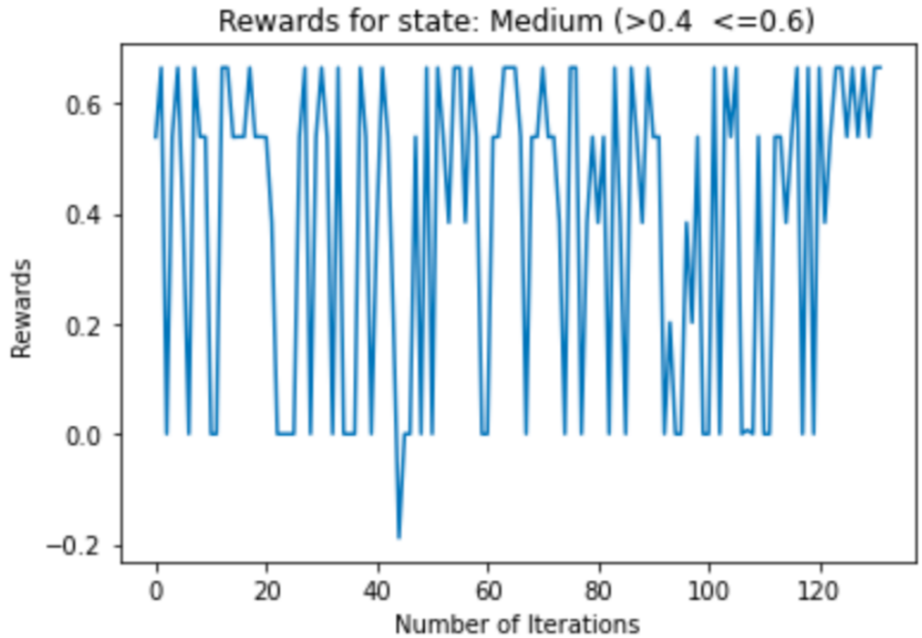

Figure 5.5: Rewards vs No.of Iterations for state S2

The graphs shown in Figures 5.8, 5.9 and 5.10, represent the exploration and exploitation phase for choosing the appropriate actions in state S1, S2, S3 respectively.In Figure 5.8, the algorithm is exploring the system till 800^{th} iteration and action {1,2} is choosing during exploitation.Similarly in Figure 5.9 and Figure5.10, action {1,3} and action {1,2} are chosen during exploitation phase.

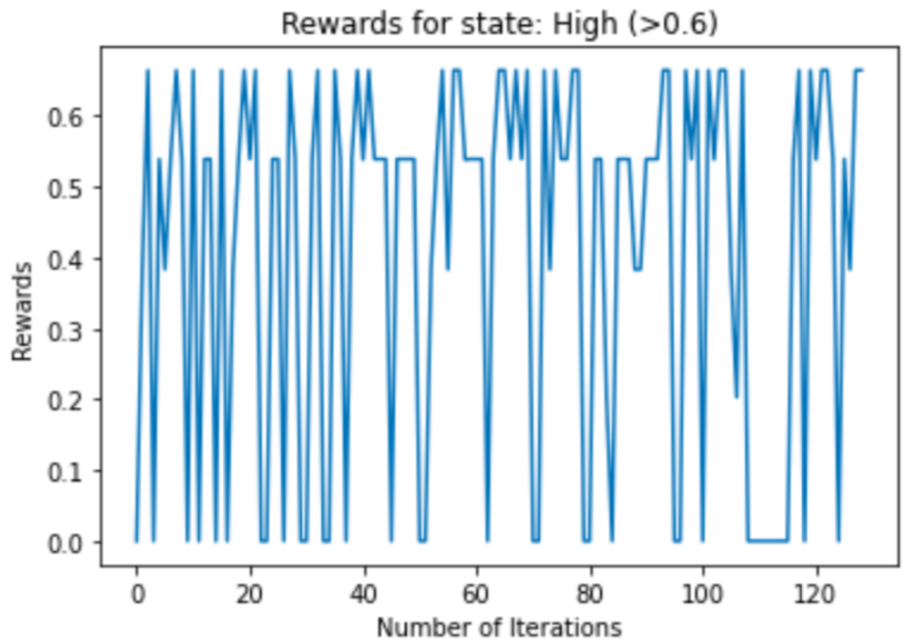

Figure 5.6: Rewards vs No.of Iterations for state S3

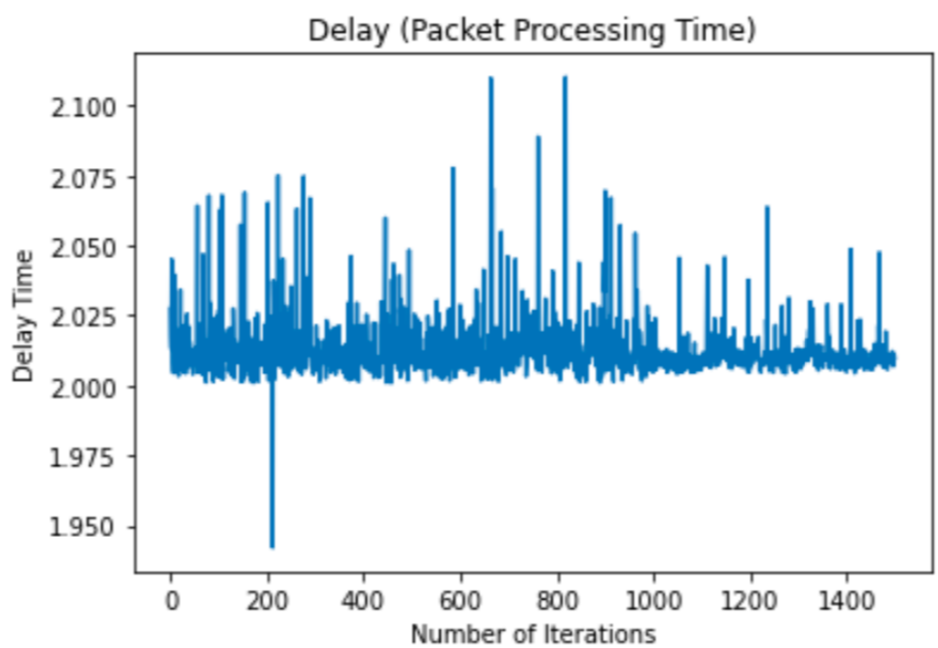

Figure 5.7: Packet Processing Delay vs No. of Iterations

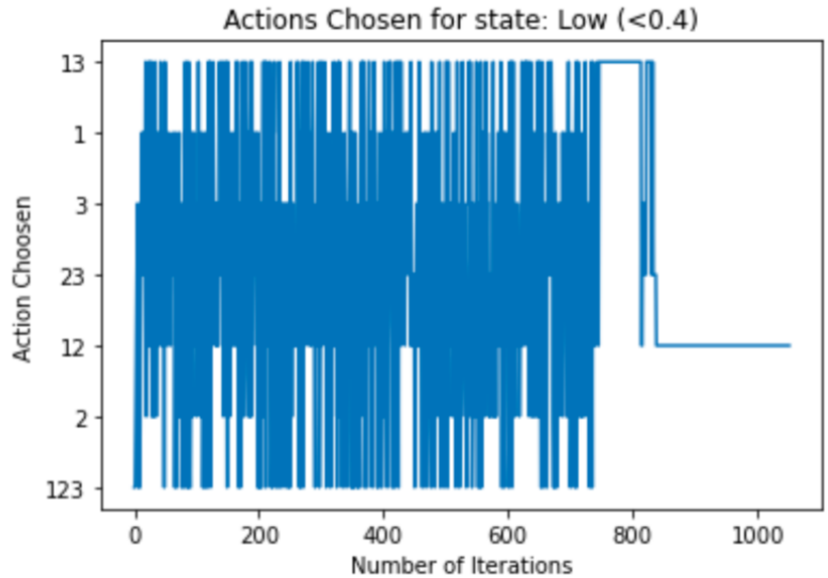

Figure 5.8: Action Chosen vs No. of Iterations for State S1

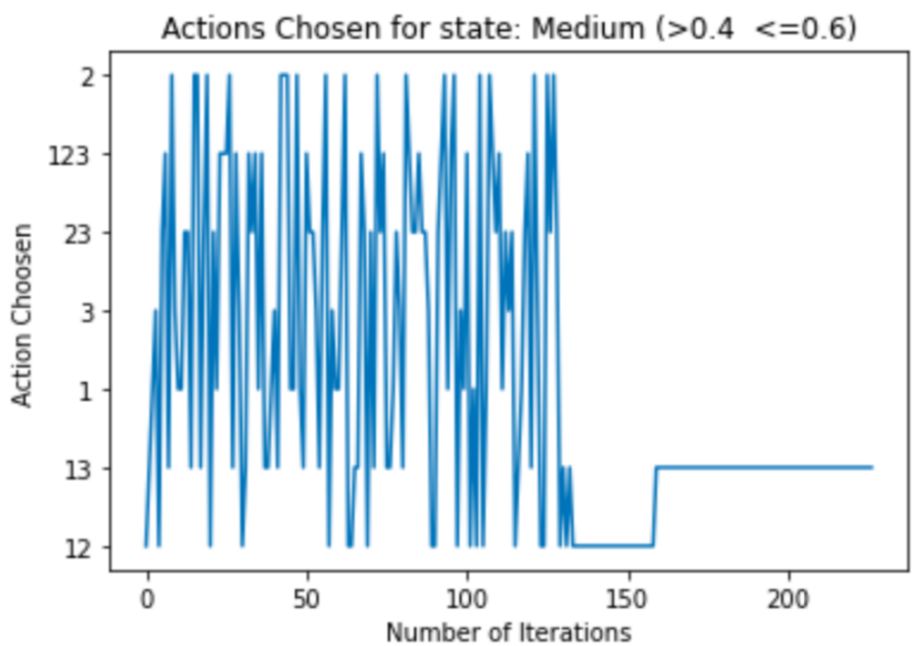

Figure 5.9: Action Chosen vs No. of Iterations for State S3

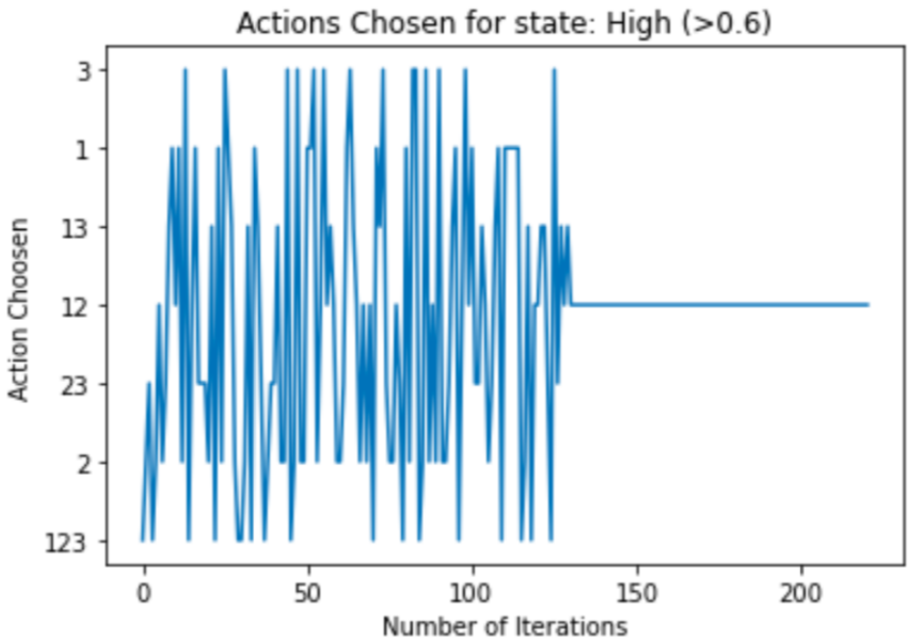

Figure 5.10: Action Chosen vs No. of Iterations for State S3

Chapter 6

CONCLUSION AND FUTURE SCOPE

Bringing the edge devices closer to the computing devices and to reduce the request processing time is the need of the hour, usage of fog networks is a viable solution. As the amount of data is rapidly increasing along with increase in the number of requests from the edge devices, the fog nodes might get overloaded ,to redistribute the load we tried to implement an effective load balancing algorithm which uses Q learning to select the optimal node to offload to.The fog network is implemented using a client-server architecture. The requests arriving at the fog node follow Poisson distribution and the processing rate of each fog node follows an exponential distribution.

The future scope of this project will involve improving the methodology so that the Q-learning algorithm reduces the amount of time needed to select the best node and offloads the number of packets depending on the current overload probability of the fog node.

REFERENCES

[1] Ahmadreza Montazerolghaem and Mohammad Hossein Yaghmaee. Load-balanced and qos-aware software-defined internet of things. *IEEE Internet of Things Journal*, 7(4):3323–3337, 2020.

[2] Jiaquan Yan, Jigang Wu, Yalan Wu, Long Chen, and Shuangyin Liu. Task offloading algorithms for novel load balancing in homogeneous fog network. In *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 79–84, 2021.

[3] Abdoulaye Tall, Zwi Altman, and Eitan Altman. Self-optimizing load balancing with backhaul-constrained radio access networks. *IEEE Wireless Communications Letters*, 4(6):645–648, 2015.

[4] Mithun Mukherjee, Lei Shu, and Di Wang. Survey of fog computing: Fundamental, network applications, and research challenges. *IEEE Communications Surveys Tutorials*, 20(3):1826–1857, 2018.

[5] Jung-yeon Baek, Georges Kaddoum, Sahil Garg, Kuljeet Kaur, and Vivianne Gravel. Managing fog networks using reinforcement learning based load balancing algorithm. pages 1–7, 2019.

[6] V Divya and R. Leena Sri. Retra: Reinforcement based traffic load balancer in fog based network. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–6, 2019.

[7] Wenle Bai and Cheng Qian. Deep reinforcement learning for joint offloading and resource allocation in fog computing. In *2021 IEEE 12th International Conference on Software Engineering and Service Science (ICSESS)*, pages 131–134, 2021.

N11-2

ORIGINALITY REPORT

8%	4%	7%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	V Divya, R. Leena Sri. "ReTra: Reinforcement based Traffic Load Balancer in Fog based Network", 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019 Publication	1%
2	Wenle Bai, Cheng Qian. "Deep Reinforcement Learning for Joint Offloading and Resource Allocation in Fog Computing", 2021 IEEE 12th International Conference on Software Engineering and Service Science (ICSESS), 2021 Publication	1%
3	Niloofar Tahmasebi Pouya, Mehdi Agha Sarram. "Blind Load-Balancing Algorithm using Double-Q- learning in the Fog Environment", 2021 11th International Conference on Computer Engineering and Knowledge (ICCKE), 2021 Publication	1%
4	export.arxiv.org Internet Source	1%

5	Jung-yeon Baek, Georges Kaddoum, Sahil Garg, Kuljeet Kaur, Vivianne Gravel. "Managing Fog Networks using Reinforcement Learning Based Load Balancing Algorithm", 2019 IEEE Wireless Communications and Networking Conference (WCNC), 2019 Publication	1 %
6	toc.proceedings.com Internet Source	1 %
7	"Table of contents", IEEE Communications Surveys & Tutorials, 2018 Publication	1 %
8	essay.utwente.nl Internet Source	1 %
9	www.coursehero.com Internet Source	1 %
10	www.ccs.neu.edu Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		