

**Republic of Iraq**  
**Ministry of Higher Education and Scientific Research**  
**University of Babylon**  
**College of Information Technology**  
**Department of Information Networks**



# **RESOURCE ALLOCATION IN FOG COMPUTING BASED ON HYBRID METAHEURISTIC ALGORITHM**

A Thesis Submitted

to the Council of the College of Information Technology for Postgraduate  
Studies of University of Babylon in Partial Fulfillment of the Requirements  
for the Degree of Master in Information Technology / Information  
Networks.

**By**

**Zahraa Husam Ebrahim Mosa**

**Supervised by**

**Asst. Prof. Dr. Mehdi Ebady Manaa**

**2023 A.D.**

**1445 A.H**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

يَرْفَعُ اللَّهُ الَّذِينَ آمَنُوا مِنْكُمْ  
وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ وَاللَّهُ  
بِمَا تَعْمَلُونَ خَبِيرٌ

صَدَقَ اللَّهُ الْعَظِيمُ

سورة المائدة . آية 11

## **Certification of the Examination Committee**

We hereby certify that we have studied the thesis entitled (**RESOURCE ALLOCATION IN FOG COMPUTING BASED ON HYBRID METAHEURISTIC ALGORITHM**) presented by the student (**ZAHRAA HUSAM EBRAHIM MOSA**) and examined her in its content and what is related to it, and that, in our opinion, it is adequate with (**Excellent**) standing as a thesis for the degree of Master in Information Technology-Information Networks.

Signature:  
Name: **Dr. Saad Talib Hasson**  
Title: Prof.  
Date:    /    / 2023  
(Chairman)

Signature:  
Name: **Dr. Abdul basit Kadhim Skukur**  
Title: Asst. Prof.  
Date:    /    / 2023  
(Member)

Signature:  
Name: **Rasim Azeez Kadhim**  
Title: Lecturer  
Date:    /    / 2023  
(Member)

Signature:  
Name: **Dr. Mehdi Ebady Manaa**  
Title: Asst. Prof.  
Date:    /    / 2023  
(Member and Supervisor)

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:  
Name: Wesam Sameer AbdAli Bhaya  
Title: Professor  
Date:    /    / 2023  
(Dean of Collage of Information Technology)

## **Supervisor Certification**

I certify that the thesis entitled (**RESOURCE ALLOCATION IN FOG COMPUTING BASED ON HYBRID METAHEURISTIC ALGORITHM**) was prepared under my supervision at the department of Information Networks/ College of Information Technology / University of Babylon as partial fulfillment of the requirements of the degree of Master in Information Technology-Information Networks.

Signature:

Supervisor Name: **Asst. Prof. Dr. Mehdi Ebady Manaa**

Date:     /     /2023

## **The Head of the Department Certification**

In view of the available recommendations, I forward the thesis entitled “**RESOURCE ALLOCATION IN FOG COMPUTING BASED ON HYBRID METAHEURISTIC ALGORITHM**” for debate by the examination committee.

Signature:

**Prof. Dr. Saad Talib Hasson Aljebori**

Head of Information Networks Department

Date:     /     /2023

## **Declaration**

I hereby declare that this thesis, submitted to the University of Babylon in partial fulfillment of requirement for the degree of Master of Information Technology-Information Networks has not been submitted as an exercise for a similar degree at any other university. I also certify that this work described here is entirely my own except for experts and summaries whose sources are appropriately cited in the references.

Signature:

Name : **Zahraa Husam Ebrahim**

Date: / / **2023**

# DEDICATION

I DEDICATE THIS THESIS

TO MY FATHER

TO MY MOTHER

TO MY SUPERVISOR

TO MY FAMILY

TO MY FRIENDS

Researcher

## Acknowledgement

In the name of God, Most Gracious, Most Merciful, At first, Praise be to God and thanks to God and the satisfaction of parents and conciliation only from God greatest praise is to **Allah** for His assistance in facing the difficulty that I met in my study, and for always helping me to achieve my aims, also for His great graces and boons all the time.

I would like to express my deepest thanks to my supervisor **Asst. Prof. Dr. Mehdi Ebady Manaa** for their valuable advice, motivation, guidance, and for so many fruitful discussions throughout the preparation of this thesis.

I would like to extend my respect and deepest gratitude to the College of Information Technology.

Sincere appreciation and love go to my family, my father's soul who he was always be with me in my heart and my dear mother that whatever I did to her will not reward her they provide me with optimism and pure affection and they give me great hope, encouragement and they have stood with me in every step in this research. I dedicate this work and give special thanks to those who encouraged me to continue my scientific career, my husband, and my children.

Finally, Sincere thanks and appreciation to all friends, colleagues and loved ones

Researcher

## **Abstract**

The use of mobile-fog-server applications in recent times has resulted in the generation of a huge amounts of data encompassing diverse data formats, including images, documents, multimedia, and other file types. It is a challenge to manage and control the volume of data utilized in fog computing. Additionally, fog-server computing has network performance issues such as latency, resource allocation, and fitness. In order to address these issues, a developed fog-server computing resource allocation approach has been employed based on meta-heuristics algorithms, which integrated Single Node/Multiple fog computing with hybrid resource allocation algorithm to simulate and evaluate network performance in mobile network.

The proposed system is applied in fog node with three scenarios as. The first is the using of Particle Swarm Optimization (PSO) algorithm to enhance the effectiveness, better search speed, and improve Cost efficiency for mobile devices. The second is the Harris Hawks Optimization (HHO) algorithm to maintain the quality of the data collected in mobile network. The third is hybrid resource allocation algorithm based on integrated two optimization methods represented by HHO and PSO to build a value Better Fitness Resource Allocation (BFRA). The output was produced by both optimizers, taking into consideration their respective characteristics, drawing on simulation outcomes and a comparative analysis of Hybrid resource allocation conventional and related methodologies, as well as PSO and HHO algorithms, the proposed hybrid multiple system is better with average makespan is 78.734 seconds, and average speed for process completion in servers is 322.7477 KB/ seconds. The average processing time of hybrid distributed in seconds is enhanced to 44% compared with related works.

# Table of Contents

Dedication .....	IV
Acknowledgement.....	V
Abstract .....	
Table of Contents .....	VI
List of Tables.....	IX
List of Figures .....	XIII
List of Algorithms .....	XV
<b>CHAPTER ONE INTRODUCTION.....</b>	<b>1</b>
1.1 Overview .....	1
1.2 Problem Statement .....	3
1.3 Related Work .....	3
1.4 Aims of the work .....	8
1.5 Thesis Outline .....	9
<b>CHAPTER TWO THEORETICAL AND TECHNICAL BACKGROUND</b>	
2.1 Introduction.....	10
2.2 Fog Computing .....	10
2.2.1 The architecture of Fog Computing.....	13
2.2.2 Benefits of Fog Computing.....	15
2.2.3 Fog Computing Issues .....	16
2.2.4 Fog-computing in IoT-Mobile network applications.....	18
2.2.5 Resource Management approaches in Fog-IoT Environment .....	18
2.3 Resource Allocation in Fog computing .....	19
2.4 Methuristics Algorithm for resource allocation.....	19
2.4.1 Harris Hawks Optimization (HHO) algorithm .....	21
2.4.2 Particle Swarm Optimization (PSO).....	24
2.5 Resource Allocation Evaluation Metrics .....	26
2.5.1 Throughput.....	26
2.5.2 Response time .....	26
2.5.3 Execution time .....	27
2.5.4 Resource Utilization .....	27
2.5.5 Latency.....	27
2.5.6 Packet Loss Ratio.....	27
2.5.7 Total Network Usage .....	28

2.5.8 Total Execution Cost .....	28
2.5.9 Fitness value .....	28
2.6 Simulation Tools.....	29
2.6.1 iFogSim simulation.....	29
2.6.2 OMNET++.....	30
<b>CHAPTER THREE THE PROPOSED APPROACH</b>	
3.1 Introduction.....	31
3.2 The implementation requirements .....	31
3.2.1 Harris hawks optimization implementation requirements .....	31
3.2.2 Meta-heuristic algorithm implementation requirements.....	33
3.2.3 Hybrid algorithm in Multiple fog-server system .....	34
3.3 The proposed Mobile-Fog computing architectures .....	35
3.3.1 The proposed Fog-Server Single Node Architecture.....	35
3.3.2 The proposed Fog-Server Multiple architecture.....	38
3.4 The proposed Hybrid Resource Allocation Development Approach .....	42
3.4.1 The Harris hawks optimization algorithm .....	43
3.4.2 The PSO Algorithm .....	47
3.4.3 The hybrid Resource Allocation .....	51
<b>CHAPTER FOUR RESULTS AND DISCUSSION.....</b>	
4.1 Introduction.....	55
4.2 The proposed System Implementation.....	55
4.2.1 The proposed Single Fog Computing Resource Allocation Scenario .....	55
4.2.2 The proposed Multiple Fog Computing Resource Allocation Scenario.....	58
4.3 The results of Single Node Fog Computing. ....	62
4.3.1 Fog computing system without Resource Allocation .....	62
4.3.2 Resource allocation with Meta-heuristic PSO algorithm.....	66
4.3.3 Resource allocation with Harris hawks optimization.....	70
4.3.4 Resource allocation with Hybrid algorithm .....	74
4.4 The results of Multiple Fog Computing.....	83
4.4.1 Fog computing system Without Resource Allocation.....	85

4.4.2 Resource allocation with Meta-heuristic PSO algorithm.....	89
4.4.3 Resource allocation with HHO .....	94
4.4.4 Resource allocation with Hybrid algorithm .....	99
4.5 Summary.....	107
<b>CHAPTER FIVE CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORKS ....</b>	
5.1 Conclusions.....	108
5.2 Suggestions for Future Works .....	108
<b>REFERENCES.....</b>	<b>110</b>

<b>List of Tables</b>		
<b>Table No.</b>	<b>Title</b>	<b>Page No.</b>
1.1	The literature survey summary.	7
4.1	The specifications of the proposed fog-server system.	55
4.2	The Specification of the Single Node Fog-server system.	56
4.3	The Multiple Fog-server system Specification.	59
4.4	The main evaluation parameters of the without resource allocation in Single Node fog-server system.	62
4.5	Channel allocation and total network usage of the without resource allocation in Single Node fog-server system.	63
4.6	Evaluation metrics of the without resource allocation in Single Node fog-server system.	64
4.7	FTP file upload/download by server 1 of without allocation in Single Node fog-server system.	65
4.8	FTP file upload/download by server 2 of without allocation in Single Node fog-server system.	65
4.9	FTP file upload/download by server 3 of without allocation in Single Node fog-server system.	65
4.10	FTP file upload/download by server 4 of without allocation in Single Node fog-server system.	66
4.11	FTP file upload/download by server 5 of without allocation in Single Node fog-server system.	66
4.12	2nd PSO case study evaluation criteria in Single Node fog-server system.	67
4.13	Channel allocation and total network usage in Single Node fog-server system.	67
4.14	2nd PSO case study channel allocation in Single Node fog-server system.	68
4.15	FTP file upload/download by server 1 of PSO in Single Node fog-server system.	69
4.16	FTP file upload/download by server 2 of PSO in Single Node fog-server system.	69
4.17	FTP file upload/download by server 3 of PSO in Single Node	70

	fog-server system.	
4.18	FTP file upload/download by server 4 of PSO in Single Node fog-server system.	70
4.19	FTP file upload/download by server 5 of PSO in Single Node fog-server system.	70
4.20	HHO case study evaluation criteria in Single Node fog-server system.	71
4.21	Channel utilization and Total Network Usage for HHO in Single Node fog-server system.	72
4.22	HHO case study channel allocation in Single Node fog-server system.	72
4.23	FTP file upload/download by server 1 of HHO in Single Node fog-server system.	73
4.24	FTP file upload/download by server 2 of HHO in Single Node fog-server system.	73
4.25	FTP file upload/download by server 3 of HHO in Single Node fog-server system.	74
4.26	FTP file upload/download by server 4 of HHO in Single Node fog-server system.	74
4.27	FTP file upload/download by server 5 of HHO in Single Node fog-server system.	74
4.28	Main evaluation Hybrid resource allocation in Single Node fog-server system.	76
4.29	Channel utilization with total network usage of the Hybrid in Single Node fog-server system.	77
4.30	Resource Utilization, Processing Time, Execution Cost, and Fitness Value of the Hybrid in Single Node fog-server system.	78
4.31	FTP file upload/download by server 1 of the Hybrid approach in Single Node fog-server system.	79
4.32	FTP file upload/download by server 2 of the Hybrid approach in Single Node fog-server system.	79
4.33	FTP file upload/download by server 3 of the Hybrid approach in Single Node fog-server system.	79
4.34	FTP file upload/download by server 4 of the Hybrid approach in Single Node fog-server system.	80
4.35	FTP file upload/download by server 5 of the Hybrid approach in Single Node fog-server system.	80
4.36	The proposed Single Node fog-server system comparison with the other related works.	83
4.37	Multiple fog-server system evaluation of without resource allocation	85
4.38	Channel allocation and total network usage of without resource allocation in Multiple fog-server system.	86

4.39	Evaluation metrics of without resource utilization in Multiple fog-server system.	87
4.40	FTP file upload/download by server 1 of without allocation in Multiple fog-server system.	88
4.41	FTP file upload/download by server 2 of without allocation in Multiple fog-server system.	88
4.42	FTP file upload/download by server 3 of without allocation in Multiple fog-server system.	89
4.43	FTP file upload/download by server 4 of without allocation in Multiple fog-server system.	89
4.44	FTP file upload/download by server 5 of without allocation in Multiple fog-server system.	89
4.45	PSO case study evaluation criteria in Multiple fog-server system.	90
4.46	Channel allocation and total network usage in Multiple fog-server system.	91
4.47	PSO Case Study channel allocation in Multiple fog-server system.	92
4.48	FTP file upload/download by server 1 of the PSO of Multiple fog-server system.	93
4.49	FTP file upload/download by server 2 of the PSO of Multiple fog-server system.	93
4.50	FTP file upload/download by server 3 of the PSO of Multiple fog-server system.	94
4.51	FTP file upload/download by server 4 of the PSO of Multiple fog-server system.	94
4.52	FTP file upload/download by server 5 of the PSO of Multiple fog-server system.	94
4.53	Main evaluation metrics of HHO in Multiple fog-server system.	95
4.54	Channel utilization and total network usage for HHO case study of Multiple fog-server system.	96
4.55	Evaluation metrics of HHO in Multiple fog-server system.	97
4.56	FTP file upload/download by server 1 of the HHO of Multiple fog-server system.	98
4.57	FTP file upload/download by server 2 of the HHO of Multiple fog-server system.	98
4.58	FTP file upload/download by server 3 of the HHO of Multiple fog-server system.	98
4.59	FTP file upload/download by server 4 of the HHO of Multiple fog-server system.	98
4.60	FTP file upload/download by server 5 of the HHO of Multiple fog-server system.	99
4.61	The Hybrid approach of Multiple fog-server main evaluation parameters.	100

4.62	Channel utilization with Total Network Usage of the Hybrid in Multiple fog-server system.	101
4.63	Evaluation metrics of hybrid case study in the Multiple fog-server system.	102
4.64	FTP file upload/download by server 1 of Hybrid of Multiple fog-server system.	103
4.65	FTP file upload/download by server 2 of Hybrid of Multiple fog-server system.	103
4.66	FTP file upload/download by server 3 of Hybrid of Multiple fog-server system.	103
4.67	FTP file upload/download by server 4 of Hybrid of Multiple fog-server system.	103
4.68	FTP file upload/download by server 5 of Hybrid of Multiple fog-server system.	103
4.69	The proposed Multiple fog-server system comparison with the other related works.	106

<b>List of Figures</b>		
<b>Figure No.</b>	<b>Title</b>	<b>Page No.</b>
1.1	Three layers of Mobile fog computing ( Cloud, Fog, and Device).	2
2.1	System model for Fog Computing	11
2.2	Fog computing architecture.	14
2.3	IoT-fog-cloud systems of fog environment heterogeneity and imbalanced workload.	17
2.4	Taxonomy of meta-heuristic optimization algorithms.	21
3.1	Single Node Mobile-Fog Computing resource allocation system architecture.	36
3.2	The proposed Single Node fog computing resource allocation system steps.	37
3.3	Multiple Fog computing resource allocation structure.	40
3.4	The proposed Multiple fog computing system steps.	41
3.5	The proposed HHO optimization algorithm	46
3.6	The proposed Particle Swarm Optimization (PSO) algorithm	50
3.7	The proposed Hybrid HHO-PSO optimization algorithm	54
4.1	Single Node Network configuration.	58
4.2	Packets sent from Fog node resource allocator to Server 1.	75
4.3	Packets sent into server 2 in the active servers.	75
4.4	Packets divided among active servers.	76
4.5	Proposed Single Node fog system comparison with latency, response time, and loss rate.	81
4.6	The proposed Single Node fog system comparison with channel allocation and network usage.	81
4.7	Proposed Single Node fog system comparison with resource	82

	utilization and fitness value.	
4.8	The proposed Single Node fog system comparison with average execution cost and processing time.	82
4.9	Multiple Fog computing network configuration.	84
4.10	Beacon messages between mobile nodes and network elements.	85
4.11	Redirect packets from Fog to the servers by PSO Optimizer.	90
4.12	Redirect packets from Fog to the servers by HHO Optimizer.	95
4.13	Redirect request from Fog to active servers with Hybrid approach.	1000
4.14	The proposed Multiple fog system comparison with latency, response time, and packet loss rate.	104
4.15	The proposed Multiple fog system comparison with Channel resource allocation and network usage.	105
4.16	The proposed Multiple fog system comparison with total average of resource utilization and fitness time.	105
4.17	The proposed Multiple fog system comparison with average processing time and execution cost.	106

<b>List of Algorithms</b>		
<b>Algorithm No.</b>	<b>Title</b>	<b>Page No.</b>
3.1	The pseudocode for Harris hawks optimization algorithm	44
3.2	Task-scheduling based on Particle Swarm Optimization (PSO) algorithm	49
3.3	The pseudocode for the proposed hybrid Better Fitness Load Balancing (BFRA)	52

## List of Abbreviations

Abbreviation	Description
<b>ACO</b>	Ant Colony Optimization
<b>ABC</b>	Artificial Bee Colony
<b>AI</b>	Artificial Intelligence
<b>BS</b>	Base Stations
<b>BHO</b>	Black Hole Algorithm
<b>CGWO</b>	Chaotic Gray Wolf Optimization
<b>CSA</b>	Cuckoo Search Algorithm
<b>DE</b>	Differential Evolution
<b>EDRAM</b>	Dynamic Resource Allocation Method based on Particle Swarm Optimization
<b>EC</b>	Edge Computing
<b>EP</b>	Evolutionary Programming
<b>5G</b>	Fifth generation technology
<b>FANETs</b>	Flying Ad-hoc Networks
<b>FSA</b>	Fish Swarm Algorithm
<b>FPA</b>	Flower Pollination Algorithm
<b>FNs</b>	Fog Computing Nodes
<b>GACO</b>	Generalized Ant Colony Optimizer
<b>GA</b>	genetic algorithm
<b>GP</b>	Genetic Programming
<b>GUI</b>	Graphical User Interface
<b>GSA</b>	Gravitational Search Algorithm
<b>HS</b>	Harmony Search
<b>HHO</b>	Harris Hawks Optimization
<b>HAS</b>	Harmony Search Algorithm
<b>IOT</b>	Industrial Internet of Things
<b>IoT</b>	Internet-of-Things
<b>IWO</b>	Invasive Weed Optimization
<b>LGFP</b>	Longest job to fastest processor
<b>MEC</b>	Multi-Access Edge Computing
<b>NOMA</b>	Non-orthogonal Multiple Access
<b>OMNET++</b>	Objective Modular Network Testbed in C++
<b>PPO</b>	Proximal Policy Optimization
<b>PSO</b>	Particle Swarm Optimization
<b>QoE</b>	Quality of Experience

<b>RL</b>	Reinforcement Learning
<b>RATEC</b>	Resource Allocation Technology for Edge Computing
<b>RSU</b>	road side unit
<b>SLA</b>	Service Level Agreement
<b>SA</b>	Simulated Annealing
<b>UAVs</b>	Unmanned Aerial Vehicles
<b>VFC</b>	Vehicle Fog Computing
<b>V2X</b>	vehicle-to-everything

## **List of Thesis Related Publications**

**Name of Conference: Sixth International Iraqi Conference on Engineering Technology (6th IICETA-2023).**

**Journal : Journal of University of Babylon for Pure and Applied Sciences**

- **1<sup>st</sup> Paper Title: Fog-based Resource Allocation Hybrid Approach using Metaheuristic for Mobile Networks.**
- **2<sup>nd</sup> Paper Title: Multiple Fog-based Resource Allocation Hybrid Approach using Metaheuristic Optimizers for Mobile Networks.**
- **Authors:**
  - Zahraa Husam Ebrahim
  - Asst. Prof. Dr. Mehdi Ebady Manaa

Information Network Department, Information Technology College, Babylon University.

Email: [zahrah.net.msc@student.uobabylon.edu.iq](mailto:zahrah.net.msc@student.uobabylon.edu.iq)

Email: [it.mehdi.ebady@itnet.uobabylon.edu.iq](mailto:it.mehdi.ebady@itnet.uobabylon.edu.iq)

# ***Chapter One***

## ***General Introduction***

## 1.1 Overview

Networks facilitate cost savings and revenue generation for individuals and corporations on a global scale. Rather than maintaining multiple copies of data across various locations within the organization and ensuring their consistent updates. An organization that employs a network infrastructure has the ability to maintain a singular, Single Node version of its data, which can be accessed and utilized by multiple users [1].

The network primarily consists of hardware components, including hubs, switches, computers, fog computing devices, servers, routers, and other infrastructure devices. These devices have a significant function in the transmission of data between locations through various technologies, including wired and wireless means [2].

Fog computing is an emerging Multiple computing model that provides end-users with storage, communication, and computationally services in close proximity. Fog computing is a novel paradigm that was first proposed by CISCO, which serves as an extension of cloud computing. It aims to cater to the increasing requirements of internet users [3]. It achieves this by processing data in close proximity to mobile network devices, rather than transmitting the data to the cloud [4]. The primary aim of resource allocation in fog computing is to assign optimal resources to tasks generated by edge devices, thereby ensuring the fulfillment of quality of service (QoS) requirements [5].

The categorization of resource allocation strategies is typically comprised of five primary classifications [6]. The aforementioned functionalities encompass application placement, scheduling of resource

deployment, resource allocation loading, and load balancing. The literature has identified several key performance metrics, including CPU utilization, response time, throughput, power consumption, latency, data traffic, service migration ratio, and number of active nodes [7].

The fog computing framework is composed of three layers, specifically the cloud, fog, and device layers, as illustrated in Figure 1.1 [8] :

- 1- The cloud layer is composed of a variety of servers, various types of servers, including cloud servers, application servers, and data servers, are commonly utilized alongside data centers and operation centers [9].
- 2- The fog computing it is comprised a variety of network equipment, including switches, routers, base stations, and gates, that possess computational capabilities [10].
- 3- The device layer encompasses a variety of portable electronic devices that are commonly utilized by individuals. It is used to collect data and transmit it to the highest level, specifically the fog layer [11].

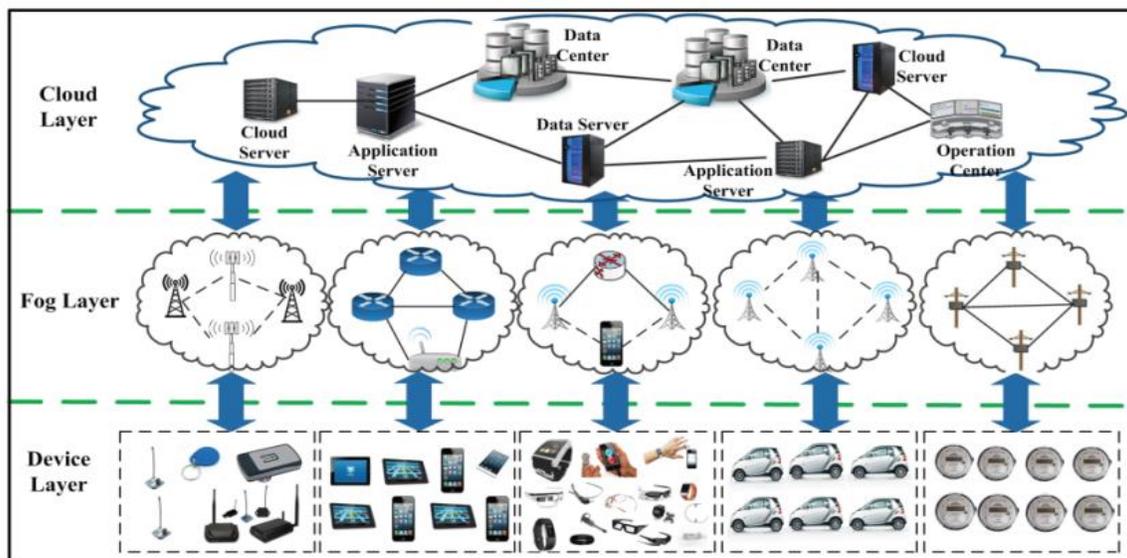


Figure 1.1: Three layers of Mobile fog computing ( Cloud, Fog, and Device)[12].

## 1.2 Problem Statement

The problems statement of the thesis are:

1. The Mobile devices generation massive amount of data that affects connection bandwidth in application such as health care, military surveillance, smart homes, and many more.
2. Low latency, and low execution time is a critical requirement for real time applications.
3. Increasing demand and requirements for enhancing the performance, fog computing scenario requires increased output, less latency, greater network performance etc.

## 1.3 Related Work

This section reviews the most relevant Fog-based resource allocation techniques for mobile networks:

The utilization of a genetic algorithm (GA) was proposed by Li et al., specifically to facilitate job planning and heterogeneous distribution of resources across multiple devices within wireless Internet of Things (IoT) networks. The decision-making process for offloading data to Fog Nodes (FNs) is a crucial aspect that must be considered by Massive IoT devices. Non-orthogonal Multiple Access (NOMA) is utilized in IoT networks to facilitate numerous device connections and expedite high-volume data transmission while operating within constraints of low latency and limited bandwidth. According to the simulations conducted, it has been determined that the proposed system exhibits favorable characteristics in terms of throughput, latency, outage likelihood, and power consumption [13].

A heuristic technique for effective issue formulation was suggested by Lee et al. to distribute limited fog resources to car applications using parked automobiles to reduce service delay. The reinforcement learning (RL) based Vehicle Fog Computing (VFC) resource allocation algorithm uses RL and the proximal policy optimization (PPO) algorithm to continuously learn the dynamic vehicle environment and adjust allocation decision resources. The Vehicle Fog Computing (VFC) resource allocation technique achieves greater service satisfaction than standard algorithms in simulations [14].

A minimal latency dump choice approach using latency-based genetic simulation was described by Wang et al. Due to the fast increase of sensors, computing may be offloaded to reduce network strain and deliver low-latency services. Numerical findings show that our suggested method outperforms competing schemes in completion time, energy consumption, and convergence speed and quality [15].

Hybrid Gradient Descent Spider Monkey Optimization (HGDSMO) algorithm was proposed by Seethalakshmi et al. to efficient resource scheduling. The proposed HGDSMO algorithm uses the Gradient Descent and foraging and social behavior of the spider monkey optimization algorithm involved in the objective of effective resource allocation. The simulation results of the proposed HGDSMO algorithm confirmed to be potent in throughput, load balancing and makespan compared to the baseline hybrid meta-heuristic resource allocation algorithms used for investigation [16].

An energy-aware metaheuristic algorithm was proposed by Abdel-Basset et al. based on a Harris Hawks optimization algorithm and a local search strategy (HHOLS) for Task Scheduling Fog computing (TSFC) to improve the QoSs provided to the users in Industrial Internet of Things (IIoT) applications. For further improvements, a local search strategy is integrated with HHOLS. They compare HHOLS with other metaheuristics using various performance metrics, such as energy consumption, makespan, cost and flow time. The proposed algorithm gives superior results in comparison with other algorithms [17].

To decrease delay, processing time, and Service Level Agreement (SLA) approaches, Naha et al. suggested a multi-criteria resource allocation approach with resource reserve. This approach considers fog computing aspects such device heterogeneity, resource restrictions, mobility, and changing user requirements. They employed numerous goal functions to discover resources for fog-related time-sensitive jobs. The used strategy reduces latency by 51%, according to empirical evidence. For fog-based time-sensitive applications, the suggested technique lowers processing time and SLA approaches [18].

An extended particle swarm optimization (EPSO) algorithm was developed by Potu et al. with an extra gradient method to optimize the task scheduling problem in cloud-fog environments. They conducted extensive experiments on the iFogSim simulator in terms of makespan and total cost. They compared the performance of the proposed EPSO method with that of other traditional techniques, such as ideal PSO and modified PSO; the results demonstrated that EPSO achieved a makespan of 342.53 s. Thus, it

can be concluded that the performance of the proposed method is comparable to that of other approaches [19].

The Chaotic Gray Wolf Optimization (CGWO) algorithm introduced by Brahmi et al. which is a meta-heuristic and evolutionary algorithm, as a potential solution for addressing the resource allocation problem. The study also includes a comparative analysis between the proposed approach and other descriptive algorithms, namely PSO and GWO for resource allocation. The simulation outcomes demonstrate the efficacy of the proposed resource allocation mechanism. The resource allocation system based on CGWO demonstrates superior total productivity compared to schemes based on PSO and GWO [20].

An improved initialization of particle swarm optimization (PSO) using heuristic algorithms was proposed by Alsaïdy et al. with Longest job to fastest processor (LJFP) and minimum completion time (MCT) algorithms are used to initialize the PSO. The performance of the proposed LJFP-PSO and MCT-PSO algorithms are evaluated in minimizing the makespan, total execution time, degree of imbalance, and total energy consumption metrics. Simulation results revealed the effectiveness and superiority of the proposed LJFP-PSO and MCT-PSO compared to the conventional PSO and comparative algorithms [21].

Table 1.1 shows different researchers' goals and their primary methodologies of fog computing resource allocating approaches.

Table 1.1: The literature survey summary.

Ref, Year	Methods	Aims	Results
[13], 2019	NOMA	To tackle the challenge of scheduling tasks and distributing diverse resources among multiple devices in wireless IoT networks.	The results showed that the average delay is 0.5, execution time is 15 seconds
[14], 2020	Heuristic algorithm, Proximal Policy Optimization (PPO) algorithm	To allocate limited fog resources to vehicle applications	VFC resource allocation method improves service satisfaction.
[15], 2020	Minimal latency dump decision algorithm	Improving the dump decision to get the optimum dump decision with the least latency.	Completion time is 1.4 seconds, and total energy is 2 J.
[16], 2020	Hybrid Gradient Descent Spider Monkey Optimization (HGDSMO)	Efficient resource scheduling.	Makespan of the HGDSMO resource allocation algorithm is estimated to be excellent by 4.82%, 5.78% and 6.94%, remarkable to the compared HGDCSO, IHS-CSO and MGWO schemes
[17], 2020	HHOLS	Energy-aware metaheuristic algorithm for industrial-Internet-of-Things task scheduling problems	Average of CPU time is 8.7 seconds of HHOLS, and makespan is 80 seconds.
[18], 2021	multi-criteria resource reservation policy	decrease delays, processing time, and SLA breaches	The approach they used reduces latency by 51%, according to empirical evidence.
[19], 2021	Extended particle swarm optimization (EPSO)	Optimize the task scheduling problem in cloud-fog environments.	Achieved a makespan of 342.53 s. Thus, it can be concluded that the performance of the proposed method is

			comparable to that of other approaches.
[20],2021	Meta-heuristic evolutionary algorithm CGWO	Fix V2X resource allocation.	PSO-based resource allocation. With $c1 = 0.75$ and $c2 = 0.75$ , the scheme reaches the best objective value of throughput.
[21], 2022	PSO, LJFP and MCT algorithms	Minimizing the makespan, total execution time, degree of imbalance, and total energy consumption metrics	Total execution time performance showed the PSO is 931.1, SJFP-PSO is 966.4, LJFP-PSO is 959.9 and MCT-PSO is 972.3. Makespan performance as PSO is 20.7, SJFP-PSO is 21.8, LJFP-PSO is 22.8 and MCT-PSO is 17.6.

## 1.4 Aim of Thesis

The main of this thesis is the implementation of a resource allocation approach in fog-server computing environment. This aim can be achieved through the following objectives:-

1. Proposing a hybrid resource allocation approach based on metaheuristic represented by PSO and HHO algorithm for good resource utilization and improve network performance by decreasing the channel usage, processing time, computational time and fitness value.
2. Implementing and simulating the proposed approach with single and multiple fog nodes.
3. Evaluating the proposed approach in terms of channel usage, processing time, computational time, execution cost, fitness value, lost packets, and requests speed for file uploading.

## 1.5 Thesis Outline

The thesis is divided into four chapters in addition to chapter one :

**Chapter Two:** It showed the architecture of Fog Computing, Load Balancing Techniques, Meta-heuristics algorithm for resource allocation as HHO and PSO algorithm, Resource Allocation Evaluation Metrics, and common simulation tools for resource allocation in Fog computing environments.

**Chapter Three:** It presents the used system and illustrates the working steps of the system as the used architecture for designing and implementing the proposed method.

**Chapter Four:** It shows the results of the suggested algorithm with the result explanations.

**Chapter Five:** The document outlines the primary findings of the study and proposes potential avenues for future research.

# *Chapter Two*

## *Theoretical and Technical Background*

## 2.1 Introduction

This chapter explained the Fog Computing concept, The architecture of Fog Computing, Resource Management approaches in Fog environment, Resource Allocation in Fog computing, Methuristics algorithm for resource allocation, Harris Hawks Optimization (HHO) algorithm, Particle Swarm Optimization (PSO), and common resource allocation evaluation metrics.

## 2.2 Fog Computing

Fog computing is a distributed model that extends beyond the cloud and provides computing and networking functionalities to the edge of the network, in close proximity to end-users and Internet of Things (IoT) devices [26]. This is achievable by leveraging extensively Multiple fog nodes. Cloud-centric architectures involve the transmission of a considerable amount of data to cloud servers for processing, analysis, and storage. The present transmission has the potential to adversely affect various aspects such as latency, mobility, and reliability [27].

The advent of location-aware and delayed-sensitive applications has presented obstacles for cloud computing platforms to meet the rigorous latency requirements of these applications [28]. The close proximity of the fog layer to IoT devices presents the possibility of substantially decreasing latency and accommodating the needs of applications that demand extremely low latency [29]. Fog computing is recognized for its ability to facilitate communication with the cloud and offer assistance, resulting in the development of novel types of applications and services. In modern fog computing environments, it is imperative for end-users to have access to applications that provide rapid response times to ensure smooth accessibility and optimal performance[30].

An effective approach to enhancing QoS metrics within a fog network is through the implementation of a proficient load balancing strategy, given the significant role that load balancing plays in this context. Due to their heterogeneous and dynamic nature, fog networks require load balancing mechanisms to distribute incoming loads among available fog nodes or cloud resources [31]. This approach helps prevent overloading or underloading of fog nodes. This mechanism has the potential to optimize throughput, performance, and resource utilization, while simultaneously minimizing response time, cost, and energy consumption. The diagram depicted in Figure 2.1 the interrelationships that exist between fog computing and Internet of Things (IoT) applications [32].

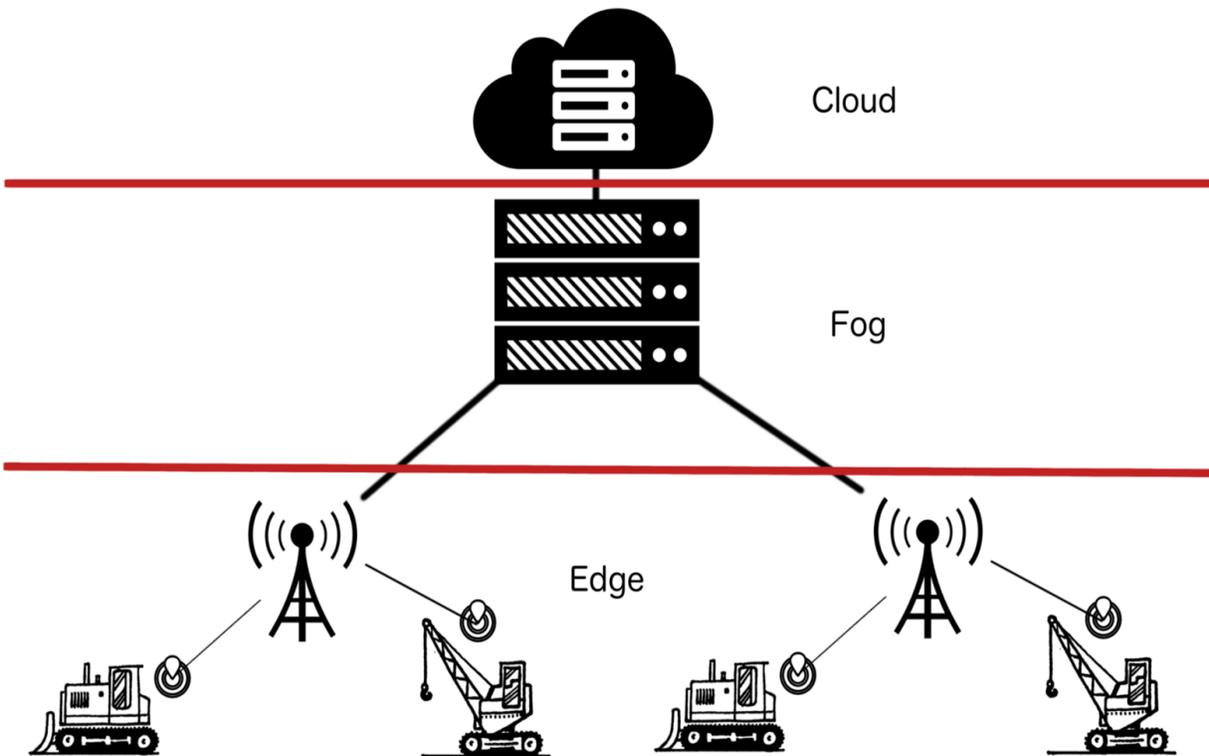


Figure 2.1: System model for Fog Computing and IoT [32].

To differentiate fog computing from different computing requirements, it is imperative that an IoT client or smart end-device exhibit specific characteristics, although not all of them are necessary when utilizing a fog computing service [33].

- Fog computing provides a low latency solution by taking into account the contextual location awareness of fog nodes. This awareness includes the consideration of the latency costs associated with communication between nodes, as well as their logical location within the overall system context. Low latency is a crucial requirement for applications in fog computing, which facilitates and directs resource-rich services at the network edge [34]. The processing and reaction time of data generated by fog nodes is notably faster in comparison to that of a Single Node data center or cloud service, primarily due to their close proximity to the intelligent user devices [35].
- The geographical distribution aspect of fog computing is characterized by the provision of services to Multiple deployments that are identifiable by their geographic location, as opposed to a more Single Node [36].
- Heterogeneity is addressed by fog computing, which supports the gathering and processing of a wide range of data from components acquired through diverse network communication capabilities [37].
- The integration of services across domains and the ability of fog computing components to interoperate are crucial for supporting certain services, as they require the collaboration of multiple providers [38].
- Real-time interactions are utilized in fog computing applications, as opposed to batch processing [39].
- Fog computing exhibits inherent adaptability in various aspects such as cluster or cluster-of-clusters, network conditions, data-load fluctuations, resource pooling, and elastic compute support, among others. This

adaptability enables agility and scalability of federated fog-node clusters [40].

- The Fog system is primarily situated within wireless IoT access networks, where wireless access is prevalent. Despite the fact that fog computing is employed in wired settings, a considerable quantity of wireless Internet of Things (IoT) sensors necessitate dispersed computation and analysis [41].
- The provision of mobility support entails the implementation of mobility techniques such as the locator/ID separation protocol. This technique facilitates the separation of the host's identity from its location identity. To achieve this, a Multiple directory system is required, which is essential for various applications in fog computing that enable direct communication with mobile devices [42].

### **2.2.1 The architecture of Fog Computing**

A heterogeneous range of architectural designs has been suggested for fog computing, predominantly originating from a three-part composition. The fog architecture is characterized by a hierarchical structure consisting of three layers [43].

- The cloud layer in computing : it is comprised of a variety of storage devices and high-performance servers, which facilitate the creation of multiple application services [44].
- The fog layer : it is situated at the periphery of the network and comprises several fog nodes, such as access points, routers, switches, and gateways. The data is Multiple across both cloud and endpoint devices. End devices can establish connections with fog nodes to access services. These devices possess the ability to perform computations, store information, and transmit data that has been sensed and received. Fog computing can facilitate the

execution of latency-sensitive applications and enable real-time analysis within its layer [44].

- User device layer : it is situated in immediate proximity to both the physical environment and the end-user. It can represent by IoT devices, including sensors, mobile phones, smart vehicles, cards, and readers. Despite possessing computing capabilities, cellphones and smart vehicles are primarily utilized as smart sensing devices. In this architectural framework, various end devices or intelligent objects are linked to fog nodes through wired or wireless connectivity technologies, including 3G, 4G, wireless LAN, ZigBee, Bluetooth, and Wi-Fi. Wireless and wired communication technologies are utilized to facilitate the interconnection and intercommunication among fog nodes [45]. Figure 2.2 shows Fog architecture.

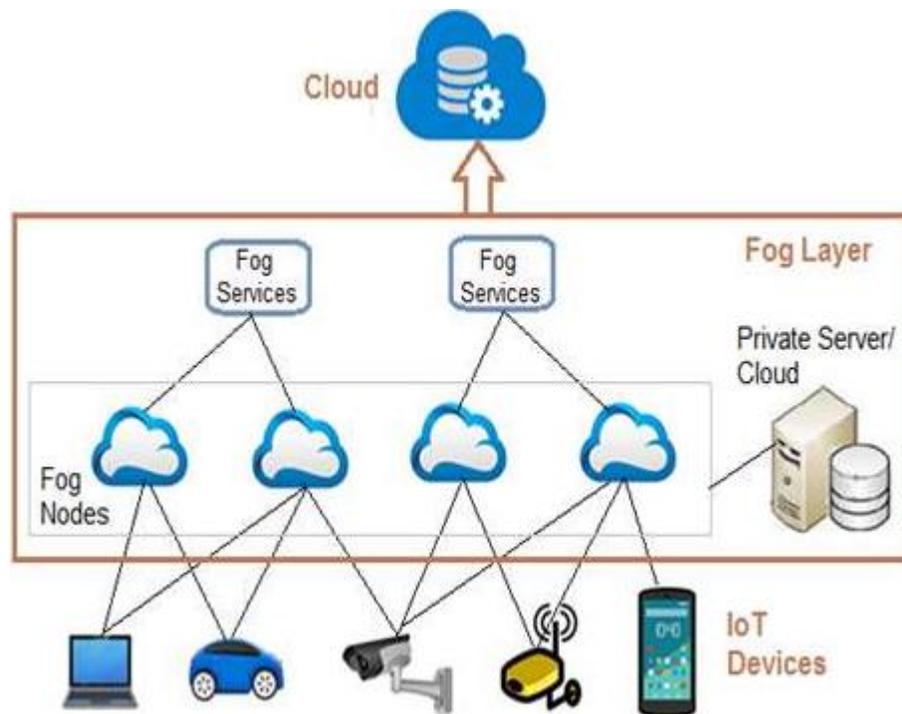


Figure 2.2 : Fog computing architecture[45].

## 2.2.2 Benefits of Fog Computing

The main benefits of the fog computing is summarized as the following :

### 1. Reduced Latency

Latency emerged as a primary concern for enterprises utilizing cloud computing. In cases where the duration for data transmission to reach the recipient is prolonged, it has the potential to not only diminish customer satisfaction levels but also pose a risk to human life [46].

### 2. Enhanced Security

Fog computing is an infrastructure for computing that is distributed Node, whereby servers are Multiple across strategically determined locations. The intricacy of these systems may present difficulties in terms of unauthorized access and interference [47].

### 3. Improved Customer Experience

The primary objective of any enterprise is to offer exceptional services to its clientele. Fog computing enables minimal latency in data transmissions, thereby facilitating prompt response and assistance for all customer requests [48].

### 4. Scalability

A primary obstacle faced by smart cities pertains to the analysis and computation of voluminous data generated incessantly by myriad sensors embedded across the urban landscape. Therefore, the implementation of fog computing can be facilitated by the conducive environment provided by smart cities. Fog nodes are designed to perform localized data processing, thereby obviating the need to transmit all data to cloud servers [49].

### 2.2.3 Fog Computing Issues

Fog computing is a Multiple computing paradigm that enables fog nodes to process and offload tasks requested by IoT-enabled devices in an IoT-based system. Notwithstanding the potential advantages, the attainment of said benefit remains a formidable task in systems characterized by a high volume of requests, which consequently give rise to protracted task queues within the fog nodes [50]:

- A. The environment of fog computing is characterized by complexity and dynamic variability. The fog computing devices, such as gateways, hubs, and switches, exhibit heterogeneity in their computational, storage, and communication capabilities [51].

Figure 2.3 depicts a common challenge encountered in fog computing environments, wherein a fog node (F2) is incapable of processing the complete input data of a service request (A) owing to resource constraints. In the interim, delegating the aforementioned task to the neighboring fog nodes F1 and F3 could potentially result in significant delays, as the queues of said fog nodes are currently burdened with substantial workloads [51].

- B. The elevated frequency of requests has the potential to extend the task queues within the potent fog nodes. This is due to the limited resources of the fog nodes in terms of computational capability and storage, which may render them incapable of processing the entire input data of the service [52].

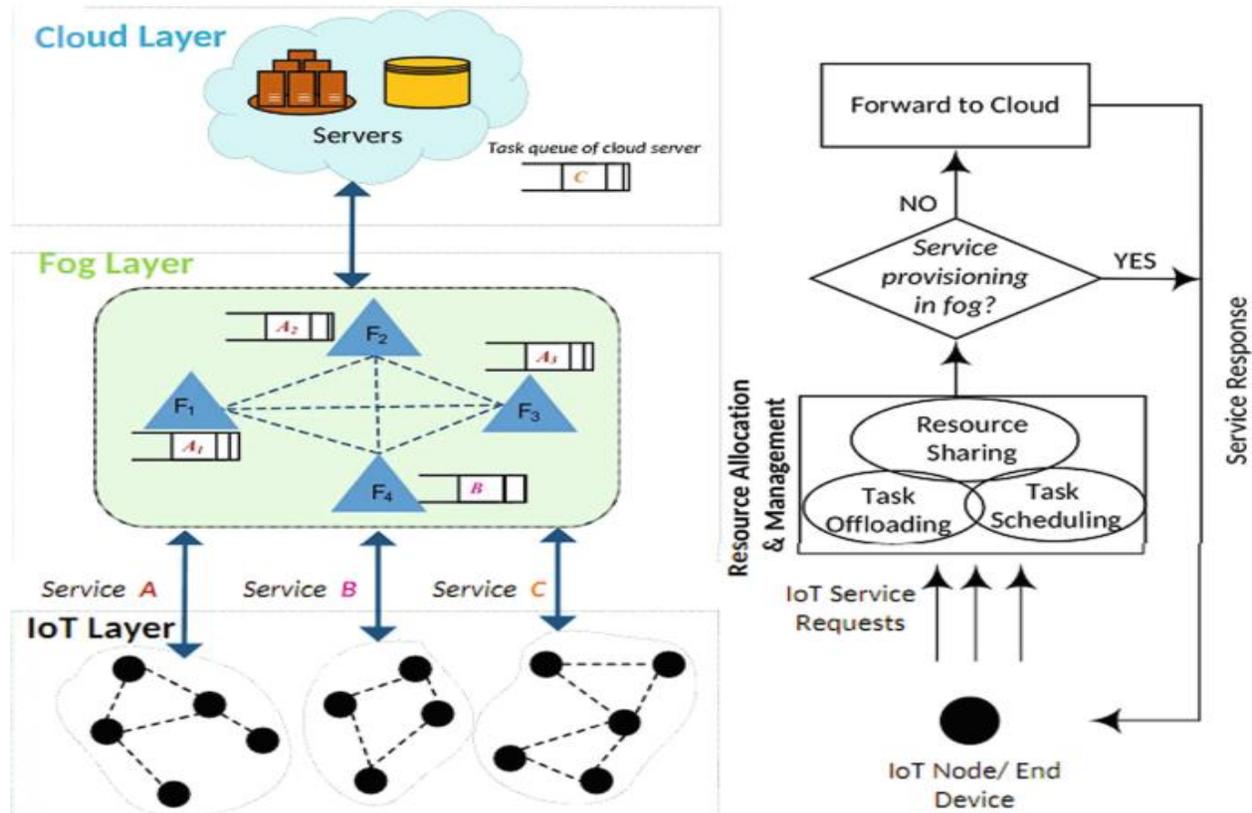


Figure 2.3: IoT-fog-cloud systems of fog environment heterogeneity and imbalanced workload[52].

- C. Mobility alters fog computing resources. In many actual applications, drones and vehicles cause a dynamic shift in fog computing resources. Fog devices departing and joining fog computing systems also caused this shift. To handle this, resource allocation systems must be flexible and adaptable [52].
- D. Due to different jobs, resource needs also fluctuate dynamically. Application, timing, and environment affect resource demand. Resource sharing, task scheduling, and task offloading are the three main fog computing resource allocation concerns. Resource sharing involves fog machines pooling resources to meet computing demands. The fog computing system is a pool of storage, CPU, and GPU resources, thus nodes must

cooperate to complete computational tasks. To enable resource sharing, fog computing domains need fog-to-fog collaboration protocols [52].

#### **2.2.4 Fog-computing in IoT-Mobile network applications**

Open fog computing supports IOT innovation. Fogs secure Multiple architecture by connecting cloud and device. Fog computing on Flying Ad-hoc Networks (FANETs) allows data storage, transmission, and computation services and applications. Fog computing can provide FANET mobile nodes with local connections and low latency [53].

Online gaming, augmented reality, and virtual reality, which need high bandwidth, low reaction time, and computer capacity, have evolved in the previous decade. Such a system uses remote cloud services to handle data, which causes delay [54].

#### **2.2.5 Resource Management approaches in Fog-IoT Environment**

The quality of life of individuals has been enhanced in recent times through the utilization of IoT applications such as smart homes, self-driving cars, smart agriculture, and smart healthcare. The proliferation of IoT applications has led to a corresponding rise in the number of IoT devices, including sensors, smart CCTV cameras, smart gadgets, and other intelligent devices. The technology under consideration facilitates the ability to determine location, provides support for user mobility, enables real-time interactions, offers low latency, high scalability, and ensures interoperability, which are not feasible with cloud-based systems. The proliferation of Internet of Things (IoT) applications and the scarcity of resources in fog/edge computing environments have rendered effective resource management a critical concern [55].

### **2.3 Resource Allocation in Fog computing**

Resource allocation in fog computing varies from the traditional to Multiple computing environment due to the presence of various QoS metrics such as CPU memory, speed, and stability. In dealing with resource allocation, multiple requests will be in the queue, waiting to be served at various stages[56].

Resource allocation is a step in the resource management process, which comprises several other steps in addition to allocation such as estimating, discovering and monitoring[57]. Resource management is relevant in several areas of research because it aims at the optimized use of available resources. This process is composed of steps that together intend to use a reasonable amount of computational resources and also to do it in an easy and efficient way [58]. The resource management process is widely discussed in fog computing since computational resources are often limited and must be well used [59], unlike cloud computing that gives the perception of having infinite resources, in the point of view of a single user [60].

One of the main requirements in managing computing resources is the ability to estimate how many resources will be needed to perform a task. Thus, given the restriction of computational capabilities of fog nodes, the estimation process plays a fundamental role in the allocation and optimal use of resources in fog computing [61].

### **2.4 Methuristics Algorithm for resource allocation**

A metaheuristic is a high-level problem-independent algorithmic framework that provides a set of guidelines or strategies to develop heuristic optimization algorithm. It is used to find, a good solution to an optimization problem that is complex and difficult to solve to optimality [62].

A metaheuristic algorithm refers to a search methodology that is specifically developed to obtain a favorable solution for an optimization problem that is intricate and challenging to achieve optimal results. In the context of limited resources, such as computational power and time, it is crucial to identify a solution that is nearly optimal despite the presence of imperfect or incomplete information in the real world. The field of metaheuristic optimization pertains to the resolution of optimization problems through the utilization of metaheuristic algorithms. Numerous optimization algorithms exist and can be categorized in various manners. Based on their focus and characteristics, these algorithms can be classified as either local or global search algorithms. Furthermore, it is noteworthy that all metaheuristic algorithms employ a specific balance between randomization and local search [62].

The categorization of meta-heuristic methodologies can be demarcated into five discrete groups, which are derived from various sources of inspiration [63].

- A- Bio-inspired involves the use of external stimuli to enhance biological processes.
- B- Nature-inspired approaches draw inspiration from natural systems and processes to solve complex problems.
- C- Physics-based methods utilize principles and theories from the field of physics to develop innovative solutions.
- D- Evolutionary techniques involve the application of principles from evolutionary biology to optimize.
- E- Swarm based algorithms it is a part of inspired by nature it used to solving tricky problems. Figure 2.4 provides an illustration of the taxonomy of

several meta-heuristic optimization methods with the proposed Harris Hawks Optimization (HHO) added within nature inspired algorithm[63].

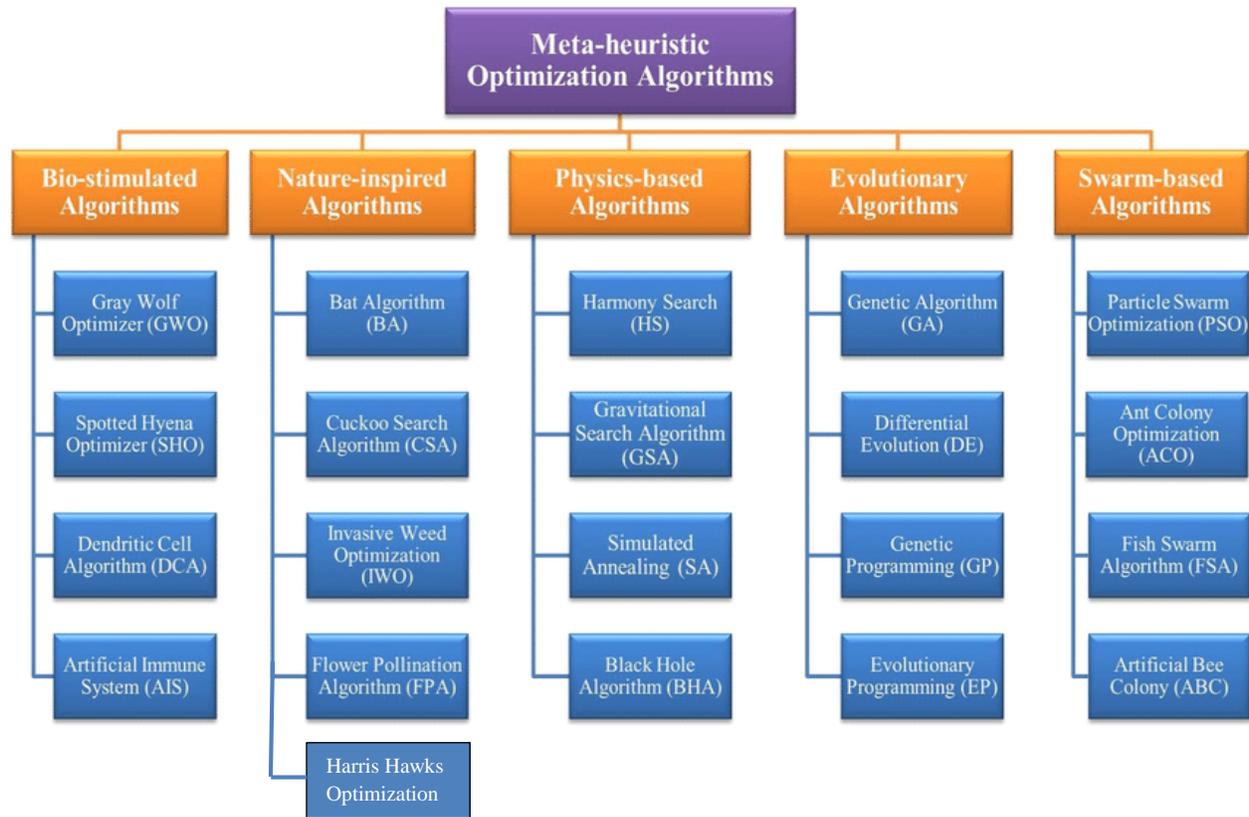


Figure 2.4: Taxonomy of meta-heuristic optimization algorithms[63].

The used meta-heuristic optimization algorithms are :

### 2.4.1 Harris Hawks Optimization (HHO) algorithm

The HHO algorithm exhibits similarities to other metaheuristic algorithms. The utilization of HHO exhibits a multitude of advantages, while its drawbacks are relatively limited in number. The advantages of HHO can be enumerated as follows [63]:

- The convergence speed is commendable.
- The neighborhood search feature exhibits a high degree of efficacy.
- There exists a favorable equilibrium between the act of exploring and exploiting.

This approach exhibits versatility in addressing a wide range of issues and is characterized by its straightforward implementation[63]. The attributes of adaptability, scalability, flexibility, and robustness are noteworthy in various contexts [63].

Usability pertains to the extent to which users are able to effectively utilize HHO for the purpose of addressing optimization scenarios. The aforementioned criterion was subsequently partitioned into three distinct sub criteria [64]:

The term "ease of use" pertains to the level of simplicity involved in the implementation and execution of HHO. As demonstrated in Algorithm 1, the HHO methodology's logic is straightforward and does not necessitate substantial computational effort for each iteration. Moreover, the computational procedures involved in HHO require minimal effort to comprehend and can be readily implemented by a majority of interested users in practical applications[65].

The utility of HHO is evidenced by its capacity to improve optimization solutions. The study utilized to demonstrate the application of HHO in 12 distinct domains. The aforementioned domains encompass enhanced HHO, the manufacturing industry, environmental quality, solar photovoltaic, power systems, various engineering fields, data mining and processing, image thresholding, networking and Multiple systems, software engineering, drug design and discovery, and the classification of mammogram images. These domains were categorized under three main fields of study, namely engineering, computer science, and medicine and public health, and were associated with various optimization problems. Also, it demonstrated a unanimous agreement among the

studies regarding the efficacy and utility of the HHO algorithm in addressing optimization problems [66].

Flexibility refers to the capacity of HHO to undergo enhancements such as modification, hybridization, multiobjective optimization, binarization, and chaotic optimization, among others [67].

One of the drawbacks associated with HHO, similar to other algorithms, is its susceptibility to becoming trapped in local optima. Additionally, there is a lack of a formalized convergence analysis framework for this method [67].

One potential issue is that when dealing with complex problems, the population of the HHO may become trapped in local optima. In such cases, it may be necessary to implement appropriate modifications to enhance the algorithm's exploration capabilities [68].

As previously stated, the HHO functions based on the server hunting technique employed by Harris Hawks. The HHO algorithm conventionally executes both exploration and exploitation phases to address potential scenarios that may arise during the hunting process. The HHO algorithm designates the candidate solutions as hawks, while the optimal solution or server is regarded as the prey [68].

Typically, the HHO performs the exploration and exploitation steps regarding the situations which may happen during the hunting. In the HHO algorithm, the candidate solutions are considered hawks, and the server is the best solution. The exploitation and exploration steps in the HHO algorithm are controlled by the  $E$  or energy factor, computed by Equation (2.1). In each iteration, the exploration will be performed when  $|E| < 1$ ; otherwise, the exploitation should be conducted. In the exploration step, the hawks move to the randomly selected

locations and wait to detect the prey. At first, an equal chance, indicated by  $q$ , is considered for each perching strategy [68]:

$$X(t+1) = \begin{cases} X_{rand}(t) - r1|X_{rand}(t) - 2r2X(t)| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r3(LB + r4(UB - LB)) & q < 0.5 \end{cases} \quad (2.1)$$

In which the hawk's current position is  $X(t)$ , the hawk's next position is  $X(t+1)$ , and the server's position is  $X_{rabbit}(t)$  or  $X_{server}(t)$ . Also,  $q$  and  $r1$  to  $r4$  are random numbers between  $(0,1)$ . Also,  $X_{rand}(t)$  is a random hawk,  $UB$  and  $LB$  are considered as upper and lower bounds, and  $X_m$  is the hawks' average position[68].

#### 2.4.2 Particle Swarm Optimization (PSO)

The PSO method is a type of evolutionary computation technique was derived from the collective behavior of bird flocks. The algorithm employs multiple particles, also known as candidate solutions, that traverse the search space in search of the optimal solution. In the interim, the individuals direct their attention towards the optimal particle, also referred to as the best solution, within their respective trajectories. Stated differently, particles take into account both their individual optimal solutions and the optimal solution discovered thus far [69].

In Particle Swarm Optimization (PSO), individual particles are required to take into account several factors, including their present position, current velocity, proximity to  $p_{best}$ , and proximity to  $g_{best}$ , to effectively adjust their position. The Particle Swarm Optimization (PSO) algorithm commences by initializing a set of particles (solutions) randomly, followed by iteratively updating generations to search for optima. During each iteration, every particle is updated by considering two optimal values. The initial option, namely fitness, has thus far proven to be the most optimal resolution. The value pertaining to fitness is also retained. The aforementioned quantity is commonly referred to as "pbest" in

academic literature. The particle swarm optimizer monitors a significant metric, namely the optimal value achieved by any particle within the population, which is regarded as the "best" value. The term "gbest" refers to the globally recognized best value [70].

PSO algorithm exhibits certain limitations, such as a tendency to converge to local optima in high-dimensional spaces and a relatively slow convergence rate during the iterative process. The utilization of PSO for addressing intricate and multi-dimensional problems is acknowledged due to its computational complexity [70].

The primary benefits of the Particle Swarm Optimization (PSO) algorithm are a reduced need for parameter tuning and the potential for suboptimal solutions. The constraint is simple and straightforward, making it suitable for multi-objective optimization. They generated the concept of function-optimization by means of a particle swarm. Consider the global optimum of an n-dimensional function defined by Equation (2.2) [71].

$$f(x_1, x_2, x_3, \dots, x_n) = f(X) \quad (2.2)$$

Where  $x_i$  is the search variable, which represents the set of free variables of the given function. The aim is to find a value  $x^*$  such that the function  $f(x^*)$  is either a maximum or a minimum in the search space. Consider the functions given by Equation (2.3) and Equation (2.4) .

$$f_1 = x_1^2 + x_2^2 \quad (2.3)$$

$$f_2 = x_1 \sin(4\pi x_2) - x_2 \sin(4\pi x_1 + \pi) + 1 \quad (2.4)$$

It is clear that the global minimum of the function  $f_1$  is at  $(x_1, x_2) = (0, 0)$ . at the origin of function  $f_1$  in the search space. That means it is a unimodel

function, which has only one minimum. However, to find the global optimum is not so easy for multi-model functions, which have multiple local minima.

The primary drawbacks of the Particle Swarm Optimization (PSO) algorithm are:

- The solution provided exhibits substandard quality.
- The updating of velocity requires the utilization of memory.
- The phenomenon of early convergence [72].

## 2.5 Resource Allocation Evaluation Metrics

The proposed system is evaluated with different evaluation metrics which explained as the following :

### 2.5.1 Throughput

The load balancer's efficacy is determined by the quantity of requests that are accomplished within a given time frame. This is a metric of successful job completion. The metric of measuring throughput is insightful as it indicates a correlation between higher throughput and increased efficiency of load balancers [73].

$$\text{Throughput} = \frac{\text{Total Number of sent and received packets}}{\text{Time}} \quad (2.5)$$

### 2.5.2 Response time

It demonstrated that the outcome is contingent upon the  $M_t$  and  $E_t$ . The term "response time" refers to the duration of time that transpires between the completion of a given task and the receipt of a corresponding response.[74]:

$$R_t = M_t + E_t + N_{DL} \quad (2.6)$$

Where  $M_t$  is the migration time,  $E_t$  is the execution time and  $N_{DL}$  is the all delay.

### 2.5.3 Execution time

The performance of a computer system is contingent where factors  $E_t$  is the execution time,  $J_s$ , job task and  $CPU_s$  velocity CPUs, as expressed in Equation (2.7)[75].

$$E_t = \frac{J_s}{CPU_s} (2.7)$$

### 2.5.4 Resource Utilization

This pertains to the state wherein all of the available resources of the fog system are utilized to their maximum capacity. Resource utilization (LU), which is calculated in equation (2.8):

$$RU = \text{Min}(1, TMA/TM) (2.8)$$

Where TMA is the total allocated MIPS of the host, and TM is the total MIPS of the host[76].

### 2.5.5 Latency

The term denotes the temporal duration that elapses from the moment the load balancer receives a request until the moment it transmits a response [77].

$$d_{\text{trans}} = L/R (2.9)$$

Where d represents as delay time in seconds and L is the packet length in bits and R is the rate of transmitted data in bits per unit of time [78].

### 2.5.6 Packet Loss Rate

The metric in question pertains to the proportion of successfully transmitted packets relative to those that were not successfully transmitted due to transmission errors. Every packet is assigned a specific deadline for execution. In the event that the deadline cannot be met, the scheduler will endeavor to minimize the number of lost packets that result from the expiration of the deadline [79].

$$\text{packet Loss rate} = (\text{transmitted} - \text{received}) / \text{transmitted} * 100\% \quad (2.10)$$

### 2.5.7 Total Network Usage

The present study examines the correlation between the overall latency and the duration of the simulation. Tuples are utilized to define the input/output relationships among modules. The allocation of network resources is contingent upon the magnitude of transferred tuples during a specific temporal interval [80].

$$\text{Total Network Usage} = \frac{\text{Total Latency}}{\text{Simulation Time}} \quad (2.11)$$

### 2.5.8 Total Execution Cost

The equation can be utilized to determine the overall cost of execution (2.12) [81].

$$\text{Total Execution Cost} = \text{Processing Time} + (\text{Latency} - \text{Response Time}) * \text{Fitness Value} \quad (2.12)$$

**2.5.9 Fitness value :** The allocation of Bandwidth for individual modules is established at the onset of the simulation. In order to determine the optimal scheduling, it is necessary to minimize the fitness value, which can be computed using Equation (2.13). The term "bits per time unit" refers to the rate at which digital information is transmitted, received, or processed, typically measured in bits per second (bps) [82].

$$\text{Fitness value} = \frac{1}{\text{sum bandwidth}(\text{send}+\text{recived})+\text{Resource Utalization}} \quad (2.13)$$

## 2.6 Simulation Tools

The common simulation tools in this thesis are :

### 2.6.1 iFogSim simulation

The OMNET++ Fog modelling tool requires a framework that is deemed essential. The framework was constructed on the fundamental architecture of CloudSim, which was utilized as its foundation. The CloudSim simulator has gained significant popularity as a tool for modeling cloud computing infrastructures. iFogSim provides an extension to the abstraction of the fundamental CloudSim classes, enabling the simulation of a customized fog computing platform that encompasses a substantial number of fog nodes and Internet of Things (IoT) devices, including sensors and actuators[83].

The iFogSim platform employs a class analysis approach that enables users with limited familiarity of CloudSim to efficiently establish regulations for cloud computing pertaining to facilities, service placement, and resource allocation. iFogSim is specifically tailored to cater to users who lack prior familiarity with CloudSim. iFogSim employs the Sense-Process-Actuate paradigm and a Multiple dataflow architecture to model application scenarios within a fog computing framework. Assessing end-to-end latency, network issues, power consumption, operational expenses, and the degree of satisfaction with the quality of service is facilitated by this approach. iFogSim has been utilized extensively in various research studies to simulate resource management, mobility administration, latency management, QoE, energy management, safety administration, and QoS management in fog computing environments[83].

### 2.6.2 OMNET++

Being the foremost open-source simulator tool that features a graphical user interface, it facilitates the comprehension of the simulator's internal events. Omnet++ offers debugging and tracing capabilities in addition to its modular framework. The simulation of Fog computing network devices with diverse configurations and the application of load balancing models within the context of fog computing is feasible by integrating the Fognetsim or iFogSim framework onto an already existing framework. Moreover, it was employed in the development of the architectural design of the Fog nodes[84].

# ***Chapter Three***

## ***The Proposed Approach***

### 3.1 Introduction

This chapter presents the primary phases involved in the implementation, configuration, and installation of the proposed system. The proposed algorithms are explained as Harris hawks optimization algorithm, Meta-heuristic PSO algorithm, and Hybrid algorithm. Various simulation parameters are employed to reduce expenses related to bandwidth while simultaneously ensuring equitable distribution of workloads.

### 3.2 The Implementation Requirements

The implementation of the proposed mobile network Fog computing system is based on the following steps:

- The study involves the development of models for wireless connection in network systems
- Resource management modeling of fog computing network.
- Modeling of queuing, time delay, and network performance.

#### 3.2.1 Harris hawks optimization implementation requirements

It aimed to decrease platform expenses by devising a suitable plan for mobile users while upholding the integrity of the data gathered in the mobile network.

The process of the proposed Harris hawks optimization algorithm presented as follow :

1. initialize the location and Cost of the Fog-Server
  - Modifying the value to negative infinity is a common approach for solving maximization problems.
  - Initiate the spatial coordinates of server.

- Initialize convergence
- 2. Timer Start of HHO tackling
  - Check boundaries
  - fitness of locations
  - Update the location of server
- 3. Revise the current exactly where of server.
  - The code iterates through a range of search agents, as specified by the user.
- 4. Exploration phase
  - if  $\text{abs}(\text{Escaping\_Cost}) \geq 1$ :
    - Harris's hawks exhibit two distinct perching strategies that appear to be random in nature:
    - The `rand_Hawk_index` variable is computed by taking the floor of the product between the number of search agents and a randomly generated decimal value.
- 5- The behavior and characteristics of perch can be influenced by those of their family members.
- 6- Select a perch location on a tall tree within the group's home range at random.
- 7- Employing a two-phase approach to target the server based on its behavioral patterns.
- 8- During the first phase a series of hawks execute multiple, brief and rapid dives resulting in the successful hunting of seven prey.

### 3.2.2 Particle swarm optimization (PSO) implementation requirements

The utilization of PSO as a meta-heuristic approach for resource allocation in mobile networks has been proposed. This approach aims to improve the efficiency, search speed, and cost-effectiveness of mobile devices.

The main steps of the used PSO Meta-heuristic approach implementation presented as follow :

- 1- Evaluating current fitness
- 2- Verifying whether the present position represents an optimal individual performance.
- 3- Updating new particle velocity
- 4- The present study involves an assessment of the constant inertia weight for current weight evaluation, which pertains to the determination of the appropriate weight to assign to the previous velocity.
- 5- Revise the position of the particle in accordance with the updated velocity modifications.
- 6- Adjust maximum fog position if necessary
- 7- Determining the best fog position for request
- 8- Establishing the swarm
- 9- Beginning the optimization loop
- 10- Cycle through particles in swarm and evaluate fitness
- 11- Determining if current fog particle is the best.
- 12- Cycle through swarm and update velocities and position

### 3.2.3 Hybrid algorithm in Multiple Fog-server System

The present study outlines a proposed Hybrid algorithm that amalgamates the Harris hawks optimization algorithm and the Meta-heuristic PSO algorithm. The algorithmic process is elucidated as follows:

1. When a connection is made, this function will be triggered, if a new fog node or endpoint is connected to this fog node, this fog node will send a fog ready message
2. When this fog node receives data, data received will be triggered
3. Save this connection to connection repository
4. Distribute this task to local processing, neighbour fog node
5. Handle the received results
6. Handle the received state information of neighbour fog nodes
7. Handling the fog ready message, respond a fog ready acknowledge message.
8. Handling the fog ready acknowledge message, calculate delay
9. When a connection is lost, this function will be triggered.
10. Analyzing the incoming task, status of the task queue and neighbour fog state table to determine where to process the task.
11. Sending the task to the best neighbour fog node in case of Multiple fog nodes
12. Sending the task to server
13. Processing the task locally by the selective server
14. Using task delay to insert the task to the task queue for processing
15. Distributing the task to local processing, neighbour fog node or server based on the result of task Inspection function.
16. Return the result to the fog node that offloads the task to this fog

17. Storing the received state information of neighbour fog node in state table in Better Fitness Resource Allocation (BFRA) attribute column.
18. Preserving the linkage between a fog node and its neighboring fog node by storing it in a repository.
19. Removing the linkage between a fog nodes.
20. Specifying the period of state sharing in Multiple fog node
21. Sending the state information to neighbour fog nodes
22. Finding the best neighbour fog node with minimal estimated waiting time, number of active connection.

### **3.3 The proposed Mobile-Fog computing architectures**

The proposed system implemented with two main system architectures are represented as follow :

#### **3.3.1 The proposed Fog-Server Single Node architecture**

The proposed architecture is comprised three layers, with a Single Node fog node at its single node. The first layer is the mobile node as clients for the purpose of transmitting and receiving data. The second layer is Fog resource allocator represents and exercises control over the network component, also to allocate resources through network transmission. The third layer is the server as network service elements, analyzed the demands of mobile devices by generating responses to their requests. The main goal of the proposed Single Node methodology, which relies on Fog load balancing, is to effectively allocate the workload among multiple servers to enhance the computational capacity utilization of each server and minimize the average task response time. This is tantamount to maximizing the overall system throughput. Furthermore, reducing the waiting time of mobile nodes

has a significant impact on the cost consumption of these devices that are limited by resource constraints. The architecture of the proposed system is illustrated in Figure 3.1.

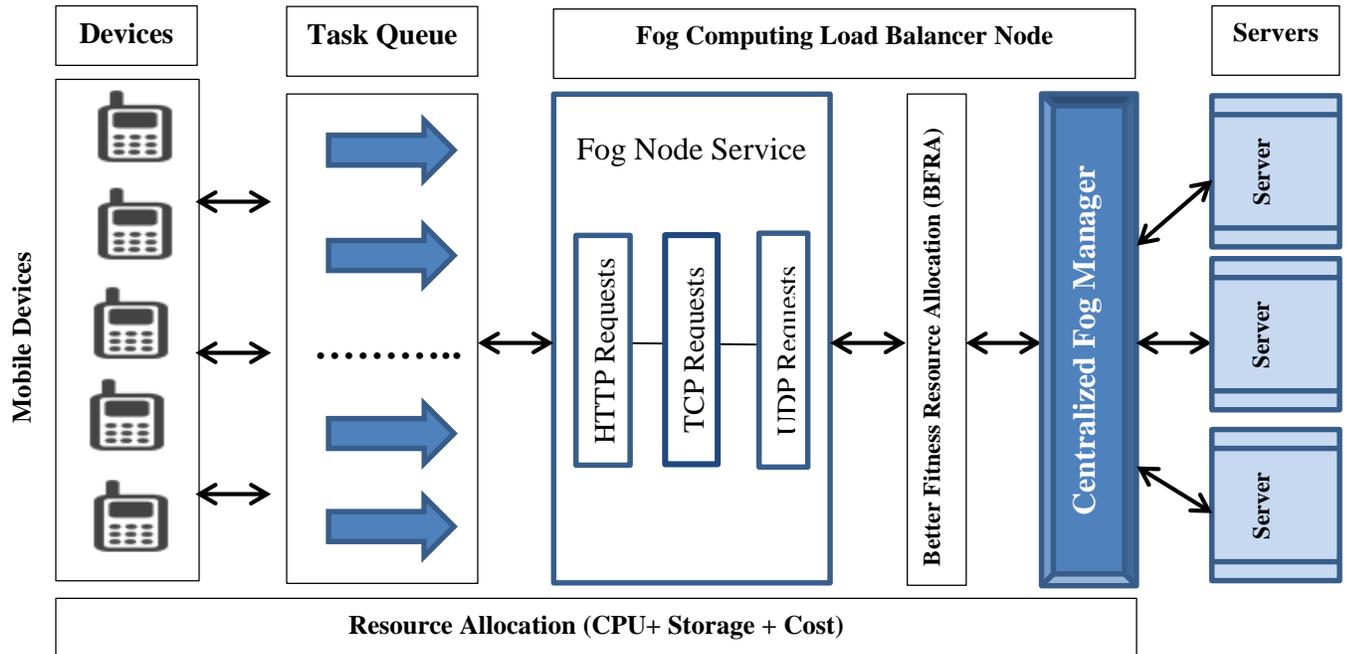


Figure 3.1: Single Node Mobile-Fog Computing resource allocation system architecture.

Mobile devices have the capability to select Fog computing nodes in order to meet their resource demands. To ensure optimal resource utilization and prevent issues such as bottlenecks, overload, and high waiting times, load balancing is a crucial factor that must be considered. However, the task of achieving resource allocation for computing nodes in the fog the environment during the execution of mobile applications remains a challenging endeavor. The variance in execution period and specifications across computing nodes within the fog infrastructure leads to suboptimal utilization of available resources. The system steps as depicted in Figure (3.2) have been proposed.

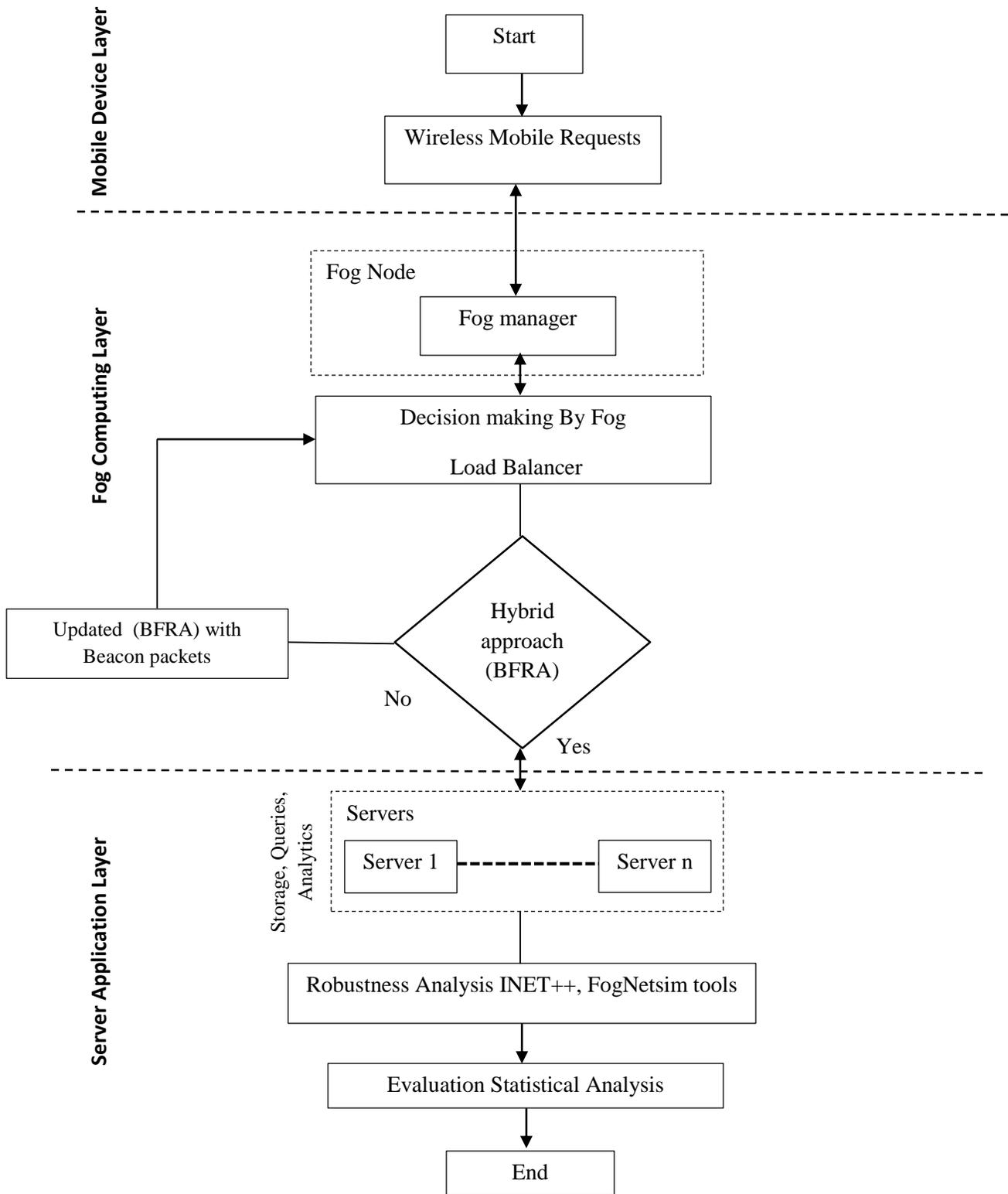


Figure 3.2 : The proposed Single Node fog computing resource allocation system steps.

### 3.3.2 The proposed Fog-Server Multiple architecture

The present study concerns the architectural design of a Multiple resource allocation system for mobile devices, which utilizes a Fog-based approach. The system comprises three distinct layers. Each of them is simulated and created with the C++ code design. The first layer is the devices layer, which consists of wireless Mobile as the Hosts to transmit / received data, access point to provide coverage area for wireless connection, router to provide routing purposes for arrived packets with base-broker as virtualize layer to provide computing services controlled by fog node as the determination of the priority of jobs to be executed first is a crucial aspect of task management.

The second layer comprises of a Multiple Fog load balancer that operates on specific Fog nodes. This layer is responsible for controlling the network components and allocating resources through network transmission. Furthermore, it facilitated optimal decision-making for time-sensitive requests (tasks) originating from the device layer, taking into account the large number of requesters, technical and quality-of-service requirements of customers, as well as the scope and constraints of server provider resources. The Hybrid optimizers (HHO and PSO) utilized in fog nodes exhibit rapid adaptability to the allocation and release of resources based on requests.

The third layer comprises servers functioning as network service components that furnish responses to individual requests made by mobile devices. Furthermore, it facilitated the consolidation of resources for the purpose of response processing and storage provision.

The primary objective of the proposed Multiple methodology, which relies on Fog load balancing, is to allocate the workload across multiple servers to enhance the computational efficiency of each server and minimize the average task response time. This is tantamount to maximizing the system throughput and optimizing network performance. The task requests are received and Multiple by the Fog nodes to all servers in the pool based on certain conditions that are determined by optimization resource allocation algorithms such as HHO algorithm, Meta-heuristic PSO algorithm, and Hybrid approach. The fundamental framework of the proposed system's Hybrid approach is contingent upon the primary function of evaluation, save individual and global optimize number of request value, update rate and fog position in the active fog pool and finally verify optimization criteria for optimal fog selection to process incoming request and redirect it to the better server selection.

The diagram presented in Figure 3.3 depicts a Multiple fog computing system that utilizes a hybrid optimization algorithm to achieve load balancing in the context of fog-based resource allocation for mobile networks. The integrated Hybrid approach of load balancing resource allocation in mobile network is enhanced the Fog-Server computing with : The concept of on-request self-service allows customers to independently allocate computing resources, such as server time and network storage capacity, based on their specific requirements, without the need for any system administration involvement. Service management involves the implementation of metering capabilities to optimize resource utilization across various levels, such as capacity, handling, bandwidth, and dynamic utilization of mobile requests.

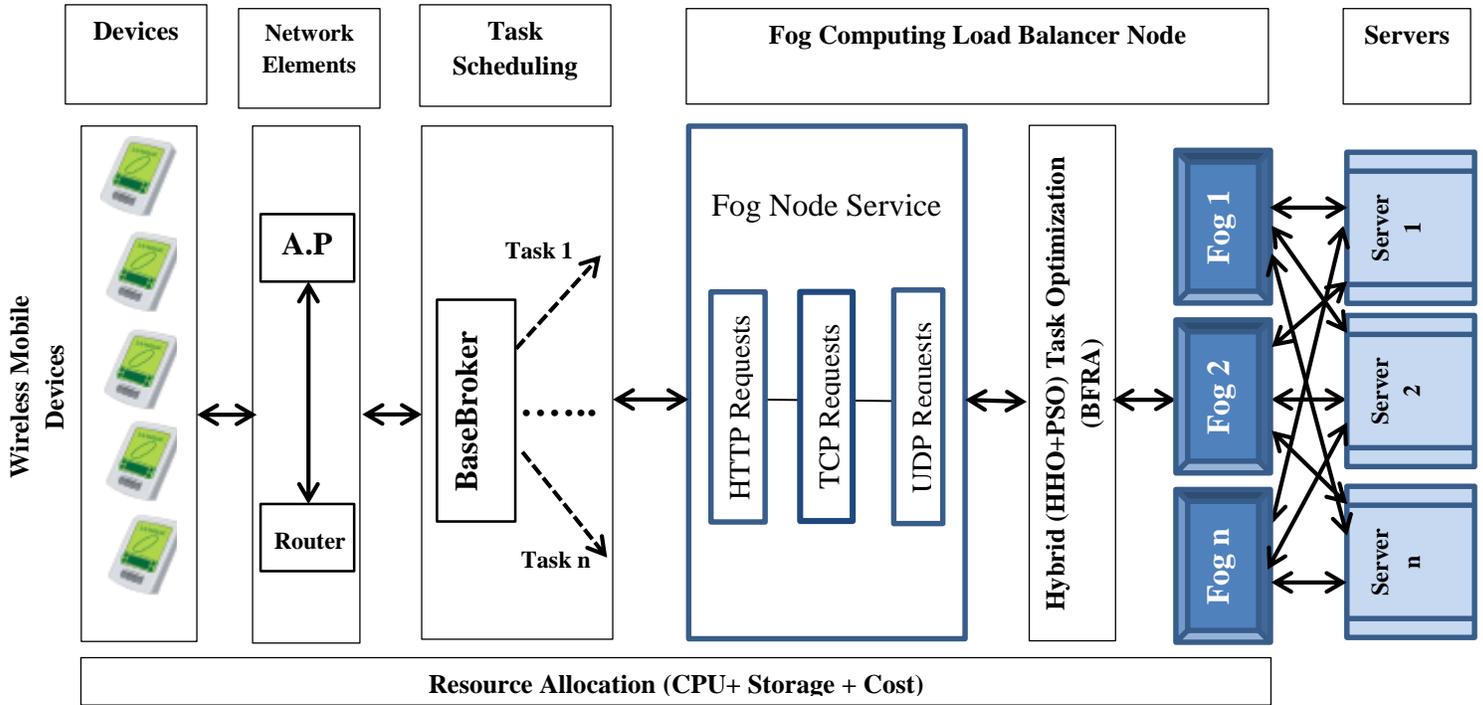


Figure 3.3: Multiple Fog computing resource allocation structure.

Fog-based-broker devices select Fog computing nodes to redirect request incoming from wireless Mobile device network to the server, and Fog load distribution resource efficiency is a crucial variable in achieving optimal performance and mitigating issues such as bottlenecks, overloaded, and low load.

The Fog node is optimized the required time to achive each task redirected by basebroker through PSO, and HHO optimizers, in addition, the optimization of computing resources is achived , including servers, applications, and storage, and services to balance load and enhance network performance overall. Besides, it enhanced mobile network computation power by decreasing waiting time which effected on Cost of mobile as resource constraint device and resource limitation. The proposed Hybrid system steps are shown in Figure (3.4).

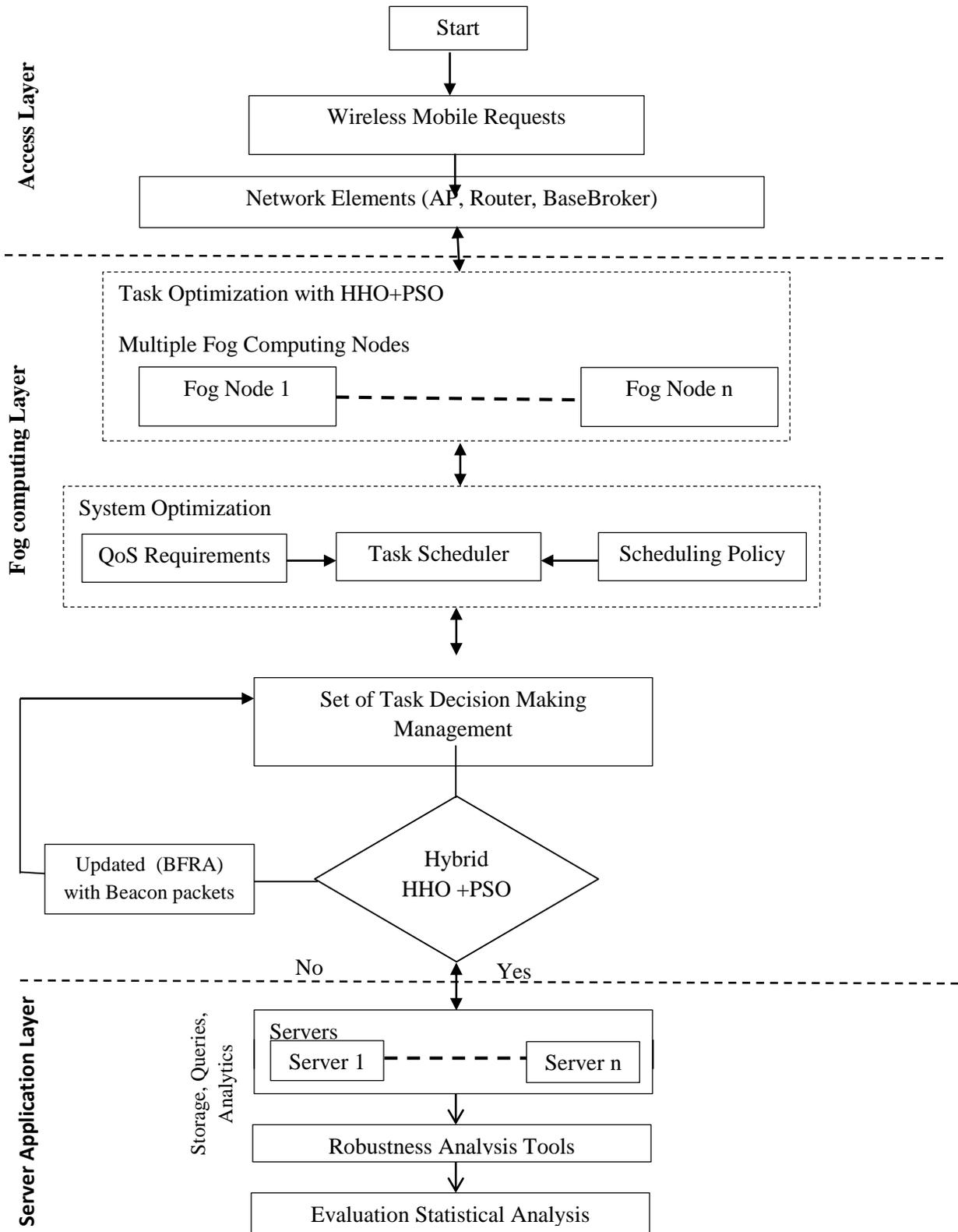


Figure 3.4 : The proposed Multiple fog computing system steps.

The utilization of the Hybrid optimization mechanism, comprising of HHO and PSO, facilitates the selection of servers with superior Fitness Resource Allocation (BFRA) value, as computed by fog nodes, for the allocation of task assignments generated by mobile devices. The direct application of synchronization strategies in Multiple Fog applications to wireless networks is not feasible. Mobile devices may exhibit high levels of instability in dynamically changing locations due to their node connectivity, which is unlike traditional Multiple structures that have stable connections among nodes.

Furthermore, the suggested approach for load balancing in Multiple fog computing is based on task allocation and optimization methods such as HHO and PSO. This approach aims to assign tasks to available resources in the fog node environment, which can be Multiple among servers. By doing so, the proposed approach can improve response time and handle high traffic data rates, while also managing large amounts of requests and data traffic in a Multiple environment. Additionally, the approach can reduce waiting times in queues through adaptive data management, particularly in fault-tolerant states for Single Node fog nodes.

### **3.4 The proposed Hybrid Resource Allocation Development Approach**

The Mobile-Fog-Server system under consideration is founded on a hybrid methodology for load distribution, which relies on the HHO algorithm and Meta-Heuristic algorithm. The efficacy of resource utilization and overall system performance is significantly impacted by the load-balancing algorithm utilized to determine the target server responsible for

handling a given request. The proposed load-balancing module initially gathers the latest server loads to achieve improved load distribution among servers with varying capabilities. The system subsequently leverages this data to identify the optimal server utilizing Better Fitness Load Balancing (BFRA) metric for executing a novel request, whereby the solution with the maximal fitness is referred to as the superior hawk. The hawk possesses precise knowledge regarding the hunting site, thereby rendering its position as the optimal Fitness Load Balancing (FLB) value for determining the prey's whereabouts. The process of gathering current server loads incurs a processing overhead. Servers are required to periodically retrieve their current load status and communicate it to the Fog load balancer. The Fog load balancer is responsible for maintaining the received load status of all servers. This process is impacted by the packet processing traffic.

### **3.4.1 The Harris hawks optimization algorithm**

It applied in the proposed environment due to :

- The initial steps of the algorithm require a thorough exploration phase, and the local search should exhibit exploitative behavior by employing effective search strategies to identify the optimal solution.
- The utilization of suitable operators, such as random bounce power, is necessary to maintain equilibrium during the exploration process.
- The utilization of time and adaptive parameters is a viable approach to address search space problems and mitigate the impact of local optima.

- This optimizer exhibits superior scalability in addressing high-dimensional problems relative to alternative optimization methods.
- This study aims to propose an efficient framework for collaborative and intelligent task sensing in a mobile system that operates on the Fog computing paradigm.

**Algorithm (3.1): the pseudocode for Harris hawks optimization algorithm**
**Input: population size, active server, required time.**
**Output: Better fitness server selection**
**Begin**
**Require** : algo,func\_details,popSize,Iterasyon

*func*=[F1,F2,F3, Fn]

*PopulationSize* = 20 (mobile nodes +fog nodes + servers)

*Iterations*= 5 (active servers)

*FuncAgain*=1

**while** HHO=TRUE **do**

   *if*(*algo*==0)

     *x*=*hho.HHO*(*getattr*(*functions*, *function\_name*),*lb*,*ub*,*dim*,*popSize*,*Iterasyon*)

     *function\_name*=*func\_details*[0]

**else**

   **for** *l* in range(0,*Iterations*):

     *CnvgHeader.append*("Iterasyon - "+str(*l*+1))

     **for** *i* in range (0, len(*algorithm*))

       **for** *j* in range (0, len(*func*))

         **if**((*algorithm*[*i*]==True) and (*func*[*j*]==True))

           **for** *k* in range (0,*FuncAgain*)

             *func\_details*=*functions.getFunctionDetails*(*j*)

             *x*=*selector*(*i*,*func\_details*,*PopulationSize*,*Iterations*)

           **End**

         **End if**

       **End**

     **End**
**End**
**End while**
**End of algorithm**

The proposed HHO swarm-based algorithm is used due to the

flexible structure, high performance, and high-quality results. It simulated in the proposed fog resource allocation framework to provide cooperative behavior to discover network resource as prey device to balance the load among different resources allocation.

The limitations of HHO have been exacerbated by the potential for entrapment in local optima and premature convergence, which hinders the determination of optimal resource allocation among neighboring devices. The main implementation stages as follow:

- The initial phase: a collection of entities (population) is produced. In the given scenario.
- Second phase: the population will undergo iterative evolution through the replacement of the current population with a newly generated population utilizing stochastic meta-heuristic mathematical models.
- Third phase: the optimization process continues until it meets a stopping criterion, such as reaching the maximum number of iterations.
- Fourth phase : calculate Fitness Function to find best (Fog-Server) position determining.
- Fifth phase: Hawks population with initial cost definition to determine jumping power of Fog-Server elements.
- Sixth phase: calculating escaping chance with the main three cases:
  - $R < 0.5$  (The probability of successful prey escape.)
  - $R \geq 0.5$  (The probability of server successfully escaping from a predator is low)
  - $|E|$  (Prey Escaping Cost Servers)

Figure 3.5 showed the HHO optimization algorithm used for resource allocation in fog computing system.



The HHO algorithm under consideration presents a model that simulates the behaviors of hawks and prey (represented by Fog-Servers) during both the exploration and exploitation phases. In the exploratory stage, the hawk observes the prey's stationary behavior and experiences limitations in accurately tracking the target.

This enables the identification of the most suitable Fog-Server node, for example. When the value of  $r$  is greater than or equal to 0.5 and the absolute value of  $E$  is less than 0.5, a severe siege takes place. Let ( $r$ ) denote a random number within the interval (0,1). The prey, namely Fog-Server, is experiencing a state of excessive load and insufficient resources to accommodate additional processes. The hawks exhibit infrequent attempts to besiege the Fog-Server, subsequently executing an unexpected rebound upon it.

Furthermore, assuming that the value of  $r$  is less than 0.5 and the absolute value of  $E$  is less than 0.5, the phenomenon of hard siege with rapid progressive dives takes place. The Fog-Server exhibits low processing costs for new task jobs, and executes a rigorous siege prior to the hawks optimizer's unexpected rebound and successful target elimination. The Hawks endeavor to minimize their mean proximity to the Fog-Server by manipulating the mobility status of individual components in a simulated network environment.

### **3.4.2 The PSO Algorithm**

The Meta-heuristic algorithm employed in this study is PSO for task scheduling. The PSO algorithm aims to allocate incoming tasks in a Multiple system to available computing resources based on predetermined criteria and objectives, such as the number of incoming requests and the

required makespan time to complete the tasks. The PSO algorithm being examined employs the portrayal of every individual in the swarm as a feasible solution to the optimization problem at hand. Throughout the optimization process, each particle experiences a procedure of updating that considers both the global best particle's position and its own local best position.

The allocation of particle resources is assessed by Basebroker through an analysis of the behavior and requirements of both the fog nodes and servers. The Fog platform facilitates the identification and selection of suitable resources. The Basebroker allocates resources in accordance with the behavior of mobile devices. The system facilitates the allocation, tracking, booking, and administration of resources within the Fog computing framework. Furthermore, it oversees the resources that are accessible within the Fog computing environment.

Upon the selection of resources, the Fog platform proceeds to schedule the application to the designated Fog resources and subsequently maintains continuous monitoring of said resources. In the event that mobile device traffic triggers dynamic changes during the loading state, the basebroker platform will reschedule the application to the Fog resources, taking into account the alterations in load requirements stemming from task scheduling. The allocation of resources in the Fog environment is determined by the submission history of application time scheduling. Additionally, the Basebroker component is responsible for managing all accounting operations pertaining to billing and usage. Algorithm (3.2) showed the PSO algorithm to scheduling the incoming request tasks.

---

**Algorithm (3.2): Task-scheduling based on Particle Swarm Optimization (PSO) algorithm**

Multi-criteria task-scheduling based on PSO:

1. Define a set of  $n$  parallel tasks of an application  $A = \{T_1, T_2, \dots, T_n\}$ , to be Multiple over a set of resources in the fog network  $Fnet = \{R_1, R_2, \dots, R_n\}$ .
  2. Initialize the importance Better Fitness Load Balancing (BFRA) of the criteria of the task scheduling.
  3. Repeat:
    - For each particle  $i = \{1, \dots, q\}$  do:
    - Determine the speed of request (velocity)
    - Determine the Fog position to optimize requests.
- Go to next iteration  $k=k+1$   
 Until  $k=k_{max}$  or a stop condition is reached
4. gbest is chosen as the optimal fog solution for the task scheduling

**End of algorithm**

The proposed PSO framework comprises two primary operations, namely position update and velocity update, as depicted in Figure 3.6. The PSO algorithm utilized in this study involves four main steps during each iteration of the process.

- The evaluation of pBest is conducted for every constituent of the population, representing the optimal position attained by the particle up to the current point. The updating of the particle position occurs only if the current position is superior to the preceding position; otherwise, the previous position remains unchanged.
- Assess the gBest, denoting the optimal position of the particles within the complete population.
- The velocity can be updated by incorporating the personal best (pBest) and global best (gBest) values. The calculation is determined by particular parameters, namely the particle index, time index, and random vectors within the range of  $[0, 1]$ .

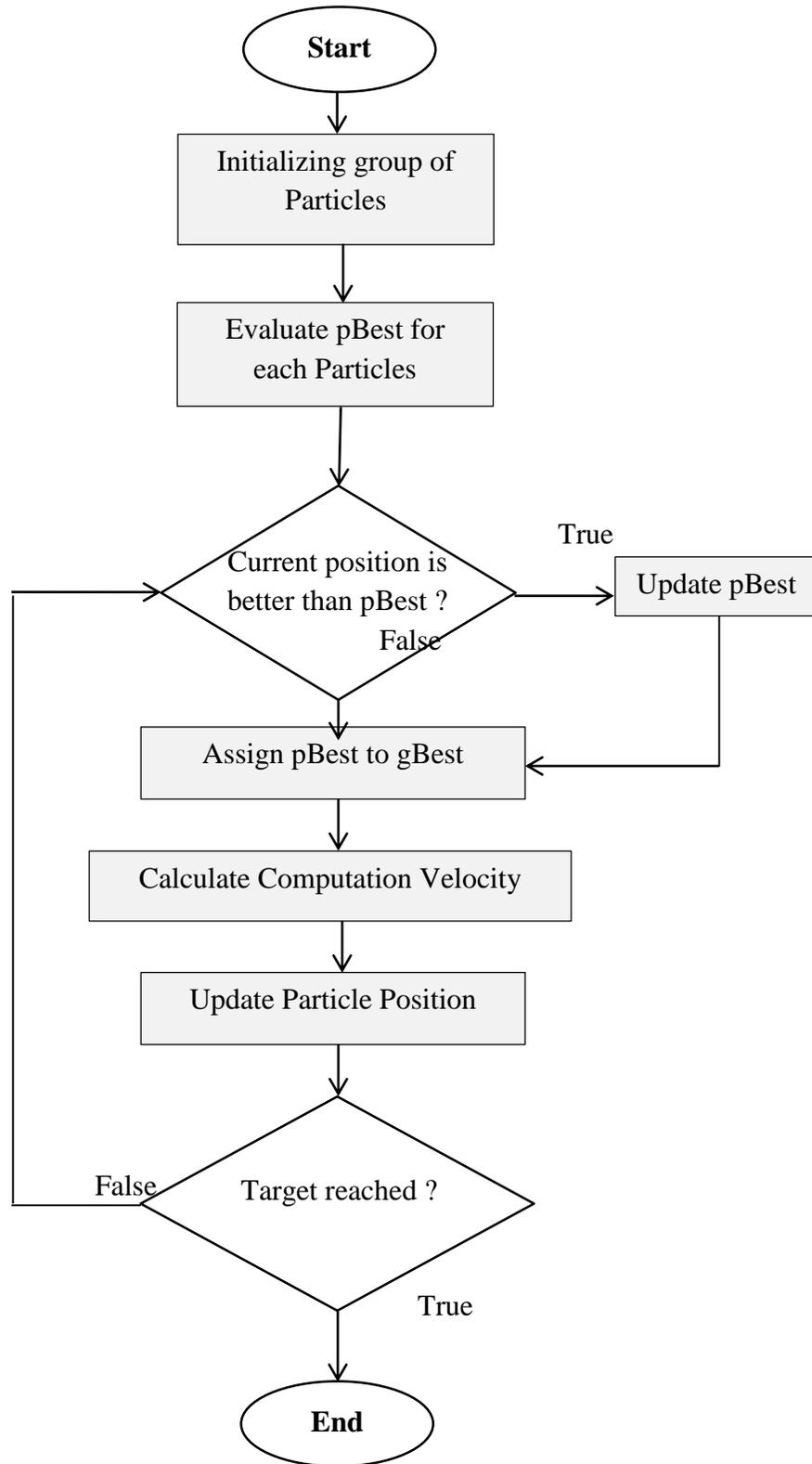


Figure 3.6: The proposed Particle Swarm Optimization (PSO) algorithm

### 3.4.3 The hybrid Resource Allocation

The proposed system employed a hybrid approach for resource allocation, utilizing two optimization methods, namely HHO and PSO. The objective is to construct a value-based Better Fitness Resource Allocation (BFRA) by integrating the characteristics of both optimizers. In a Multiple environment, each Fog-Server node periodically evaluates the network resources to determine the optimal Fog-Server for processing incoming requests. The primary objectives of the Hybrid system under consideration are :

- Decrease Cost: The cost incurred to process the task.
- High Scalability: This shows the capability of the load balancing mechanism which can be applicable to the machines and the tasks.
- High Flexibility: The joining of new nodes and revocation of the nodes mechanism is flexible.
- Maximum Resource Utilization: The resource utilization is maximized in the Fog-Server system.
- Decrease Processing Time: The total time takes to execute a service request.
- Increase availability: The capability of the computing system can be took up to maintain the system performance.
- High throughput: It is done through the service requests that are processed in the Fog-Server system.
- Compute the fitness values for each individual hawk.
- Choose the optimal individual position to serve as the prey location.
- Updating the position in the fusion map-compass operator.
- Mobile devices initiate requests to access particular types of

applications, such as HTTP requests, UDP, and FTP file upload requests.

- A request has been transmitted to fog nodes for the purpose of redirecting it to a designated server.
- Fog nodes periodically transmit Beacon BFRA request packets to servers, resulting in improved value with respect to processing costs and server specifications for each active server. Additionally, Fog nodes continue to transmit check packets at regular intervals to remain updated on the latest server state and to assess the optimal server for processing subsequent requests.

Algorithm (3.3) presents the pseudocode for the proposed hybrid BFRA in a fog computing environment.

**Algorithm (3.3): the pseudocode for the proposed hybrid Better Fitness Resource Allocation (BFRA)**

**Input:** The size of population  $N$ , maximum number of iterations  $t_{max}$ .

**Output:** The position of the Fog-Server and the corresponding fitness function value.

**Begin**

- Initialize users' position and mobility models.
- Calculate contribution level needs in each block based on user's density
- Initialize the position of the hawks  $X_i$  and the Fog-Server  $X_{Fog-Server}$ .
- Initialize the fitness values of the hawks  $f_i$  and the Fog-Server  $f_{Fog-Server}$ .
- Set the dimensions of the optimization problem  $dim$ .

**For** each mobile request **do**

**While** (termination condition is not met ( $t < t_{max}$ )) **do**

- Check the boundary and evaluate the fitness value of each optimizer hawk  $f_i$ .
- Update the location  $X_{Fog-Server}$  and fitness value  $f_{Fog-Server}$  of Fog-Server if there is a better one.
- Calculates the task imbalances among Fog-Server nodes with :

$$\text{Task imbalances} = \frac{\text{Highest duration of tasks (Tmax)} - \text{Lowest duration of tasks (Tmin)}}{\text{Average duration of tasks (Tavg)}}$$

- Evaluating load of each Fog-Server

$$\text{BFRA} = \frac{\text{Number of tasks assigned at time } t}{\text{Service rate of that Fog-Server at time } t}$$

**For** (each optimizer hawk ( $i = 1 : N$ )) **do**

<ul style="list-style-type: none"> <li>- <b>Define</b> the initial population class (<b>BFRA</b>) with population Size, max Iteration, filename</li> <li>- <b>Run</b> BFRA for request task allocation with Beacons messages</li> <li>- <b>Allocate</b> a request task based on the best solution (assign request task to active nodes)</li> <li>- <b>Calculate</b> cost and coverage for each request task</li> </ul> <p>Update the cost of the Fog-Server</p> <p><b>If</b> (<math>jEj \geq 1</math>) <b>then</b> % <i>Exploration Stage</i></p> <p>Update the position</p> <p style="padding-left: 2em;"><b>Else If</b> (<math>jEj &lt; 1</math>) <b>then</b> % <i>Exploitation Stage</i></p> <p style="padding-left: 4em;"><b>If</b> (<math>r \geq 0.5</math> and <math>jEj \geq 0.5</math>. ) <b>then</b> % <i>Soft besiege</i></p> <p style="padding-left: 6em;">Update the position</p> <p style="padding-left: 4em;"><b>Else If</b> (<math>r \geq 0.5</math> and <math>jEj &lt; 0.5</math>) <b>then</b> % <i>Hard besiege</i></p> <p style="padding-left: 6em;">Update the position</p> <p style="padding-left: 4em;"><b>Else If</b> (<math>r &lt; 0.5</math> and <math>jEj \geq 0.5</math>) <b>then</b> % <i>Soft besiege with progressive rapid dives</i></p> <p style="padding-left: 6em;">Update the position</p> <p style="padding-left: 4em;"><b>Else If</b> (<math>r &lt; 0.5</math> and <math>jEj &lt; 0.5</math>) <b>then</b> % <i>Hard besiege with progressive rapid dives</i></p> <p style="padding-left: 6em;">Update the position</p> <p style="padding-left: 4em;"><b>End If</b></p> <p style="padding-left: 2em;"><b>End If</b></p> <p style="padding-left: 2em;"><b>End For</b></p> <p style="padding-left: 2em;"><b>End</b></p> <p><b>Display</b> result in term of <b>cost</b> and <b>coverage</b></p> <p><b>End of algorithm</b></p>
---

The assessment of the population's cost usage is demonstrated in Figure 3.7. In the fog computing environment, each Fog-Server is categorized as either active or idle. The total cost consumption of a Fog-Server is the combination of its active and idle cost consumption. BFRA value is calculated to optimize cost consumption and improved the QoS of Fog-Server in Multiple fog computing resource allocation environment.

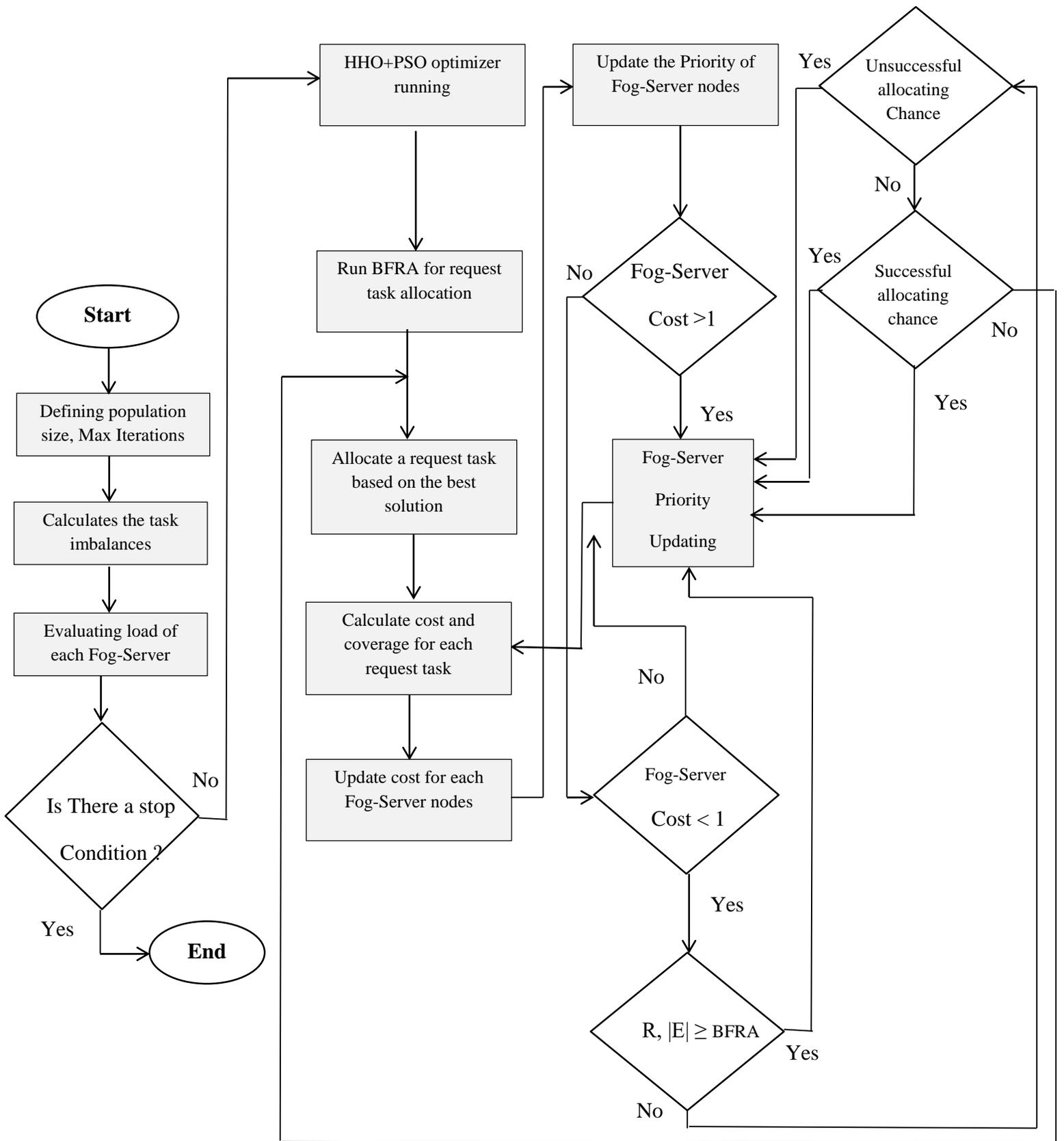


Figure 3. 7: The proposed Hybrid HHO-PSO optimization algorithm

# ***Chapter Four***

## ***The Implementation, Results and Discussion***

## 4.1 Introduction

This chapter presents an analysis of the results obtained from the proposed system in chapter three. It implemented with two resource allocation algorithms for optimization the load traffic in mobile network, as the Harris hawks optimization algorithm, PSO algorithm, and Hybrid algorithm. The proposed system scenarios based on the fog computing node with the two main cases : the 1<sup>st</sup> case study based on the one Single Node fog node to distribute load among the five servers. The 2<sup>nd</sup> case study based on five Multiple fog node to distribute request among them depending on the task scheduling approach, and then redirect incoming requests of services to the optimal convergence server available in the network.

## 4.2 The proposed System Implementation

The proposed system is implemented with the characteristics showed in Table (4.1) and the programming language used for the system's code is C++ within the OMNET++ simulation tool.

*Table 4.1: the specifications of the proposed fog-server system.*

<b>Operating Systems</b>	Windows 10 pro, 64-Bit
<b>CPU</b>	Core (TM) I5-4210U
<b>RAM</b>	8.00 GB
<b>Implementation Tools</b>	OMNET++ 4.6, INET 3.3.0, and FogNetSim++, iFogSim extension

### 4.2.1 The proposed Single Fog Computing Resource Allocation Scenario

It relies on a fog node to regulate traffic transmission and redirect requests to optimal servers in a state of resource allocation load balancing.

The network comprises a total of 8 mobile nodes, 5 servers, and wireless connections among all network elements. Table 4.2 displays the network components configuration details that impact the allocation of resources.

*Table 4.2: The Specification of the Single Node Fog-server system.*

<b>Node IP</b>	<b>Node Type</b>	<b>Network interface</b>	<b>Local port</b>
<b>10.0.0.2/30</b>	Mobile 1	eth0-eth0	1000
<b>10.0.0.6/30</b>	Mobile 2	eth0-eth1	1005
<b>10.0.0.22/30</b>	Mobile 3	eth0-eth5	1006
<b>10.0.0.30/30</b>	Mobile 4	eth0-eth7	1007
<b>10.0.0.14/30</b>	Mobile 5	eth0-eth3	1008
<b>10.0.0.18/30</b>	Mobile 6	eth0-eth4	1009
<b>10.0.0.15/30</b>	Mobile 7	eth0-eth6	1012
<b>10.0.0.17/30</b>	Mobile 8	eth0-eth7	1013
<b>10.0.0.1/30</b>	Fog Node	eth0-to-eth8	1010
<b>10.0.0.10/30</b>	Server 1	eth0-eth2	1002
<b>10.0.0.26/30</b>	Server 2	eth0-eth6	1003
<b>10.0.0.34/30</b>	Server 3	eth0-eth8	1004
<b>10.0.0.40/30</b>	Server 4	eth0-eth8	1001
<b>10.0.0.46/30</b>	Server 5	eth0-eth8	1011

The mobile network environment is established and configured using Fog resource allocation management, which is based on network service for storage capacity and request reply. The primary configuration is set accordingly:

**Step 1:** Initialize variables with position, swarm size, target function, and node velocity.

**Step 2 :** Configuring mobile network with the main parameters as local port number, interfaces, message length, packet type, and time management.

**Step 3 :** Building system topology with four tiers are (Mobile, network elements gateway, Fog, and Server).

**Step 4 :** Network setting with Fog configuration and network servers and exchanging network packets to build route path.

**Step 5 :** Calculating network overload of each server by exchanging beacon messages of fog node available servers dynamically. Through, the load table is added to the Fog node and data path cost calculation is built with the flow traffic for each interface connected with the available servers. Incoming packet with destination address checked and evaluated with the load table to see where to forward it, otherwise the Fog decide how deal with this packet.

Moreover, the network behavior for the Single Node scenario is applied with the main step to build the network results as follow:

- 1- Mobile node create a request and sent to specific available server to process their request as HTTP web access, File uploading service and storage purposes through Single Node Fog node.
- 2- The fog node is capable of receiving requests from mobile nodes and subsequently adjusting flow rules to ensure that the load is evenly Multiple across all available servers in a transparent transmission manner.
- 3- The fog node sends periodic ICMP echo messages to available servers in order to assess their cost, and the server that provides the optimal response is selected based on the value returned. The proposed concept under consideration is Better Fitness Load Balancing (BFLB).

- 4- The fog check link provides information on the status of each of the five servers, as well as updated cost values that are contingent upon the characteristics of connection traffic and network overloading.
- 5- The response message is sent by servers to mobile nodes in reply to each request.

Figure 4.1 showed the network configuration of the Single Node approach.

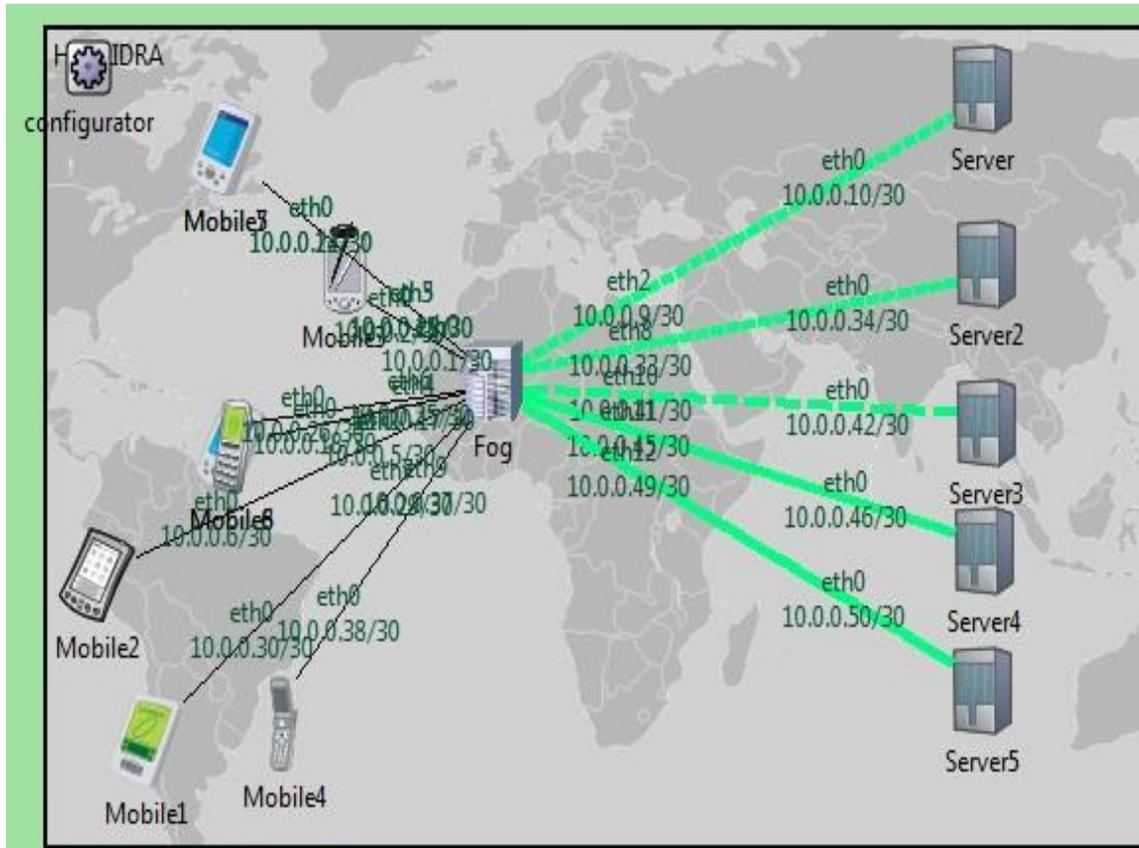


Figure 4.1 : Single Node Network configuration.

#### 4.2.2 The proposed Multiple Fog Computing Resource Allocation Scenario

The Multiple scenario of mobile fog-server computing system based on mobility five fog nodes to control traffic in the network and each of them redirect the incoming request from optimized basebroker to the

selective mobility server with take into consideration the amount of the overload in each server, in addition, 8 wireless mobile connected with network element as access point, gateway router, base broker, five fog nodes, and five servers. Table 4.3 displays the technical details of the network components that impact the allocation of resources. These details include the network setting.

Table 4.3: The Multiple Fog-server system Specification.

Node IP	Node Type	Network interface		Local port
10.0.0.2/30	Mobile 1	Wireless connection with Mobility service		1000
10.0.0.6/30	Mobile 2			1005
10.0.0.22/30	Mobile 3			1006
10.0.0.30/30	Mobile 4			1007
10.0.0.14/30	Mobile 5			1008
10.0.0.18/30	Mobile 6			1009
10.0.0.19/30	Mobile 7			1021
10.0.0.21/30	Mobile 8			1022
10.0.0.46/30	Fog Node 1	Eth1-to-eth3	Mobility service	1010
10.0.0.30/30	Fog Node 2	eth0-to-eth3		1013
10.0.0.62/30	Fog Node 3	eth0-to-eth3		1016
10.0.0.32/30	Fog Node 4	eth0-to-eth3		1017
10.0.0.65/30	Fog Node 5	eth0-to-eth3		1018
10.0.0.45/30	Server 1	eth0-to-eth2	Mobility service	1002
10.0.0.29/30	Server 2	eth0-to-eth2		1003
10.0.0.53/30	Server 3	eth0-to-eth2		1004
10.0.0.51/30	Server 4	eth0-to-eth2		1019
10.0.0.52/30	Server 5	eth0-to-eth2		1020
/	Access point	/		/
10.0.0.17/30	Gateway Router	eth0-to-eth1		1014
10.0.0.1/30	BaseBroker(virtual)	Eth0-to-eth3		1015

The network setting is first established through the implementation of a Multiple Fog allocation of resources management system that leverages network servers to effectively manage network congestion across the available servers. The optimization process happened between basebroker and fog node as Harris hawks optimization algorithm, Meta-heuristic PSO algorithm, and Hybrid algorithm with the main setting as:

**Step 1:** Initializing position, number of iteration, current population of active servers, and escaping energy.

**Step 2 :** Network configuration with local port number, interfaces, message length, packet type, and time management with synchronize task management .

**Step 3:** Building system topology with the mobility feature of Fog-Server elements and with 8 wireless mobiles.

**Step 4:** Multiple Fog-Server configuration with the network setting for wireless mobile nodes.

**Step 5:** Exchanging setting messages from wireless mobile with access point and gateway router to build network table and update mobile locations with the main phases as follow :

A- Router received packets from wireless mobile through access point and pass the packet to virtualize layer as basebroker device which passed the incoming messages to the Multiple Fog nodes (five Fog nodes).

B- Fog nodes received Wireless mobile packets and calculating cost for each servers dynamically.

C- The flow table is added to each Fog and data path is built with the flow traffic. Incoming packet with destination address checked with the flow table to see where to forward it, or Fog decide how deal with it.

Additionally, the Multiple fog computing system that has been suggested is executed by employing a resource allocation optimization algorithms are PSO and HHO optimizers. It aims to achieve load balancing, reduce response time, and enhance resource utilization. The primary steps involved in this process are as follows:

- 1- The mobile node initiated requests to the servers to process their request as HTTP web access or File uploading service through network gateway and basebroker then Fog nodes, as the network elements such as access point, gateway router, and basebroker configured to pass data messages to the Fog nodes layer.
- 2- Fog nodes received request from mobile and modify flow rules and balance the load among all available connected servers and estimate current cost of each server.
- 3- Fog nodes periodically transmit ICMP echo messages to the present servers in order to ascertain the most recently updated cost value (load) of the servers. The server that responds with the lowest processing cost is deemed to be the optimal server.
- 4- The fog nodes assess the task statuses of five individual servers and contemporaneously revise the cost (BFLB) metric based on the

characteristics of connection traffic and the degree of overloading present in each of the operational servers.

- 5- Fog nodes depend on the incoming request cost to distribute the request in some cases of high overloaded data to multiple servers to reply with the response to each request by the mobile nodes.

### 4.3 The results of Single Node Fog Computing

The proposed system comprises four case studies. The first case study is presented without the use of resources allocation algorithm, while the second case study involves resource allocation using a PSO meta-heuristic approach. The third case study employs the resource allocation with HHO algorithm while the fourth case study utilizes a hybrid approach that combines both PSO and HHO algorithms.

#### 4.3.1 Fog computing system without Resource Allocation

It is involved the overloading and high traffic load, which effects on the network parameters as high waiting of response time, and complexity of resource utilization. Table 4.4 displays the HTTP web requests made by mobile nodes to connected servers via fog nodes.

*Table 4.4: The main evaluation parameters of the without resource allocation in Single Node fog-server system.*

HTTP Request 1024 Byte	Throughput / Bps		Latency / ms	Response Time/ms	Packet Loss Rate (%)
	Frames/sec Sent	Frames/sec Received			
Mobile 1	22.03098	21.2148	48722.4	66.74256	0.81618
Mobile 2	2.6508	1.152	1179.36	1.4148	1.4988
Mobile 3	24.288	22.752	25153.92	60.8088	1.536
Mobile 4	25.9296	24.96	17845.92	53.7516	0.9696

Mobile 5	21.4452	20.2584	16513.92	68.8068	1.1868
Mobile 6	3.3804	2.5068	1271.52	0.6348	0.8736
Mobile 7	27.74467	25.9584	19095.13	57.51421	1.78627
Mobile 8	22.94636	21.06874	17669.89	73.62328	1.87762
Server 1	1.5288	1.44	1271.772	/	0.0888
Server 2	59.9016	53.9496	36641.78	/	5.952
Server 3	37.794	35.6352	23388.65	/	2.1588
Server 4	36.842	33.0578	23856.42	/	3.7842
Server 5	10.473	8.4257	9412.93	/	2.0473
Fog	151.447	138.9135	96809.44	/	12.5335
Average Packet Loss Rate (%)					2.6506

*Table 4.5: channel allocation and total network usage of the without resource allocation in Single Node fog-server system.*

Device Type	Requests	Channel Idle (%)	Channel Usage (%)	Total Network Usage in KB
Mobile Node	Mobile 1	82.4928	17.5072	0.16240
	Mobile 2	81.7464	18.2536	0.00393
	Mobile 3	83.1324	16.8676	0.08384
	Mobile 4	72.9156	27.0844	0.05948
	Mobile 5	83.2068	16.7932	0.05504
	Mobile 6	83.2188	16.7812	0.00423
	Mobile 7	71.645	28.355	0.06365
	Mobile 8	78.655	21.345	0.05889
	Server 1	83.16	16.84	0.00423
	Server 2	83.1384	16.8616	0.12213

Servers	Server 3	79.8816	20.1184	0.07796
	Server 4	78.589	21.411	0.07952
	Server 5	75.902	24.098	0.03137
Fog (avg all interfaces)		79.821	20.178	0.32269

Table 4.6 illustrates the allocation of channel resources for mobile nodes, highlighting the advantages that mobile devices derive from channel utilization. Additionally, the table presents the duration of time that packets remain in queue before being processed by the servers. The optimization of fitness value is achieved by maximizing the total sum, which amounts to 0.4413 in the absence of resource allocation.

*Table 4.6: evaluation metrics of the without resource allocation in Single Node fog-server system..*

Device Type	Requests	Resource Utilization (%)	Processing Time in sec	Execution Cost in Sec	Fitness Value
Mobile Node	Mobile 1	0.1164	32.76	33.882	0.02306
	Mobile 2	7.7136	3.12	3.222281	0.08683
	Mobile 3	0.876	1.56	2.083693	0.02087
	Mobile 4	8.5164	42.12	42.41944	0.01683
	Mobile 5	0.00972	43.68	44.07419	0.02397
	Mobile 6	8.6328	18.72	18.80753	0.06887
	Mobile 7	8.5164	42.12	42.42593	0.01607
	Mobile 8	0.00972	31.68	32.07961	0.02271
Servers	Server 1	8.1564	31.2	/	0.08988
	Server 2	0.05004	84.24	/	0.00878
	Server 3	9.041	4.05	/	0.01212

	Server 4	0.1500	3.12	/	0.01427
	Server 5	8.071	74.24	/	0.03707
Fog (avg all interfaces)		4.604	31.73	/	/

Tables 4.7, Table 4.8 to Table 4.11 present data on file uploads and downloads to servers of varying file sizes, which were used to assess data transfer speeds and the duration of the entire process, as measured by the end time of the final job.

*Table 4.7: FTP file upload/download by server 1 of without allocation in Single Node fog-server system.*

Server 1: 10.0.0.10/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	109.44	2.1252
4 MB	224.01	6.7728
20 MB	224.8644	105.978

*Table 4.8: FTP file upload/download by server 2 of without allocation in Single Node fog-server system.*

Server 2: 10.0.0.26/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	131.67	2.1588
4 MB	233.4144	6.7932
20 MB	231.7044	106.2312

*Table 4.9: FTP file upload/download by server 3 of without allocation in Single Node fog-server system.*

Server 3: 10.0.0.34/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	122.2644	2.214

4 MB	229.14	6.7788
20 MB	230.85	27.1512

*Table 4.10: FTP file upload/download by server 4 of without allocation in Single Node fog-server system.*

Server 4: 10.0.0.51/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	133.8645	2.19478
4 MB	237.3046	6.90642
20 MB	235.5661	108.0017

*Table 4.11: FTP file upload/download by server 5 of without allocation in Single Node fog-server system.*

Server 5: 10.0.0.52/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	124.3021	2.2509
4 MB	232.959	6.89178
20 MB	234.6975	27.60372

### 4.3.2 Resource allocation with Meta-heuristic PSO algorithm

The second case study presents a proposed system that utilizes the Meta-heuristic PSO algorithm for resource allocation. The system aims to address the issue of overload within the fog computing environment by distributing mobile request packets to the optimal available server. This is achieved by considering the minimum required time and scheduling tasks. Table 4.12 illustrates the distribution of HTTP requests across five servers.

Table 4.12: 2<sup>nd</sup> PSO case study evaluation criteria in Single Node fog-server system.

HTTP Request 1024 Byte	Throughput / Bps		Latency / ms	Response Time/ms	Packet Loss Rate (%)
	Frames/sec Sent	Frames/sec Received			
Mobile 1	26.43718	26.15776	36541.8	50.05692	0.27942
Mobile 2	3.18096	2.0824	884.52	1.0611	1.09856
Mobile 3	29.1456	28.0024	18865.44	45.6066	1.1432
Mobile 4	31.11552	30.652	13384.44	40.3137	0.46352
Mobile 5	25.73424	25.01008	12385.44	51.6051	0.72416
Mobile 6	4.05648	3.70816	953.64	0.4761	0.34832
Mobile 7	33.2936	31.85008	14321.35	43.13566	1.44352
Mobile 8	27.53563	25.98249	13252.42	55.21746	1.55314
Server 1	1.83456	1.828	953.829	/	0.00656
Server 2	71.88192	66.43952	27481.34	/	5.4424
Server 3	45.3528	43.46224	17541.49	/	1.89056
Server 4	44.2104	40.36936	17892.32	/	3.84104
Server 5	12.5676	11.81084	7059.698	/	0.75676
Fog	181.7364	167.3962	72607.08	/	14.3402
Average Packet Loss Rate (%)					2.3808

Table 4.13: channel allocation and total network usage in Single Node fog-server system.

Device Type	Requests	Channel Idle (%)	Channel Usage (%)	Total Network Usage in KB
	Mobile 1	90.74208	9.25792	0.12180
	Mobile 2	89.92104	10.07896	0.00294
	Mobile 3	91.44564	8.55436	0.06288

Mobile Node	Mobile 4	80.20716	19.79284	0.04461
	Mobile 5	91.52748	8.47252	0.04128
	Mobile 6	91.54068	8.45932	0.00317
	Mobile 7	78.8095	21.1905	0.04773
	Mobile 8	86.5205	13.4795	0.04417
Servers	Server 1	91.476	8.524	0.00317
	Server 2	91.45224	8.54776	0.09160
	Server 3	87.86976	12.13024	0.05847
	Server 4	86.4479	13.5521	0.05964
	Server 5	83.4922	16.5078	0.02353
Fog (avg all interfaces)		87.804	12.195	0.242023

Table 4.14 showed channel allocation of resources for mobiles and fog computing outcomes that improved compared to the 1st case study by increasing channel availability and utilization of resources and decreasing Processing Time and Execution Cost. The fitness value is 0.3684, which is better than without resource allocation.

*Table 4.14: 2nd PSO case study channel allocation in Single Node fog-server system.*

Device Type	Requests	Resource Utilization (%)	Processing Time in sec	Execution Cost in Sec	Fitness Value
Mobile Node	Mobile 1	0.130368	24.57	25.26188	0.01896
	Mobile 2	8.639232	2.34	2.403538	0.07192
	Mobile 3	0.98112	1.17	1.493701	0.01720
	Mobile 4	9.538368	31.59	31.77708	0.01402
	Mobile 5	0.010886	32.76	33.00298	0.01970

	Mobile 6	9.668736	14.04	14.09467	0.05736
	Mobile 7	9.538368	31.59	31.78119	0.01339
	Mobile 8	0.010886	23.76	24.00652	0.01868
Servers	Server 1	9.135168	23.4	/	0.07813
	Server 2	0.056045	63.18	/	0.00722
	Server 3	10.12592	3.0375	/	0.01010
	Server 4	0.168	2.34	/	0.0118
	Server 5	9.03952	55.68	/	0.02992
Fog (avg all interfaces)		5.156	23.804	/	/

Tables 4.15, 4.16, and 4.19 presented data pertaining to the uploading and downloading of files to the available servers with less fitness value with different file size and it showed the PSO useful for file uploading through less makespan compared with 1<sup>st</sup> case study of without resource allocation.

*Table 4.15: FTP file upload/download by server 1 of PSO in Single Node fog-server system.*

Server 1: 10.0.0.10/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	131.328	1.5939
4 MB	268.812	5.0796
20 MB	269.8373	79.4835

*Table 4.16: FTP file upload/download by server 2 of PSO in Single Node fog-server system.*

Server 2: 10.0.0.26/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	158.004	1.6191

4 MB	280.0973	5.0949
20 MB	278.0453	79.6734

*Table 4.17: FTP file upload/download by server 3 of PSO in Single Node fog-server system.*

Server 3: 10.0.0.34/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	146.7173	1.6605
4 MB	274.968	5.0841
20 MB	277.02	20.3634

*Table 4.18: FTP file upload/download by server 4 of PSO in Single Node fog-server system.*

Server 4: 10.0.0.51/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	160.6374	1.64608
4 MB	284.7655	5.17981
20 MB	282.6793	81.00128

*Table 4.19: FTP file upload/download by server 5 of PSO in Single Node fog-server system.*

Server 5: 10.0.0.52/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	149.1625	1.68817
4 MB	279.5508	5.16883
20 MB	281.637	20.70279

### 4.3.3 Resource allocation with Harris hawks optimization

The present study outlines a system proposal that utilizes the Harris hawks optimization technique for resource allocation in the context of task scheduling in fog servers to minimize execution cost, processing time

and fitness depending on the less makespan for analyzing request the determination of the optimal server for incoming requests is based on the assigned task value and the required response time for task completion on each server. Table 4.20 presents the primary assessment metrics utilized in the HHO.

The channel allocation scheme for fog system is presented in Table 4.21 environment utilizing the HHO optimization algorithm.

*Table 4.20: HHO case study evaluation criteria in Single Node fog-server system.*

HTTP Request 1024 Byte	Throughput / Bps		Latency / ms	Response Time/ms	Packet Loss Rate (%)
	Frames/sec Sent	Frames/sec Received			
Mobile 1	34.36833	33.59509	27406.35	42.54838	0.77324
Mobile 2	4.135248	3.29712	663.39	0.901935	0.83812
Mobile 3	37.88928	35.99312	14149.08	38.76561	1.89616
Mobile 4	40.45018	39.4376	10038.33	34.26665	1.01258
Mobile 5	33.45451	32.1031	9289.08	43.86434	1.35141
Mobile 6	5.273424	4.410608	715.23	0.404685	0.86281
Mobile 7	43.28168	42.9951	10741.01	36.66531	0.28658
Mobile 8	35.79632	33.36724	9939.315	46.93484	2.42908
Server 1	2.7464	2.384928	715.3718	/	0.36147
Server 2	93.4465	84.66138	20611.01	/	8.78512
Server 3	58.95864	56.09091	13156.12	/	2.86773
Server 4	57.47352	55.07017	13419.24	/	2.40335
Server 5	16.33788	15.64409	5294.774	/	0.69379
Fog	236.2573	230.2051	54455.31	/	6.0522
Average Packet Loss Rate (%)					2.1350

Table 4.21: Channel utilization and Total Network Usage for HHO in Single Node fog-server system.

Device Type	Requests	Channel Idle (%)	Channel Usage (%)	Total Network Usage in KB
Mobile Node	Mobile 1	96.1866	3.813395	0.09135
	Mobile 2	95.3163	4.683698	0.00221
	Mobile 3	96.93238	3.067622	0.04716
	Mobile 4	85.01959	14.98041	0.03346
	Mobile 5	97.01913	2.980871	0.03096
	Mobile 6	97.03312	2.966879	0.00238
	Mobile 7	83.53807	16.46193	0.03580
	Mobile 8	91.71173	8.28827	0.03313
Servers	Server 1	96.96456	3.03544	0.00238
	Server 2	96.93937	3.060626	0.06870
	Server 3	93.14195	6.858054	0.04385
	Server 4	91.63477	8.365226	0.04473
	Server 5	88.50173	11.49827	0.01764
Fog (avg all interfaces)		93.0722	6.9277	0.181517

As showed in Table 4.22 the total sum of fitness value is 0.2898 and it is decreased compared with PSO case due to increased resource utilization and total sum of throughput send and received.

Table 4.22: HHO case study channel allocation in Single Node fog-server system.

Device Type	Requests	Resource Utilization (%)	Processing Time in sec	Execution Cost in Sec	Fitness Value
	Mobile 1	0.156442	18.4275	18.82942	0.01468
	Mobile 2	10.36708	1.755	1.792232	0.05618
	Mobile 3	1.177344	0.8775	1.065777	0.01332

Mobile Node	Mobile 4	11.44604	23.6925	23.80221	0.01094
	Mobile 5	0.013063	24.57	24.71152	0.01525
	Mobile 6	11.60248	10.53	10.56356	0.04697
	Mobile 7	11.44604	23.6925	23.80227	0.01023
	Mobile 8	0.013063	17.82	17.96348	0.01445
Servers	Server 1	10.9622	17.55	/	0.06213
	Server 2	0.067254	47.385	/	0.00561
	Server 3	12.1511	2.278125	/	0.00786
	Server 4	0.2016	1.755	/	0.00887
	Server 5	10.84742	41.76	/	0.02334
Fog (avg all interfaces)		6.188548154	17.85331731	/	/

Tables 4.23, Table 4.24, and Table 4.27 presented the data on uploaded and downloaded files with different size from 512 KB to 20 MB assigned to the optimal servers.

*Table 4.23: FTP file upload/download by server 1 of HHO in Single Node fog-server system.*

Server 1: 10.0.0.10/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	157.5936	1.11573
4 MB	322.5744	3.55572
20 MB	323.8048	55.63845

*Table 4.24: FTP file upload/download by server 2 of HHO in Single Node fog-server system.*

Server 2: 10.0.0.26/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	189.6048	1.13337

4 MB	336.1168	3.56643
20 MB	333.6544	55.77138

*Table 4.25: FTP file upload/download by server 3 of HHO in Single Node fog-server system.*

Server 3: 10.0.0.34/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	176.0608	1.16235
4 MB	329.9616	3.55887
20 MB	332.424	14.25438

*Table 4.26: FTP file upload/download by server 4 of HHO in Single Node fog-server system.*

Server 4: 10.0.0.51/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	192.7649	1.15226
4 MB	341.7186	3.62587
20 MB	339.2152	56.7009

*Table 4.27: FTP file upload/download by server 5 of HHO in Single Node fog-server system.*

Server 5: 10.0.0.52/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	178.995	1.18172
4 MB	335.461	3.61818
20 MB	337.9644	14.49195

#### 4.3.4 Resource allocation with Hybrid algorithm

The fourth case study pertains to a hybrid approach for distributing server overload through the utilization of Better Fitness Load

Balancing (BFLB) values. These values are assigned to each server and traffic is redirected to the server with the least fitness value based on the PSO and HHO optimizers. Figure 4.2 showed the sent packets from Fog node to the server 1.

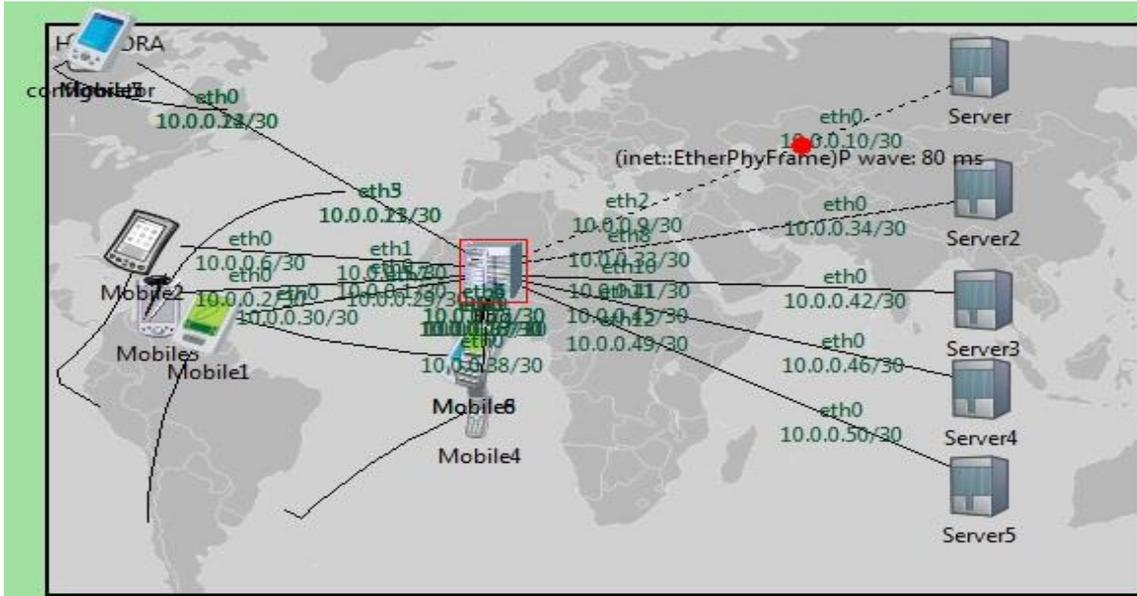


Figure 4.2 : Packets sent from Fog node resource allocator to Server 1.

Figure 4.3 showed the packets sent from Fog node to the another Server in the active network resources.

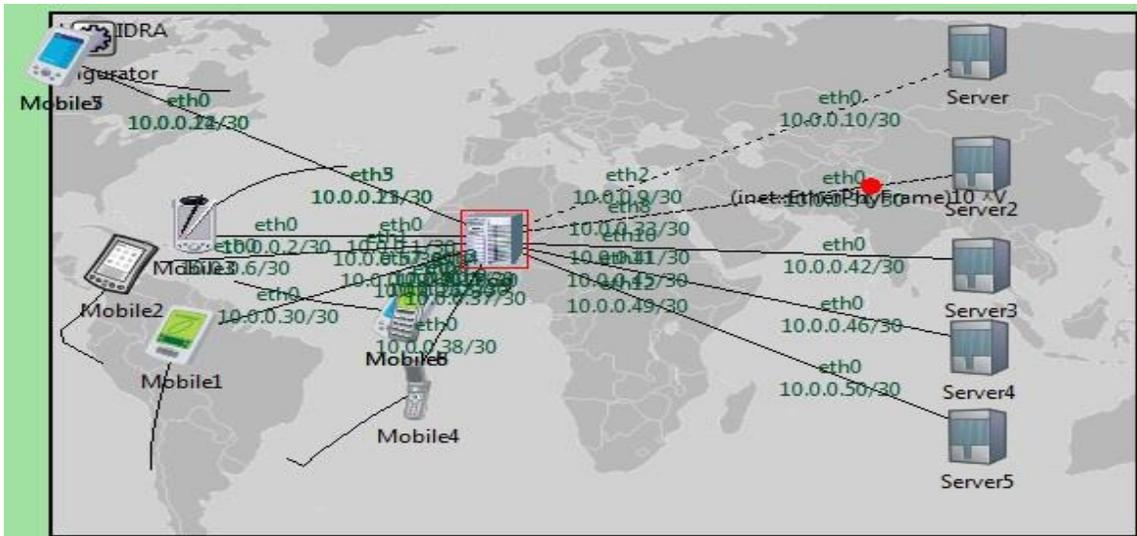


Figure 4.3 : Packets sent into server 2 in the active servers.

Figure 4.4 showed the request divided into more than one server in the network.

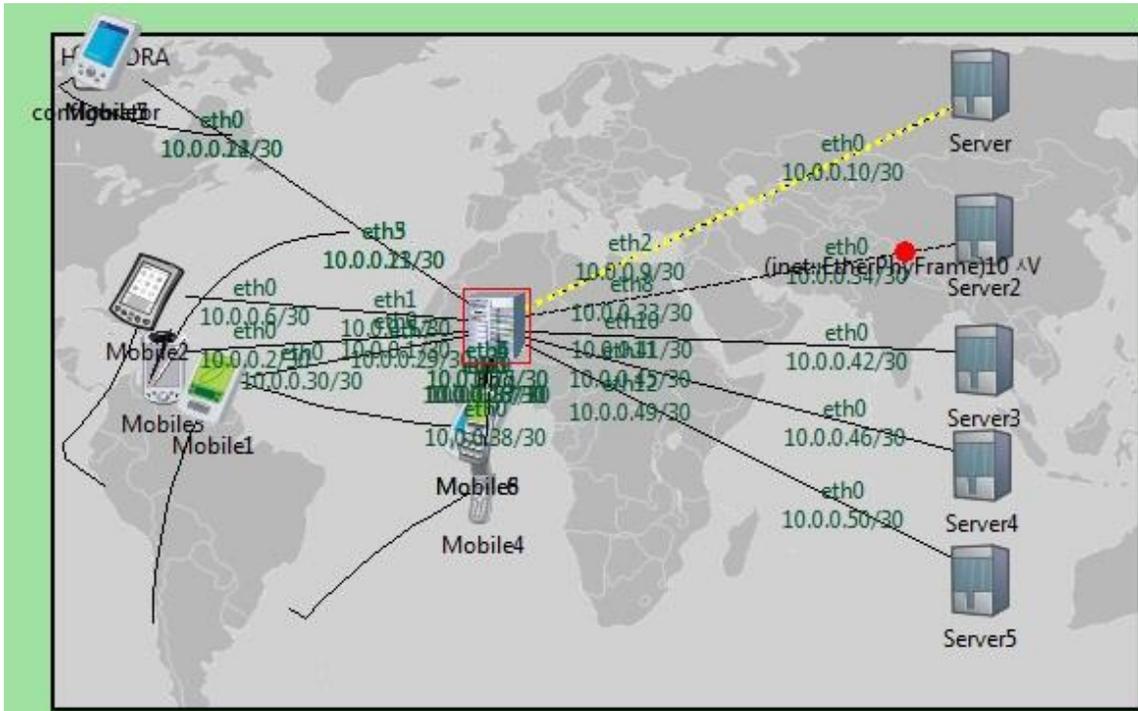


Figure 4.4: Packets divided among active servers.

Table 4.28 presents the evaluation parameters utilized in the fourth case study.

Table 4.28: main evaluation Hybrid resource allocation in Single Node fog-server system.

HTTP Request 1024 Byte	Throughput / Bps		Latency / ms	Response Time/ms	Packet Loss Rate (%)
	Frames/sec Sent	Frames/sec Received			
Mobile 1	41.242	40.11411	23843.52	37.54269	1.12789
Mobile 2	4.962298	2.556544	577.1493	0.795825	2.40575
Mobile 3	45.46714	43.99174	12309.7	34.20495	1.4754
Mobile 4	48.54022	47.12512	8733.347	30.23528	1.4151
Mobile 5	40.14541	39.32372	8081.5	38.70383	0.82169
Mobile 6	6.328109	5.09273	622.2501	0.357075	1.23537

Mobile 7	51.93802	49.99412	9344.679	32.35175	1.9439
Mobile 8	42.95558	40.84069	8647.204	41.4131	2.11489
Server 1	2.861914	2.09568	622.3735	/	0.76623
Server 2	112.1358	110.3937	17931.58	/	1.7421
Server 3	70.75037	68.10909	11445.82	/	2.64128
Server 4	68.96822	66.2842	11674.74	/	2.68402
Server 5	19.60546	18.17291	4606.453	/	1.43255
Fog	283.5088	280.4461	47376.12	/	3.0627
Average Packet Loss Rate (%)					1.7763

The channel usage together with the queue duration was presented in Table 4.29 for the fourth case study.

*Table 4.29: Channel utilization with total network usage of the Hybrid in Single Node fog-server system.*

Device Type	Requests	Channel Idle (%)	Channel Usage (%)	Total Network Usage in KB
Mobile Node	Mobile 1	98.11033	1.889668	0.079478
	Mobile 2	97.22263	2.777374	0.001923
	Mobile 3	98.87103	1.128972	0.041032
	Mobile 4	86.71998	13.28002	0.029111
	Mobile 5	98.95951	1.040487	0.026938
	Mobile 6	98.97378	1.026218	0.002074
	Mobile 7	85.20883	14.79117	0.031148
	Mobile 8	93.54596	6.454035	0.028824
Servers	Server 1	98.90385	1.096149	0.002074
	Server 2	98.87816	1.121843	0.059771
	Server 3	95.00479	4.995211	0.038152

	Server 4	93.46747	6.532535	0.038915
	Server 5	90.27176	9.728235	0.015354
Fog (avg all interfaces)		94.933698	5.066301	0.157920

As showed in Table 4.30 the fitness value is 0.2484, and it is decreased compared with HHO due to the increased number of transmit and received packets and resource utilization is larger than previous cases. Fitness value is better when minimized which means throughput and resource utilization is high compared with other cases.

*Table 4.30: Resource Utilization, Processing Time, Execution Cost, and Fitness Value of the Hybrid in Single Node fog-server system.*

Device Type	Requests	Resource Utilization (%)	Processing Time in sec	Execution Cost in Sec	Fitness Value
Mobile Node	Mobile 1	0.18773	14.742	15.03393	0.012263
	Mobile 2	12.4405	1.404	1.432876	0.050102
	Mobile 3	1.412813	0.702	0.837092	0.011005
	Mobile 4	13.73525	18.954	19.03356	0.009141
	Mobile 5	0.015676	19.656	19.75719	0.012581
	Mobile 6	13.92298	8.424	8.448538	0.039457
	Mobile 7	13.73525	18.954	19.03451	0.008645
	Mobile 8	0.015676	14.256	14.35868	0.011931
Servers	Server 1	13.15464	14.04	/	0.055211
	Server 2	0.080705	37.908	/	0.004492
	Server 3	14.58132	1.8225	/	0.006517
	Server 4	0.24192	1.404	/	0.00738
	Server 5	13.0169	33.408	/	0.019687
Fog (avg all interfaces)		7.426258462	14.28265385		/

Tables 4.31, Table 4.32, and Table 4.35 presented data on server speed and the corresponding required makespan to the upload and download files by optimal servers with different file size streaming.

*Table 4.31: FTP file upload/download by server 1 of the Hybrid approach in Single Node fog-server system.*

Server 1: 10.0.0.10/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	169.2555	0.892584
4 MB	346.4449	2.844576
20 MB	347.7664	44.51076

*Table 4.32: FTP file upload/download by server 2 of the Hybrid approach in Single Node fog-server system.*

Server 2: 10.0.0.26/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	203.6356	0.906696
4 MB	360.9894	2.853144
20 MB	358.3448	44.6171

*Table 4.33: FTP file upload/download by server 3 of the Hybrid approach in Single Node fog-server system.*

Server 3: 10.0.0.34/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	189.0893	0.92988
4 MB	354.3788	2.847096
20 MB	357.0234	11.4035

*Table 4.34: FTP file upload/download by server 4 of the Hybrid approach in Single Node fog-server system.*

Server 4: 10.0.0.51/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	207.0295	0.921808
4 MB	367.0058	2.900697
20 MB	364.3171	45.36072

*Table 4.35: FTP file upload/download by server 5 of the Hybrid approach in Single Node fog-server system.*

Server 5: 10.0.0.52/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	192.2406	0.945378
4 MB	360.2851	2.894548
20 MB	362.9738	11.59356

The findings indicate that the hybrid approach outperformed the POS algorithm and HHO algorithms in the context of HTTP and FTP file transfer, particularly when dealing with files of varying sizes. The PSO algorithm exhibits superior performance in the absence of resource allocation, whereas the HHO algorithm outperforms PSO in scenarios involving HTTP and FTP requests. Furthermore, the hybrid methodology exhibited superior resource utilization, execution costs, and makespan in comparison to the individual algorithms. The comparison of the system was illustrated in Figure 4.5 and Figure 4.8.

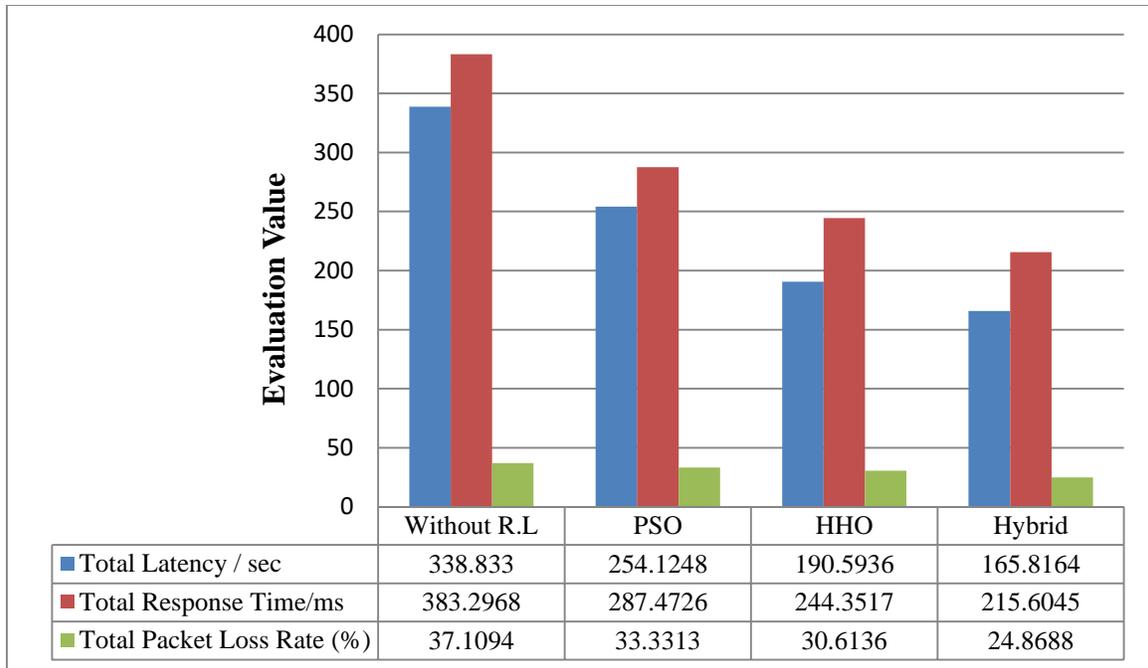


Figure 4.5: Proposed Single Node fog system comparison with latency, response time, and loss rate.

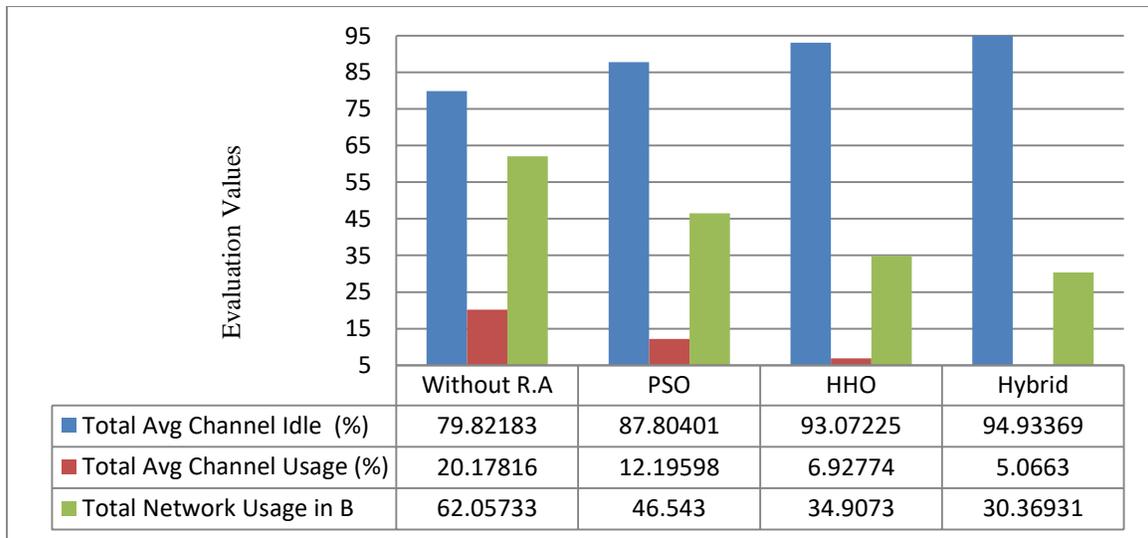


Figure 4.6: Proposed Single Node fog system comparison with channel allocation and network usage.

The results presented in Figure 4.7 illustrate the total utilization of resources and fitness value required to execute the transmitting job for the

utilized case studies. The findings indicate that the suggested hybrid strategy outperforms the other resource allocation methods examined in this study. The Hybrid approach exhibits an average total utilization of resources of 7.4262% and a superior overall fitness value of 0.2484.

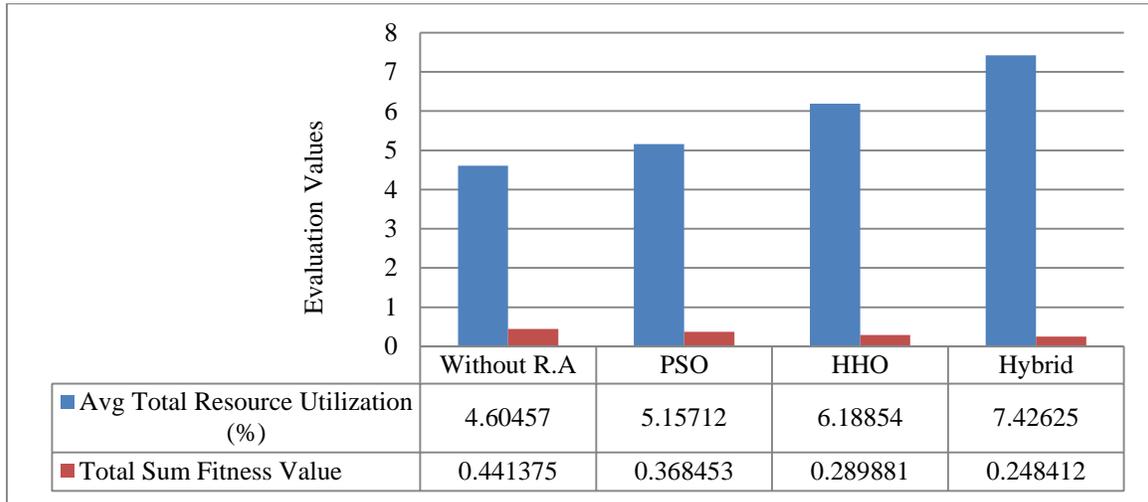


Figure 4.7: Proposed Single Node fog system comparison with resource utilization and fitness value.

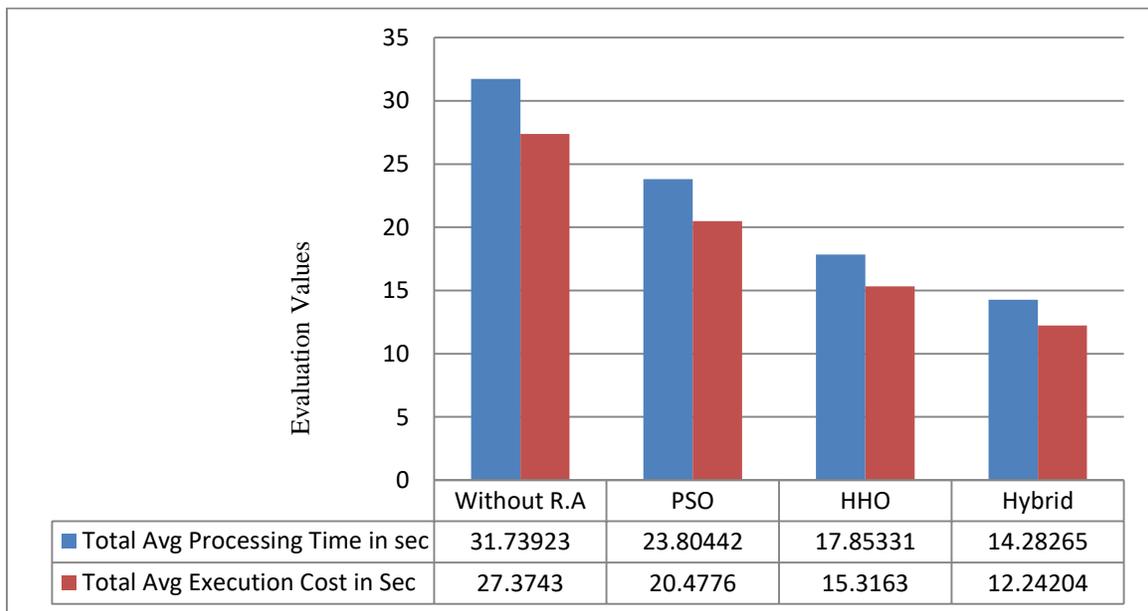


Figure 4.8: Proposed Single Node fog system comparison with average execution cost and processing time.

When contrasted with previous studies in the same field, the suggested method demonstrated superior assessment results, as can be shown in Table 4.36.

*Table 4.36 : The proposed Single Node fog-server system comparison with the other related works.*

<b>Ref.Year</b>	<b>Algorithm</b>	<b>Average Makespan in Sec</b>	<b>Average Execution Cost in Sec</b>	<b>Average Processing Time in Sec</b>
[18], 2020	Harris Hawks Optimization algorithm based on local search strategy (HHOLS)	20 Sec	/	/
[17], 2020	Hybrid gradient descent spider monkey optimization (HGDSMO)	1858.14 Sec	/	19.21 Sec
[21], 2022	MCT-PSO minimum completion time (MCT)-particle swarm optimization (PSO)	17.6 Sec	326.7	/
<b>HHO</b>		<b>14.7018 Sec</b>	<b>174.096</b>	<b>17.853 Sec</b>
<b>The proposed Hybrid system</b>		<b>11.7614 Sec</b>	<b>122.338</b>	<b>14.282 Sec</b>

#### 4.4 The results of Multiple Fog Computing

The proposed system is founded on four distinct case studies. The first case study does not involve the allocation of resources for fog-server network services. The second case study utilizes the PSO algorithm for

resource allocation. The third case study employs the HHO for resource allocation. Lastly, the fourth case study utilizes a hybrid approach that combines PSO and HHO. This approach is implemented in each fog node used in Multiple fog computing in order to optimize and schedule incoming requests to specific fog nodes. Resource allocation algorithm used to manage tasks in processing (active requests) that executed on the Fog, also have different direction to specific server. Therefore, in order to acquire an optimal choice of fog, it is required to discover an ideal way to place the various responsibilities evenly on the various fogs, taking into consideration the various properties of the fog and the application. Figure 4.9 showed Multiple fog network configuration.

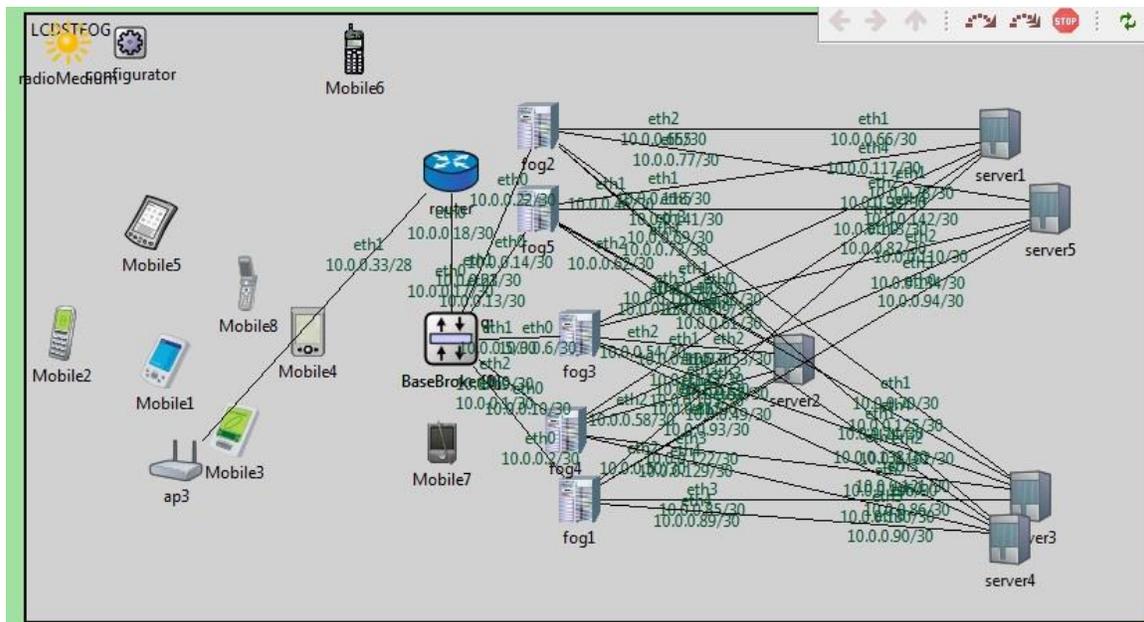


Figure 4.9 : Multiple Fog computing network configuration.

Figure 4.10 showed beacon message from access-point to mobile nodes to associated each mobile node into network elements.

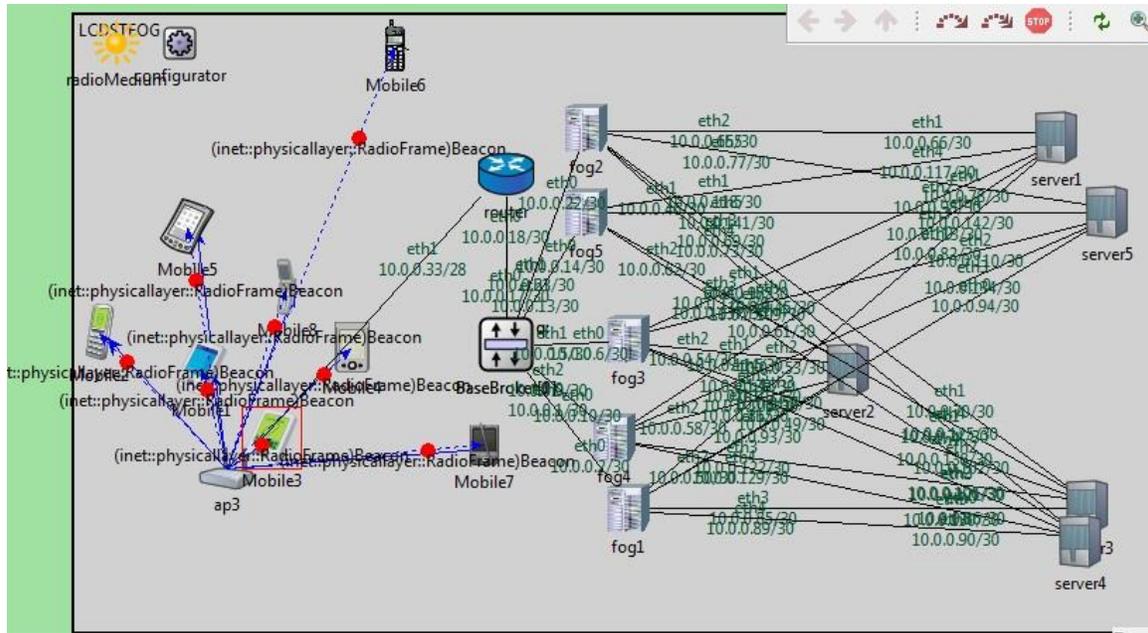


Figure 4.10: Beacon messages between mobile nodes and network elements.

#### 4.4.1 Fog computing system Without Resource Allocation

The strategy for implementing overload is applied to simulate the state of high computation processes without any allocation of resource and traffic is high with makespan and high required time as fitness value which required to allocate resource to manage data request among available servers. The initial case study pertains to a scenario of excessive load, necessitating the reduction of response-processing durations, enhancement of resource utilization, and optimization of QoS metrics. Table 4.37 illustrates the HTTP web requests made by nodes that move to the associated servers via fog node.

Table 4.37: Multiple fog-server system evaluation of without resource allocation

HTTP Request 1024 Byte	Throughput / Bps		Latency / ms	Response Time/ms	Packet Loss Rate (%)
	Frames/sec Sent	Frames/sec Received			
Mobile 1	24.23408	23.76058	42875.71	58.73345	0.4735

Mobile 2	2.91588	1.29024	1037.837	1.245024	1.62564
Mobile 3	26.7168	25.48224	22135.45	53.51174	1.23456
Mobile 4	28.52256	27.9552	15704.41	47.30141	0.56736
Mobile 5	23.58972	22.68941	14532.25	60.54998	0.90031
Mobile 6	3.71844	2.807616	1118.938	0.558624	0.910824
Mobile 7	30.51914	28.07341	16803.71	50.6125	2.44573
Mobile 8	25.241	22.59699	15549.5	64.78849	2.64401
Server 1	1.68168	1.6128	1119.159	/	0.06888
Server 2	65.89176	60.42355	32244.77	/	5.46821
Server 3	41.5734	39.91142	20582.01	/	1.66198
Server 4	40.5262	37.02474	20993.65	/	3.50146
Server 5	11.5203	9.436784	8283.378	/	2.083516
Fog	166.5917	155.5831	85192.31	/	11.0086
Average Packet Loss Rate (%)					2.471041

Table 4.38: channel allocation and total network usage of without resource allocation in Multiple fog-server system.

Device Type	Requests	Channel Idle (%)	Channel Usage (%)	Total Network Usage in KB
Mobile Node	Mobile 1	84.96758	15.03242	0.142919033
	Mobile 2	84.19879	15.80121	0.003459457
	Mobile 3	85.62637	14.37363	0.073784833
	Mobile 4	75.10307	24.89693	0.052348033
	Mobile 5	85.703	14.297	0.048440833
	Mobile 6	85.71536	14.28464	0.003729793
	Mobile 7	73.79435	26.20565	0.056012367
	Mobile 8	81.01465	18.98535	0.051831667

Servers	Server 1	85.6548	14.3452	0.00373053
	Server 2	85.63255	14.36745	0.107482567
	Server 3	82.27805	17.72195	0.0686067
	Server 4	80.94667	19.05333	0.069978833
	Server 5	78.17906	21.82094	0.02761126
Fog (avg all interfaces)		82.21649	17.78351	0.283974367

Table 4.39 presents information regarding the allocation of resources, processing time, and the execution cost for mobile nodes. The table also highlights the benefits that mobile devices receive from the channel with channel utilization. Additionally, the table provides data on the amount of time that packets spend in the queue before being processed by the servers. Fitness value maximized as the total sum is 0.40929 of without resource allocation in Multiple fog-server environment.

*Table 4.39: evaluation metrics of without resource utilization in Multiple fog-server system.*

Device Type	Requests	Resource Utilization (%)	Processing Time in sec	Execution Cost in Sec	Fitness Value
Mobile Node	Mobile 1	0.121056	27.846	28.73574	0.02078
	Mobile 2	8.022144	2.652	2.736772	0.08178
	Mobile 3	0.91104	1.326	1.741803	0.01883
	Mobile 4	8.857056	35.802	36.04171	0.01531
	Mobile 5	0.010109	37.128	37.44059	0.0216
	Mobile 6	8.978112	15.912	15.98414	0.0645
	Mobile 7	8.857056	35.802	36.05045	0.01483
	Mobile 8	0.010109	26.928	27.25163	0.0209

Servers	Server 1	8.482656	26.52	/	0.08491
	Server 2	0.052042	71.604	/	0.00791
	Server 3	9.40264	3.4425	/	0.011
	Server 4	0.156	2.652	/	0.01287
	Server 5	8.39384	63.104	/	0.03407
Fog (avg all interfaces)		4.788758	26.97835	/	/

The files that were uploaded and downloaded were displayed in Table 4.40, Table 4.41, and Table 4.44 to the servers with different file size to measure speed of data transfer and makespan as the required time as the end time of the last job to complete the entire process.

*Table 4.40: FTP file upload/download by server 1 of without allocation in Multiple fog-server system.*

Server 1: 10.0.0.10/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	117.5386	1.763916
4 MB	240.5867	5.621424
20 MB	241.5044	87.96174

*Table 4.41: FTP file upload/download by server 2 of without allocation in Multiple fog-server system.*

Server 2: 10.0.0.26/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	141.4136	1.791804
4 MB	250.6871	5.638356
20 MB	248.8505	88.1719

*Table 4.42: FTP file upload/download by server 3 of without allocation in Multiple fog-server system.*

Server 3: 10.0.0.34/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	131.312	1.83762
4 MB	246.0964	5.626404
20 MB	247.9329	22.5355

*Table 4.43: FTP file upload/download by server 4 of without allocation in Multiple fog-server system.*

Server 4: 10.0.0.51/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	143.7705	1.887511
4 MB	254.8651	5.939521
20 MB	252.998	92.88146

*Table 4.44: FTP file upload/download by server 5 of without allocation in Multiple fog-server system.*

Server 5: 10.0.0.52/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	133.5005	1.958283
4 MB	250.198	5.995849
20 MB	252.0651	24.01524

#### **4.4.2 Resource allocation with Meta-heuristic PSO algorithm**

The second case study presents a proposed system that utilizes a Multiple fog-server architecture. This system employs the Particle Swarm Optimization (PSO) algorithm for resource allocation to effectively manage workload distribution and balance the overload of resources within the fog-

server computation environment by distribute incoming mobile requests packets to the optimal server while minimizing the total makespan, fitness value and execution cost compared with Single Node PSO and with the case of overloading transmission packets. Figure 4.11 showed PSO optimizer in Fog node resource allocator redirect packets into optimal server.

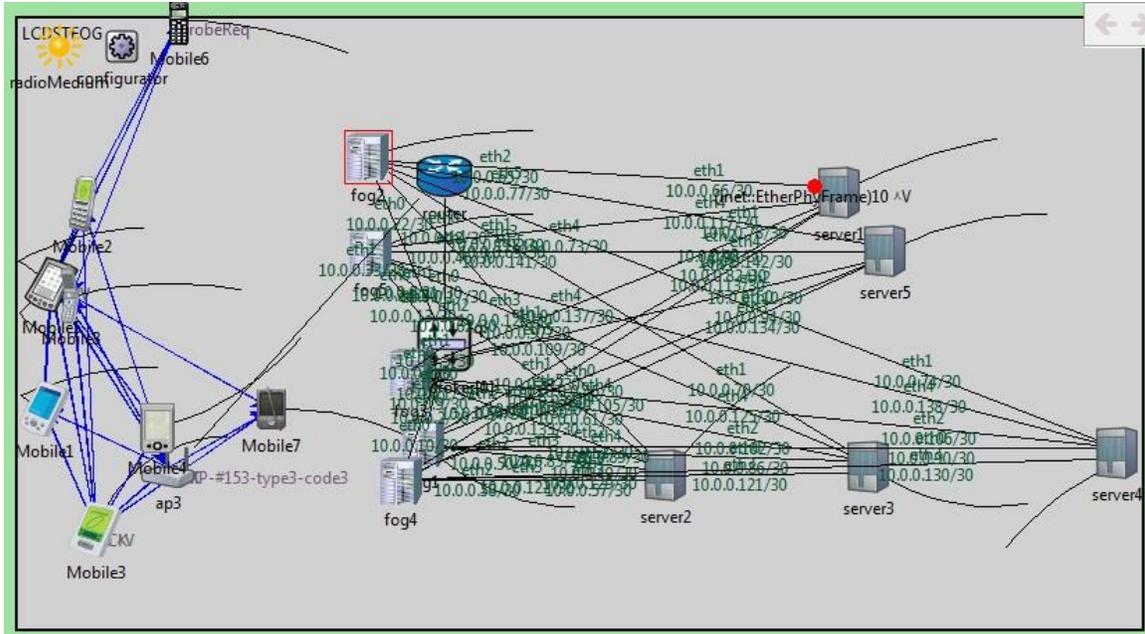


Figure 4.11 : Redirect packets from Fog to the servers by PSO Optimizer.

As depicted in Table 4.45 the allocation of HTTP requests was Multiple across a total of five servers.

Table 4.45: PSO case study evaluation criteria in Multiple fog-server system.

HTTP Request 1024 Byte	Throughput / Bps		Latency / ms	Response Time/ms	Packet Loss Rate (%)
	Frames/sec Sent	Frames/sec Received			
Mobile 1	29.60964	29.29669	32156.78	44.05009	0.31295
Mobile 2	3.562675	2.332288	778.3776	0.933768	1.230387
Mobile 3	32.64307	31.36269	16601.59	40.13381	1.28038

Mobile 4	34.84938	34.33024	11778.31	35.47606	0.51914
Mobile 5	28.82235	28.01129	10899.19	45.41249	0.81106
Mobile 6	4.543258	4.153139	839.2032	0.418968	0.390119
Mobile 7	37.28883	35.67209	12602.79	37.95938	1.61674
Mobile 8	30.83991	29.10039	11662.13	48.59136	1.73952
Server 1	2.054707	2.04736	839.3695	/	0.007347
Server 2	80.50775	78.41226	24183.58	/	2.09549
Server 3	50.79514	48.67771	15436.51	/	2.11743
Server 4	49.51565	47.21368	15745.24	/	2.30197
Server 5	14.07571	13.22814	6212.534	/	0.84757
Fog	203.5448	187.6837	63894.23	/	15.8611
Average Packet Loss Rate (%)					2.223657

*Table 4.46: channel allocation and total network usage in Multiple fog-server system.*

Device Type	Requests	Channel Idle (%)	Channel Usage (%)	Total Network Usage in KB
Mobile Node	Mobile 1	93.46434	6.535658	0.107189267
	Mobile 2	92.61867	7.381329	0.002594592
	Mobile 3	94.18901	5.810991	0.055338633
	Mobile 4	82.61337	17.38663	0.039261033
	Mobile 5	94.2733	5.726696	0.036330633
	Mobile 6	94.2869	5.7131	0.002797344
	Mobile 7	81.17379	18.82622	0.0420093
	Mobile 8	89.11612	10.88389	0.038873767
	Server 1	94.22028	5.77972	0.002797898
	Server 2	94.19581	5.804193	0.080611933

Servers	Server 3	90.50585	9.494147	0.051455033
	Server 4	89.04134	10.95866	0.052484133
	Server 5	85.99697	14.00303	0.020708447
Fog (avg all interfaces)		90.43813	9.561866	0.212980767

Table 4.47 showed the channel resources allocation, processing time, execution cost with fitness value time for mobiles and the results of the allocation of resources in fog computing has been found to be superior to that of the first case study. This is due to the increase in channel the availability and resource utilization, as well as the decrease in processing time and execution cost. Total fitness is 0.3370 so it is better with decreased fitness value compared with state of without resource allocation, also it is better compared with PSO of Single Node system due to the increase available servers after task scheduling of Multiple system .

*Table 4.47: PSO case study channel allocation in Multiple fog-server system.*

Device Type	Requests	Resource Utilization (%)	Processing Time in sec	Execution Cost in Sec	Fitness Value
Mobile Node	Mobile 1	0.136886	20.8845	21.42849	0.01694
	Mobile 2	9.071194	1.989	2.040949	0.06682
	Mobile 3	1.030176	0.9945	1.249215	0.01538
	Mobile 4	10.01529	26.8515	26.99981	0.01263
	Mobile 5	0.01143	27.846	28.03692	0.01759
	Mobile 6	10.15217	11.934	11.9785	0.05305
	Mobile 7	10.01529	26.8515	27.00291	0.01205
	Mobile 8	0.01143	20.196	20.38971	0.01668
	Server 1	9.591926	19.89	/	0.07302

Servers	Server 2	0.058847	53.703	/	0.00629
	Server 3	10.63222	2.581875	/	0.00908
	Server 4	0.1764	1.989	/	0.01032
	Server 5	9.491496	47.328	/	0.02718
Fog (avg all interfaces)		5.414981	20.23376	/	/

Tables 4.48 to Table 4.52 showed upload and download files to the available servers with less fitness value with different file size and it showed the PSO useful for file uploading requests through less makespan and acceptable speed compared with 1<sup>st</sup> case study of without resource allocation and with PSO of Single Node fog-server system.

*Table 4.48: FTP file upload/download by server 1 of the PSO of Multiple fog-server system.*

Server 1: 10.0.0.10/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	141.0463	1.386693
4 MB	288.7041	4.419252
20 MB	289.8053	69.15065

*Table 4.49: FTP file upload/download by server 2 of the PSO of Multiple fog-server system.*

Server 2: 10.0.0.26/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	169.8543	1.408617
4 MB	301.1046	4.432563
20 MB	298.8987	69.31586

*Table 4.50: FTP file upload/download by server 3 of the PSO of Multiple fog-server system.*

Server 3: 10.0.0.34/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	157.7211	1.444635
4 MB	295.5906	4.423167
20 MB	297.7965	17.71616

*Table 4.51: FTP file upload/download by server 4 of the PSO of Multiple fog-server system.*

Server 4: 10.0.0.51/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	172.6852	1.43209
4 MB	306.1229	4.506435
20 MB	303.8802	70.47111

*Table 4.52: FTP file upload/download by server 5 of the PSO of Multiple fog-server system.*

Server 5: 10.0.0.52/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	160.3497	1.468708
4 MB	300.5171	4.496882
20 MB	302.7598	18.01143

#### 4.4.3 Resource allocation with HHO

The system proposed in this instance is founded on the optimization technique known as HHO resource allocation for task scheduling in fog servers to minimize execution cost, processing time and fitness depending on the less makespan for analyzing request as the determination of the optimal server for incoming requests is based on the

assigned value of required time (response time) for task completion on each server. Figure 4.12 showed HHO redirect request from fog to servers.

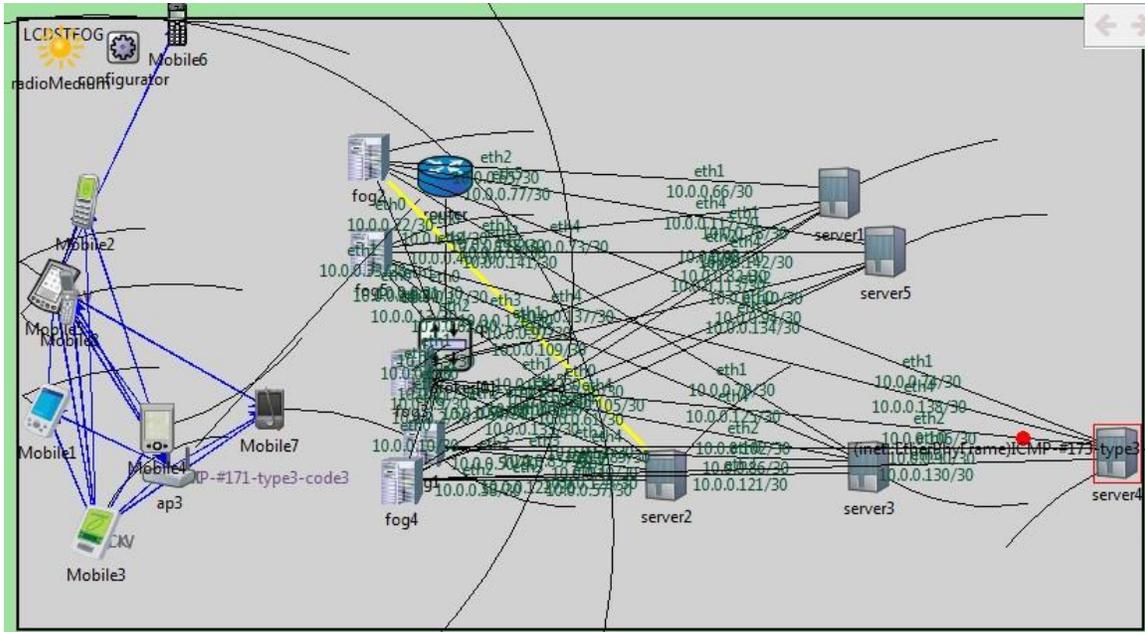


Figure 4.12 : Redirect packets from Fog to the servers by HHO Optimizer.

Table 4.53 presents the channel allocation for a fog computing environment utilizing the HHO optimization algorithm.

Table 4.53: main evaluation metrics of HHO in Multiple fog-server system.

HTTP Request 1024 Byte	Throughput / Bps		Latency / ms	Response Time/ms	Packet Loss Rate (%)
	Frames/sec Sent	Frames/sec Received			
Mobile 1	38.49253	37.6265	24117.59	37.44257	0.86603
Mobile 2	4.631478	3.692774	583.7832	0.793703	0.938704
Mobile 3	42.43599	40.31229	12451.19	34.11374	2.1237
Mobile 4	45.3042	44.17011	8833.73	30.15465	1.13409
Mobile 5	37.46905	35.95547	8174.39	38.60062	1.51358
Mobile 6	5.906235	4.939881	629.4024	0.356123	0.966354

Mobile 7	48.47548	48.15451	9452.089	32.26547	0.32097
Mobile 8	40.09188	37.37131	8746.597	41.30266	2.72057
Server 1	3.075968	2.671119	629.5272	/	0.404849
Server 2	104.6601	95.82075	18137.69	/	8.83935
Server 3	66.03368	63.82182	11577.39	/	2.21186
Server 4	64.37034	63.67859	11808.93	/	0.69175
Server 5	18.29843	17.52138	4659.401	/	0.77705
Fog	264.6082	260.8297	47920.67	/	3.7785
Average Packet Loss Rate (%)					1.949097

*Table 4.54: Channel utilization and total network usage for HHO case study of Multiple fog-server system.*

Device Type	Requests	Channel Idle (%)	Channel Usage (%)	Total Network Usage in KB
Mobile Node	Mobile 1	96.47516	3.52484	0.080391967
	Mobile 2	95.60225	4.397751	0.001945944
	Mobile 3	97.22318	2.776823	0.041503967
	Mobile 4	85.27465	14.72535	0.029445767
	Mobile 5	97.21019	2.789813	0.027247967
	Mobile 6	97.32422	2.675781	0.002098008
	Mobile 7	83.78868	16.21132	0.031506963
	Mobile 8	91.98687	8.013135	0.029155323
Servers	Server 1	97.25545	2.744546	0.002098424
	Server 2	96.23019	3.769812	0.060458967
	Server 3	93.42138	6.578624	0.0385913
	Server 4	91.90967	8.090326	0.0393631

	Server 5	88.76724	11.23276	0.015531337
Fog (avg all interfaces)		93.26686	6.733145	0.159735567

As showed in Table 4.55 fitness value is 0.26501 and it is decreased compared with PSO case due to increased resource utilization and total sum of throughput send and received.

*Table 4.55: evaluation metrics of HHO in Multiple fog-server system.*

Device Type	Requests	Resource Utilization (%)	Processing Time in sec	Execution Cost in Sec	Fitness Value
Mobile Node	Mobile 1	0.164264	15.66338	15.97907	0.01311
	Mobile 2	10.88543	1.49175	1.5221	0.05206
	Mobile 3	1.236211	0.745875	0.893762	0.01191
	Mobile 4	12.01834	20.13863	20.22535	0.00985
	Mobile 5	0.013716	20.8845	20.99531	0.01362
	Mobile 6	12.1826	8.9505	8.977813	0.04342
	Mobile 7	12.01834	20.13863	20.22529	0.0092
	Mobile 8	0.013716	15.147	15.25939	0.01291
Servers	Server 1	11.51031	14.9175	/	0.05795
	Server 2	0.070617	40.27725	/	0.00499
	Server 3	12.75866	1.936406	/	0.00701
	Server 4	0.21168	1.49175	/	0.0078
	Server 5	11.38979	35.496	/	0.02118
Fog (avg all interfaces)		6.497976	15.17532	/	/

Tables 4.56, Table 4.57, to Table 4.60 present data on the uploading and downloading of files with different size from 512 KB to 20 MB assigned to the optimal servers in Multiple fog-server system.

*Table 4.56: FTP file upload/download by server 1 of the HHO of Multiple fog-server system.*

Server 1: 10.0.0.10/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	162.3214	0.970685
4 MB	332.2516	3.093476
20 MB	333.5189	48.40545

*Table 4.57: FTP file upload/download by server 2 of the HHO of Multiple fog-server system.*

Server 2: 10.0.0.26/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	195.2929	0.986032
4 MB	346.2003	3.102794
20 MB	343.664	48.5211

*Table 4.58: FTP file upload/download by server 3 of the HHO of Multiple fog-server system.*

Server 3: 10.0.0.34/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	181.3426	1.011245
4 MB	339.8604	3.096217
20 MB	342.3967	12.40131

*Table 4.59: FTP file upload/download by server 4 of the HHO of Multiple fog-server system.*

Server 4: 10.0.0.51/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	198.5478	1.002466
4 MB	351.9702	3.154507

20 MB	349.3917	49.32978
-------	----------	----------

*Table 4.60: FTP file upload/download by server 5 of the HHO of Multiple fog-server system.*

Server 5: 10.0.0.52/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	184.3649	1.028096
4 MB	345.5248	3.147817
20 MB	348.1033	12.608

#### 4.4.4 Resource allocation with Hybrid algorithm

The Hybrid Multiple Fog-Server Computing approach has been proposed as a solution to address service delay issues. This approach leverages the untapped computational resources in the vicinity of fog nodes, which are utilized as servers, and the resources in the vicinity of basebrokers, which are used as fog nodes. Additionally, this approach emphasizes the optimal utilization of computational resources that are available within the network, specifically the fog nodes-servers. The fourth case study pertains to a hybrid approach for distributing incoming mobile requests to servers.

This approach involves the utilization of BFLB values, which are assigned to each server and used to redirect traffic to the server with the least fitness value. The approach integrates PSO and HHO optimizers to achieve this objective. Figure 4.13 showed redirect request from Fog to servers and huge request is divided among servers.

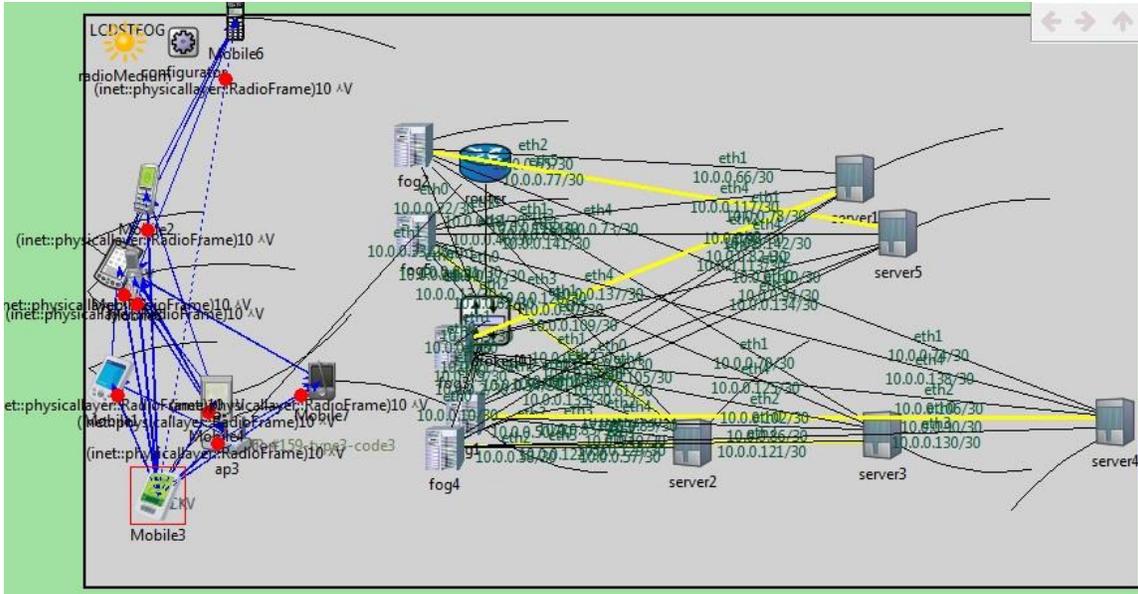


Figure 4.13 : Redirect request from Fog to active servers with Hybrid approach.

Table 4.61 presents the evaluation parameters utilized in the fourth case study.

Table 4.61: The Hybrid approach of Multiple fog-server main evaluation parameters.

HTTP Request 1024 Byte	Throughput / Bps		Latency / ms	Response Time/ms	Packet Loss Rate (%)
	Frames/sec Sent	Frames/sec Received			
Mobile 1	46.19104	45.9278	20266.99	29.65873	0.26324
Mobile 2	5.557774	4.863329	490.5769	0.628702	0.694445
Mobile 3	50.9232	49.27075	10463.25	27.02191	1.65245
Mobile 4	54.36505	53.78013	7423.345	23.88587	0.58492
Mobile 5	44.96286	44.04257	6869.275	30.57603	0.92029
Mobile 6	7.087482	6.703858	528.9126	0.282089	0.383624
Mobile 7	58.17058	56.99341	7942.977	25.55788	1.17717
Mobile 8	48.11025	47.74157	7350.123	32.71635	0.36868
Server 1	3.205344	2.347162	529.0175	/	0.858182
Server 2	125.5921	124.6409	15241.84	/	0.9512

Server 3	79.24041	76.28218	9728.947	/	2.95823
Server 4	77.24441	76.2383	9923.529	/	1.00611
Server 5	21.95812	20.35366	3915.485	/	1.60446
Fog	317.5299	314.0996	40269.7	/	3.4303
Average Packet Loss Rate (%)					1.203807

The channel usage together with the queue duration was displayed in Table 4.62 for the fourth case study.

*Table 4.62: Channel utilization with Total Network Usage of the Hybrid in Multiple fog-server system.*

Device Type	Requests	Channel Idle (%)	Channel Usage (%)	Total Network Usage in KB
Mobile Node	Mobile 1	99.09143	0.908567	0.067556633
	Mobile 2	98.19486	1.805144	0.001635256
	Mobile 3	99.85974	0.14026	0.0348775
	Mobile 4	97.58718	2.41282	0.024744483
	Mobile 5	99.94911	0.050895	0.022897583
	Mobile 6	99.96352	0.036482	0.001763042
	Mobile 7	96.06092	3.939082	0.02647659
	Mobile 8	94.48142	5.51858	0.02450041
Servers	Server 1	99.89289	0.107111	0.001763392
	Server 2	99.86694	0.133058	0.050806133
	Server 3	95.95484	4.045162	0.032429823
	Server 4	94.40214	5.597855	0.03307843
	Server 5	91.17448	8.825522	0.013051617
Fog (avg all interfaces)		97.4215	2.578503	0.134232333

As showed in Table 4.63 the total sum fitness value is 0.21863, and it is decreased compared with HHO due to the increased number of transmit and received packets and resource utilization is larger than previous cases. Fitness value is better when minimized which means throughput and resource utilization is high compared with other cases.

*Table 4.63: Evaluation metrics of hybrid case study in the Multiple fog-server system.*

Device Type	Requests	Resource Utilization (%)	Processing Time in sec	Execution Cost in Sec	Fitness Value
Mobile Node	Mobile 1	0.204626	11.0565	11.27567	0.01083
	Mobile 2	13.56015	1.053	1.073431	0.0417
	Mobile 3	1.539966	0.5265	0.629088	0.00983
	Mobile 4	14.97142	14.2155	14.27558	0.00812
	Mobile 5	0.017087	14.742	14.8188	0.01123
	Mobile 6	15.17605	6.318	6.336248	0.03452
	Mobile 7	14.97142	14.2155	14.27631	0.00768
	Mobile 8	0.017087	10.692	10.76832	0.01043
Servers	Server 1	14.33856	10.53	/	0.05027
	Server 2	0.087968	28.431	/	0.00399
	Server 3	15.89364	1.366875	/	0.00583
	Server 4	0.263693	1.053	/	0.0065
	Server 5	14.18842	25.056	/	0.0177
Fog (avg all interfaces)		8.094622	10.71199		/

Tables 4.64, Table 4.65 to Table 4.68 presented data on server speed and the corresponding required makespan to the upload and download files by optimal servers with different file size streaming.

*Table 4.64: FTP file upload/download by server 1 of Hybrid of Multiple fog-server system.*

Server 1: 10.0.0.10/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	177.7183	0.669438
4 MB	363.7671	2.133432
20 MB	365.1547	33.38307

*Table 4.65: FTP file upload/download by server 2 of Hybrid of Multiple fog-server system.*

Server 2: 10.0.0.26/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	213.8174	0.680022
4 MB	379.0389	2.139858
20 MB	376.262	33.46283

*Table 4.66: FTP file upload/download by server 3 of Hybrid of Multiple fog-server system.*

Server 3: 10.0.0.34/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	202.3256	0.69741
4 MB	379.1853	2.135322
20 MB	382.015	8.552625

*Table 4.67: FTP file upload/download by server 4 of Hybrid of Multiple fog-server system.*

Server 4: 10.0.0.51/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	221.5216	0.67292
4 MB	392.6962	2.117509
20 MB	389.8193	33.11333

*Table 4.68: FTP file upload/download by server 5 of Hybrid of Multiple fog-server system.*

Server 5: 10.0.0.52/30		
FTP File Size Upload/Download	Speed in KB/sec	Makespan in Sec
512 KB	209.5423	0.690126
4 MB	392.7108	2.11302
20 MB	395.6414	8.463299

The results indicate that the hybrid approach outperformed the PSO and HHO algorithms in terms of HTTP and FTP file transfer, particularly for files of varying sizes. The Particle Swarm Optimization (PSO) algorithm exhibits superior performance in scenarios where resource allocation is not a factor, whereas the Hybrid Harmony Search-Particle Swarm Optimization (HHO) algorithm outperforms PSO in situations involving HTTP and FTP requests. Furthermore, the hybrid methodology exhibited superior resource utilization, the execution cost, and makespan compared to the individual algorithms. The comparison of the system was demonstrated in Figure 4.14 and Figure 4.17.

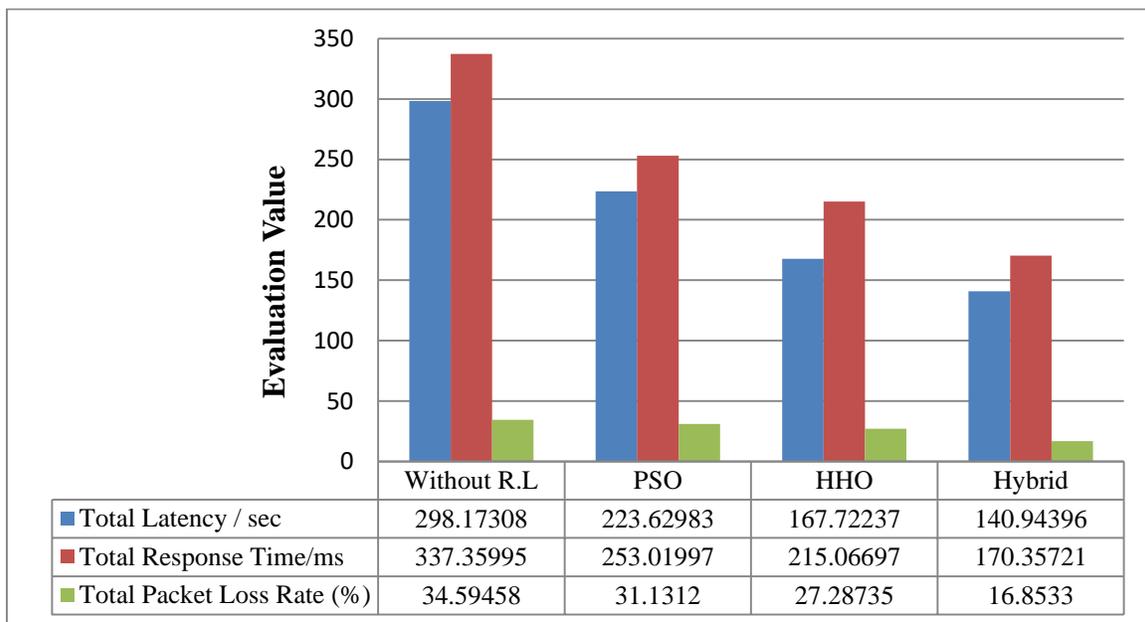


Figure 4.14: The proposed Multiple fog system comparison with latency, response time, and packet loss rate.

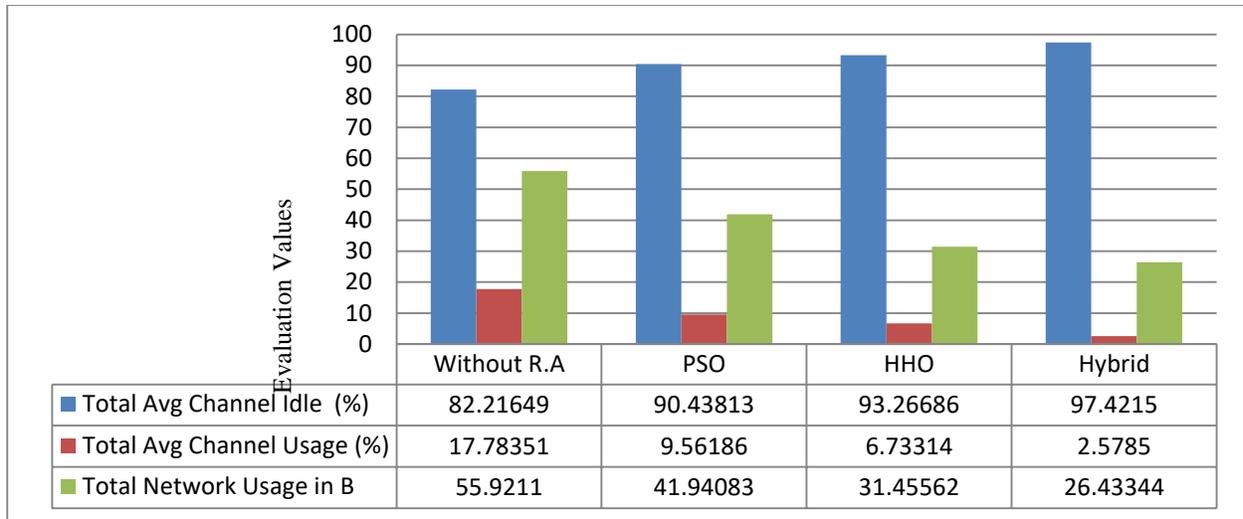


Figure 4.15: The proposed Multiple fog system comparison with Channel resource allocation and network usage.

As depicted in Figure 4.16, the transfer of total utilization of resources and fitness value during the execution of transmission position for the utilized case studies was demonstrated. The findings indicate that the suggested hybrid strategy outperformed the other cases of resource allocation. The Hybrid method exhibits an average total utilization of resources of 8.09462% and a superior total sum fitness value of 0.21863.

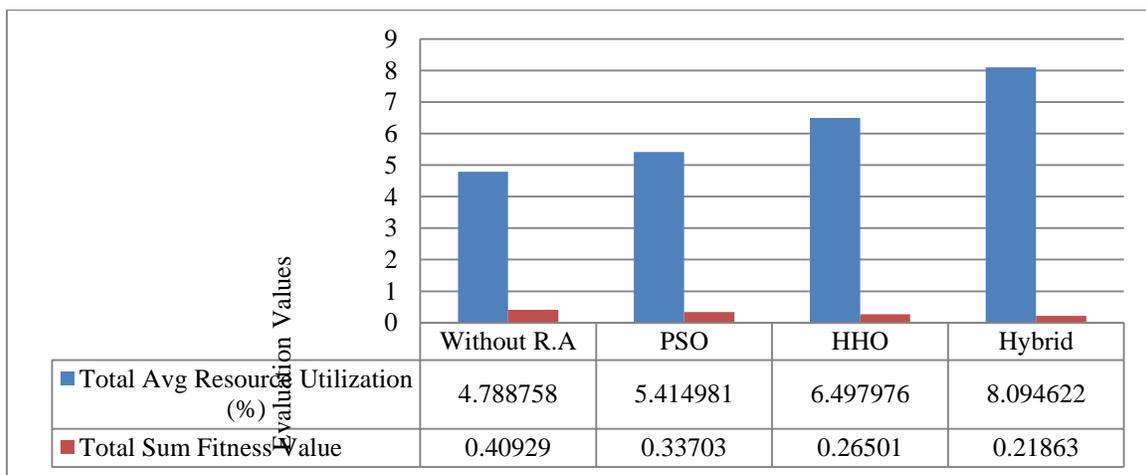


Figure 4.16: The proposed Multiple fog system comparison with total average of resource utilization and fitness time.

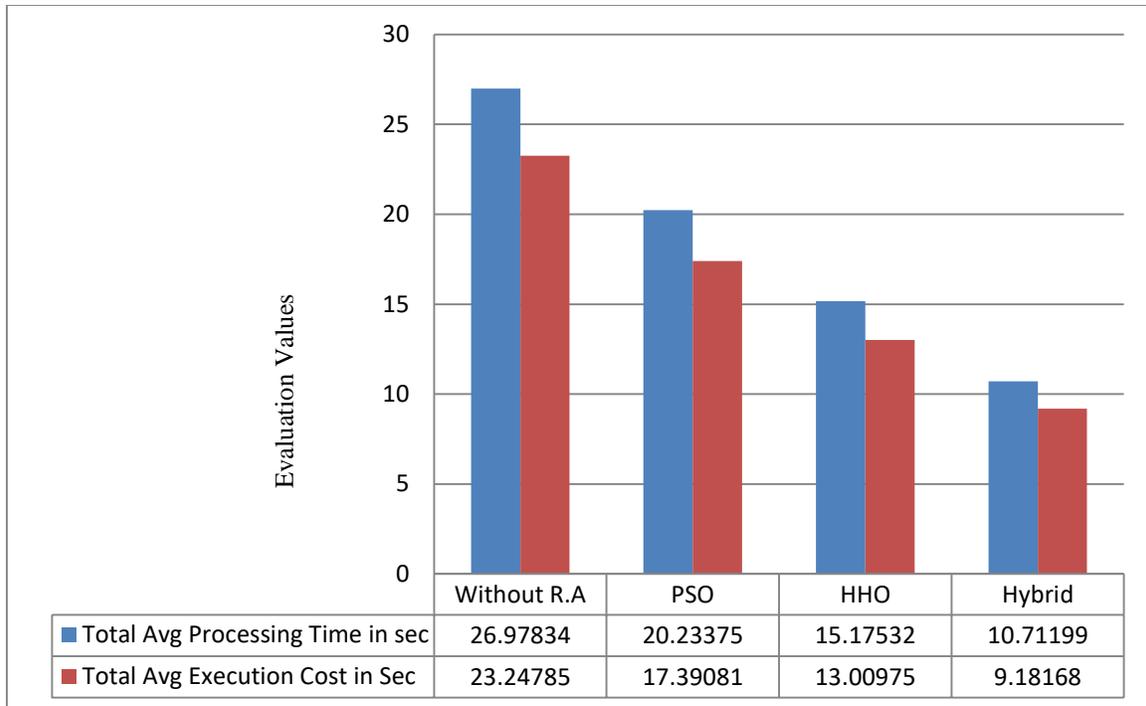


Figure 4.17: The proposed Multiple fog system comparison with average processing time and execution cost.

Upon comparison with other relevant works, the proposed system exhibited superior evaluation outcomes, as evidenced by the data presented in Table 4.69.

Table 4.69 : The proposed Multiple fog-server system comparison with the other related works.

Ref.Year	Algorithm	Simulation Tool	Total Makespan in Sec	Average Processing Time in Sec
[17], 2020	Hybrid gradient descent spider monkey optimization (HGDSMO)	Hadoop task scheduler tool	1858.14	19.21
[19], 2021	Extended Particle Swarm Optimization	iFogSim	342.53	/

	(EPSO)			
<b>PSO</b>		<b>FogNetSim+</b>	<b>274.084</b>	<b>20.233</b>
<b>HHO</b>		<b>+, iFogSim</b>	<b>191.858</b>	<b>15.175</b>
<b>The proposed Hybrid system</b>		<b>extension</b>	<b>131.024</b>	<b>10.711</b>

## 4.5 Summary

The Multiple fog-server computing framework under consideration has been implemented through four distinct case studies. The allocation of resources in fog servers is implemented with the aim of enhancing performance metrics, including but not limited to reducing delay, response time, time to process, fitness value, and makespan times, while simultaneously increasing throughput through optimal resource utilization. A comparative analysis is conducted to validate the effectiveness and efficiency of the suggested system for the hybrid allocation of resources between fog and server.

The findings indicate that the suggested system has enhanced the network's performance by ensuring the effective handling of mobile requests through the utilization of Single Node and Multiple services an optimizer for hybrid fog-server computing. It showed execution cost time of without resource allocation of Single Node is 27.374 seconds, and Multiple is 23.247 seconds, Single Node fog-server with PSO is 20.477 seconds, and 17.390 seconds of Multiple, besides Single Node fog-server execution cost of HHO is 15.316 seconds, and HHO of Multiple fog-server is 13.009 seconds, The proposed Hybrid execution cost of Single Node is 12.242 seconds while the Hybrid approach of resource allocation execution cost time is 9.181 seconds.

# ***Chapter Five***

## ***Conclusions and Suggestions for Future Works***

## 5.1 Conclusions

The current chapter offers a summary of the findings of the suggested approach and suggests potential avenues for further investigation. These suggestions can be succinctly summarized as follows :

- 1- The implementation hybrid resource allocation system within fog-server architecture to reduce implementation costs, makespan time and fitness in the proposed mobile network.
- 2- The results showed the proposed resource allocation fog-server computing system enhanced overload and computation on each server by balancing incoming wireless mobile requests among available servers as the total latency of PSO is 223.629 seconds, HHO is 167.722 seconds, and decreased latency with Hybrid as 140.943 seconds. Besides, total response times are 253.019 ms, 215.066 , 170.357 for PSO, HHO, and Hybrid respectively.
- 3- In addition, Average total packet loss rate is 31.131, 27.287, and 16.853 for PSO, HHO, and Hybrid respectively. Furthermore, Total Network Usage computation is decreased into 26.433 in Hybrid module while, 31.4556 , 41.940 for HHO, and PSO respectively.

## 5.2 Suggestions for Future Works

There are various aspects that can be considered for the prospective extension of present research by employing the subsequent propositions:

- 1- The implementation of an effective access control system that is enhanced with suitable monitoring capabilities and risk assessment techniques to identify and prevent unauthorized user activities. It aims to

detect anomalous user behavior and subsequently deactivate any illegitimate actions.

- 2- The utilization of fog-to-fog scalability communication in clusters remains an unresolved matter in the field of fog computing pertaining to the allocation of resources.
- 3- The integration of IoT and fog computing services has initiated a positive impact on the quality of services within the healthcare sector. These enhancements aim to enhance productivity, optimize the use of energy, and minimize operational costs.
- 4- The management of communication and information resources in actual infrastructure necessitates an increase in order to facilitate real-time monitoring of big data in proximity to the user. This is essential to ensure the reliability, stability, and durability required to process large volumes of data at the network edge, thereby enhancing node failure and service quality.
- 5- The edge Intelligence involves the utilization of fog computing to enable the implementation of machine learning and AI algorithms in closer proximity to the data source, specifically at the network's edge.

## REFERENCES

- [1] Qasem, M. H., Abu-Srhan, A., Natoureh, H., & Alzaghoul, E. (2020). Fog computing framework for smart city design.
- [2] Mahmud, R., Ramamohanarao, K., & Buyya, R. (2020). Application management in fog computing environments: A taxonomy, review and future directions. *ACM Computing Surveys (CSUR)*, 53(4), 1-43.
- [3] Hazra, A., Adhikari, M., Amgoth, T., & Srirama, S. N. (2020). Stackelberg game for service deployment of IoT-enabled applications in 6G-aware fog networks. *IEEE Internet of Things Journal*, 8(7), 5185-5193.
- [4] Meng, Y., Naeem, M. A., Almagrabi, A. O., Ali, R., & Kim, H. S. (2020). Advancing the state of the fog computing to enable 5g network technologies. *Sensors*, 20(6), 1754.
- [5] Al Yami, M., & Schaefer, D. (2019, April). Fog computing as a complementary approach to cloud computing. In *2019 International Conference on Computer and Information Sciences (ICCIS)* (pp. 1-5). IEEE.
- [6] Ruan, L., Guo, S., Qiu, X., & Buyya, R. (2020). Fog computing for smart grids: Challenges and solutions. *arXiv preprint arXiv:2006.00812*.
- [7] Khaturia, M., Jha, P., & Karandikar, A. (2020). Connecting the unconnected: Toward frugal 5G network architecture and standardization. *IEEE Communications Standards Magazine*, 4(2), 64-71.
- [8] Varghese, B., Wang, N., Nikolopoulos, D. S., & Buyya, R. (2020). Feasibility of fog computing. *Handbook of Integration of Cloud Computing, Cyber Physical Systems and Internet of Things*, 127-146.
- [9] Jaiswal, R., Davidrajuh, R., & Rong, C. (2020). Fog computing for realizing smart neighborhoods in smart grids. *Computers*, 9(3), 76.

- [10] Al Yami, M., & Schaefer, D. (2019, April). Fog computing as a complementary approach to cloud computing. In *2019 International Conference on Computer and Information Sciences (ICCIS)* (pp. 1-5). IEEE.
- [11] Yi, C., Huang, S., & Cai, J. (2019). Joint resource allocation for device-to-device communication assisted fog computing. *IEEE Transactions on Mobile Computing*, 20(3), 1076-1091.
- [12] Attaoui, W., & Sabir, E. (2018). Multi-criteria virtual machine placement in cloud computing environments: a literature review. *arXiv preprint arXiv:1802.05113*.
- [13] Li, X., Liu, Y., Ji, H., Zhang, H., & Leung, V. C. (2019). Optimizing resources allocation for fog computing-based Internet of Things networks. *IEEE Access*, 7, 64907-64922.
- [14] Lee, S. S., & Lee, S. (2020). Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information. *IEEE Internet of Things Journal*, 7(10), 10450-10464.
- [15] Wang, Q., & Chen, S. (2020). Latency-minimum offloading decision and resource allocation for fog-enabled Internet of Things networks. *Transactions on Emerging Telecommunications Technologies*, 31(12), e3880.
- [16] Seethalakshmi, V., Govindasamy, V., & Akila, V. (2020). Hybrid gradient descent spider monkey optimization (HGDSMO) algorithm for efficient resource scheduling for big data processing in heterogenous environment. *Journal of Big Data*, 7, 1-25.
- [17] Abdel-Basset, M., El-Shahat, D., Elhoseny, M., & Song, H. (2020). Energy-aware metaheuristic algorithm for industrial-Internet-of-Things task scheduling problems in fog computing applications. *IEEE Internet of Things Journal*, 8(16), 12638-12649.
- [18] Naha, R. K., & Garg, S. (2021). Multi-criteria--based Dynamic User Behaviour--aware Resource Allocation in Fog Computing. *ACM Transactions on Internet of Things*, 2(1), 1-31.

- [19] Potu, N., Jatoth, C., & Parvataneni, P. (2021). Optimizing resource scheduling based on extended particle swarm optimization in fog computing environments. *Concurrency and Computation: Practice and Experience*, 33(23), e6163
- [20] Brahmi, I., Hamdi, M., & Zarai, F. (2021). Chaotic grey wolf optimization-based resource allocation for vehicle-to-everything communications. *International Journal of Communication Systems*, 34(13), e4908.
- [21] Alsaidy, S. A., Abbood, A. D., & Sahib, M. A. (2022). Heuristic initialization of PSO task scheduling algorithm in cloud computing. *Journal of King Saud University-Computer and Information Sciences*, 34(6), 2370-2382.
- [22] Mukherjee, S., Gupta, S., Rawley, O., & Jain, S. (2022). Leveraging big data analytics in 5G-enabled IoT and industrial IoT for the development of sustainable smart cities. *Transactions on Emerging Telecommunications Technologies*, 33(12), e4618.
- [23] Iftikhar, S., Ahmad, M. M. M., Tuli, S., Chowdhury, D., Xu, M., Gill, S. S., & Uhlig, S. (2023). HunterPlus: AI based energy-efficient task scheduling for cloud–fog computing environments. *Internet of Things*, 21, 100667.
- [24] Rani, S., Kataria, A., & Chauhan, M. (2022). Fog computing in industry 4.0: Applications and challenges—A research roadmap. *Energy conservation solutions for fog-edge computing paradigms*, 173-190.
- [25] Sharma, S., & Saini, H. (2019). Efficient solution for load balancing in fog computing utilizing artificial bee colony. *International Journal of Ambient Computing and Intelligence (IJACI)*, 10(4), 60-77.
- [26] Shaheen, Q., Shiraz, M., Aziz Butt, S., Gani, A., & Khan, M. A. (2021). An analytical survey of wsns integration with cloud and fog computing. *Electronics*, 10(21), 2625.
- [27] Xie, Y., Zhu, Y., Wang, Y., Cheng, Y., Xu, R., Sani, A. S., ... & Yang, Y. (2019). A novel directional and non-local-convergent particle swarm optimization based

workflow scheduling in cloud–edge environment. *Future Generation Computer Systems*, 97, 361-378.

- [28] Yang, Y., Luo, X., Chu, X., Zhou, M. T., Yang, Y., Luo, X., ... & Zhou, M. T. (2020). Fog computing architecture and technologies. *Fog-Enabled Intelligent IoT Systems*, 39-60.
- [29] Sharifzadeha, M., Malekpoura, H., & Shojab, E. (2022). Cloud Computing and Its Impact on Industry 4.0: An Overview. *Industry 4.0 Vision for Energy and Materials: Enabling Technologies and Case Studies*, 99-120.
- [30] Singh, J., Singh, P., & Gill, S. S. (2021). Fog computing: A taxonomy, systematic review, current trends and research challenges. *Journal of Parallel and Multiple Computing*, 157, 56-85.
- [31] Pareek, K., Tiwari, P. K., & Bhatnagar, V. (2021, March). Fog computing in healthcare: A review. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1099, No. 1, p. 012025). IOP Publishing.
- [32] Anawar, M. R., Wang, S., Azam Zia, M., Jadoon, A. K., Akram, U., & Raza, S. (2018). Fog computing: An overview of big IoT data analytics. *Wireless Communications and Mobile Computing*, 2018.
- [33] Abdussami, A. A., & Farooqui, M. F. (2020). A systematic literature review on fog computing. *International Journal of Advanced Science and Technology*, 29(7), 12755-12769.
- [34] Kashyap, V., & Kumar, A. (2022). Load balancing techniques for fog computing environment: Comparison, taxonomy, open issues, and challenges. *Concurrency and Computation: Practice and Experience*, 34(23), e7183.
- [35] Nayeri, Z. M., Ghafarian, T., & Javadi, B. (2021). Application placement in Fog computing with AI approach: Taxonomy and a state of the art survey. *Journal of Network and Computer Applications*, 185, 103078.

- [36] Kishor, A., & Chakarbarty, C. (2021). Task offloading in fog computing for using smart ant colony optimization. *Wireless personal communications*, 1-22.
- [37] Shakarami, A., Shakarami, H., Ghobaei-Arani, M., Nikougoftar, E., & Faraji-Mehmandar, M. (2022). Resource provisioning in edge/fog computing: A comprehensive and systematic review. *Journal of Systems Architecture*, 122, 102362.
- [38] Kaur, N., Kumar, A., & Kumar, R. (2021). A systematic review on task scheduling in Fog computing: Taxonomy, tools, challenges, and future directions. *Concurrency and Computation: Practice and Experience*, 33(21), e6432.
- [39] Abdulqadir, H. R., Zeebaree, S. R., Shukur, H. M., Sadeeq, M. M., Salim, B. W., Salih, A. A., & Kak, S. F. (2021). A study of moving from cloud computing to fog computing. *Qubahan Academic Journal*, 1(2), 60-70.
- [40] Iorga, M., Feldman, L., Barton, R., Martin, M. J., Goren, N. S., & Mahmoudi, C. (2018). Fog computing conceptual model.
- [41] Atlam, H. F., Walters, R. J., & Wills, G. B. (2018). Fog computing and the internet of things: A review. *big data and cognitive computing*, 2(2), 10.
- [42] Mahmud, R., Kotagiri, R., & Buyya, R. (2018). Fog computing: A taxonomy, survey and future directions. *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*, 103-130.
- [43] Haghi Kashani, M., Rahmani, A. M., & Jafari Navimipour, N. (2020). Quality of service-aware approaches in fog computing. *International Journal of Communication Systems*, 33(8), e4340.
- [44] Moysiadis, V., Sarigiannidis, P., & Moscholios, I. (2018). Towards Multiple data management in fog computing. *Wireless Communications and Mobile Computing*, 2018.

- [45] Aazam, M., Zeadally, S., & Harras, K. A. (2018). Fog computing architecture, evaluation, and future research directions. *IEEE Communications Magazine*, 56(5), 46-52.
- [46] Mahmood, Z., & Ramachandran, M. (2018). Fog computing: Concepts, principles and related paradigms. In *Fog computing: concepts, frameworks and technologies* (pp. 3-21). Cham: Springer International Publishing.
- [47] Zhang, P., Liu, J. K., Yu, F. R., Sookhak, M., Au, M. H., & Luo, X. (2018). A survey on access control in fog computing. *IEEE Communications Magazine*, 56(2), 144-149.
- [48] Al Yami, M., & Schaefer, D. (2019, April). Fog computing as a complementary approach to cloud computing. In *2019 International Conference on Computer and Information Sciences (ICCIS)* (pp. 1-5). IEEE.
- [49] Bermbach, D., Pallas, F., Pérez, D. G., Plebani, P., Anderson, M., Kat, R., & Tai, S. (2018). A research perspective on fog computing. In *Service-Oriented Computing–ICSOC 2017 Workshops: ASOCA, ISyCC, WESOACS, and Satellite Events, Málaga, Spain, November 13–16, 2017, Revised Selected Papers* (pp. 198-210). Springer International Publishing.
- [50] Gasmi, K., Dilek, S., Tosun, S., & Ozdemir, S. (2022). A survey on computation offloading and service placement in fog computing-based IoT. *The Journal of Supercomputing*, 78(2), 1983-2014.
- [51] Dai, X., Xiao, Z., Jiang, H., Alazab, M., Lui, J. C., Min, G., ... & Liu, J. (2022). Task offloading for cloud-assisted fog computing with dynamic service caching in enterprise management systems. *IEEE Transactions on Industrial Informatics*, 19(1), 662-672.
- [52] Tran-Dang, H., & Kim, D. S. (2022). A survey on matching theory for Multiple computation offloading in iot-fog-cloud systems: Perspectives and open issues. *IEEE Access*.

- [53] Ashraf, M., Shiraz, M., Abbasi, A., & Albahli, S. (2022). Multiple application execution in fog computing: A taxonomy, challenges and future directions. *Journal of King Saud University-Computer and Information Sciences*, 34(7), 3887-3909.
- [54] Borelli, H., Costa, F. M., & Carvalho, S. T. (2022, December). Use of multilevel resource clustering for service placement in fog computing environments. In *2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)* (pp. 360-365). IEEE.
- [55] Wadhwa, H., & Aron, R. (2023). Optimized task scheduling and preemption for Multiple resource management in fog-assisted IoT environment. *The Journal of Supercomputing*, 79(2), 2212-2250.
- [56] Varshney, S., Sandhu, R., & Gupta, P. K. (2021, October). QoE-based resource management of applications in the fog computing environment using AHP technique. In *2021 6th international conference on signal processing, computing and control (ISPCC)* (pp. 669-673). IEEE.
- [57] Martinez, I., Hafid, A. S., & Jarray, A. (2020). Design, resource management, and evaluation of fog computing systems: a survey. *IEEE Internet of Things Journal*, 8(4), 2494-2516.
- [58] Singh, J., Warraich, J., & Singh, P. (2021, December). A survey on load balancing techniques in fog computing. In *2021 International Conference on Computing Sciences (ICCS)* (pp. 47-52). IEEE.
- [59] Baburao, D., Pavankumar, T., & Prabhu, C. S. R. (2021). Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method. *Applied Nanoscience*, 1-10.
- [60] Kashani, M. H., Ahmadzadeh, A., & Mahdipour, E. (2020). Load balancing mechanisms in fog computing: A systematic review. *arXiv preprint arXiv:2011.14706*.

- [61] Seyfollahi, A., & Ghaffari, A. (2020). Reliable data dissemination for the Internet of Things using Harris hawks optimization. *Peer-to-Peer Networking and Applications*, 13, 1886-1902.
- [62] Attiya, I., Abd Elaziz, M., & Xiong, S. (2020). Job scheduling in cloud computing using a modified harris hawks optimization and simulated annealing algorithm. *Computational intelligence and neuroscience*, 2020.
- [63] Montazerolghaem, A., Khosravi, M., Rezaee, F., & Khayyambashi, M. R. (2022, November). An optimal workflow scheduling method in cloud-fog computing using three-objective Harris-Hawks algorithm. In *2022 12th International Conference on Computer and Knowledge Engineering (ICCKE)* (pp. 300-306). IEEE.
- [64] Ding, W., Abdel-Basset, M., Eldrandaly, K. A., Abdel-Fatah, L., & De Albuquerque, V. H. C. (2020). Smart supervision of cardiomyopathy based on fuzzy Harris Hawks optimizer and wearable sensing data optimization: a new model. *IEEE Transactions on Cybernetics*, 51(10), 4944-4958.
- [65] Zivkovic, M., Bezdan, T., Strumberger, I., Bacanin, N., & Venkatachalam, K. (2021). Improved harris hawks optimization algorithm for workflow scheduling challenge in cloud-edge environment. In *Computer Networks, Big Data and IoT: Proceedings of ICCBI 2020* (pp. 87-102). Singapore: Springer Singapore.
- [66] Al-Safi, H., Munilla, J., & Rahebi, J. (2022). Patient privacy in smart cities by blockchain technology and feature selection with Harris Hawks Optimization (HHO) algorithm and machine learning. *Multimedia Tools and Applications*, 81(6), 8719-8743.
- [67] Liu, Q., Kosarirad, H., Meisami, S., Alnowibet, K. A., & Hoshyar, A. N. (2023). An Optimal Scheduling Method in IoT-Fog-Cloud Network Using Combination of Aquila Optimizer and African Vultures Optimization. *Processes*, 11(4), 1162.
- [68] Hussien, A. G., Abualigah, L., Abu Zitar, R., Hashim, F. A., Amin, M., Saber, A., ... & Gandomi, A. H. (2022). Recent advances in harris hawks optimization: A comparative study and applications. *Electronics*, 11(12), 1919.

- [69] Abdullah, G., & Hassan, Z. A. H. (2020, November). Using of particle swarm optimization (PSO) to addressed reliability allocation of complex network. In *Journal of Physics: Conference Series* (Vol. 1664, No. 1, p. 012125). IOP Publishing.
- [70] Sharif, M., Amin, J., Raza, M., Yasmin, M., & Satapathy, S. C. (2020). An integrated design of particle swarm optimization (PSO) with fusion of features for detection of brain tumor. *Pattern Recognition Letters*, 129, 150-157.
- [71] Al Hwaitat, A. K., Almaiah, M. A., Almomani, O., Al-Zahrani, M., Al-Sayed, R. M., Asaifi, R. M., ... & Alsaaidah, A. (2020). Improved security particle swarm optimization (PSO) algorithm to detect radio jamming attacks in mobile networks. *International Journal of Advanced Computer Science and Applications*, 11(4).
- [72] Freitas, D., Lopes, L. G., & Morgado-Dias, F. (2020). Particle swarm optimisation: a historical review up to the current developments. *Entropy*, 22(3), 362.
- [73] Okwu, M. O., Tartibu, L. K., Okwu, M. O., & Tartibu, L. K. (2021). Particle swarm optimisation. *Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications*, 5-13.
- [74] Radhika, E. G., & Sadasivam, G. S. (2021). A review on prediction based autoscaling techniques for heterogeneous applications in cloud environment. *Materials Today: Proceedings*, 45, 2793-2800.
- [75] Desebrock, C., & Spence, C. (2021). The Self-Prioritization Effect: Self-referential processing in movement highlights modulation at multiple stages. *Attention, Perception, & Psychophysics*, 83, 2656-2674.
- [76] Chalapathi, G. S. S., Chamola, V., Vaish, A., & Buyya, R. (2021). Industrial internet of things (iiot) applications of edge and fog computing: A review and future directions. *Fog/edge computing for security, privacy, and applications*, 293-325.
- [77] Posner, J. (2021). *Load Balancing, Fault Tolerance, and Resource Elasticity for Asynchronous Many-Task Systems* (Doctoral dissertation).

- [78] Ye, X. (2022). *Development of emergency response systems by intelligent and integrated approaches for marine oil spill accidents* (Doctoral dissertation, Memorial University of Newfoundland).
- [79] Dhanaraj, R. K., Lalitha, K., Anitha, S., Khaitan, S., Gupta, P., & Goyal, M. K. (2021). Hybrid and dynamic clustering based data aggregation and routing for wireless sensor networks. *Journal of Intelligent & Fuzzy Systems*, 40(6), 10751-10765.
- [80] Jazayeri, F., Shahidinejad, A., & Ghobaei-Arani, M. (2021). Autonomous computation offloading and auto-scaling the in the mobile fog computing: a deep reinforcement learning-based approach. *Journal of Ambient Intelligence and Humanized Computing*, 12, 8265-8284.
- [81] Wang, Y., & Zuo, X. (2021). An effective cloud workflow scheduling approach combining PSO and idle time slot-aware rules. *IEEE/CAA journal of automatica sinica*, 8(5), 1079-1094.
- [82] Devi, K. L., & Valli, S. (2021). Multi-objective heuristics algorithm for dynamic resource scheduling in the cloud computing environment. *The Journal of Supercomputing*, 77, 8252-8280.
- [83] Gill, M., & Singh, D. (2021). A comprehensive study of simulation frameworks and research directions in fog computing. *Computer Science Review*, 40, 100391.
- [84] Afloogee, A. H. N. (2022). Design and development of IoT based simulation framework for wireless sensor network towards environment monitoring.

## الخلاصة

أدى استخدام تطبيقات خادم الضباب المحمول في الآونة الأخيرة إلى توليد كميات هائلة من البيانات التي تشمل تنسيقات بيانات متنوعة، بما في ذلك الصور والمستندات والوسائط المتعددة وأنواع الملفات الأخرى. إن إدارة حجم البيانات المستخدمة في حوسبة الضباب والتحكم فيها يمثل تحديًا. بالإضافة إلى ذلك، تواجه حوسبة خادم الضباب مشكلات في أداء الشبكة مثل زمن الوصول وتخصيص الموارد واللياقة البدنية. من أجل معالجة هذه المشكلات، تم استخدام نهج متطور لتخصيص موارد حوسبة خادم الضباب استنادًا إلى خوارزميات الاستدلال الفوقي، والتي قامت بدمج حوسبة الضباب أحادية العقدة/متعددة مع خوارزمية تخصيص الموارد الهجينة لمحاكاة وتقييم أداء الشبكة في شبكة الهاتف المحمول.

تم تطبيق النظام المقترح في عقدة الضباب بثلاثة حالات دراسية. الأولى هو استخدام خوارزمية Particle Swarm Optimization (PSO) لتعزيز الفعالية وتحسين سرعة البحث وتحسين كفاءة التكلفة للأجهزة المحمولة. والثانية هي خوارزمية Harris Hawks Optimization (HHO) للحفاظ على جودة البيانات المجمعة في شبكة الهاتف المحمول. والثالثة هي خوارزمية تخصيص الموارد المختلطة بناءً على طريقتين متكاملتين للتحسين يمثلهما HHO و PSO لبناء قيمة أفضل لتخصيص موارد أفضل مرونة (BFRA). تم إنتاج المخرجات بواسطة كلا المحسنين، مع الأخذ في الاعتبار خصائص كل منهما، وبالاعتماد على نتائج المحاكاة والتحليل المقارن للمنهجيات التقليدية لتخصيص الموارد الهجينة وما يتصل بها، بالإضافة إلى خوارزميات PSO و HHO، فإن النظام الهجين المتعدد المقترح أفضل مع متوسط المدى هي 78.734 ثانية، ومتوسط سرعة إتمام العملية في الخوادم هو 322.7477 كيلو بايت/ ثانية. تم تحسين متوسط وقت معالجة الأعمال الهجينة الموزعة بالثواني إلى 44% مقارنة بالأعمال ذات الصلة.



جمهورية العراق  
وزارة التعليم العالي والبحث العلمي  
جامعة بابل  
كلية تكنولوجيا المعلومات  
قسم شبكات المعلومات

تخصيص الموارد في الحوسبة الضبابية باستخدام خوارزمية  
ميتاهيورستك هجينة

رسالة مقدمة

إلى مجلس كلية تكنولوجيا المعلومات في جامعة بابل والتي هي جزء من متطلبات  
الحصول على درجة الماجستير في تكنولوجيا المعلومات / شبكات المعلومات

من قبل الطالب

زهراء حسام ابراهيم موسى

باشراف

أ.م.د مهدي عبادي مانع