PRIORITY BASED MULTI-STAGE LAXITY-AWARE WORKLOAD DISTRIBUTION FOR COLLABORATIVE VEHICULAR EDGE COMPUTING

ZESEYA SHARMIN

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

PRIORITY BASED MULTI-STAGE LAXITY-AWARE WORKLOAD DISTRIBUTION FOR COLLABORATIVE VEHICULAR EDGE COMPUTING

ZESEYA SHARMIN

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE (APPLIED COMPUTING)

FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Zeseya Sharmin

Registration/Matric No.: 17198716

Name of Degree: Master

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"): Priority Based Multi-Stage Laxity-Aware Workload Distribution for Collaborative Vehicular Edge Computing Field of Study: Vehicular Edge Computing

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature　　　　Date: 07/07/2021

Subscribed and solemnly declared before,

Witness's Signature　　　　Date: 07/07/2021

Name:
Designation:

PRIORITY BASED MULTI-STAGE LAXITY-AWARE WORKLOAD DISTRIBUTION FOR COLLABORATIVE VEHICULAR EDGE COMPUTING

ABSTRACT

Technological developments have made it possible for smart machines to access ground-breaking applications. The backend cloud servers is unreliable because of the resultant overhead network to cope with increasing processing requirements. This is usually minimized by the use of edge positions to satisfy the growing demands for computation. An significant function is to maximize productivity and reduce the volume of data transferred into the cloud for fog collection, analysis and storage. Edge locations have commonly been aimed at promoting latency related solutions for end-users. However resource-restricted regions are frequently flooded by many ongoing demands, whereas the output of delay sensing systems is difficult to sustain at the lowest end-to-end delay. To provide the quality of services (QoS) to resource-limited end-user using computing resources within the data transmission range and to handle the imbalanced workloads because of the traffic density the micro-level fog unit has formed a fog federation in a network that uses underutilized resources to provide service efficiency and achieved energy reduction through this technique by comparing with the traditional non-federated model where multi-access edge computing is used to process data between fog node and the end-users. Most modern innovative vehicular services are delay-sensitive and computationally complex. They pose challenging obstacles for vehicular networks since vehicles are resource stricken with limited computing and storage capacity. Earlier attempts using edge servers get choked up with increasing vehicular traffic. Moreover, workload balancing at available resources, especially there is limited support for priority tasks. In this paper, we propose a collaborative fog computing system to improve balanced fog

resource utilization by offloading tasks across the fog federation. The participating fog nodes implement a workload-based offloading decision model, enabling collaboration and suitable fog node selection within the collaborative environment. Furthermore, the system implements a priority-aware multi-queue task scheduling to provide high service throughput. The simulation results demonstrate improved performance for the proposed collaborative fog computing system in terms of queuing delay, delay rate, number of task offloading, and pending tasks.

Keywords: Fog computing, Fog federation, Laxity, Multi-access edge computing, Workload distribution.

PENGAGIHAN BEBAN KERJA MULTI-TAHAP KERENTANAN BERASASKAN KEUTAMAAN UNTUK PENGKOMPUTERAN TEPI KENDERAAN BERKOLABORATIF

ABSTRAK

Kemajuan teknologi telah membolehkan akses ke aplikasi inovatif untuk peranti yang disambungkan. Untuk menangani keperluan pengiraan yang semakin meningkat, pusat data awan backend menjadi penyelesaian yang tidak cekap kerana overhead rangkaian yang disebabkan. Ini secara umum dikurangkan menggunakan lokasi tepi yang digunakan untuk memenuhi permintaan pengkomputeran yang semakin meningkat. Untuk meningkatkan kecekapan dan mengurangkan jumlah data yang diangkut ke awan untuk pemprosesan, analisis, dan penyimpanan kabut pengkomputeran memainkan peranan penting. Lokasi Edge biasanya bertujuan untuk mempromosikan penyelesaian berkaitan latensi untuk pengguna akhir. Walau bagaimanapun, kawasan yang dilarang sumber daya sering dibanjiri oleh banyak tuntutan yang sedang berlangsung, sedangkan keluaran sistem penginderaan kelewatan sukar dipertahankan pada penundaan hujung ke ujung terendah. Untuk memberikan kualiti perkhidmatan (QoS) kepada pengguna akhir terhad sumber menggunakan sumber pengkomputeran dalam julat penghantaran data dan untuk menangani beban kerja yang tidak seimbang kerana kepadatan lalu lintas unit kabut tingkat mikro telah membentuk fog federasi dalam rangkaian yang menggunakan kurang digunakan sumber untuk memberikan kecekapan perkhidmatan dan pengurangan tenaga yang dicapai melalui teknik ini jika dibandingkan dengan model bukan gabungan tradisional di mana pengkomputeran tepi pelbagai akses digunakan untuk memproses data antara node kabut dan pengguna akhir. Sebilangan besar perkhidmatan kenderaan inovatif moden sensitif terhadap kelewatan dan komputasi kompleks. Mereka menimbulkan rintangan yang

menantang untuk jaringan kenderaan kerana kenderaan dilanda sumber daya dengan kapasiti komputer dan penyimpanan yang terhad. Percubaan sebelumnya menggunakan pelayan tepi tersekat dengan peningkatan lalu lintas kenderaan. Lebih-lebih lagi, pengimbangan beban kerja dengan sumber yang ada, terutama ada sokongan terhad untuk tugas keutamaan. Dalam makalah ini, kami mencadangkan sistem pengkomputeran kabut kolaboratif untuk meningkatkan penggunaan sumber kabut yang seimbang dengan memunggah tugas di seluruh fog federasi. Nod kabut yang mengambil bahagian melaksanakan model keputusan pemunggahan berdasarkan beban kerja, memungkinkan kolaborasi dan pemilihan node kabut yang sesuai dalam lingkungan kolaboratif. Tambahan pula, sistem ini melaksanakan penjadualan tugas antrian yang diutamakan untuk memberikan perkhidmatan yang tinggi. Hasil simulasi menunjukkan peningkatan prestasi untuk sistem pengkomputeran kabut kolaboratif yang dicadangkan dari segi kelewatan giliran, kadar kelewatan, jumlah pemuatan tugas, dan tugas tertunda.

ACKNOWLEDGEMENTS

First of all, I would like to express my heartiest thanks and gratefulness to almighty Allah for his divine blessing makes me possible to compete the final year thesis successfully.

I am really pleased and greatful to my supervisor Dr. Rafidah Md. Noor, Associate Professor and Deputy Dean (Science) of Institute for Advanced Studies (IAS), Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya (UM). Her deep knowledge keeps me interested in the field of vehicular networking and supports me to carry out this thesis. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many interior draft and correcting them at all stages have made it possible to complete this thesis.

I take this opportunity to express gratitude to Dr. Asad Waqar Malik, Assistant Professor, Department of Computing, National Institute of Science and Technology, Pakistan, for his helps and supports. I also thanks my parents for the unceasing encouragement, support and attention.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

Acronyms	Meaning
AWGN	Added White Gaussian Noise
DM	Decision Manager
DPTO	Delay-dependent Priority-aware Offloading
FCFS	First Come First Serve
FDN	Fog Delivery Network
F-FDN	Federated Fog Delivery Network
FM	Manager of Federation
FTTH	Fiber to the Home
IoT	Internet of Things
ITS	Intelligent Transportation System
LAN	Local Area Network
LBP-ACS	Laxity and Colony Method Algorithm
LLF	Least Laxity First
LTE	Long-Term Evolution
MAB	Multi-armed Bandit
MEC	Multi-Access Edge Computing
NFA	Neighbouring Fog Algorithm
PEPA	Performance Evaluation Process Algebra
PV	Public Vehicles
QoS	Quality of Services
RAN	Radio Access Network
RSU	Roadside Units

Acronyms	**Meaning**
RWA	Random Walk Algorithm
SNR	Signal to Noise Ratio
VANET	Vehicular Ad-hoc Network
VCC	Vehicular Cloud Computing
VEC	Vehicular Edge Computing
VFC	Vehicular Fog Computing
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything
WAN	Wide Area Network
WM	Workload Manager

LIST OF SYMBOLS

Symbol	Meaning
B	Bandwidth
C	Channel Capacity
d_i	Deadline
E	End Device
F	Fog Node
$\mathscr{F}$	Federated
$\mathscr{F}'$	Non-federated
h	Channel Gains
L	Laxity
l	Path Loss
N_0	Noise Spectral Density
P	Channel Transmission Power
Q	Queue
S	Scenario
s	Data Size
t_i	End-to-End Delay
$\mathscr{U}$	Uniformly Offload
v	Compute Capacity
W	Workload
X and Y	Data Transmission Rate
α	Initial Offset
β	Path Attenuation Index

Symbol	**Meaning**
δ	Offloading Decision
ϕ_i	Priority
ε_i	Shadow-fading Effect
μ	Execution Times
λ	Decision

CHAPTER 1: INTRODUCTION

With the advancement in communication technology, vehicles are more connected in the era of the internet of things (IoT). This enables the development of innovative vehicular services including automotive maintenance system, fleet management, connected cars, autonomous driving, and in-vehicle infotainment and telematics (Vegni & Loscri, 2015). These applications usually require a significant amount of computation; whereas the vehicles are often resource stricken with limited computing and storage capacity. Therefore, they are unable to provide good service quality, which is a bottleneck when supporting innovative services. Recently, to facilitate such innovative services, multi-access edge computing (MEC) is used as an alternative computing resource. The MEC servers are deployed at the edge of the network with additional computing capacity to support resource-constraint vehicles (Rodrigues et al., 2016). A typical vehicular offloading model with MECs is illustrated in Figure 1. The connected vehicles offload compute-intensive tasks to nearby MEC servers to reduce service delay. In recent years, researchers have focused on computation offloading in vehicular networks supported by the concept of smart cities. However, very limited works exist where tasks are prioritized based on their priority. In most cases, the offloaded tasks discussed are in terms of computational complexity.

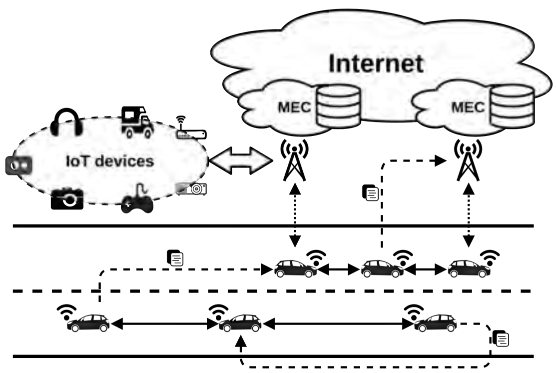

Figure 1.1: Typical vehicular offloading model where vehicles and multi-access edge computing (MECs) are used as alternative compute resource.

Fog Computing is a clustered computing framework in somewhere between data source and cloud data, analysis, storage and applications are situated. Fog Computing takes the benefits and intensity of the cloud closer to where data is generated and practiced like edge computing. Many people overlap the words fog computing and edge computing, both because the information and processing are more in line with the location in which data are generated. Sometimes this is done to increase performance, but for protection and compliance purposes too. In 2012, Fog Computing (Bonomi et al., 2012) was introduced as the perfect model to enable the data collection and knowledge transmission of resource-constrained IoT systems. The topological proximity to such instruments is the biggest enabler of certain advantages that were not available when the remote cloud was continuously unloaded. Cloud Computing technology and concepts are spread through Fog Computing, which is not meant to replace the centralized Cloud, but to co-exist and collaborate with it everywhere along the Cloud-to-Things spectrum and particularly at the network edge, in close proximity to the IoT computers. Due to the growing use of cloud storage, problems such as inconsistent latency, lack of mobility support and knowledge of the position remain unresolved. Fog computing solves these issues by supplying end-users with scalable infrastructure and facilities at the edge of the network (Yi et al., 2015), while cloud computing offers more resources in the main network.

The implementation of connected vehicles has a rich networking and communicating scenario: vehicles to vehicles, vehicles to points of access (wifi, 3G, LTE, highway units, intelligent lights) and network devices to points of connection. The Fog offers a range of attributes, making it the perfect framework to provide a robust range of Electric mobility services for infotainment, surveillance, road support and analytics: geo-distribution, accessibility and location knowledge. Low latency, heterogeneity and real time support.

The above is explained by an adaptive traffic light scheme (Bonomi et al., 2012). The intelligent node of light traffic communicates locally with a variety of sensors that sense the presence of foot-and-motorcyclists. It also interacts to align the green traffic wave with adjacent lights. The smart light sends warning signals to incoming cars based on this information and also adjusts its own period to deter accidents.

Fog Federation are used to maximise the value and optimal usage of available resources. The concept of federation refers generally to the participation of independent entities in a common agreement which provides a cooperative structure for optimizing benefit (Zhanikeev, 2015). In addition to offering computer resources and access to centralized servers (i.e. cloud) to communications networks edge users, fog computing was planned to reduce the request response time of operation. The fog paradigm must guarantee the following characteristics in order to achieve its goal: knowledge of position and geographic spread, low latency, support for accessibility, support for heterogeneity data, realtime interaction, federation and service/application scalability. For secure control and management of multiple fog domain instance fog federation is introduced.

A traditional fog-federation consisting of a number of fog units working together to maintain QoS and workload balance (Kapsalis et al., 2017). Each device can connect with terminals within its limited range. Often they are linked by a high-speed network. In this research, we use the idea of micro-level fog deployment in which the associated fog units at intersections forming a fog federation are deployed. The main purpose of using federation within this research is to avoid fog units in order to efficiently distribute the workload.

Multi-Access Edge Computing (MEC) can be characterized as a model for business-oriented cloud computing platforms that serve responsive, context-conscious application delay within multiple types of the access network (e.g., LTE, 5G, WiFi, FTTH etc.) (Tanaka et al., 2018). MEC must be built to optimize its capacity as an infrastructure in addition to an effective management framework in order to serve a wide variety of IoT technologies and their eco-systems. Within the expanding sustainable environment, MEC aims to converge IT and mobile communications services and to have a radio access network within edge cloud infrastructure. For mobile end users, MEC delivers congestion and the effective use of cutting-edge mobile backhaul and core networks (Taleb et al., 2017). MEC technology seeks to expand the power of cloud infrastructure to the edge of the radio access network and thereby provide access to the capabilities of the radio network in real time, with high bandwidth and low speed. In the capacity of MEC to provide network edge cloud platform and gateway services, IoT is defined as a key case for MEC (Porambage et al., 2018). Due to its dense geographical distribution and strong support for mobility, MEC is promoting the production of numerous applications and services that require ultra-low latency and high level of quality of service. Thus, MEC is a significant enabler of real-time operations in IoT applications and services.

Laxity within the cloud-fog is a theory, was formulated in the today's IoT research field which enables an IoT device to consume electricity and to respond to delays and needs almost in real time. However, until now it has not been thoroughly discussed how to plan computing functions that are to be discharge into fog nodes or cloud nodes (Enokido et al., 2010). A related task planning problem in a narrow cloud-fog-environment optimisation process has been introduced to solve the dynamic task planning problem with some of the priority restrictions of IoT implementations in terms of energy usage and energy savings, subject to reaching the mixed-term deadline. In order for this problem to be solved, there is a laxity and colony method algorithm (LBP-ACS) (Xu et al., 2019) is used to handle the sensitivity of task delay. The laxity-based priority algorithm is adopted to construct a task scheduling sequence with reasonable priority.

Each task is assigned with the laxity and the minimum laxity process first. The laxity of a task is defined by the difference of deadline and remaining time of computation. When the process loads and has less laxity compared to the task in progress, the task will be closed with more laxity and the remaining time will be allocated after calculation and the task with less laxity will continue to be performed within its deadline. It is an ideal system algorithm which processes the periodic tasks in real time.

1.1 Problem Statement

In this section, we provide insights into the shortcomings of the existing offloading strategies that are preventing the recent Fog Computing paradigm from distributing workloads (Santoro et al., 2017). Mobility (Bittencourt et al., 2017), addressing processes, as well as the need for novel control plane techniques (Ren et al., 2018), involving different resources exploration and visualization functionality, to name a few, are some of the drawbacks.

Edge locations have commonly been aimed at promoting latency related solutions for end users (Mukherjee & Amitav, 2018). However resource-restricted regions are frequently flooded by many ongoing demands, whereas the output of delay sensing systems is difficult to sustain at a lowest end-to-end delay.

Holding the fog part topologically close to the associated IoT nodes (Gubbi et al., 2013) does not seem like a hard job at first glance. Nevertheless, the question is much more complicated than what it seems to be. The first problem is the task offloading distribution system when numerous evolving requirements often overflow within resource-restricted regions with its fog node.

Last but not least, delay sensitive and computational complexity within the vehicular network is recognise as a modern innovative (Gu & Zhang, 2021). The main challenge for vehicular network is resource utilization within its limited computing and storage capacity along with workload balancing at available resources.

1.2 Research Motivation

Internet of Things (IoT) being a part of our everyday life and our environment, we believe the vast majority of connected devices to rise rapidly. The emergence of the IoT means that a huge volume of data is generated by millions of devices and sensors each other. Notably, the ever-increasing number of connected vehicles in urban areas, all offloading tasks at the nearby MEC server, leads to workload imbalance among the MEC servers (Beraldi et al., 2017). Consequently, this results in the degradation of service quality. Over the years, significant effort has been placed to evenly distribute the workload among computing servers in a cloud data center (Mishra et al., 2020). As per our literature review, limited works cover workload imbalance issues for priority tasks among fog devices, especially in fog federation. The motivation to tackle this issue is to maximize service quality of fog computing system and to support delay-sensitive and innovative services in vehicular networks.

1.3 Research Question(s)

To meet the objectives of this research, the underlying questions are required to be answered-

1. How resource utilization can be improved by using fog federation and laxity?
2. Why multi-stage laxity based collaborative fog system is used in the workload distribution system?
3. How to evaluate the proposed method and what performance metrics are used?

1.4 Research Aim(s)

Due to increased computational activity, schedule time effect on system efficiency increases, in particular when using dynamic priority algorithms. This overhead can be minimized by the use of dedicated hardware that performs the time needed calculations.

In dealing with cloud computing, one critical question arises about the delay in data transmission from vehicles to the cloud server and recovering the information after it is stored and processed. The aim of this work is to research, investigate and develop a collaborative fog computing system to improve balanced fog resource utilization by offloading tasks across the fog federation. The participating fog nodes implement a workload-based offloading decision model, enabling collaboration and suitable fog node selection within the collaborative environment. To efficiently utilize the fog computing power, deployed at the edge of the road network, we propose a collaborative laxity-based priority-aware computing model. In this work, fog devices are deployed, connected through wire and wireless connection.

1.5 Research Objective(s)

The main objective of the model is to schedule tasks efficiently so that tasks of all types can meet their predefined deadlines, as stated earlier to improve the overall service quality of the fog computing system. For achieve the aforementioned aim, the under-listed objectives are defined:

1. To develop an enhanced multi-stage laxity based collaborative fog computing model that improves existing workload distribution in VANET.
2. To improve the laxity-aware multi-queue task scheduling model for reducing the workload distribution system.
3. To achieve high service throughput the proposed model is implemented.

Table 1.1: Mapping of research objectives and research questions

Research Question	Research Objectives
How resource utilization can be improved by using fog federation and laxity?	Develop a priority-based queuing model for collaborative fog computing
Why priority multi-stage laxity based collaborative fog system is used?	To improve existing workload distribution system
Why the proposed model is implemented?	To achieve high throughput

1.6 Proposed Solution

Fog computing is an enhanced cloud computing edition that extends some of its resources to end-users (Yang et al., 2020). The rising levels of wireless connectivity require mobility and wide geographical distribution that Fog Computing fulfills (Kaur & Sachdeva, 2020). When the tasks are offloaded from one fog node to the other fog nodes, it creates a burden on these fog nodes because of the limited computational resources. Each task has a priority constrain relationship with the others. When the users own fog device can not meet the demand, it enables to leasing cloud resource to handle the tasks. The associated task scheduling strategy based on laxity in cloud-fog environment, which takes into account the energy consumption and tries to fulfil reduce energy consumption on the condition of satisfying the mix deadline.

To provide the quality of services (QoS) to resource-limited end-user using computing resources within the data transmission range and to handle the imbalanced workloads because of the traffic density the micro-level fog unit (Shamseddine et al., 2021) has formed a fog federation in a network that uses underutilized resources to provide service efficiency and 72% energy reduction is achieved through this technique in compare with the traditional non-federated model.

For achieving high level of throughput and to improve the offloading task we propose a multi-tier priority-based fog computing system. At the fog nodes, we use online laxity-aware multi-queue task scheduling for improved task fairness and throughput. Here, task laxity is a dynamic measure has a direct relationship to the task completion deadline and is updated at every time instant. Furthermore, we implement a scheme to exploit federated fog resources based on workload.

In this study, we propose a multi-tier priority-based fog computing model. At the fog nodes, we use online laxity-aware multi-queue task scheduling for improved task fairness and throughput. Here, task laxity is a dynamic measure having a direct relationship to the task completion deadline.

The proposed offloading model is a multi-stage decision process, where the end devices take the offloading decision which is followed by another offloading stage to better-utilized fog node through fog grid. The main objective of the model is to share resources to execute the priority-aware tasks efficiently and maximize the tasks to meet their predefined deadlines. As stated earlier, the purpose is to improve the overall service quality through a federated fog grid.

Most of the recent works consider priority task offloading either vertically or horizontally. In imbalanced workload scenarios, frequent offloading leads to overloaded computing resources, resulting in lower system performance. Our proposed model scales vertically and horizontally while fulfilling task completion deadlines. Furthermore, we implemented a model to exploit federated fog resources based on workload.

1.7 Research Contribution(s)

To efficiently utilize the fog computing power, deployed at the edge of the road network, we propose a collaborative laxity-based priority-aware computing model. In this work, fog devices are deployed, connected through wire connection. The most significant contributions of the work proposed are described below:

- Design a collaborative fog computing model to improve balanced fog resource utilization by offloading tasks across the fog federation.

- Implement a priority-aware multi-queue task scheduling model at the fog nodes to provide high service throughput.
- Provide a workload-based offloading decision model, enabling collaboration and suitable fog node selection within the collaborative environment.

1.8 Scope of Study

The focus of this research is to shared the resource for task efficiency and achieved high throughput by enabling collaborative workload distribution system. While fog computing is not inherent to cloudlets, it captures and secures data from vehicles traveling across a large geographic area in a variety of environmental conditions, allowing for geomobility. To reduce latency, fog computing analyzes extremely time-critical data and makes a decision near the vehicles. Sometimes, it became tough to take decision weather offload to fog or local. To reduce this, we ensure the task should be schedule first then took decision.

1.9 Thesis Organization

This research consists of six chapters. The first chapter introduce us with multi-access edge computing. Apart from that, problem statement, research motivation, aims and objectives along with research question, proposed solution and the scope of study. The rest of the chapters of the research are organized as follows-

- Chapter 2: The literature study on the subject matter i.e offloading techniques, fog federation, vehicular edge and priority based computing. Also, comparison of existing and reviewed priority based workload distribution in collaborative fog system.

- Chapter 3: We describe the methodology of multi-stage priority based workload distribution system. The task model, communication model, queuing model, task execution granularity model, tiered computational model, offloading decision model and collaborative task offloading methods are described in this chapter.
- Chapter 4: The proposed collaborative fog computing model and its implementation details are described in this chapter. Our proposed solution is simulated in Anylogic. Experimental tools and software details are also described in this chapter.
- Chapter 5: The result of performance evaluation, depicted from the simulation and extensive experiments, show that the proposed system is significant improvements compared to state-of-art-works.
- Chapter 6: Finally, we draw conclusions and mentions directions for our future works.

CHAPTER 2: LITERATURE REVIEW

In getting a deeper understanding of the subject area, this research includes a literature review which covers the recent contributions in vehicular edge computing and priority-based computing (Yang et al., 2019), a review of well known and already existing offloading mechanisms and a comparison of the reviewed priority based workload distribution in collaborative fog system. The first section provides brief introduction about vehicular edge computing and then in the second section explain laxity based computing and how it works. In third section, it describes about task offloading techniques in vehicular network. Fourth section tells us about federation and compare with some existing works. Last but not the least is section five where summarising of chapter two is done by comparing some existing work.

2.1 Vhicular Ad-hoc Network (VANET)

As technology has rapidly exploded with the advent of cloud computing over the past few years, vehicle networks have been also formed and become ubiquitous day by day. VANET in recent years have gained prominence. The vast amount of vehicles have brought significant global problems, including traffic collisions, road congestion, fuel consumption and environmental pollution (Verma et al., 2021). In developed and emerging countries, traffic collisions are ongoing problems and end in a substantial loss of life and property. Intelligent Transportation Syèmes (ITS) has released VANETs in order to address these problems and make the journey smoother, effective, hassle-free and enjoyable, in order to create a safer road transport system (Aramrattana et al., 2019). VANETs concentrate on road safety and effective road traffic control, while providing drivers and passengers with convenience and comfort throughout their journeys (Meneguette et al., 2013). VANETs use RSUs as a portal to connect to cloud providers on the move via a virtualization layer. Cloud technologies are used for traffic and multimedia information in car nodes. Vehicle-to-vehicle connectivity (V2V) and vehicle-to-infrastructure (V2I) communications are possible via the sharing of information within VANET. A VANET that uses the cloud can preprocess the positions of all vehicles, receive immediate traffic flows and evaluate the target area.

In VANETs, security and non-security applications are primarily used in 2 kinds. Security applications in VANETs are used for transmitting safety signals, for example various warning messages that help cars on the road to escape collisions and to deter unsafe conditions for pedestrians. The safety alerts cover incidents including traffic collision reports, road-building reports and emergency vehicle warnings (Grassi et al., 2013). These applications need a low latency and high reliability. Nonsecurity implementations, on the other hand, have an easy and convenient approach Experience driving. Applications for non-security are split into two categories: traffic and infotainment control. Applications for traffic control are used to boost traffic flow and to overcome road congestion. Knowledge and entertainment technologies, such as data collection, video streaming or video calls are used to provide Internet connectivity to travelers. In contrast to safety applications, these applications do not require high reliability and low latency.

VANETs are based on V2I communications, in which vehicles can broadcast data on a regular basis to keep RSU up to date. All RSU sends traffic data to a cloud or a central server, which allows for global study and detection of road congestion levels. When a big amount of data is sent to any computing service providers, communication costs may increase. Using fog computing, a data clustering methodology is used to reduce traffic information at the edge of vehicular networks. Two strategies for reducing the traffic data stream are defined by (Peixoto et al., 2021) et.al. for data clustering framework.

Although VANETs use the cloud to address current issues, some key problems, along

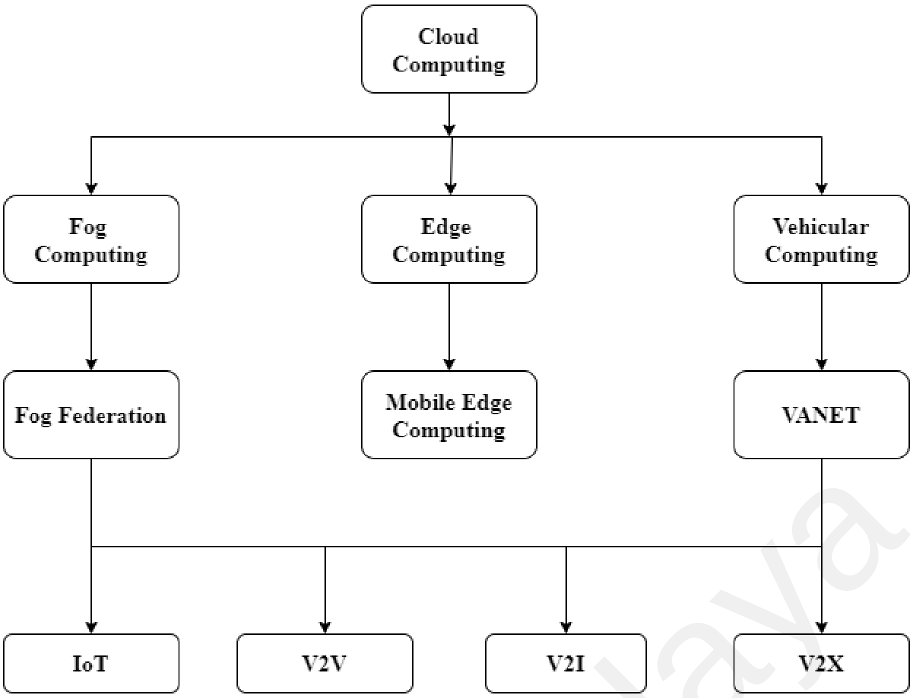

Figure 2.1: Relationship of Cloud, Fog, Edge and VANET

with the rise of smart vehicles, also have to be overcome. Based on the rapid growth and success of VANETs, some primary problems and specifications of potential VANETs are established. Future VANETs and their uses will go beyond the present trend and combine emerging new technology into new features. The following are some of the main challenges of future VANETs-

- **Interrupted compatibility:** access control monitoring and maintenance between vehicles and networks is a challenging task. Interrupted links in vehicle networks must be stopped due to high mobility of vehicles or a high loss of packets.
- **Location awareness and high mobility:** future VANETs need high mobility and awareness about location of interacting vehicles. To deal with an emergency situation, each vehicle should have the correct location of other vehicles on the network.
- **Heterogeneously operated vehicles:** a large number of heterogeneous smart vehicles will be available in future. Another problem of potential VANETs is the handling of heterogeneous vehicles and their intermittent linkages.

- **Security and privacy:** the protection of the data information and location of the user is often a concern. The vehicles that connect inside the infrastructure should allow users to determine which information should be transmitted and what privacy should be protected. Privacy can be guaranteed by locally reviewing confidential data, rather than sending it to the cloud.
- **Intelligent network:** the need to support network intelligence may be one of the problems for future VANETs. In future VANET's, a significant number of sensors would have been installed and, before exchanging the details with other areas of the network, the edge cloud receives and prepossesses, for instance, traditional cloud servers.

Basically, two types of vehicle connectivity typically exist: V2V and V2I. Additionally, other organizations, such as motor cars, pedestrians and utilities, can capture information from their surroundings in order to process and exchange it so that more intelligent resources can be offered (i.e., to obtain information from other vehicles or from another sensor equipment within range). Cooperative collision warning and automated movement comprise these systems. In our previous comprehensive analysis of fog and big data value, a unique virtual vehicle coordination system was addressed that would help to deploy potential cities. Figure 2.2, represent overall vehicle communication.

Wireless technologies facilitate the production of modern vehicle software and facilities, enabling mobile vehicle connectivity and communications between vehicles (V2V) and networks knots (V2I). In general, V2V communications are aimed at communicating small messages primarily geared towards improving protection. Instead, V2I calls allow users to access the Internet and benefit from applications of higher level. The integration of the V2V and V2I is known as V2X communications, which will further boost the advantages of smart transport networks (ITS).

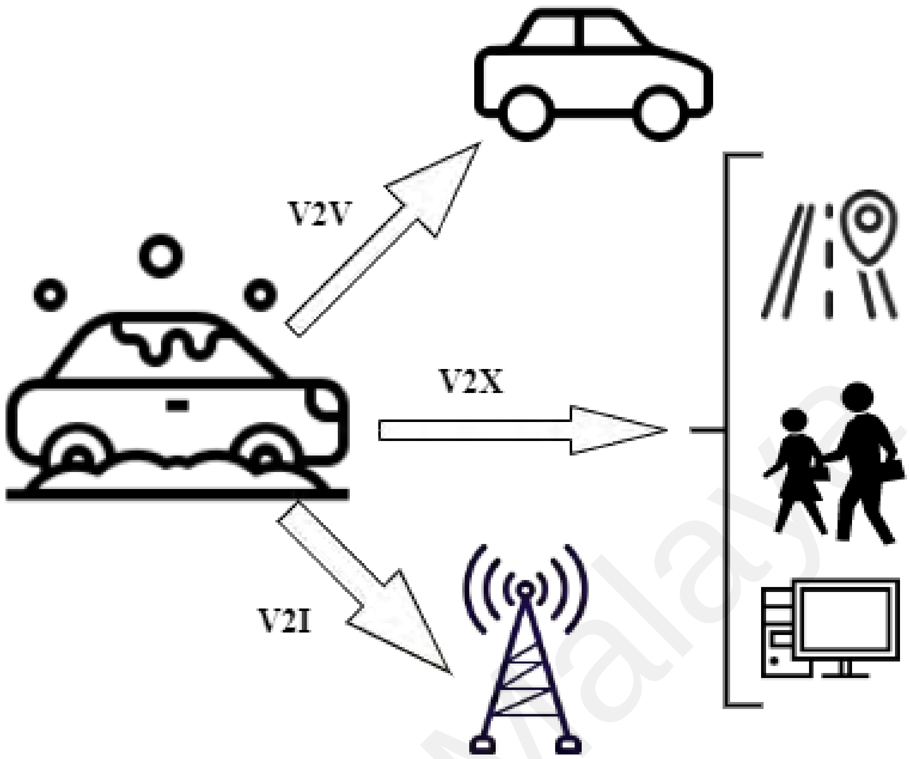

Figure 2.2: Types of Vehicles Communications

2.2 Vehicular Edge Computing (VEC)

In recent years, a new networking model has been applied to the vehicle network, Vehicular Edge Computing (VEC), to expand its capacities in computation. With the emergence of ever rising modern car applications, the final obstacle to satisfy the demands of connectivity and computing is becoming increasingly popular. With VEC, service providers are hosting facilities directly in close proximity to intelligent vehicles, which reduce latency and improve service efficiency (QoS). VANETs are mostly cloud based. Cloud Computing offers unified computing and storage facilities with a cloud infrastructure or a server entity (Ageed et al., 2020). Cloud computing helps users with virtual servers that allow remote storage and data facilities. Data stored can be collected from anywhere without a significant volume of energy and computation in vehicles. Users will also share an immense volume of data between the vehicles. There is a primary issue regarding the delay of moving data from vehicles to the cloud server and the retrieval of information after preparation and processing when interacting with the cloud computing sector. This aspect adds to the need for technologies that maintain low latency and continuous operation, and to the growth in the number of vehicles with increasing mobility (Shojafar et al., 2016) . Furthermore, the connectivity between the vehicles and the cloud server requires high bandwidth. The high network load also contributes to more energy usage on different wireless devices, which has a direct effect on bandwidth costs. It has thus become difficult to satisfy computational and connectivity demands, as vehicle applications are rapidly advancing. Another main problem facing Vehicle Cloud Computing providers combining cloud computing with VANET is to fulfill the required QoS.

VEC refers to the use of network edge devices, multi-access edge computing (MECs), or roadside units (RSUs) for task computation. Due to their limited communication range, at times, it becomes impossible to deliver task output to the source vehicle, resulting in a delivery failure. To overcome this issue, vehicular fog computing (VFC) framework is introduced to share the available resources among the vehicles and to improve service quality. The V2V framework facilitates the sharing of the computing power of underutilized vehicles with overloaded vehicles. However, due to vehicle mobility, this faces various challenges related to vehicle speed and direction and its random mobility.

In (Madan et al., 2020), a flying IoT concept is used to manage overloaded RSUs, reducing task service delay. Similarly in (Rahman et al., 2020), a V2V task offloading framework is proposed to exploit available computing resources at nearby vehicles. The underlying opportunistic vehicle selection improved the overall system efficiency. Notably, the computing load at RSUs increase exponentially with the increase in connected end devices; thus, this adds additional service delay. In (Ye et al., 2016), a scalable fog computing paradigm is proposed using a network comprising bus fog servers. The servers act as compute resources for nearby overloaded RSUs. Similarly, in (Shah et al., 2019), selected vehicles are designated as fog computing nodes to facilitate other vehicles. This helps alleviate the workload on nearby RSUs and MEC servers.

Similarly, VEC enables task offloading to the network edge easing the computational load on vehicles and satisfying real-time application needs. In (Zhang et al., 2016), a contract-based resource sharing framework for MECs is used to fulfill task computation demands. Fan et al. (Fan & Ansari, 2018) study cost and response time as the main performance measures for fog-based deployment. They set up an M/M/1 queuing model for a mobile cloud system and introduce a workload allocation strategy to consider network latency and service delay. Similarly, service delay and queuing cost for an offloaded task is explored in (Liu et al., 2017). The work uses linear programming techniques to manage energy consumption for both the fog and mobile edge computing. In summary, most of the existing works are on the efficient utilization of edge devices to maximize their service rate.

2.3 Fog Paradigm

As part of our daily lives and our environment IoT is a quickly expanding part of the overwhelming majority of mobile devices. When the IoT appears, millions of computers and sensors each other produce an immense amount of data. It is sent to the cloud to be analysed and measured, but due to latency, bandwidth and storage issues we need paradigms at the edge of the computers. Fog computing is an improved form of cloud computing that expands those tools to end users. (Yang et al., 2020). Fog Computing is addressing the growing need for connectivity and wide regional distribution. (Kaur & Sachdeva, 2020). IoT is predicted to connect trillions of devices and people to produce promising advantages.

In 2020, over 20 billion connected devices with unpredictable effects on the gross domestic product and connected resources will produce over 44GB of data, presenting interesting privacy, accessibility, scalability and other challenges (Brito et al., 2018). Increasing demand for network devices in the IoT sector increased the need for reliable data connectivity for delay-sensitive applications. In the field of IoT, fog computing plays an important role in reducing these requirements and it also allows software and services which does not match the model of the cloud. It serves as a link between the Cloud and the internal networks (Desai & Thakkar, 2019) and the natural extension of Vehicle Fog Computing (VFC) in Intelligent Transport Systems (ITS).

2.3.1 Micro-level fog paradigm

Edge locations have commonly been aimed at promoting latency related solutions for end users. However resource-restricted regions are frequently flooded by many ongoing demands, whereas the output of delay sensing systems is difficult to sustain at a lowest end-to-end delay. Close vehicles have recently been used to mitigate this issue of efficiency. To provide the quality of services (QoS) to resource-limited end-user using computing resources within the data transmission range and to handle the imbalanced workloads because of the traffic density the microlevel fog unit (Sharmin et al., 2020) has formed a fog federation in a network that uses underutilized resources to provide service efficiency and 72% energy reduction is achieved through this technique in compare with the traditional non-federated model.

Micro-level edge computing systems have a restricted range of connectivity at crossings since they have a maximum number of vehicles available for sharing resources. In (Iqbal et al., 2020), the smart vehicles do a dual function to end-users and resource sharing computing units to handle the workloads and support delay-sensitive applications. To deliver processing resources at the edge of the network, mobile edge computing can allow delay-sensitive applications (Ge & Xu, 2020). However, the edge resources were inadequate to quickly process real-time mobility due to the enormous quantities of data produced on the Internet of Vehicles (IoV) (Fangchun et al., 2014). With the needs of IoV a multi-layered network architecture proposed by the (Qiao et al., 2018) to meet the low latency and real-time mobility. To satisfy the deadline requirement of real-time task, the multi-level queuing model are able to achieved the tradeoff between the throughput and long delay (Li et al., 2019).

2.3.2 Federated Fog

Nowadays, the number of vehicles are increased rapidly and it becomes a great challenge for researchers and engineers to find out how to accord with communication and computational insistence more efficiently and competently. For better utilization of individual communication and computational resources of each vehicle (Hou et al., 2016) assume the idea of utilizing vehicles as the framework for communication and computation, named vehicular fog computing (VFC).

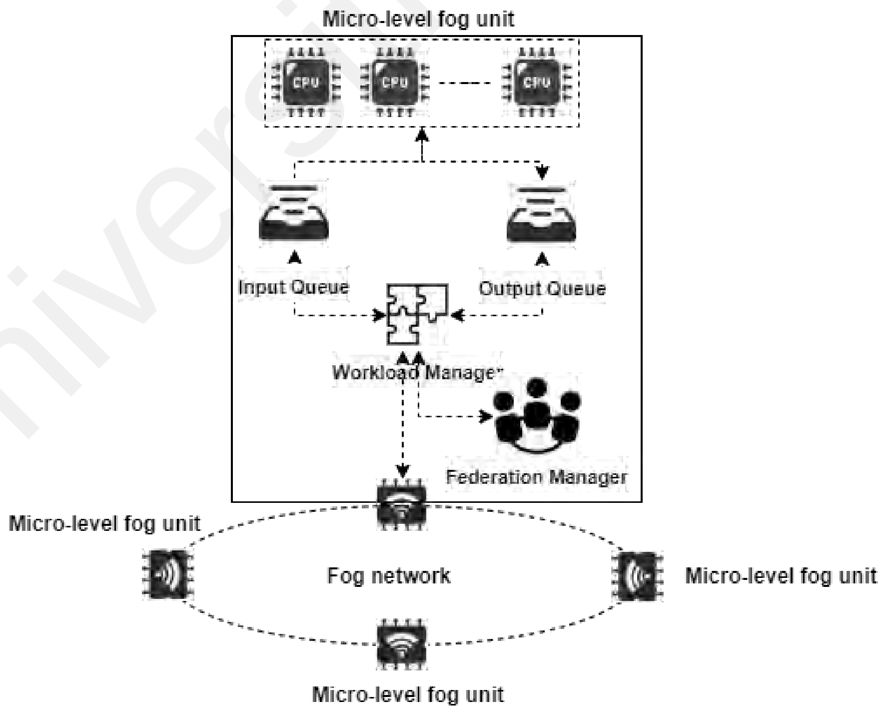

Figure 2.3: High-level fog consortium model

For delays related applications, the concept of fog computing has been extensively observed. In comparison, the data transmitted throughout the backend network is limited.

However, it is difficult to sustain QoS at an edge position for a large number of devices in any area. It is important to control QoS in its coverage area by micro-level fog placing (Sharmin et al., 2020). This recent study of federated fog computing enables functionality at various geo-locations rather than at a centralized location accessible via a single network link, as illustrated in Figure 2.3. In addition, it offers lateral expansion versatility over fragmented fields. The direct contact and extendable system minimizes the reliable link request response time and thus achieves enhanced QoS as the functionality lies closer to the service request source.

Fog runs on a network edge rather than viewing or operating from a centralized cloud. That's why it takes less time. Fog has three layers i.e fog node, fog domain and fog federation (Al-Khafajiy et al., 2020). Fog nodes implements fog computing services and it is analogous to a server in cloud computing. Fog domain extends the cloud computing to the edge for secure control and management of domain specific network functions. For secure control and management of multiple fog domain instance fog federation is introduced.

There is currently a fog federation system or program, which manages and federates fog services across various operating realms, and is like the hybrid cloud federation scheme (Masdari & Zangakani, 2020). New systems are needed, particularly when they belong to various operating realms, for the federation of fog nodes. The federation scheme should take fog nodes of various vendors/operators into account for resource sharing models. New price models for federated fog services can also be described. Finally, it is possible to propose policies within the system of the federal government for modern fog resource share systems (Anglano et al., 2018). Current versions are expected to use fog processing to handle the requests, while fog nodes will download computations between themselves or to a cloud. IoT machines (clients) use cloud-computing or fog calculation (servers) to process their requests. We contend that the architecture of the entire peer-to-peer (P2P) fog computing resource platform is a strong guidance. In the P2P fog paradigm, fog nodes exchange P2P resources, where single nodes with fog share their resources (e.g. computation or storage) without the use of third-party intermediaries. Fog federations or related applications to hybrid cloud federation systems are not supported by some of the existing system. In this research fog federation gives some potential solution by-

- Introduce additional strategies for the federation of fog nodes around multiple subsectors.
- Fog nodes from various providers/operators built resource sharing models.
- Defining current price concepts for the sharing of federated fog services.

2.3.3 Federated Vehicular Fog Network

The challenge has been a safe and error-free connectivity between moving vehicles in a vehicular network with an increasing number of smart vehicles. At the edge, such networks are raised using RSUs connected to the backend cloud data centers (Kuo et al., 2013), within the VFC paradigm, but constrained bandwidth and high cost of deployment. Vehicle fog computing is currently used to further optimize the connectivity and computing capabilities available in vehicle networks through efficient load sharing between neighboring fog units (Hou et al., 2016; Menon & Prathap, 2017). However, to support delay-sensitive applications in (Veillon et al., 2019), fog delivery networks (FDN) architecture is extended to include federated fog devices, termed as federated fog delivery network (F-FDN). There are multiple configured FDNs connected to the backend cloud data center throughout the architecture. A fog-based storage architecture known as Nebula is primarily aimed at geographic, neighborhood-based and informational storage applications (Ryden et al., 2014). There is no question that for any delay-sensitive application, successful cloud storage management and computer resources available in nearby fog devices are necessary.

The space and communication requirements of vehicular networks are historically limited, whereas recent ad-hoc vehicle networks (VANETs) are cloud-dependent. The following includes low latency, uninterrupted vehicle cloud computing (VCC) facilities. But as services become all-round every day, they require a high bandwidth, which is a difficulty for reaching the QoS, to connect with the cloud server. With the advent of ever growing linked cars, the operational performance is being improved by using a new networking model called vehicular fog computing (VFC) (Cheng et al., 2017). Vehicles here offload vehicle-to-the-all (V2X) computation at the network's edge. In comparison, transitional storage on the roadside may be introduced (RSUs). These offloading mechanisms require over-all connectivity, thereby involving new architecture methods to distribute computational and communication capital to offers sustainable user interface and counter energy-performance tradeoffs.

2.4 Task Offloading

Task offloading is one of the most common issues in fog computing. A few analysts tended to the energy issues and defined assignment offloading as deterministic optimization problem. In (Fuqian et al., 2018), authors proposed a framework of task offloading from a user equipment to multiple fog nodes in fog-enabled networks and try to solve the task offloading problem via an index policy i.e Whittle's index policy. Previous studies presume that an application's tasks are independent to encourage the offloading process. However, there is an internal dependent relationship between tasks in an application. A task may have prior tasks that must be achieved before the task starts, as well as predecessor tasks that can not be done until the task is completed. Assuming the interaction between tasks, authors (Yuan et al., 2012) use call graph between application methods, which implies dependency on tasks. Sometimes it becomes more complex in the offloading assignment of multiple applications for each one comprising several dependent tasks in a system. Even though the offloading of dependent tasks under previous constrictions is complex, prior works point to a specific metric, energy or start making-up.

A new metric is implemented by (Yinuo et al., 2019) to take into account the characteristics of different devices to calculate the total cost of the offloading process. To minimize the system cost all task have to complete within its deadline and he term of maximum delay and energy consumption of the UbiComp (Saraswat et al., 2019) process can be obtained from the periodic function within the particular deadline. Application deadline and cost minimization are jointly considered in (Weihong et al., 2021). The researchers strive to save maximum energy on a mobile device by unloading activities, according to the application deadline. Furthermore, we use two traditional task offloading algorithms - random walk algorithm (RWA) and neighboring fogs algorithm (NFA) for comparison.

- **Random walk algorithm (RWA) –** In a random walk, the tasks are moved to the random fog locations to balanced the workload (Zhu et al., 2017). Here, every fog unit uniformly offloads tasks within the fog nodes, the set of RSUs R, mathematically, $\mathscr{U}\{r \in R\}$. There are no specifics collection create for fog node selection.
- **Neighboring fogs algorithm (NFA) –** The workload is shared with only adjacent fog nodes (Bozorgchenani et al., 2017) in this method. The origin fog node consistently offloads the tasks in dispersion through its nearest fog nodes $R' \subset R$, ones with less propagation delay, mathematically, $\mathscr{U}\{r \in R'\}$.

Vehicles may add their computational power to the network in-vehicle edge computing (VEC) systems and support other vehicles or users to manage their computing tasks.

Sometimes it becoms more challenging to optimize the delay performance of tasks to design task offloading algorithms. (Xiao et al., 2019), propose an adaptive learning-based task offloading (ALTO) algorithm and a learning-based task replication algorithm (LTRA) based on the multi-armed bandit (MAB) theory. To minimize the average offloading delay (Sun et al., 2021) extended the classic MAB algorithms.

Based on ant colony optimization, a distributed (much lower complexity) task offloading algorithm is proposed by (Dorigo & Stützle, 2019). Moreover, vehicular state exchanges are requires over there. For overcome this types of ambivalence in the vehicle cloud computing system and improve the accuracy of the service, replicated task offloading is proposed in (Zhiyuan et al., 2017), where task replicas are assigned to multiple service vehicles at the same time.

2.4.1 Offloading Task in Vehicular Network

Task offloading is a feasible solution to enhance device efficiency with the widely-used introduction of smart vehicles. In other words, vehicles might share their computing power with the network to assist another vehicles or end users in neighboring computer nodes. Note that it would be challenging to find an acceptable assignment strategy for task offloading due to the growing complexity of fog networks. However, a significant literature explores different evaluation models for the elimination of tasks and consequently enhances the methods performance. A relevant study in (Fuqian et al., 2018) proposed a task offloading techniques from the user infrastructure to nearby fog nodes within fog-enabled networks by using a heuristic-based dynamic allocation index. Similarly, two-tier federated process for vehicular networks like ones in (Lin et al., 2019; Yinuo et al., 2019) uses system application to estimate completing deadlines along with the total cost of offloading in terms of energy and delay. The aim is to reduce accumulated latency for the offloading of tasks to different devices in a resource limitation environment. However, the majority of these other models neglect traditional problems with the network and mobility. More recent models like the one in (Mostafa, 2019) proposed a resource utilization facility based on run-time estimation for the fog environment. Similar works use adaptive learning, MAB, and probabilistic methods like ant colony optimization to minimize the average offloading delay. Another category of techniques uses pricing models. For instance, in (Nguyen et al., 2019; Wu et al., 2019) resource-based price models are connected to spending and thus have better means for accessing and effectively allocating dispersed services.

Each of the fog unit has a multi-core device with an input queue, output queue, workload manager, federation manager, decision manager, and communication models, as illustrated in Fig. 2.4. The workload Manager (WM) tracks, processes and returns after performance of the tasks. It receives a problem, puts it in the input queue and sends its results to the source vehicle from the output queue. The Federation Manager (FM) is responsible for sharing and retaining information about the current workload with other fog groups. The FM also gathers all the information relating to its local registries on a daily basis. The Decision Maker (DM) chooses the outsourcing node from the existing state of all fog devices. The criterion of selection can differ depending on the DM algorithm. In order to balance the workload equally between all fog units we are recommending a price-based distribution algorithm. Finally, the communications module establishes a link in the contact area with other fog units and vehicles.

2.4.2 Offloading Task in Cloud Federation

The cloud and edge computing federation approaches are used to increase benefit and minimize the use of resources available. The term federation applies generally to the incorporation of self-employed individuals into a collective arrangement that provides an atmosphere of cooperation to optimize benefit. In (Hammoud et al., 2020), the authors intend to use evolutionary game theory as a very profitable cloud formation model. The

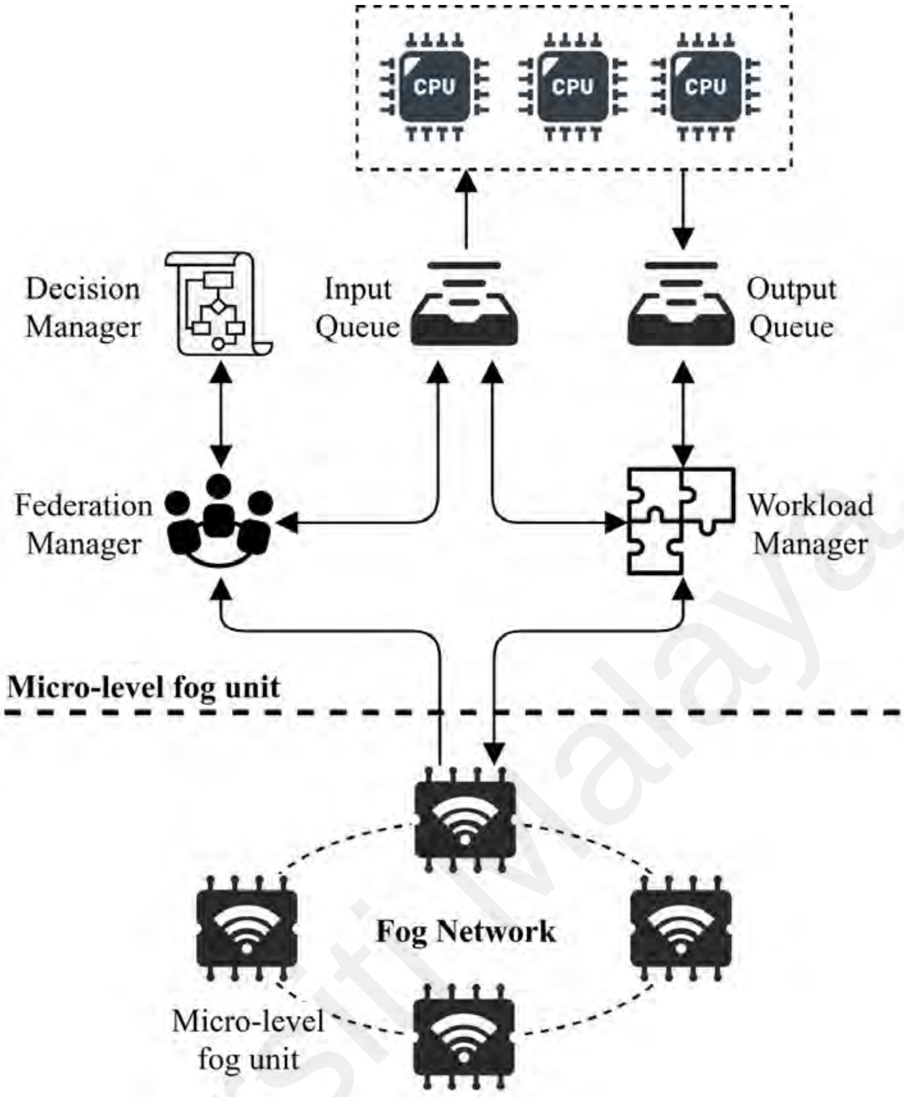

Figure 2.4: Fog units internal architecture and connectivity

paradigm ensures the equilibrium between its participants to ensure that their services are distributed to others, so that benefit can be maximized. The test reveals that the model functions well with respect to benefit and QoS compared with standard genetic algorithms. Similarly, in (Moghaddam et al., 2019), the cloud federation's hedonic coalition forming algorithm seeks to reduce resource usage and increase overall benefit. A different approach incorporates a Cloud-Fog Infrastructure Load Transfer Preparation Strategy to minimize time and network consumption. (Ottenwälder et al., 2013). However, the task does not require the offloading of fog users.

In a hybrid cloud-fog architecture, (Lee et al., 2019) researched the topic of fog network creation and task delivery. They are differentiated from other experiments in the propagation of fog nodes, taking the dynamic construction of a fog network into consideration. Since the coordinates of the closest fog node are unknown, the writers use an online solution to easily extract information from the fog network and thereby reduce device latency. Your online algorithm enables a certain fog node to track the uncertain world and to decide how computational activities are to be offloaded. A recent study (Tang & He, 2018) explores the download of computers in an unpredictable wireless world is the task. oThe researchers analyze the computational offloading because the activity of smartphone users is arbitrary. In this non-cooperative game, users battle for scarce communications tools.

Due to the accessibility to vehicles, edge servers will reduce connectivity costs and produce a high efficiency in offloading services. Within Given the rapid reaction time, edge servers generally face resource limitations relative to traditional, heavily computerized, cloud-based servers. It takes time for the edge servers to execute the tasks computation. This is particularly true for road segment edge servers that have a high vehicle density relative to other segments. In (Ke et al., 2017), the study was carried out of an computational offloading infrastructure which enhances the system throughput of the V2I and V2V transmission frameworks. In addition, an effective predictive mode reduction method was also introduced, taking into consideration of time consumption for the success of the tasks and vehicle mobility. In this model, the tasks are downloaded through direct upload and predictive relay transmissions to the MEC servers.

Saqib et al. tested a stable computer offload model called FogR (Saqib & Hamid, 2016). It believed that the fog networks could be more reliable in intelligent traffic systems in emergencies. If a certain edge node fails, it would then communicate with every other edge/fog node nearby. In case of an emergency, if a car tries to reach its destination via traffic, it can do so easily via smart traffic systems. In order to provide user-friendly experience, Y. Bi et al. suggested a cross-layer and neighboring vehicle that helped to ensure they are linked online while running along metropolitan roads (Bi, 2018) . The vehicle will take assistance from its adjacent vehicles to get the qualified vehicle, to determine the purpose of AP and to collect the necessary information before joining the target AP coverage. The vehicle will obtain the IP address by selecting the target AP with this support from its neighboring vehicle.

2.5 Task Scheduling

Due to the increased complexity of the fog networks, it is difficult to find the best fog or cloud tools for task execution. A resource selection service based on a run-time predictions within a fog environments developed by (Mostafa, 2019). Sometimes cloud computing has been challenging to scalable to infrastructure of the system. A low cost solution techniques has been proposed (Larcher et al., 2019) in Fog-Computing paradigm. The researchers (Zhangand He et al., 2019) suggest a completely polynomial-time-frame estimation scheme to minimize cumulative latency when offloading based tasks to various devices within resource cost constraints. It is believed that the machines have unlimited processing power to continuously perform any number of tasks, which is impractical. In (Habak et al., 2015), authors suggest a similar offloading method for a mobile device cluster and layout standard task scheduling statistical models to optimize the cumulative effective estimate. In practice, most contemporary pricing scenarios record several pricing origins in research. (Wu et al., 2019) provides a systematic review of cloud pricing in an interdisciplinary approach, where they mapped a total of 60 pricing models under 9 cloud pricing categories. Most researchers in pricing discuss value-based pricing and cost-based pricing is still popular because it can help decision-makers set a floor to charge customers for the minimum price to cover capital expenditure at least. Resource-based pricing have some specific properties that are correlated with the expenditure components and the goal of resource-based pricing is to provide a better method for consumers to access and allocate the distributed available resources and efficiently. That's why in (Wu et al., 2019), authors categories resource based pricing in cost-based pricing strategies. When the infinite scale of resources is needed without a prerequisite condition then the utility based pricing can help every individual to access the cloud service directly.

For resource sharing and infraction to schedule network services (NSes) in virtual networks an optimization model is proposed (Zhangand He et al., 2019) which allows the process in order of NSes in run-time and the process duration of each function of an NS is allowed to be discrete.

2.6 Multi-Access Edge Computing (MEC)

Mobile cloud infrastructure is a cloud branch to mobile computing. Similarly, multi-access edge (MEC) computing represents an enhancement to mobile computing. ETSI describe MEC as a platform that allows 4G and 5G, IT and cloud networking capability in close proximity to mobile subscribers via the radio access network (RAN) (Giust et al., 2018). The paradigm has been developed to include an extended range of applications outside of technology activities, however, multiaccess processing is previously known as "mobile edge computing". Examples of edge computing technologies for multi-access are video analytics, connected cars, tracking of wellbeing and increased reality.

MEC expands cutting-edge computing by supplying low-energy electronic computers with processing and storage tools. MEC enables the additional edge device capability of the RAN operators to existing base stations. Tiny data centers with virtualization capabilities can also be found in MEC in the same manner as edge computing. The usable computational resources are modest relative to cloud computing because of the underlying hardware in the MEC and edge computing. In addition, MEC can support low-latency applications. MEC apps are able to benefit from real-time radio and network information and can also give smartphone users a customized and contextual experience.

The edge of the network runs both the edge of edge computing and MEC infrastructure and can work with minimal to no Internet connection. However, MEC allows WAN, WiFi and cellular networks networking, while Edge Machine can normally have some connectivity (e.g., LAN, WiFi, cellular). The research in MCC focuses on the link between cloud service users (on mobile devices) and cloud service providers, while research in MEC is focused on (RAN) network infrastructure providers. In comparison, MEC largely differs from the MCC in its operation. The new 5G network is supposed to gain substantially from MEC (Hu et al., 2015). Likewise, 5G is called a MEC-enabled technology because it provides lower latency and higher capacity across mobile devices and supports a wider range of finer, granular mobile devices. MEC provides cut-and-drop computing for a wide range of latency and more powerful mobile core network mobile applications (Taleb et al., 2017). MEC also facilitates mobile network framework that is mission-critically vulnerable to delays (Hu et al., 2015).

2.7 Laxity Based Computing

Apart from the heterogeneity of compute resource requirements, taking into consideration task priority is important. Often this involves the use of a multi-level queuing model is adopted to handle the task priority. For instance, in (Li et al., 2019), the concept is used to improve task processing at the fog computing nodes. Similarly in (Nan et al., 2018), a framework is proposed to minimize processing delay and energy cost based on Lyapunov optimization. Furthermore, effective scheduling of offloaded tasks is critical for efficient utilization of computing resources. Undoubtedly, task deadline is a key parameter when making an offloading decision (Mukherjee et al., 2019). In (Karmakar et al., 2020), a novel task scheduling algorithm for dynamic workflows is designed to reduce costs of using cloud data centers. Furthermore, in (Enokido & Takizawa, 2020), laxity time is used as task priority for task scheduling to improve the performance of real-time tasks inside data centers.

As mentioned earlier, IoT devices are resource constraint, with compute alternatives available at fog nodes, MECs, and cloud data center. Therefore, to meet task completion deadlines, a task source needs to select an appropriate network for timely task execution. Considering the significance of this decision, there are few studies concerning offloading decision models. In (Cao et al., 2009), priority-based offloading is introduced to achieve high throughput of cloud servers. Similarly, in (Goudarzi et al., 2016), a priority-based computation offloading scheme based on the branch-and-bound algorithm is proposed. Another relevant study for IoT device in (Mithun et al., 2019) uses a delay-dependent priority-aware offloading (DPTO) strategy is proposed for task scheduling, to minimize the overall service delay while meeting task deadlines.

Other studies attempt to find a trade-off between energy consumption and execution time with minimal cost, for instance, in (Huang et al., 2012), a dynamic offloading algorithm is proposed to improve overall energy consumption and wait time. Similarly, to reduce service delay and energy consumption at MEC, a novel priority-based task caching offloading policy is proposed in (Nur et al., 2019). Here, task priority is based on parameters including requested tasks, completion deadline, data size, and required computing resources. Undoubtedly, such priority is critical to improving resource allocation because resource demand, cost and availability often are imbalanced, with high service expectations from the end users.

2.8 Workload balancing

The offloading of tasks between fog nodes can involve some risks for safety and privacy. The danger is to offloaded activities of details that are essential to protection and privacy. Even there may be a vulnerability risk if a fog node is overwhelmed and begins to offloading data critical to protection and privacy information to another fog nodes, for example via malicious user requests (that can be reached by the customer). A guidance for study, therefore, is the design and implementation of safe loading and balance schemes. An early study (Lai et al., 2018) to protect privacy is included in the offloading of MEC. In addition, a light and powerful system for IoT receivers is programmed to check the precision and fairness of the tasks being loaded.

The authors in (Beraldi et al., 2017)suggest a co-operative policy for load balance between two edge data centers. The model is based on the simplest rule: when a service request comes to a data center when its buffer is full, the request is discharged and served by the data center to the other cooperating data center. The analysis in (Fricker et al., 2016), on the other hand, analyzes a offloading strategy between several data centers on a ring topology situated at the edge of the network. The study also predicts and calculates the benefit gained in ring topology by co-operation between adjacent fog data centers.

Vehicle networks guarantee efficient connectivity in order to enhance the spread of vehicle results. Many vehicles carry out the distribution of results, contributing to a rise in load. The new scheduling algorithms have been developed so that the different challenges of the queue length can be adapted. The traditional shortest queue policy is one such algorithm. Time-based scheduling does not mean a minimum of time, so it is more effective and dependable. Chen et al. previously suggested a scheduling based on two dynamics: reaction time and queue length (Chen & Wang, 2017). This allows the vehicle communications to have a distinct contact environment. They has devised a vehicle-cloud three-layer architecture that is derived from edge/fog computing. The architecture is based on the PEPA system of composition (Srivastava et al., 2020). Because of its computationally and abstraction properties, PEPA assists in modelling large-scale structures.

In (Lai et al., 2018), a scheduling system for distributed public vehicles (PVs) that combines sophisticated computing technology and vehicle sensing is proposed. This model incorporates the metadata storage elements, expense approximations, responses to requests and preparation for PV systems. Edge nodes collect metadata and preserve them, and they serve as intermediaries. The information is then obtained via vehicle networks and retrieved. The PV system supports a heuristic algorithm for integration and a cooperative approach for transmission of requests between vehicles nodes, edge nodes and cloud as well as for scheduling routes for PV.

A conventional fog-federation consisting of a variety of fog-units that operate with QoS and to manage workload. Each device can connect through its limited set of terminals. They are also connected with a high-speed cable. In this work, we use the idea of microfog deployment where linked fog units are mounted at crossroads forming a fog federation. The principal aim is to distribute workload efficiently in order to defend the federation from overwhelmed fog units.

2.9 Summary of related work

Most of the recent works consider task offloading either vertically or horizontally. Therefore, the focus is on the efficient utilization of computing resources available at the next layer. However, there exist few studies where task priority is also considered while provisioning these resources. In imbalanced workload scenarios, frequent offloading leads to overloaded computing resources, resulting in lower system performance. In the study, we propose a framework that scales vertically and horizontally while fulfilling task completion deadlines. Moreover, the task model used is prioritized, and hence we introduce laxity-aware scheduling to avoid starvation. The comparison of the proposed work with recent contributions is listed in Table 2.1.

Table 2.1: Summary of the Literature Review

Authors (year)	Compute Model	Strategy	Simulation Tool	Queue Design	Collaborative Computing
(Madan et al., 2020)	V2X	On-demand	Anylogic	Single	RSU and MEC
(Rahman et al., 2020)	V2V	Workload Distribution	Anylogic	Single	NA
(Sharmin et al., 2020)	Mirco-level fog units	Queuing-based pricing	Anylogic	Single	Fog and Cloud
(Enokido et al., 2020)	Edge Node	Laxity-based	Not Specified	Single	NA
(Talaat et al., 2020)	Fog Nodes	Resource Optimization	MATLAB	Single	Fog and Cloud
(Li et al., 2019)	Fog Nodes	Resource Optimization	CloudSim	Multi-level	NA
(Nur et al., 2019)	Mobile Device	Priority-based	Cloudsim	Not Specified	NA
(Mukherjee et al., 2019)	Fog Nodes	Priority-based	Not Specified	Single	Fog and Cloud
(Adhikari et al., 2019)	Fog Nodes	Priority-based	Not Specified	Multi-level	Fog and Cloud
(Wang et al., 2019)	Edge device	Energy Consumption	Not Specified	Single	NA
(Mukherjeem et al.,2019)	End-Users	Resource Optimization	Not Specified	Single	Fog and Cloud
(Khattak et al., 2019)	Fog Nodes	Resource Efficiency	IFogSim	Single	Fog and Cloud
(H. Chen et al., 2018)	NA	Workload Distributions	Discrete Simu.	Single	NA
(Zhang et al., 2018)	Mobile device	Energy Optimization	Not Specified	Multi-level	NA
(Xiaolong et al., 2018)	Fog Nodes	Resource Efficiency	CloudSim	Single	Fog and Cloud
(Liu et al., 2017)	Fog Nodes	Energy Optimization	Not Specified	Multi-level	Fog and Cloud
(Goudarzi et al., 2016)	Mobile Device	Branch bound/ILP	MATLAB	Single	NA
Proposed Model	Fog nodes	Laxity-based Multi-stage	Anylogic	Multi-level	Fog consortium

CHAPTER 3: METHODOLOGY

3.1 Introduction

Fog nodes with limited compute capacity are unable to guarantee that all tasks will be executed simultaneously but they get buffered in a waiting task queue. Each end device in the fog computing system generates computational tasks with random computational requirement configuration, that is tasks with input/output data sizes in KBs with computational requirements in Mbits.

Consider a resource set comprising E end devices, and F fog nodes. Each resource supports on board compute capability and wireless connectivity. Note that the interconnection among the fog nodes is wired.

3.2 Task Model

We assume that the end devices randomly and uniformly generate computation-intensive tasks. We further assume that the total CPU cycles required to process the these tasks are different. Without the loss of generality, we categorize the tasks into three priority types: hard, firm, and soft, based on descending task priorities. Each task i with data size s_i, compute resources requirement c_i, and priority ϕ_i has a task completion deadline d_i given as,

$$d_i = t_i \; \xi_i \tag{3.1}$$

Here, t_i represents the application that dictates the deadline, the time interval to return the result. If missing this deadline leads to a critical situation, the deadline is hard. If the result has utility even after the deadline has passed, the deadline is classified as soft, otherwise it is firm. Thus, adding the offset models the upper bound on the given task of certain priority such that $\xi_i \in \xi_h, \xi_f, \xi_s$ and $\xi_m < \xi_f < \xi_s$.

3.2.1 Tasks Execution Granularity model

The time model computes the computation time when servicing a task, either locally at the end device or offloaded to fog nodes. Consider task i with data size s_i and required compute cycles c_i then local computation time t_i at end device j is defined as,

$$t_i = \frac{1}{\nu_i}\left(c_i \;\; {}_{x \in Q_j} c_x\right) \tag{3.2}$$

where ${}_{x \in Q_j} c_x$ is the compute resources requirement of waiting tasks in local queue Q_j, and ν_j is compute capacity of the end device. Similarly, the computation time t_i for task i when offloaded from end device j to fog node k is computed as (Yin et al., 2019),

$$t_i = \frac{s_i}{C_{jk}} \frac{1}{\nu_k}\left(c_i \;\; {}_{x \in Q_k} c_x\right) \tag{3.3}$$

where ${}_{x \in Q_k} c_x$ is the compute task requirement of waiting tasks at fog node queue Q_k, C_{jk} is the data transmission rate between the end device j and fog node k, and ν_k is compute capacity of the fog node.

3.3 Communication model

As mentioned earlier, the data transmission between the fog nodes (F) and end devices (E) is wireless, and among fog nodes is wired. The communication model calculates the channel capacity C of the data transmission as,

$$C = \begin{cases} X_{i \leftrightarrow j} & \text{if } i \in E \text{ ,} j \in F \text{ (wireless)} \\ Y_{i \leftrightarrow j} & \text{if } i, j \in F \text{ (wired)} \end{cases} \tag{3.4}$$

where X and Y are the symmetric data transmissions rates between fog node and end device, and among fog nodes, respectively. For simplicity, Y is robust set to 15 Mbps whereas X is based on Shannon theory (Jurgens & Crutchfield, 2021) given as, $W_{mmWave} log_2 1\ SNR$ where W_{mmWave} is the millimeter wave bandwidth and SNR is the signal-to-noise ratio. Here, the SNR is computed as $\frac{P_{tx}h}{N_0 W_{mmWave}}$ using transmission power P_{tx}, antenna channel gain h, and power spectral density of added white gaussian noise (AWGN) (Sapkal & Kulkarni, 2018). Considering X is subject to path loss $l = \alpha\ \beta 10 log_{10} d\ \varepsilon$ with initial offset α, path attenuation index β and shadow-fading effect ε, the channel gains h of LOS and NLOS (Jiang et al., 2021) for communicating devices at Euclidean distance (Siyuan et al., 2021) d_E are $10^{-l_{LOS} d_E}$ and $10^{-l_{NLOS} d_E}$, respectively. Thus, the single-hop communication delay for task i is $\frac{s_i}{C}$ where s_i is the task data size.

3.4 Queuing Model

Based on the heterogeneous task model, we used a multilevel virtual queuing model at the fog node. At the first level, there were multi queues with predefined levels corresponding to the supported task types, that is the arriving tasks with the same type get placed in the same virtual queue. At simulation start, the fog node is idle with empty queues, capable of handling incoming tasks in parallel. For the study, we considered an M/M/s/FCFS/∞ queue (Dhar et al., 2020) with the supported task types arriving at the fog node with high-priority tasks having non-preemptive priority against low-priority tasks, that is a high-priority task gets serviced ahead of only waiting low-priority tasks, not any tasks already being serviced. Here, the sequence of arriving tasks at the fog node is modeled as independent Poisson processes, with mean task arrival rates λ_1 and λ_2 with execution times exponentially distributed. Similarly, the average service time is exponentially distributed, $\frac{1}{\mu}$. The multi-server model comprises s identical compute units available at the fog node, exhibits state probability defined as $p = \frac{\lambda_i}{s\mu}$. The total wait time in the queues is calculated as $\frac{s^2}{\mu 1-p_i}$. Note that when there are enough high-priority tasks to occupy all available fog nodes, the number of low-priority tasks waiting in the queue increases by λ_2. At the next level, a ready task queue is kept populated by moving tasks from the first-level queue. Here, we use a kind of aging technique where tasks approaching their completion deadlines are moved to control task starvation and increase the throughput of the fog computing model.

3.5 Tiered Computational Model

We considered a two-tier fog computing model comprising the end tier and the fog tier. The end tier is composed of end devices E acting as the source of input data with corresponding compute tasks. Depending on the data size and complexity of the compute tasks, the tasks can be classified as either compute- or communication-intensive.

Here, we assumed that each end device has sufficient computing power for handling hard tasks locally, avoiding the uncertainty of the data-communication channel and its associated data-transfer delays. However, the application task generation rate constraints the local resources. Thus, the local computation are mostly used for the earliest-deadline-first tasks.

The fog tier is composed of fog nodes F placed at the edge of the vehicular network, providing services with significantly smaller communication latency compared to when communicating with the cloud. The fog nodes are deployed at intersections, directly connected through a wired link. However, the communication between fog units and end devices is via a wireless link. Furthermore, geographically nearby fog nodes self-organize acting collectively to form connected fog locations, individually referred to as federates. At the next level, the federates when connected form a fog federation. Such an organization enables opportunities to offloaded tasks from the end tier with different resource requirements. In general, the task processing in the tiered computational model involves two components working in parallel: the first component performs incoming task buffering at the fog node. This is done using a classical task queue, with tasks executed according to their arrivals in the queue. Once a task is scheduled for execution, the second component provisions fog resources to the task.

3.6 Decision Model

The offloading decision at any end device is based on the state of its available resources and task completion time. Recall that there are three types of tasks with different priorities, and the end device processor is capable of executing either of them. Since the hard tasks do not tolerate delays, they are always executed locally on source device whereas firm and soft tasks can be executed on local device as well as offloaded for execution on fog processors. Given that an end device is linked to strictly one fog node in its data-transmission range then the offloading decision model δ_i can be stated as,

$$\delta_i = \begin{cases} 1 & t_i < d_i \text{ and } \phi_i \neq Hard \\ 0 & otherwise \end{cases} \tag{3.5}$$

where false (0) represents task execution on local end device and true (1) defines an extended task computation space $F = \{local_f, fog_i\}$. As mentioned earlier, hard tasks are delay-sensitive and must meet their completion deadline. Therefore, they are executed on local end device to avoid task failures due to offloading delay, caused due to uplink/downlink times, channel fading, communication noise, and fog service delay. In contrast, offloading firm and soft tasks help achieve better execution services for hard tasks, in turn improving the throughput of the fog computing model.

3.7 Collaborative task offloading model

Traditionally, fog paradigm is used to provide additional computational resources to end devices, in turn alleviating the service delay. The existing paradigm can be further improved by extending it to allow workload sharing across the fog layer. Therefore, to effectively use the available computational resources, the collaborative task offloading model can be formulated as,

$$\textbf{P1:}\ \text{minimize}\ {}_{i \in T} t_i \tag{3.6}$$

$$\begin{aligned} \text{subject to }\ \textbf{C1:}&\ {}_{k} w_k \leqslant W, \forall k \in F \\ \textbf{C2:}&\ L_{jk} = 1, \forall j \in E, \forall k \in F \\ \textbf{C3:}&\ t_i \leqslant d_i, \forall i \in T \end{aligned} \tag{3.7}$$

In 3.7, **C1** is a workload constraint where the assigned workload w_k to fog nodes $k \in F$ should not exceed their total capacity W. **C2** places a constraint on the number of connections L_{jk} an end device $j \in E$ makes with a fog node $k \in F$, which is only one. **C3** limits the end-to-end delay for each task in the task set T.

3.8 Summary

Based on the types of task, the queue are divided into different groups. The queue have the fixed priority and it must be scheduled and general. There are various algorithm for queuing model. The overhead energy consumption estimate is defined for local and offloaded computing based on a time consumption model. Task are assigned to the processes based on there priority given by the users. In this research, we fixed priority rank for the each process and the lower priority get the interrupt by the higher priority. Each task is assigned with the laxity and the minimum laxity process first. If a task is allocated the resources or the task is offloaded to the cloud center, it leaves the queue and executed at the fog server or cloud center. A multi-core unit with milti-level input queue, output queue, workload manager, federation manager, decision management and communication module in each fog unit.

CHAPTER 4: PROPOSED MODEL

4.1 Introduction

The proposed offloading model is a multi-stage decision process, in the first stage is algorithm, the end device offloads a task to an in-range fog node. This may be followed by another stage where the task is offloaded to any underutilized fog node within the fog federation. The main objective of the model is to schedule tasks efficiently so that tasks of all types can meet their predefined deadlines, as stated earlier to improve the overall service quality of the fog computing model.

4.2 Priority-based Collaborative Fog Computing model

The proposed offloading model is a multi-stage decision process, in the first stage (Algorithm 3), the end device offloads a task to an in-range fog node. This may be followed by another stage where the task is offloaded to any underutilized fog node within the fog federation. The main objective of the model is to schedule tasks efficiently so that tasks of all types can meet their predefined deadlines, as stated earlier to improve the overall service quality of the fog computing system.

Algorithm 1 Multi-stage collaborative task offloading model as a decider

Input v: vehicle; t: task; Q: local task queue
Output status message

```
while true do
    t ← Generate(v)                          ▷ vehicle generates task
    if t IS HARD or v NOT CONNECTED then
        Enqueue(Q,t)                         ▷ add to local queue
    else                                     ▷ v has fog node in range using eq. 3.5
        f ← Fog(v)                           ▷ get fog node
        Send(v,f,t)                          ▷ send task to fog node
    end if
    if (r ← Recv(v)) ≠ ϕ then                ▷ receive task output
        Send(v,r)                            ▷ forward results to respective vehicle
    end if
end while
```

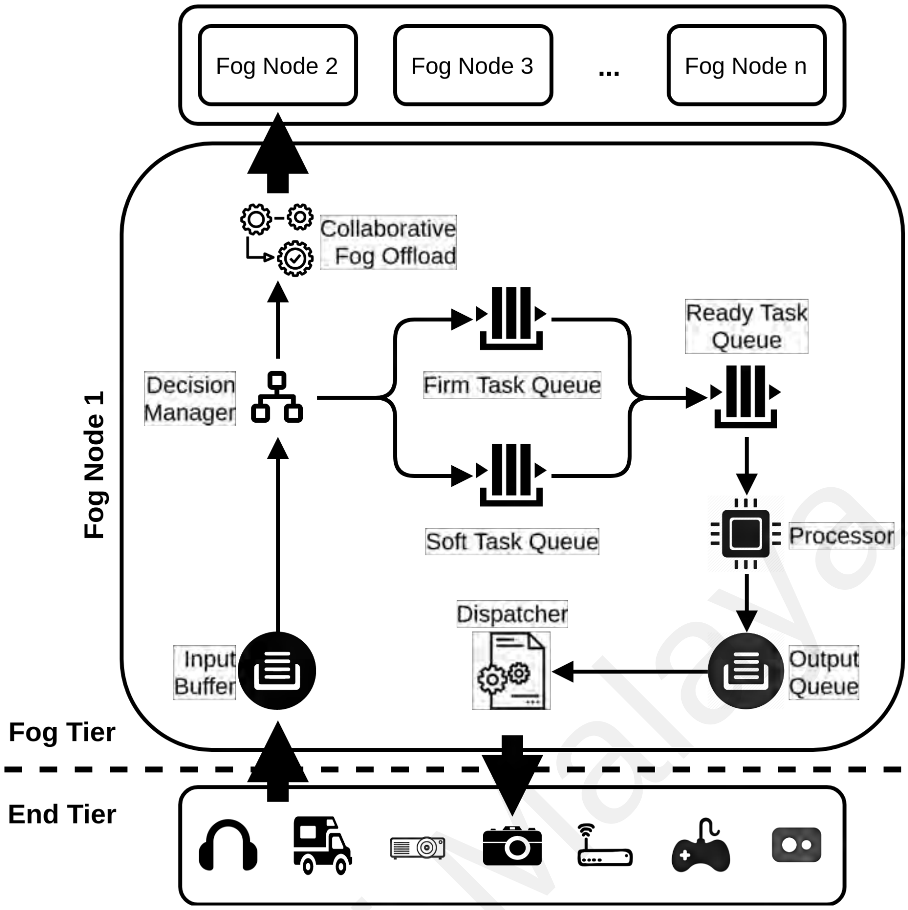

Figure 4.1: Priority-based queuing model for collaborative fog computing.

4.3 Task processing workflow

Fig. 4.1 illustrates the proposed priority-based collaborative fog computing model. At first, the arriving tasks are buffered in an input queue. We assumed that these tasks are processed in a first-come-first-serve (FCFS) manner by a decision manager (Adhikari et al., 2019). At the time of task arrival, the decision manager decides whether the tasks are scheduled for execution locally at the fog node or offloaded to another fog node within the fog federation. On the contrary, this decision can be taken when the task is scheduled for execution on the fog node, thus, can add the queuing delay at multiple fog nodes; thus, affecting the task completion rate.

4.4 Online Heterogeneous Task Scheduling

When the tasks are scheduled to be locally executed at the fog node, these tasks are grouped in separate queues based on their priority. In this study, we used an aging technique where the tasks approaching their completion deadlines are placed to the ready/execution queue. We defined laxity as the amount of remaining time after task completion if the task is scheduled for execution at the current/present time instant. To explain, the laxity time l_i for a task i is the difference between the deadline d_i and task computation time t_i, and is expressed as,

$$l_i = d_i - t_i \,. \tag{4.1}$$

Here, the task execution priority of the tasks at the fog node increases with decreasing laxity. In this study, we use the same task scheduling strategy for both medium and soft tasks queues, irrespective of the fact that the multi-queue has the advantage in independent task scheduling strategies for the heterogeneous tasks. Nonetheless, the goal is to control the starvation of low-priority tasks and to increase the overall task completion rate of the proposed model. The scheduled tasks eventually end up in a ready task queue for execution locally at the fog node. We proposed an online priority scheduling algorithm based on the least-laxity-first (LLF) scheduling strategy (Weihong et al., 2021). Upon the arrival of a task from the type-grouped task queues into the ready queue, a scheduling event is triggered. The event recomputes laxities of all queued tasks and the arriving task since it changes the laxity time at every time instant. A task with the least laxity is considered the highest priority task, and subsequently gets scheduled for execution on the fog node.

4.5 Online Resource Allocation and Task Offloading Policy

The decision model for offloading a task for a vehicle is based on the current state of its computer unit. In other words, if this is the case the task is offloaded into the nearby fog unit, the number of pending tasks in the task queue and if the car is in the communication area of any fog-units. Remember that in real life situations, the current workload on the fog machine is unclear to the car. Moreover, processing systems at fog sites have a greater computing power than vehicles and enable the discharge of delay related activities. The workload manager (WM) monitors and reports on the tasks that have been performed, stored and transferred. It receives a mission and sends the reports to the destination vehicle from the feedback queue. The manager of the federation (FM) shall exchange and retain data about existing workloads with other fog units. In order to access the regional registers, the FM regularly gathers all relevant information. The Decision Manager (DM) selects the outsourcing node for the current status of all fog units.

4.6 Collaborative Fog Computing Model

The computing resources at fog nodes are limited, they may fail to ensure service quality to all incoming requests, in particular, it becomes challenging for overloaded fog nodes. To cater to this, we propose a collaborative fog computing environment where tasks are outsourced to underutilized fog node, improving the throughput of the fog computing model. The decision at the fog node is again a multi-stage model, as listed in Algorithm 2. The first stage is referred to as the decider whereas the second stage as the selector. The decider as discussed earlier in the task processing workflow. It makes the offloading decision whether to offload the task to another fog node or execute it locally. This stage is followed by the selector that selects a suitable fog node for offloading.

Algorithm 2 Multi-stage collaborative task offloading model as a selector

Input $self$: local fog node
Q: local task queue; Q_1: firm task queue; Q_2: soft task queue
Output status message

```
while true do
    t ← Recv()                                  ▷ receive task
    if t.type IS REQUEST then
        δ ← Propose(self)                       ▷ assess proposal using eq. 4.2
        if δ then                               ▷ proposal to offload accepted
            k ← Min(F)                          ▷ get least loaded fog node
            Send(t,self,k,'REQUEST')            ▷ outsource task to k
        else
            if t IS FIRM then
                Q1.enqueue(t)
                Q1.LaxSort()
            else
                Q2.enqueue(t)
                Q2.LaxSort()
            end if
            i ← Q1.peek()
            j ← Q2.peek()
            if Lax(i) ≤ Lax(j) then             ▷ compute laxity using eq. 4.1
                Q.enqueue(Q1.dequeue())
            else
                Q.enqueue(Q2.dequeue())
            end if
            Q.LaxSort()
            if self HAS FREE_CORES then
                Schedule(Q.dequeue())
                r ← Recv(self)
                Send(r,self,t.src,'OUTPUT')
            end if
        end if
    else                                        ▷ t is result of outsource task
        Send(t,self,t.src,'OUTPUT')
    end if
end while
```

4.7 Offloading Decision

In the first stage, for some task i, the local fog node j proposes to offload it to another fog node. To do this, the decider classifies fog node j as either overloaded or underloaded based on a workload decision variable x_j. The proposal decision model δ_i is stated as,

$$\delta_i = \begin{cases} 1 & x_j > 0.5 \\ 0 & \text{otherwise}\,, \end{cases} \tag{4.2}$$

where, true (1) represents the acceptance followed by the selection of suitable fog node when local fog node is more than its half then local offloading decision happened, false

(0) represents the rejection of the proposal to offload task from the local fog node. Given that when the workloads are different, the workload decision variable x_j is defined as the workload at the local fog node j after min-max feature scaling among all fog nodes $\forall k \in F$.

4.7.1 Fog node selection for offloading

Considering that the fog node is eligible to offload, the selector initiates the selection of suitable fog node for task offloading. Here, the selector uses a balls-into-bins game for the selection of suitable fog node, a well-known process model for task distribution among a group of servers (Mitzenmacher & Upfal, 2017). Suppose we sequentially assign p balls into q bins. Initially, the balls are thrown into a randomly selected bin, later on, workload w is used to select the next bin. In this study, we modify the classical model to work in a collaborative fog environment. We skip the use of random distribution for initial binning since every fog node gets task requests from its geographically local end devices; therefore, the workload at every bin is known at the time of binning decision. That is, a ball gets allocated to the least loaded bin, alleviating maximum load. Here, the maximum load of W is the largest number of balls in any bin $f \in F$. Assuming that $p > q$, the expectation of maximum load W is given as,

$$\mathbb{E}W = \frac{p}{q} \frac{\text{loglog } q}{\log\ n}, \tag{4.3}$$

where n is the number of available fog nodes. In summary, the decider implements a strategy to avoid frequent offloads, reducing network congestion. Overall, this adds fairness and balance to workload distribution in the collaborative fog environment. Note that there is no task migration or task handover in the proposed model. However, vehicle handover happens with vehicle mobility. The proposed model uses the federated fog model to schedule the given task. Moreover, upon task execution, the results are returned back to the source fog location for onwards delivery to the source vehicle using a multi-hop vehicle-to-vehicle communication model.

4.8 Time complexity

It is evident from Algorithm 2 that the only time consuming steps are the ordering of the tasks by laxity times followed by selection of the least overloaded fog node for offloading. At every iteration, a fog node checks its priority-based waiting queues for pending tasks, if any then de-queues and en-queues one to the task queue. Here, we considered, a typical task queue implementation involving operations of time complexity O(1) time. This is followed by reordering of the queue based on laxity times. Given n pending tasks, the re-computation of laxity times requires O(n) steps, followed by ordering of the task queue based on these times involves a typical sorting algorithm of time complexity O($nlogn$). The task with the least laxity is selected for offloading to a fog node with least workload. Let m be the number of fog nodes then finding least overloaded one takes O($mlogm$) time. Therefore, the total time complexity of the algorithm is O($nlogn$) assuming that $m < n$.

4.9 Tools and software Used

To benchmark the aforementioned algorithms using a vehicular simulation, we used AnyLogic[1], an agent-based simulation platform with an extensive road network traffic library. Anylogic is a java IDE. Low-level modeling constructs (variables, expressions, parameters, processes, and so on), presentation structures (lines, edges, ovals, and so on), analytical techniques (datasets, histograms, plots), connection tools, standard pictures, and experiment models are all included. For ruining the simulation, Java software installed into the experimental device. It can support both Linux and Windows OS.

[1] https://www.anylogic.com/

4.10 Summary

The resource capacity of fog node must be less than the amount of allocated resources. These limited resources at the fog node can't complete all the tasks in its queue within its deadline as the real-time task often need deadline requirement for their execution. By considering the urgency of real-time task, the laxity time at the fog tier and cloud tier can be used to the task to deleted from or leave the waiting queue from the fog node. In this research, we proposed three policy for multi-level queuing. To reduce the time complexity we explore the parallel virtual queuing model which buffers at the arrival task in the same fog node into a separate virtual queue.

CHAPTER 5: EVALUATION AND RESULTS

5.1 Introduction

For evaluation, the performance metrics is used to calculate the proposed workload distribution of the task offloading techniques are considered as: queuing time (Wallace, 2021), end to end delay (Hameed et al., 2021), the rate of offload and a deviation in workload (Abbasi et al., 2021). We tested with two alternative of the proposed strategy, federated ($\mathscr{F}$) and non-federated ($\mathscr{F}'$). In this chapter, we are going to describe about the simulation setup and scenario and discuss about the result.

5.2 Simulation Setup

To demonstrate the effectiveness of the proposed collaborative fog computing model for priority tasks (denoted as $\mathscr{F}$), we correlate it against a trivial non-federated fog environment (denoted as $\mathscr{F}'$). Furthermore, to compare our performance with collaborative scenario, we consider the random walk algorithm (RWA) (Zhu et al., 2017) and nearest fog algorithm (NFA) (Bozorgchenani et al., 2017). Here, Manhattan road network topology is used for vehicular movement. Here, all roads are bi-directional with nine intersections and eight entry and exit points. Each intersection comprises a fog node with heterogeneous processing capacity up to eight cores. All fog nodes are directly connected to one another via a wired network, in contrast, the vehicles communicate with nearby fog nodes wirelessly.

The vehicle arrival rate λ defines the number of vehicles arriving in the simulation per hour. The vehicles collect random direction until they exit the simulation over an exit point. To simulate uneven workload at different intersections, we categorized the entry/exit points into three groups (S_1, S_2 and S_3) with varying arrival rates. The motivation is to represent realistic urban areas where fog nodes are overloaded during peak hours. However, to provide a consistent service quality, we used a collaborative fog computing model to alleviate overloaded fog nodes by offloading tasks to under loaded fog nodes. The simulation parameters used for the experimental evaluation are listed in Table 5.1.

Table 5.1: Simulation configuration and parameters.

Parameter	**Value**
Simulation area	3 km^2
Total simulation time	1 hr
Simulation repetition	5 (five) times
Vehicle speed	10–60 km/h
Vehicle acceleration/deceleration	1.6/2.6 ms^2
Vehicle compute capacity	50 MHz
Compute request size	15–50 Mbits
Task generation interval	Random
Vehicle mobility	Random
# of fog units	9 (nine)
fog-unit range	100m
fog-unit compute capacity	2.6–3.5 GHz; 4–8 cores
Scenario S_1	(λ_1=100, λ_2=200, λ_3=300)
Scenario S_2	(λ_1=200, λ_2=300, λ_3=400)
Scenario S_3	(λ_1=300, λ_2=400, λ_3=500)
System	2.30 GHz Intel Core i3 4 GB RAM
OS	Microsoft Windows 10
Simulator	AnyLogic PLE v8.5

5.2.1 Scenario

The traffic of the vehicle varies with the arrival rate in the simulation. The simulation area was 3 km^2. Any vehicle which enters the simulation has a computer unit and storage on-board. The vehicles with their measurement and storage capacities build tasks during their lifetime. A task offload decision are established based on the pending tasks and/or unambiguous connectivity along the micro-fog unit. As specified earlier, there were eight access points where the vehicles can enter the simulation. We classified these entry points into three groups with distinctive arrival rates which are defined as λ_1, λ_2 and λ_3. For evaluation, we illustrated three scenarios with fluctuating combinations of arrival rates, as listed in Table 5.1. Mention that, the fog nodes have up to cores with heterogeneous computing competences, but every vehicles have identical four cores. The total simulation time for each iteration was 1 hour and here we run 5 iterations.

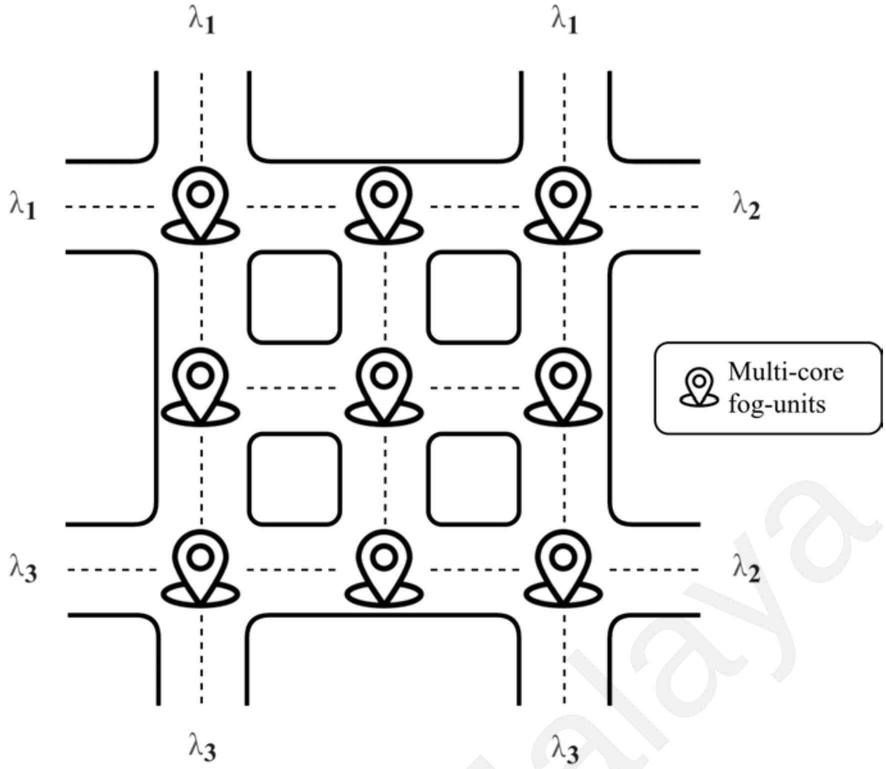

Figure 5.1: The topology of the network used to evaluate the proposed work. For the entry points, The arrival rates are defined as- ($\lambda_1, \lambda_2, \lambda_3$).

5.2.2 Network topology

The proposed work is illustrated in Fig. 5.1 by using some bench-marking. With the use of nine multi-core fog units connected via a wired network, all roads were bidirectional. Vehicles are communicated by wireless connection. The arrival rate stand for as λ is defined as the number of vehicles arriving the simulation for every entry point per hour. The entering vehicles take random direction until it exits the simulation in consequence of any exit point. To benchmark different algorithms, we defined four workload scenarios placed on the arrival rate as indexed in Table 5.1. This is performed to simulate imbalanced workload position,in particular, with standalone RSUs deficiency to handle the incoming requests. Such imbalance is frequent in realistic scenarios in regions with dense vehicular traffic easily overloading the nearby RSU. On the other hand, resources at the neighbouring RSU remain under-utilized in a less impenetrable environment. Hence, an RSU-based collaborative resource distribution facilitates handling resource inquiry in varying vehicular environments.

5.3 Queuing Delay

Queuing delay is the amount of time a task waits in the task queue before execution. Due to the time-critical nature of offloaded tasks, minimum queuing delay is desirable to ensure service quality. Figure 5.2 is compared the delay among four approaches including the proposed fog collaborative approach. In scenario S_1, the queuing delay is similar for all approaches due to limited workload; however, as the workload increases, we observed significant differences in queuing delay as shown in scenarios S_2 and S_3. Notably, scenario S_3 that generates the maximum workload, results in the highest queuing delay for the non-collaborative approach $\mathscr{F}'$, followed by RWA and NFA with 50% and 43% decrease, respectively. Whereas, the proposed collaborative approach $\mathscr{F}$ reduces the queuing delay significantly by 72% due to even distribution of workload, improving utilization of underutilized fog nodes. The max queuing delay is observed in $\mathscr{F}'$ as all the received tasks at fog units are locally executed, therefore, with increasing arrival rate and fixed processing capacity, the queuing delay also increases. Whereas, in NWA shows a slight better queuing delay on the maximum workload, due to its collaborative task-sharing approach i.e. the tasks are offloaded to directly connected neighbors. Moreover, here we assumed that the neighbors cannot offload the task further to avoid looping scenarios. Further, the RWA has a better queuing delay due to its flexible random fog node selection criteria. The fog unit can offload tasks to a randomly selected unit. The proposed $\mathscr{F}$ shows better results as it considered workload at all the fog units.

5.4 Delivery rate

It is the ratio of the number of tasks offloaded to fog nodes and the number of successfully returned task outputs to the task source vehicles. It is the measure for dealing

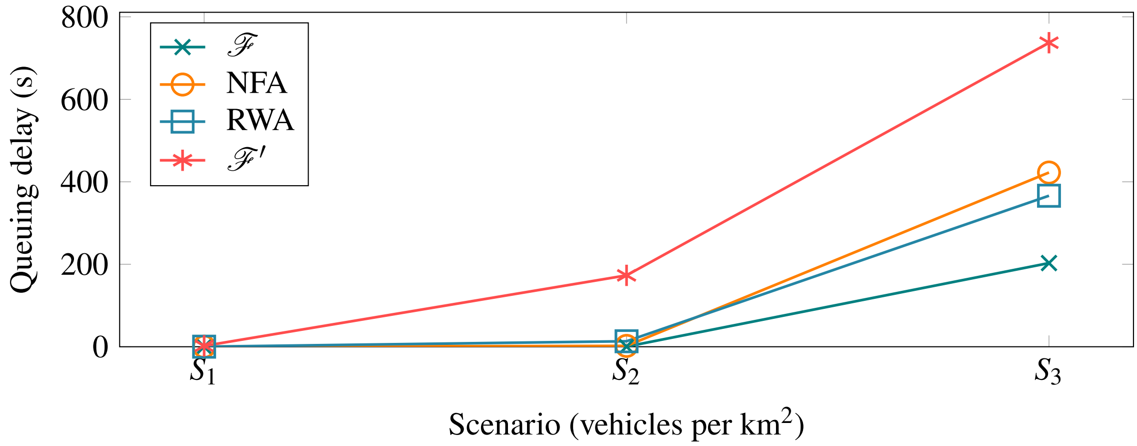

Figure 5.2: Queuing time for different workload scenarios.

with the amount of activities in each model deploy from one federate (the local RSU) to other. Since the vehicles are continuously moving, they may end up being out of the data communication range of the fog node. This leads to delivery failures when the fog node attempts to return the task output to the source vehicle. Figure 5.3 shows the delivery rate for the four approaches including the collaborative fog computing approach that shows the proposed multi-level queuing federated model $\mathscr{F}$ deliver the approaching tasks from vehicles in RSU's spectrum while the remaining tasks are computed locally on the RSU. On the other hand, 70% delivery rate for RWA and NFA is quite similar compared to $\mathscr{F}$ but $\mathscr{F}'$ is reverse compared to others. In the lightly loaded scenario S_1, all approaches performed at the same level; however, with an increased workload in scenarios S_2 and S_3, the differences in the delivery rate becomes apparent. The delivery rate for the non-federated approach $\mathscr{F}'$ reduces significantly which is 50% with an increasing workload, due to task execution at local fog node; the delivery rate is 45% higher as compared federated approach as they offload the task to other fog units; thus, unable to deliver task results directly to the source vehicle. The lower delivery rate of $\mathscr{F}$ is because every RSU calculates a local index to determine its position. The nodes do not load tasks around the federation unless a seller within the next federations is available, otherwise the duty for local computation stays with the federation. However, to handle such scenarios, we have adopted the multi-hop delivery option, where the vehicles are used to deliver the result to the source vehicle. Thus, only three-hop delivery is considered after that, the tasks are marked as failure. In comparison, the fog collaborative model $\mathscr{F}$ performs better compared to $\mathscr{F}'$, RWA, and NFA.

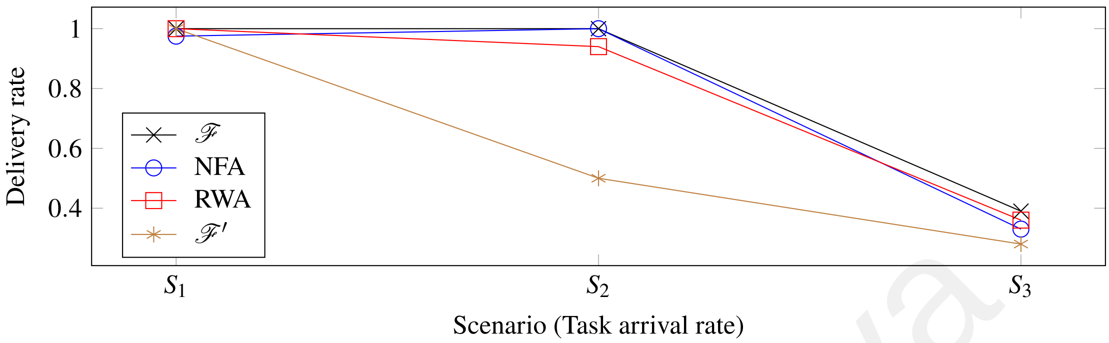

Figure 5.3: Delivery rate for different workload scenarios.

5.5 Pending task rate

Pending task rate corresponds to the number of tasks waiting in task queue for execution. In the simulation, end devices generate tasks that are executed locally or offloaded to fog nodes. Consequently, at the simulation end, some tasks might end up queued at the end device or in-range fog node. Moreover, in the case of RWA, NFA, and $\mathscr{F}$, some tasks may remain queued at a fog node in the fog federation. Figure 5.4 shows the pending tasks in a heavy loaded task offloading scenario S_3. Here, the non-collaborative approach $\mathscr{F}'$ results in the highest pending task rate, followed by RWA and NFA with 35% and 31% decrease, respectively. In the proposed collaborative approach $\mathscr{F}$, there are only 9% pending tasks compared to 26% in non-collaborative approach $\mathscr{F}'$, an improvement of 67%.

5.6 Coverage

In a collaborative environment, there are two task types (firm and soft) circulating over the network. Here, we defined coverage as *the ratio of tasks executed at the fog node that are successfully delivered to the end device.* Fig. 5.5 illustrates the performance of the

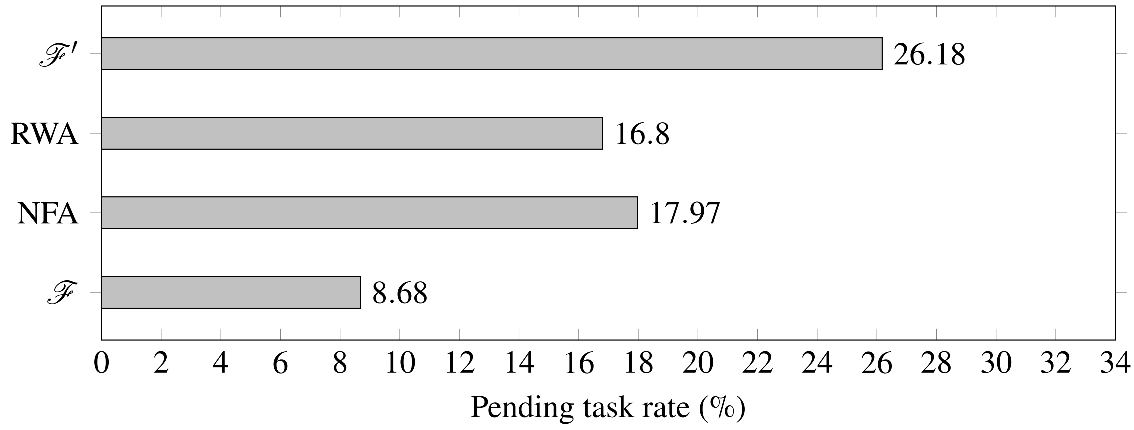

Figure 5.4: Pending task rate for workload scenario S_3.

delivery ratios for different task types. Note that a same number of firm and soft tasks are offloaded to the fog nodes by the end devices. In a non-collaborative $\mathscr{F}'$ approach, the tasks are executed on the local fog node, so on average the coverage for both task types is the same at 28%; therefore, the average of both the tasks is 28%. However, in a collaborative environment, there is an improvement in coverage; notably, the proposed $\mathscr{F}$ approach clearly outperforms the other approaches, an improvement of 78% compared to the non-collaborative approach. Moreover, we observe that firm tasks get more priority over soft tasks due to their stricter task completion deadlines. Consequently, the coverage rate of the firm and soft tasks stands at 63% and 36%, respectively; and the average rate achieved is around 50%. Even though, in all the approaches, the ready task queue is sorted based on the laxity time; however, due to the efficient collaborative fog unit selection model, the coverage rate of the firm tasks improves significantly.

5.7 Tasks Execution Granularity

Fig. 5.6 shows the overall task execution at different levels, such as the number of tasks executed at the end device, in-range fog node, and federated fog node. In the non-collaborative approach $\mathscr{F}'$, 98% tasks are offloaded to its local fog nodes. Note that for all approaches, the number of tasks executed locally at the end devices are relatively similar.

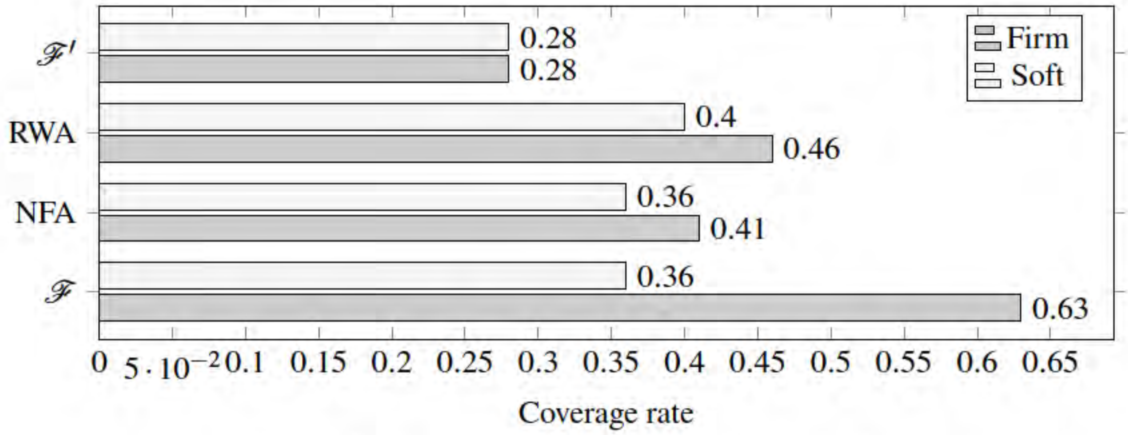

Figure 5.5: Coverage rate for workload scenario S_3.

For the collaborative environment, we observed that despite the number of federated tasks in $\mathscr{F}$ is 12%less than RWA and 57% less than RWA, its performance in terms of queuing delay, coverage rate and pending task rate is significantly better compared to other approaches. This is mainly due to the selection of suitable federated fog units.

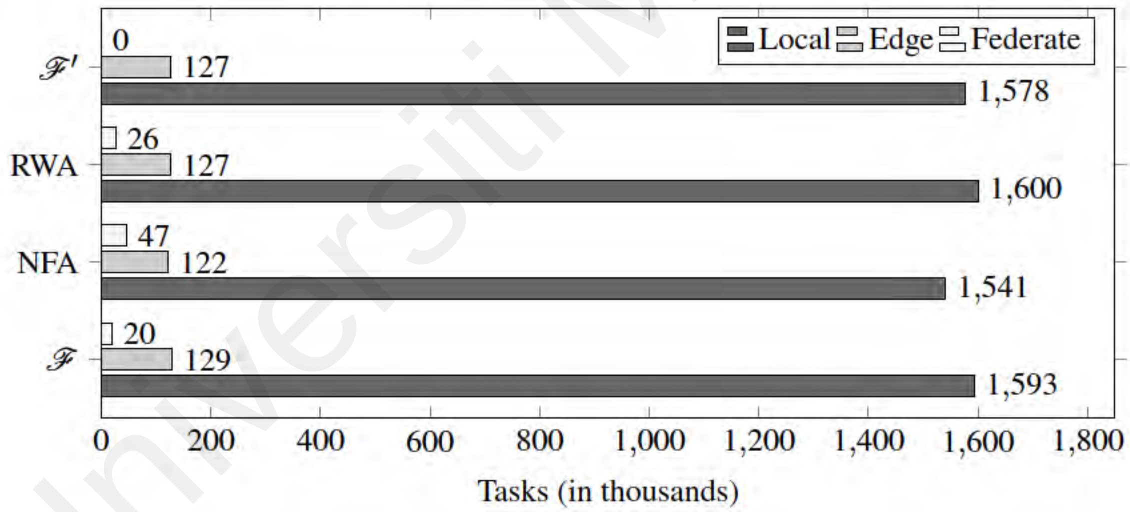

Figure 5.6: Task statistics for scenario S_3.

5.8 Summary

The experimental results demonstrate an improved service throughput for the priority tasks. The priority-based offloading model executes all hard-priority tasks locally, whereas firm- and soft-priority tasks are either executed locally or offloaded to the available fog nodes. In comparison to the traditional non-federated approach where the tasks are offloaded for execution to the fog nodes in proximity, as well as, the traditional federated approaches, the proposed collaborative fog computing model significantly reduced the number of waiting tasks due to its efficient multi-stage workload-based decision mechanism. Moreover, the online laxity-aware scheduling at the fog nodes reduces the queuing delay resulting in a high delivery rate. These results show that, the proposed fog federation improves the service quality, in terms of delivery rate compared to a non-federated approach.

CHAPTER 6: CONCLUSION

In this chapter, the conclusion of the research, main findings and contributions of the research are presented and is concluded with showing future works.

6.1 Achievement

In this work, we studied a balanced task offloading to satisfy deadline requirements of the priority-aware tasks. The proposed multi-stage task offloading model enables balanced workload distribution in the fog federation. In particular, for suitable fog resource selection, we used the *balls-into-bins* game for workload balancing. The results demonstrate that the completion ratio of priority task improves compared to the traditional non-federated and federated approaches. As a part of future work, we are planning to explore adding cloud tier for low priority task execution. Moreover, we plan to experiment with more sophisticated load balancing and task scheduling of priority tasks over heterogeneous fog resources.

In chapter-2, the literature of vehicular edge computing, laxity, fog system and there structure are reviewed in this research. The studies of existing workload distribution system and task offloading are also conducted and the characteristic of the fog computing is compared based on taxonomy, advantages and limitation of the priority based modelling.

In chapter-3, the methodology and problem formulation are described. The proposed model of workload distribution has been done by using priority based scheduling and task are distributed FCFS manner. To implement the proposed model, two tier fog computing model, its workflow, communication model and resource allocation policy are described.

In chapter-4, the collaborative fog computing model are discussed briefly. The algorithm of priority aware and collaborative computing model are also explained in this chapter. The methods of fog node selecting for offloading based on its priority, are also discussed.

In chapter-5, workload distribution system are simulated by using ANYLOGIC simulation tools. The performance metrics of the proposed system are- number of alive nodes, fog units, its range and computing capacity, number of task are compared with the performance metrics of RWA and NWA. At the end of this chapter, we summarise the compared result among non-federation and federation approaches. The proposed collaborative fog computing approaches significantly reduces the number of waiting task in multi-stage workload distribution system. The results shows that, laxity-aware scheduling reduces queuing delay and improves high delivery rate compared to non-federated approaches.

6.2 Contribution

The main contribution of this research is to improve task fairness and throughput using a multi-tier priority-based fog computing model. At the fog nodes, we used online laxity-aware multi-queue task scheduling. Here, task laxity is a dynamic measure having a direct relationship to the task completion deadline. Furthermore, we implemented a model to exploit federated fog resources based on workload. The research contribution are summarised as follows:-

- Developed a model to enhances the task delivery rate and throughput using AnyLogic simulation tool.
- The source code are written in JAVA programming language.
- Evaluated and analyzed the number of task arrival rate in fog units.
- To achieved high throughput, priority based multi-stage laxity scheduling model is implemented to provide workload-distributed offloading decision model.

6.3 Future Work

In this research, we adopted a cloud and fog communication architecture to take advantage of fog computing. To achieved the most value from such an architecture, computing tasks must be strategically allocated to each cloud or fog layer processing node. For the scheduling problems of complex tasks in IoT applications with priority constraints. This research deals with the related fog computing scheduling mission. In the fog setting, the related work scheduling approach based on the laxity method is proposed, which takes energy usage into account and aims to achieve a decrease in energy consumption on condition that the mixing deadline is reached. Simulations and numerical studies have demonstrated that higher performance than other existing approaches can be shown in our work.

In future, we are planning to deploy our proposal algorithm in real world systems. We will consider an IoT implementation scenario with user-defined review queries (tasks) that need to be executed to execute the queries on several fog nodes at the edge or public cloud nodes. Through the expected implementation, the success in the real-world operation can be closely analyzed and the shortcomings found to strengthen our idea. In the other hand, the scheduling of tasks that require independent tasks and related tasks should be considered.

6.4 Summary

The model facilitates the exchange of knowledge on workloads for the fog unit model, with the goal of meeting the time limit. The model suggests a multi level federal selection model, namely that only when a provider is categorized as the source, should share federated services from several potential suppliers. The result showed that, in contrast to standard methods, the proposed model based on time of queue distributes activities around a federation in a balance.

Here, federated technique is used to balance workload across under-utilized computing capability, which is used to maintain the recognized QoS to end-users and neighboring federates are used for workload balancing. The proposed technique shows a significant reduction in workload imbalance compared to all other techniques. In such a way, the fog units with uniformly distributed workloads to enhance the QoS.

REFERENCES

Abbasi, M., Mohammadi-Pasand, E., & Khosravi, M. R. (2021). Intelligent workload allocation in iot–fog–cloud architecture towards mobile edge computing. *Computer Communications*, *169*, 71–80.

Adhikari, M., Mukherjee, M., & Srirama, S. N. (2019). Dpto: A deadline and priority-aware task offloading in fog computing framework leveraging multilevel feedback queueing. *IEEE Internet of Things Journal*, *7*(7), 5773–5782.

Ageed, Z., Mahmood, M. R., Sadeeq, M., Abdulrazzaq, M. B., & Dino, H. (2020). Cloud computing resources impacts on heavy-load parallel processing approaches. *IOSR Journal of Computer Engineering (IOSR-JCE)*, *22*(3), 30–41.

Al-Khafajiy, M., Baker, T., Asim, M., Guo, Z., Ranjan, R., Longo, et al. (2020). Comitment: a fog computing trust management approach. *Journal of Parallel and Distributed Computing*, *137*, 1–16.

Anglano, C., Canonico, M., Castagno, P., Guazzone, M., & Sereno, M. (2018). A game-theoretic approach to coalition formation in fog provider federations. In *2018 third international conference on fog and mobile edge computing (fmec)* (pp. 123–130).

Aramrattana, M., Larsson, T., Jansson, J., & Nåbo, A. (2019). A simulation framework for cooperative intelligent transport systems testing and evaluation. *Transportation research part F: traffic psychology and behaviour*, *61*, 268–280.

Beraldi, R., Mtibaa, A., & Alnuweiri, H. (2017). Cooperative load balancing scheme for edge computing resources. In *2017 second international conference on fog and mobile edge computing (fmec)* (pp. 94–100).

Bi, Y. (2018). Neighboring vehicle-assisted fast handoff for vehicular fog communications. *Peer-to-Peer Networking and Applications*, *11*(4), 738–748.

Bittencourt, L. F., Diaz-Montes, J., Buyya, R., Rana, O. F., & Parashar, M. (2017). Mobility-aware application scheduling in fog computing. *IEEE Cloud Computing*, *4*(2), 26–35.

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the mcc workshop on mobile cloud computing* (pp. 13–16).

Bozorgchenani, A., Tarchi, D., & Corazza, G. E. (2017). An energy and delay-efficient partial offloading technique for fog computing architectures. In *Globecom 2017-2017 ieee global communications conference* (pp. 1–6).

Brito, M. S., Hoque, S., Steinke, R., Willner, A., & Magedanz, T. (2018). Application of the fog computing paradigm to smart factories and cyber-physical systems. *Trans. Emerging Telecommun. Technol.*, *29*(4), e3184.

Cao, Q., Wei, Z.-B., & Gong, W.-M. (2009). An optimized algorithm for task scheduling based on activity based costing in cloud computing. In *2009 3rd international conference on bioinformatics and biomedical engineering* (pp. 1–3).

Chen, & Wang, L. (2017). Exploring fog computing-based adaptive vehicular data scheduling policies through a compositional formal method—pepa. *IEEE Communications Letters*, *21*(4), 745–748.

Chen, H., Zhu, J., Wu, G., & Huo, L. (2018). Cost-efficient reactive scheduling for real-time workflows in clouds. *The Journal of Supercomputing*, *74*(11), 6291–6309.

Cheng, H., Lu, R., & Choo, K.-K. R. (2017). Vehicular fog computing: architecture, use case, and security and forensic challenges. *IEEE Commun. Mag.*, *55*(11), 105–111.

Desai, S., & Thakkar, A. (2019). The fog computing paradigm: A rising need of iot world. In *Proceedings of the 2nd international conference on data engineering and communication technology* (pp. 387–393).

Dhar, R., Chakraborty, A., & Sarkar, J. (2020). Queuing theory and optimization in banking models. In *Cybernetics, cognition and machine learning applications* (pp. 205–213). Springer.

Dorigo, M., & Stützle, T. (2019). Ant colony optimization: overview and recent advances. In *Handbook of metaheuristics* (pp. 311–351). Springer.

Enokido, T., Suzuki, K., Aikebaier, A., & Takizawa, M. (2010). Laxity based algorithm for reducing power consumption in distributed systems. In *2010 international conference on complex, intelligent and software intensive systems* (pp. 321–328).

Enokido, T., & Takizawa, M. (2020). The improved redundant energy consumption laxity-based algorithm with differentiating starting time of process replicas. In *International conference on emerging internetworking, data & web technologies* (pp. 181–188).

Fan, & Ansari, N. (2018). Workload allocation in hierarchical cloudlet networks. *IEEE Commun. Lett.*, *22*(4), 820–823.

Fangchun, Y., Wang, S., Li, J., Liu, Z., & Sun, Q. (2014). An overview of internet of vehicles. *China communications*, *11*(10), 1–15.

Fricker, C., Guillemin, F., Robert, P., & Thompson, G. (2016). Analysis of an offloading scheme for data centers in the framework of fog computing. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, *1*(4), 1–18.

Fuqian, Y., Zhu, Z., Zhao, S., Yang, Y., & Luo, X. (2018). Optimal task offloading in fog-enabled networks via index policies. In *2018 ieee global conference on signal and information processing (globalsip)* (pp. 688–692).

Ge, W., & Xu, F. (2020). Regional intelligent resource allocation in mobile edge computing based vehicular network. *IEEE Access*, *8*, 7173–7182.

Giust, F., Verin, G., Antevski, K., Chou, J., Fang, Y., Featherstone, et al. (2018). Mec deployments in 4g and evolution towards 5g. *ETSI White paper*, *24*(2018), 1–24.

Goudarzi, M., Movahedi, Z., & Pujolle, G. (2016). A priority-based fast optimal computation offloading planner for mobile cloud computing. *International Journal of Information and Communication Technology Research*, *8*(1), 43–49.

Grassi, G., Pesavento, D., Wang, L., Pau, G., Vuyyuru, R., Wakikawa, R., & Zhang, L. (2013). Acm hotmobile 2013 poster: vehicular inter-networking via named data. *ACM SIGMOBILE Mobile Computing and Communications Review*, *17*(3), 23–24.

Gu, X., & Zhang, G. (2021). Energy-efficient computation offloading for vehicular edge computing networks. *Computer Communications*, *166*, 244–253.

Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer* *systems*, *29*(7), 1645–1660.

Habak, K., Ammar, M., Harras, K. A., & Zegura, E. (2015). Femto clouds: Leveraging mobile devices to provide cloud service at the edge. In *2015 ieee 8th international conference on cloud computing* (pp. 9–16).

Hameed, A. R., ul Islam, S., Ahmad, I., & Munir, K. (2021). Energy-and performance-aware load-balancing in vehicular fog computing. *Sustainable Computing: Informatics and Systems*, *30*, 100454.

Hammoud, A., Mourad, A., Otrok, H., Wahab, O. A., & Harmanani, H. (2020). Cloud federation formation using genetic and evolutionary game theoretical models. *Future Gener. Comput. Syst.*, *104*, 92–104.

Hou, X., Li, Y., Chen, M., Wu, D., Jin, D., & Chen, S. (2016). Vehicular fog computing: A viewpoint of vehicles as the infrastructures. *IEEE Trans. Veh. Technol.*, *65*(6), 3860–3873.

Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., & Young, V. (2015). Mobile edge computing—a key technology towards 5g. *ETSI white paper*, *11*(11), 1–16.

Huang, D., Wang, P., & Niyato, D. (2012). A dynamic offloading algorithm for mobile computing. *IEEE Trans. Wireless Commun.*, *11*(6), 1991–1995.

Iqbal, S., Malik, A. W., Rahman, A. U., & Noor, R. M. (2020). Blockchain-based reputation management for task offloading in micro-level vehicular fog network. *IEEE Access*, *8*, 52968–52980.

Jiang, C., Xu, B., & Hsu, L.-T. (2021). Probabilistic approach to detect and correct gnss nlos signals using an augmented state vector in the extended kalman filter. *GPS Solutions*, *25*(2), 1–14.

Jurgens, A. M., & Crutchfield, J. P. (2021). Shannon entropy rate of hidden markov processes. *Journal of Statistical Physics*, *183*(2), 1–18.

Kapsalis, A., Kasnesis, P., Venieris, I. S., Kaklamani, D. I., & Patrikakis, C. Z. (2017). A cooperative fog approach for effective workload balancing. *IEEE Cloud Computing*, *4*(2), 36–45.

Karmakar, K., Das, R. K., & Khatua, S. (2020). Resource scheduling for tasks of a workflow in cloud environment. In *International conference on distributed computing and internet technology* (pp. 214–226).

Kaur, K., & Sachdeva, M. (2020). Fog computing in iot: An overview of new opportunities. In *Proceedings of icetit 2019* (pp. 59–68). Springer.

Ke, Z., Mao, Y., Leng, S., He, Y., & Zhang, Y. (2017). Predictive offloading in cloud-driven vehicles: Using mobile-edge computing for a promising network paradigm. *IEEE Vehicular Technology Magazine*, *12*(2).

Khattak, H. A., Arshad, H., ul Islam, S., Ahmed, G., Jabbar, S., Sharif, A. M., & Khalid, S. (2019). Utilization and load balancing in fog servers for health applications. *EURASIP Journal on Wireless Communications and Networking*, *2019*(1), 1–12.

Kuo, W.-H., Tung, Y.-S., & Fang, S.-H. (2013). A node management scheme for r2v connections in rsu-supported vehicular adhoc networks. In *2013 international conference on computing, networking and communications (icnc)* (pp. 768–772).

Lai, Y., Yang, F., Zhang, L., & Lin, Z. (2018). Distributed public vehicle system based on fog nodes and vehicular sensing. *IEEE Access*, *6*, 22011–22024.

Larcher, L., Ströele, V., & Dantas, M. (2019). A fog-cloud approach to enhance communication in a distance learning cloud based system. In *International conference on p2p, parallel, grid, cloud and internet computing* (pp. 275–286).

Lee, G., Saad, W., & Bennis, M. (2019). An online optimization framework for distributed fog network formation with minimal latency. *IEEE Transactions on Wireless Communications*, *18*(4), 2244–2258.

Li, L., Guan, Q., Jin, L., & Guo, M. (2019). Resource allocation and task offloading for heterogeneous real-time tasks with uncertain duration time in a fog queueing system. *IEEE Access*, *7*, 9912–9925.

Lin, Y.-D., Hu, J.-C., Kar, B., & Yen, L.-H. (2019). Cost minimization with offloading to vehicles in two-tier federated edge and vehicular-fog systems. In *2019 ieee 90th vehicular technology conference (vtc2019-fall)* (pp. 1–6).

Liu, L., Chang, Z., Guo, X., Mao, S., & Ristaniemi, T. (2017). Multiobjective optimization for computation offloading in fog computing. *IEEE IoT J.*, *5*(1), 283–294.

Madan, N., Malik, A. W., Rahman, A. U., & Ravana, S. D. (2020). On-demand resource provisioning for vehicular networks using flying fog. *Veh. Commun.*, *24*, 100252.

Masdari, M., & Zangakani, M. (2020). Efficient task and workflow scheduling in inter-cloud environments: challenges and opportunities. *The Journal of Supercomputing*, *76*(1), 499–535.

Meneguette, R. I., Bittencourt, L. F., & Madeira, E. R. M. (2013). A seamless flow mobility management architecture for vehicular communication networks. *Journal of Communications and Networks*, *15*(2), 207–216.

Menon, V. G., & Prathap, P. J. (2017). Moving from vehicular cloud computing to vehicular fog computing: Issues and challenges. *International Journal on Computer Science and Engineering*, *9*(2), 14–18.

Mishra, S. K., Sahoo, B., & Parida, P. P. (2020). Load balancing in cloud computing: a big picture. *J. King Saud Univ. Comput. Inf. Sci.*, *32*(2), 149–158.

Mithun, M., Guo, M., Lloret, J., Iqbal, R., & Zhang, Q. (2019). Deadline-aware fair scheduling for offloaded tasks in fog computing with inter-fog dependency. *IEEE Communications Letters*, *24*(2), 307–311.

Mitzenmacher, M., & Upfal, E. (2017). *Probability and computing: Randomization and probabilistic techniques*. Cambridge, U.K.: Cambridge Univ. Press.

Moghaddam, M. M., Manshaei, M. H., Saad, W., & Goudarzi, M. (2019). On data center demand response: A cloud federation approach. *IEEE Access*, *7*, 101829–101843.

Mostafa, N. (2019). Cooperative fog communications using a multi-level load balancing. In *2019 fourth international conference on fog and mobile edge computing (fmec)* (pp. 45–51).

Mukherjee, & Amitav. (2018). Energy efficiency and delay in 5g ultra-reliable low-latency communications system architectures. *IEEE Network*, *32*(2), 55–61.

Mukherjee, Kumar, S., Zhang, Q., Matam, R., Mavromoustakis, C. X., Lv, Y., & Mastorakis,

G. (2019). Task data offloading and resource allocation in fog computing with multi-task delay guarantee. *IEEE Access*, *7*, 152911–152918.

Nan, Y., Li, W., Bao, W., Delicato, F. C., Pires, P. F., & Zomaya, A. Y. (2018). A dynamic tradeoff data processing framework for delay-sensitive applications in cloud of things systems. *J. Parallel Distrib. Comput.*, *112*, 53–66.

Nguyen, D. T., Le, L. B., & Bhargava, V. K. (2019). A market-based framework for multi-resource allocation in fog computing. *IEEE/ACM Transactions on Networking*, *27*(3), 1151–1164.

Nur, F. N., Islam, S., Moon, N. N., Karim, A., Azam, S., & Shanmugam, B. (2019). Priority-based offloading and caching in mobile edge cloud. *Journal of Communications Software and Systems*, *15*(2), 193–201.

Ottenwälder, B., Koldehofe, B., Rothermel, K., & Ramachandran, U. (2013). Migcep: operator migration for mobility driven distributed complex event processing. In *Proceedings of the 7th acm international conference on distributed event-based systems* (pp. 183–194).

Peixoto, M., Maia, A., Mota, E., Rangel, E., Costa, D., Turgut, D., & Villas, L. (2021). A traffic data clustering framework based on fog computing for vanets. *Vehicular Communications*, *31*, 100370.

Porambage, P., Okwuibe, J., Liyanage, M., Ylianttila, M., & Taleb, T. (2018). Survey on multi-access edge computing for internet of things realization. *IEEE Communications Surveys & Tutorials*, *20*(4), 2961–2991.

Qiao, G., Leng, S., Zhang, K., & He, Y. (2018). Collaborative task offloading in vehicular edge multi-access networks. *IEEE Commun. Mag.*, *56*(8), 48–54.

Rahman, A. U., Malik, A. W., Sati, V., Chopra, A., & Ravana, S. D. (2020). Context-aware opportunistic computing in vehicle-to-vehicle networks. *Veh. Commun.*, *24*, 100236.

Ren, W., Sun, Y., Luo, H., & Guizani, M. (2018). A novel control plane optimization strategy for important nodes in sdn-iot networks. *IEEE Internet of Things Journal*, *6*(2), 3558–3571.

Rodrigues, T. G., Suto, K., Nishiyama, H., & Kato, N. (2016). Hybrid method for minimizing service delay in edge cloud computing through vm migration and transmission power control. *IEEE Trans. Comput.*, *66*(5), 810–819.

Ryden, M., Oh, K., Chandra, A., & Weissman, J. (2014). Nebula: Distributed edge cloud for data intensive computing. In *2014 ieee international conference on cloud engineering* (pp. 57–66).

Santoro, D., Zozin, D., Pizzolli, D., De Pellegrini, F., & Cretti, S. (2017). Foggy: A platform for workload orchestration in a fog computing environment. In *2017 ieee international conference on cloud computing technology and science (cloudcom)* (pp. 231–234).

Sapkal, A., & Kulkarni, U. (2018). Modified backpropagation with added white gaussian noise in weighted sum for convergence improvement. *Procedia computer science*, *143*, 309–316.

Saqib, M. T., & Hamid, M. A. (2016). Fogr: A highly reliable and intelligent computation offloading on the internet of things. In *2016 ieee region 10 conference (tencon)* (pp. 1039–1042).

Saraswat, S., Gupta, H. P., Dutta, T., & Das, S. K. (2019). Energy efficient data forwarding scheme in fog-based ubiquitous system with deadline constraints. *IEEE Transactions on Network and Service Management*, *17*(1), 213–226.

Shah, S. S., Ali, M., Malik, A. W., Khan, M. A., & Ravana, S. D. (2019). vfog: A vehicle-assisted computing framework for delay-sensitive applications in smart cities. *IEEE Access*, *7*, 34900–34909.

Shamseddine, H., Nizam, J., Hammoud, A., Mourad, A., Otrok, H., Harmanani, H., & Dziong, Z. (2021). A novel federated fog architecture embedding intelligent formation. *IEEE Network*, *35*(3), 198-204.

Sharmin, Z., Malik, A. W., Rahman, A. U., & Noor, R. M. (2020). Toward sustainable micro-level fog-federated load sharing in internet of vehicles. *IEEE Internet of Things Journal*, *7*(4), 3614–3622.

Shojafar, M., Cordeschi, N., & Baccarelli, E. (2016). Energy-efficient adaptive resource management for real-time vehicular cloud services. *IEEE Transactions on Cloud computing*, *7*(1), 196–209.

Siyuan, W., Tian, Z., Dong, K., & Xie, Q. (2021). Inconsistency of neighborhood based on voronoi tessellation and euclidean distance. *Journal of Alloys and Compounds*, *854*, 156983.

Srivastava, S., Banicescu, I., & Sanders, W. S. (2020). Addressing the robustness of resource allocation in the presence of application and system irregularities via pepa based modeling. In *International conference on computational science* (pp. 590–603).

Sun, Y., Zhou, S., & Niu, Z. (2021). Distributed task replication for vehicular edge computing: Performance analysis and learning-based algorithm. *IEEE Transactions on Wireless Communications*, *20*(2), 1138-1151.

Talaat, F. M., Saraya, M. S., Saleh, A. I., Ali, H. A., & Ali, S. H. (2020). A load balancing and optimization strategy (lbos) using reinforcement learning in fog computing environment. *J. Ambient Intell. Hum. Comput.*, 1–16.

Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials*, *19*(3), 1657–1681.

Tanaka, H., Yoshida, M., Mori, K., & Takahashi, N. (2018). Multi-access edge computing: A survey. *Journal of Information Processing*, *26*, 87–97.

Tang, L., & He, S. (2018). Multi-user computation offloading in mobile edge computing: A behavioral perspective. *IEEE Network*, *32*(1), 48–53.

Vegni, A. M., & Loscri, V. (2015). A survey on vehicular social networks. *IEEE Commun. Surv. Tutorials*, *17*(4), 2397–2419.

Veillon, V., Denninnart, C., & Salehi, M. A. (2019). F-fdn: Federation of fog computing systems for low latency video streaming. In *2019 ieee 3rd international conference on fog and edge computing (icfec)* (pp. 1–9).

Verma, H. K., Sharma, K. P., et al. (2021). Evolution of vanets to iov: Applications and challenges. *Tehnički glasnik*, *15*(1), 143–149.

Wallace, V. L. (2021). Numerical markov queuing analysis: A twenty-five year odyssey.

In *Numerical solution of markov chains* (pp. 1–9). CRC Press.

Wang, Tao, X., Zhang, X., Zhang, P., & Hou, Y. T. (2019). Cooperative task offloading in three-tier mobile computing networks: An admm framework. *IEEE Trans. Veh. Technol.*, *68*(3), 2763–2776.

Weihong, C., Xie, G., Li, R., & Li, K. (2021). Execution cost minimization scheduling algorithms for deadline-constrained parallel applications on heterogeneous clouds. *Cluster Computing*, *24*(2), 701–715.

Wu, C., Buyya, R., & Ramamohanarao, K. (2019). Cloud pricing models: Taxonomy, survey, and interdisciplinary challenges. *ACM Comput. Surv.*, *52*(6), 108.

Xiao, L., Zhuang, W., Zhou, S., & Chen, C. (2019). Learning while offloading: Task offloading in vehicular edge computing network. In *Learning-based vanet communication and security techniques* (pp. 49–77). Springer.

Xiaolong, X., Fu, S., Cai, Q., Tian, W., Liu, et al. (2018). Dynamic resource allocation for load balancing in fog environment. *Wireless Commun. Mobile Comput.*, *2018*, 1–16.

Xu, J., Hao, Z., Zhang, R., & Sun, X. (2019). A method based on the combination of laxity and ant colony system for cloud-fog task scheduling. *IEEE Access*, *7*, 116218–116226.

Yang, Chao, Y., Liu, Chen, X., Zhong, W., & Xie, S. (2019). Efficient mobility-aware task offloading for vehicular edge computing networks. *IEEE Access*, *7*, 26652–26664.

Yang, Luo, X., Chu, X., & Zhou, M.-T. (2020). Fog computing architecture and technologies. In *Fog-enabled intelligent iot systems* (pp. 39–60). Springer.

Ye, D., Wu, M., Tang, S., & Yu, R. (2016). Scalable fog computing with service offloading in bus networks. In *2016 ieee 3rd international conference on cyber security and cloud computing (cscloud)* (pp. 247–251).

Yi, S., Li, C., & Li, Q. (2015). A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 workshop on mobile big data* (pp. 37–42).

Yin, Z., Chen, H., & Hu, F. (2019). An advanced decision model enabling two-way initiative offloading in edge computing. *Future Gener. Comput. Syst.*, *90*, 39–48.

Yinuo, F., Zhai, L., & Wang, H. (2019). Cost-efficient dependent task offloading for multiusers. *IEEE Access*, *7*, 115843–115856.

Yuan, Z., Liu, H., Jiao, L., & Fu, X. (2012). To offload or not to offload: an efficient code partition algorithm for mobile cloud computing. In *2012 ieee 1st international conference on cloud networking (cloudnet)* (pp. 80–86).

Zhang, Guo, B., Shen, Y., Li, D.-G., & Li, J.-K. (2018). An energy-efficient algorithm for multi-site application partitioning in mcc. *Sustainable Comput. Inf. Syst.*, *18*, 45–53.

Zhang, Mao, Y., Leng, S., Vinel, A., & Zhang, Y. (2016). Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks. In *2016 8th international workshop on resilient networks design and modeling (rndm)* (pp. 288–294).

Zhangand He, F., Sato, T., & Oki, E. (2019). Optimization of network service scheduling with resource sharing and preemption. In *2019 ieee 20th international conference on high performance switching and routing (hpsr)* (pp. 1–6).

Zhanikeev, M. (2015). A cloud visitation platform to facilitate cloud federation and fog computing. *IEEE Annals of the History of Computing*, *48*(05), 80–83.

Zhiyuan, J., Zhou, S., Guo, X., & Niu, Z. (2017). Task replication for deadline-constrained vehicular cloud computing: Optimal policy, performance analysis, and implications on road traffic. *IEEE IoT J.*, *5*(1), 93–107.

Zhu, Q., Si, B., Yang, F., & Ma, Y. (2017). Task offloading decision in fog computing system. *China Commun.*, *14*(11), 59–68.